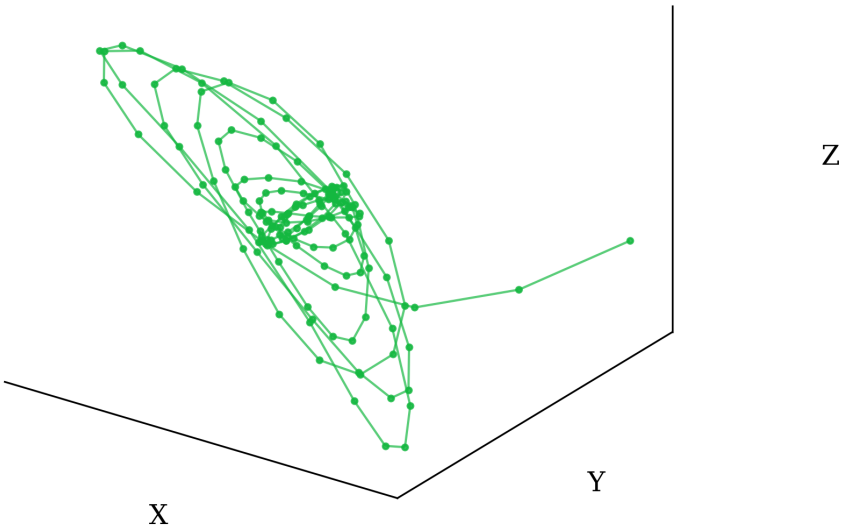

Tree Motion in Three Dimensions

Video-Based Reconstruction using a Calibrated Multi-Camera System

Master's thesis in Engineering Mathematics and Computational Science, MSc (MPENM)

CARL GILLMERT

Department of Electrical Engineering
CHALMERS UNIVERSITY OF TECHNOLOGY
Gothenburg, Sweden 2026
www.chalmers.se

Master's thesis in Engineering Mathematics and Computational Science, MSc (MPENM)
2026

Tree Motion in Three Dimensions

Video-Based Reconstruction using a Calibrated Multi-Camera System

CARL GILLMERT

Department of Electrical Engineering
Chalmers University of Technology
Gothenburg, Sweden 2026

Tree Motion in Three Dimensions: Video-Based Reconstruction using a Calibrated Multi-Camera System
CARL GILLMERT

Supervisor: Franziska Hunger, franziska.hunger@fcc.chalmers.se
Gustav Kettil, gustav.kettil@fcc.chalmers.se

Examiner: Mats Viberg, Electrical Engineering, mats.viberg@chalmers.se

Master's Thesis in Engineering Mathematics and Computational Science,
MSc (MPENM) 2026
Department of Electrical Engineering
Chalmers University of Technology
SE-412 96 Gothenburg
Telephone +46 31 772 1000

Cover: Reconstructed 3D trajectory of a marker attached to a spruce tree following a pull-and-release excitation.

Typeset in LaTeX
Printed by Chalmers Reproservice
Gothenburg, Sweden 2026

Tree Motion in Three Dimensions: Video-Based Reconstruction using a Calibrated Multi-Camera System
CARL GILLMERT
Department of Electrical Engineering
Chalmers University of Technology

Abstract

Wind-induced tree damage poses a significant risk in both managed forests and urban environments. While computational models could improve how these risks are assessed, their development and validation require reliable experimental data, including spatially resolved measurements of tree motion. A low-cost, non-invasive video-based method is presented for reconstructing time-resolved three-dimensional tree motion from a calibrated three-camera system. Physically attached colour markers are detected through image segmentation, initialised using triangulation and tracked using an Extended Kalman Filter with a Singer acceleration model, requiring no training data. The method was tested and validated in a controlled motion-tracking experiment against an industrial robot arm and a terrestrial LiDAR scanner. It achieves sub-centimetre accuracy with point-wise variation close to the limit set by the camera resolution. The method produced reproducible three-dimensional trajectories and relatively consistent frequency estimates wherever the markers remained sufficiently visible when it was applied to pull-and-release experiments on a spruce, pine and birch tree. The reconstructed motion revealed location-dependent dynamics, with stem and branch features showing different apparent frequencies and decay behaviour. Branch features generally sustained oscillations longer than stem features, consistent with the role branches are thought to play in damping tree motion. Tracking reliability depended strongly on marker visibility, with spruce performing best and pine proving most challenging due to dense foliage and marker occlusion. The thesis establishes that a small, low-cost multi-camera system can deliver accurate, physically interpretable three-dimensional measurements at multiple points simultaneously, providing a foundation for extending the method to mature trees under natural wind loading outdoors.

Keywords: tree dynamics, multi-view tracking, non-destructive measurement, Extended Kalman Filter, marker tracking

Acknowledgements

I would like to thank Fraunhofer-Chalmers Centre for giving me the opportunity to work as a contracted student over the past year and to carry out this thesis project. In particular, I thank Franziska Hunger and Gustav Kettil for welcoming me into the project, for their guidance and for introducing me to the fascinating world of tree modelling. A special thanks to Markus Berg for his help with all experiments. I would also like to thank my examiner, Mats Viberg, for his time and valuable feedback throughout the work.

This thesis is part of Digital Twin Cities Centre supported by Vinnova under the grant No. 2024-03904, the Driving Urban Transition project Urban ElemenTREE with financial support from the Swedish Research Council for Sustainable Development Formas under the grant 2024-01221 and Brattåsstiftelsen under the grant F24:07.

To my dear fiancée Saga, thank you for your endless love and support over the past years.

Carl Gillmert, Gothenburg, June 2026

List of Acronyms

Below is the list of acronyms that have been used throughout this thesis listed in alphabetical order:

2D	Two-dimensional
3D	Three-dimensional
CCL	Connected-component labelling
DFT	Discrete Fourier transform
DHO	Damped harmonic oscillator
EKF	Extended Kalman filter
HSV	Hue, saturation, value
EKF	Extended Kalman filter
LiDAR	Light detection and ranging
PC	Principal component
PCA	Principal component analysis
PSD	Power spectral density
RGB	Red, green, blue
RMS	Root mean square
RPE	Reprojection error
SVD	Singular value decomposition
TCP	Tool centre point

Nomenclature

Below is the nomenclature of indices, sets, parameters and variables that have been used throughout this thesis.

Indices

t	Time or discrete frame index.
i	Camera index.
j	Sample, calibration-image or track index.
k	Feature, corner, component, point or frequency-bin index.
l	Segment index in the Welch PSD estimator.
c	HSV colour-channel index, $c \in \{H, S, V\}$.

Sets

Ω	Image domain.
Ω_k	Pixel set belonging to connected component k.
$\mathcal{S}$	Set of acquired LiDAR scans.

Counts

N	Number of cameras or number of tracks.
n	State dimension or number of associated detections in a frame.
m	Measurement dimension.
J	Number of calibration images or number of samples in a signal.
K_p	Number of checkerboard inner corners.
M	Number of observations in a frame or number of point correspondences.

$N_{\text{seg}},\ N_{\text{seg},it}$	Number of connected components, overall and for camera i at frame t.
N_w	Number of Welch segments.
N_{jk}	Number of LiDAR points in scan $(\boldsymbol{P}_j, L_k)$.

Coordinate Frames

$\mathcal{W}$	World coordinate frame.
$\mathcal{C},\ \mathcal{C}_i$	Camera reference frame and local coordinate frame of camera i.
$\mathcal{T}$	Tree coordinate frame.
$\mathcal{R}$	Robot coordinate frame.
$\mathcal{L}$	LiDAR coordinate frame.

Parameters

K	Intrinsic camera calibration matrix.
f	Focal length.
f_x, f_y	Focal lengths in pixel units.
c_x, c_y	Principal point coordinates in pixel units.
$\boldsymbol{d}$	Lens distortion parameter vector.
k_1, k_2, k_3	Radial distortion coefficients.
p_1, p_2	Tangential distortion coefficients.
$R, \boldsymbol{t}$	Rotation matrix and translation vector in a rigid body transformation.
$R_i, \boldsymbol{t}_i$	Extrinsic parameters of camera i.
s	Checkerboard square side length, structuring element and viewing ray parameter.
l, u	Lower and upper HSV threshold bounds.
T	Sampling interval.
f_s	Sampling frequency.
F	State transition matrix.
Q	Process noise covariance matrix.
R	Measurement noise covariance matrix.
r	Measurement noise standard deviation per coordinate in pixels.

α	Inverse acceleration correlation time in the Singer model.
σ_m^2	Steady-state acceleration variance in the Singer model.
γ	Mahalanobis gating threshold.
L	Segment length in the Welch PSD estimator.
h	Segment step size in the Welch PSD estimator.
$\lambda_{\text{out}}, \lambda_{\text{black}}$	Cell-occupancy penalty weights in LiDAR-to-world registration.
A	Initial amplitude in the damped harmonic oscillator model and the stacked triangulation matrix $A\tilde{\boldsymbol{X}} = \mathbf{0}$.
δ	Decay rate in the damped harmonic oscillator model.
f_0	Oscillation frequency in the damped harmonic oscillator model.
ϕ	Phase in the damped harmonic oscillator model.

Variables

$\boldsymbol{X} = (X, Y, Z)$	Three-dimensional point in the world frame.
$\boldsymbol{X}^{\mathcal{C}}$	Three-dimensional point in a camera frame.
$\boldsymbol{X}^{\mathcal{T}}$	Three-dimensional point in the tree frame.
$\tilde{\boldsymbol{X}}$	Homogeneous representation of a point.
$\boldsymbol{C}^{\mathcal{C}}$	Camera centre (optical centre) in camera frame.
$\lambda,\ \lambda_i$	Depth (projective scale factor) of a point. λ_i for camera i.
u, v	Pixel coordinates.
$\boldsymbol{x}^{\text{px}}$	Pixel coordinate vector.
$\boldsymbol{x}^{\text{n}}$	Normalised image coordinate vector.
$\boldsymbol{x}^{\text{d}}$	Distorted normalised image coordinate vector.
$\pi(\cdot)$	Camera projection model.
$f(\cdot)$	Lens distortion function.
f	HSV image with three channels, $f : \Omega \subset \mathbb{Z}^2 \to \mathbb{R}^3$.
f_b	Binary image, $f_b : \Omega \subset \mathbb{Z}^2 \to \{0, 1\}$.
M_{it}	Binary segmentation mask for camera i at frame t.
$\boldsymbol{x}_{itk}^{\text{px}}$	Image-space observation (centroid) k in camera i at frame t.
$x(t)$	Hidden state vector.
$\hat{x}(t)$	Estimated state vector.
$y(t)$	Measurement vector.
$h(x(t))$	Non-linear measurement function.

$P(t)$	State covariance matrix.
$H(t)$	Measurement Jacobian.
$r(t)$	Innovation vector.
$S(t)$	Innovation covariance matrix.
$K(t)$	Kalman gain matrix.
$d_{ij}^2(t)$	Squared Mahalanobis distance.
$C_{ij}(t)$	Assignment cost.
$\boldsymbol{p}_{\text{base}}, \boldsymbol{p}_{\text{top}}$	Tree base and top stem points.
$\boldsymbol{P}_j$	Endpoint position j in the controlled motion experiment.
Δt	Temporal offset between two trajectories.
$\boldsymbol{v}_1, \boldsymbol{v}_2, \boldsymbol{v}_3$	Principal component directions.
$\lambda_1, \lambda_2, \lambda_3$	PCA eigenvalues.
Λ	Diagonal matrix of PCA eigenvalues or of singular values in a singular value decomposition.
Σ	PCA sample covariance matrix or the diagonal matrix of singular values in a singular value decomposition.
$s_i(t_j)$	Projection of a trajectory onto principal component i.
$s(t_j)$	Scalar signal used for spectral analysis.
f_k	Discrete frequency bin.
$\hat{S}(f_k)$	Estimated power spectral density.

Contents

List of Acronyms **ix**

Nomenclature **xi**

List of Figures **xxi**

List of Tables **xxv**

1 Introduction **1**
1.1 Background . . . 1
1.2 Related Work . . . 2
1.3 Purpose . . . 4
1.4 Objectives . . . 4
1.5 Limitations . . . 5

2 Theory **7**
2.1 Camera Geometry and Calibration . . . 7
2.1.1 Projective Geometry and Homogeneous Coordinates . . . 8
2.1.2 World and Camera Coordinate Systems . . . 8
2.1.3 Rigid Body Transformation . . . 9
2.1.4 The Pinhole Camera Model . . . 10
2.1.4.1 Perspective Projection . . . 10
2.1.4.2 Pixel Coordinate System . . . 11
2.1.4.3 Normalised Image Coordinates . . . 12
2.1.4.4 The Intrinsic Calibration Matrix . . . 12
2.1.4.5 Camera Matrix Formulation . . . 13
2.1.5 Lens Distortion Model . . . 13
2.1.6 Camera Projection Model . . . 14
2.1.7 Camera Calibration . . . 15
2.1.7.1 Corner Detection . . . 16
2.1.7.2 Intrinsic Calibration . . . 16
2.1.7.3 Extrinsic Calibration . . . 17
2.1.7.3.1 Stereo Calibration . . . 17
2.1.7.3.2 Multi-Camera Composition . . . 18
2.1.8 Triangulation . . . 19
2.2 Image Segmentation . . . 20
2.2.1 Colour Space Representation . . . 20

2.2.2 Colour-Based Thresholding . . . 21
2.2.3 Morphological Operations . . . 22
2.2.4 Connected Components and Centroid Computation . . . 22
2.3 Multi-View Motion Tracking . . . 23
2.3.1 State-Space Models . . . 24
2.3.1.1 Singer Acceleration Model . . . 25
2.3.1.2 Camera Measurement Model . . . 26
2.3.2 Extended Kalman Filter . . . 27
2.3.2.1 Prediction . . . 28
2.3.2.2 Linearisation . . . 28
2.3.2.3 Update . . . 28
2.3.3 Data Association . . . 29
2.3.3.1 Measurement Gating . . . 30
2.3.3.2 Assignment Problem . . . 30
2.3.4 Fixed-Interval Smoothing . . . 31
2.4 Coordinate Frame Alignment . . . 32
2.4.1 Point Set Registration . . . 33
2.5 Post-processing of Reconstructed Trajectories . . . 34
2.5.1 Temporal Alignment . . . 34
2.5.1.1 Linear Interpolation . . . 34
2.5.1.2 Time-Shift Estimation . . . 35
2.5.2 Principal Component Analysis . . . 35
2.5.3 Frequency Estimation . . . 36
2.5.3.1 Discrete Fourier Transform . . . 36
2.5.3.2 Energy-Weighted Welch PSD . . . 36
2.5.3.3 Damped Harmonic Oscillator . . . 38

3 Methods **39**
3.1 Camera Calibration . . . 39
3.1.1 Calibration Setup . . . 40
3.1.2 Corner Detection . . . 40
3.1.3 Intrinsic Calibration . . . 40
3.1.3.1 Data Acquisition . . . 40
3.1.3.2 Parameter Estimation . . . 41
3.1.3.3 Uncertainty Estimation . . . 41
3.1.4 Extrinsic Calibration . . . 41
3.1.4.1 Data Acquisition . . . 41
3.1.4.2 Parameter Estimation . . . 42
3.2 Multi-View Motion Tracking . . . 42
3.2.1 Colour-Based Segmentation . . . 44
3.2.2 Track Initialisation . . . 45
3.2.3 Process Model . . . 46
3.2.4 Prediction and Projection . . . 46
3.2.5 Data Association . . . 46
3.2.6 Measurement Update . . . 47
3.2.7 Smoothing . . . 47

3.3 Controlled Motion Experiment . . . 48
3.3.1 Experimental Scene and Equipment . . . 48
3.3.2 Tracked Feature . . . 49
3.3.3 Motion Protocol . . . 49
3.3.4 Coordinate Frame Registration . . . 50
3.3.4.1 Robot to World . . . 50
3.3.4.2 Video to World . . . 51
3.3.4.3 LiDAR to World . . . 51
3.3.5 Data Processing . . . 53
3.3.5.1 Video . . . 53
3.3.5.2 Robot . . . 54
3.3.5.3 LiDAR . . . 54
3.3.6 Sensor Comparison . . . 55
3.4 Tree Motion Experiments . . . 55
3.4.1 Experimental Scene and Equipment . . . 56
3.4.2 Tracked Features . . . 57
3.4.3 Pull-and-Release Protocol . . . 59
3.4.4 Tree Coordinate Frame . . . 59
3.4.5 Data Processing . . . 60
3.4.5.1 Spectral Analysis and Oscillator Fitting . . . 61
3.4.5.1.1 Signal Truncation . . . 61
3.4.5.1.2 Energy-Weighted Welch PSD . . . 61
3.4.5.1.3 Damped Harmonic Oscillator . . . 61

4 Results **63**
4.1 Intrinsic Calibration . . . 63
4.2 Controlled Motion Experiment . . . 64
4.2.1 LiDAR Scan Quality . . . 64
4.2.2 Extrinsic Calibration . . . 65
4.2.3 Coordinate Frame Registration . . . 65
4.2.4 Sensor Comparisons . . . 67
4.2.5 Camera Calibration Sensitivity . . . 70
4.3 Tree Motion Experiments . . . 71
4.3.1 Extrinsic Calibration . . . 71
4.3.2 Tracking Success . . . 72
4.3.3 Motion Trajectories . . . 73
4.3.4 Time Domain Response . . . 81
4.3.4.1 Motion in the Principal Directions . . . 81
4.3.4.2 Stem Features . . . 83
4.3.4.3 Stem and Branch Comparison . . . 83
4.3.4.4 Trajectory Reproducibility . . . 84
4.3.5 Frequency and Damping . . . 86
4.3.5.1 Energy-Weighted Welsh PSD . . . 86
4.3.5.2 Damped Harmonic Oscillator . . . 91

5 Discussion and Conclusion **97**
5.1 Controlled Motion Experiment . . . 97

5.1.1 Camera Calibration Quality 97
5.1.2 LiDAR Scan Quality 98
5.1.3 Coordinate Frame Registration 98
5.1.4 Z-axis Bias 99
5.1.5 Motion Tracking Accuracy 99
5.2 Tree Experiments 100
5.2.1 Tracking Performance 101
5.2.2 Reconstructed Motion and Reproducibility 101
5.2.3 Stem and Branch Dynamics 101
5.2.4 Frequency and Damping Estimates 102
5.2.5 Interpretation of the Tree Experiment Results 103
5.3 Conclusion 104
5.4 Future Works 105

Bibliography **107**

A Appendix 1 **I**
A.1 Intrinsic Calibration Result I

List of Figures

2.1 The rigid body transformation between the world coordinate system $\{\boldsymbol{e}_x, \boldsymbol{e}_y, \boldsymbol{e}_z\}$ centred at $\boldsymbol{C}$ and the camera coordinate system $\{\boldsymbol{e}_x^{\mathcal{C}}, \boldsymbol{e}_y^{\mathcal{C}}, \boldsymbol{e}_z^{\mathcal{C}}\}$ centred at $\boldsymbol{C}^{\mathcal{C}}$. 10

2.2 Perspective projection in the camera coordinate system. The 3D point $\boldsymbol{X}^{\mathcal{C}}$ is projected along the viewing ray onto the image plane $Z^{\mathcal{C}} = f$ producing the image point $\boldsymbol{x}$. 11

2.3 Pixel coordinate system with the origin in the upper left corner. . . . 11

2.4 The intrinsic matrix K mapping from normalised image coordinates to pixel coordinates. 13

2.5 Example of a checkerboard pattern with 10×7 inner corners. 15

2.6 Stereo calibration setup. Two cameras with optical centers $\boldsymbol{C}^{\mathcal{C}_1}$ and $\boldsymbol{C}^{\mathcal{C}_2}$ simultaneously observe a 3D point $\boldsymbol{X}$, which projects to image coordinates $\boldsymbol{x}_1^{\text{px}}$ and $\boldsymbol{x}_2^{\text{px}}$ respectively. The rigid body transformation $(R_{21}, \boldsymbol{t}_{21})$ describes the pose of camera 2 relative to camera 1 and is the quantity estimated during stereo calibration. 18

2.7 The HSV colour space represented as a cylinder. Hue encodes the colour type along the angular axis, saturation its intensity along the radial axis, and value its brightness along the vertical axis. 21

3.1 Overview of the motion tracking method. Blue boxes denote inputs, grey boxes denote processing steps and the dark-shaded box denotes the final output. Segmentation is performed as a pre-processing step over all frames before tracking begins. The dashed region marks the per-frame loop executed over all synchronised video frames. 43

3.2 Controlled motion experimental setup. Camera positions are denoted C_1, C_2 and C_3. LiDAR scanner positions are denoted L_1, L_2 and L_3. Motion target endpoint positions are denoted $\boldsymbol{P}_0$, $\boldsymbol{P}_1$, $\boldsymbol{P}_2$ and $\boldsymbol{P}_3$. The checkerboard calibration target, labelled $\mathcal{W}$, was placed flat on the table beneath the robot workspace and defines the common world coordinate frame to which all sensor data were registered. 49

3.3 Tree experimental setup. Camera positions shown as C_1, C_2 and C_3 and the pull direction from where the rope was attached on the trees. The coordinate frame $\mathcal{T}$ is centred at the tree base, i.e., the point at which the stem was fixed to the table, with X aligned with the pull direction, Z aligned with the stem axis and Y aligned with the direction opposing the viewing ray from C_2. 57

3.4 Initial feature correspondences across camera views for birch. 58
3.5 Initial feature correspondences across camera views for spruce. 58
3.6 Initial feature correspondences across camera views for pine. 59

4.1 Reprojection of the 3D checkerboard inner corners into each camera view using the jointly optimised board pose. The board pose was estimated by minimising the reprojection error across all three cameras simultaneously. 66
4.2 Filtered white LiDAR points projected onto the fitted checkerboard plane and aligned to board coordinates via a white-square occupancy optimisation, used to estimate the LiDAR-to-world coordinate transformation. Green and red points indicate white LiDAR points on white and black squares respectively, while grey points fall outside the board boundary. 67
4.3 Per-axis video-robot residuals e_X, e_Y and e_Z over time, evaluated using intrinsic calibration set 3 and the post-experiment extrinsic calibration recording. The robot trajectory (right axis) is overlaid in the background to illustrate the correlation between residual magnitude and trajectory position. 69
4.4 Successfully tracked features from the second pull and release test for birch projected on to the coordinate planes. The tree was pulled along the X-axis. Features F1-F3 are along the stem, while features F6-F12 are on branches. 74
4.5 Successfully tracked features from the second pull-and-release test for spruce projected on to the coordinate planes. The tree was pulled along the X-axis. Features F1-F6 are along the stem, while features F7-F11 are on branches. 75
4.6 Successfully tracked features from the second pull-and-release test for pine projected on to the coordinate planes. The tree was pulled along the X-axis. Feature F1 is on the stem, while features F8 and F11 are on branches. 76
4.7 Tracked feature F10 from the second pull and release test for birch projected on to the coordinate planes with PCA directions and their explained variance. 78
4.8 Tracked feature F7 from the second pull and release test for spruce projected on to the coordinate planes with PCA directions and their explained variance. 79
4.9 Tracked feature F1 from the second pull and release test for pine projected on to the coordinate planes with PCA directions and their explained variance. 80
4.10 Displacement of F12 in the PC1 and PC2 directions over time for birch pull-and-release tests 3. 81
4.11 Displacement of F8 in the PC1 and PC2 directions over time for spruce pull-and-release tests 1. 82
4.12 Displacement of F8 in the PC1 and PC2 directions over time for pine pull-and-release tests 1. 82

4.13 PC1 displacement of stem features during pull-and-release test 3 and 1 for the birch and spruce respectively. The explained variance of the PC1 component for each feature is shown in parentheses in the legend. 83

4.14 PC1 displacement of representative stem and branch features during pull-and-release tests for the birch, spruce, and pine. The birch result shows stem feature F1 and branch feature F7 during test 1, the spruce result shows stem feature F3 and branch feature F10 during test 1, and the pine result shows stem feature F1 and branch feature F8 during test 1. For each feature, the first principal component captures the dominant direction of motion independently, with the explained variance fraction given in the legend. 84

4.15 Displacement of F12 across the tree pull-and-release tests in the PC1 direction over time for birch. The small vertical differences between curves arise from each trajectory having its own PCA transformation. 85

4.16 Displacement of F8 across the tree pull-and-release tests in the PC1 direction over time for spruce. Test 1 is offset in time because the recordings were not synchronised across tests. Tests 2 and 3 nearly coincide, making them hard to distinguish. 85

4.17 Displacement of F11 across the tree pull-and-release tests in the PC1 direction over time for pine. The small vertical differences between curves arise from each trajectory having its own PCA transformation. 86

4.18 Energy-weighted Welch power spectral density of the first principal component (PC1) across the three pull-and-release tests for the birch. Features F1-F5 are stem features while features F6-F12 are branch features. Each feature's PC1 signal is truncated to its usable length T^* prior to spectral estimation, suppressing noise-dominated frames. The explained variance of PC1 is shown for each feature. Ridgelines within each subplot correspond to individual tracked features where dominant frequency peaks are marked. Only successfully tracked features are shown. 88

4.19 Energy-weighted Welch power spectral density of the first principal component (PC1) across the three pull-and-release tests for the spruce. Features F1-F6 are stem features while features F7-F11 are branch features. Each feature's PC1 signal is truncated to its usable length T^* prior to spectral estimation, suppressing noise-dominated frames. The explained variance of PC1 is shown for each feature. Ridgelines within each subplot correspond to individual tracked features where dominant frequency peaks are marked. Only successfully tracked features are shown. 90

4.20 Energy-weighted Welch power spectral density of the first principal component (PC1) across the three pull-and-release tests for the pine. Feature F1 is a stem feature while feature F8 and F11 are branch features. Each feature's PC1 signal is truncated to its usable length T^* prior to spectral estimation, suppressing noise-dominated frames. The explained variance of PC1 is shown for each feature. Ridgelines within each subplot correspond to individual tracked features where dominant frequency peaks are marked. Only successfully tracked features are shown. 91
4.21 Displacement of branch feature F11 on the pine in the PC1 direction over time from test 1 with the fitted DHO curve shown in black. . . . 92
4.22 Displacement of stem feature F1 on the pine in the PC1 direction over time from test 3 with the fitted DHO curve shown in black. . . . 92
4.23 Fitted damped harmonic oscillator parameters for each successfully tracked feature along the first principal component (PC1) of birch. The upper panel shows the frequency f_0 [Hz] and the lower panel shows the damping coefficient δ on a logarithmic scale. Each marker represents a single pull-and-release test, with marker shape distinguishing tests and colour indicating the feature. The annotated value above each cluster is the mean across tests. The fit was performed on the signal truncated to the usable window T^*. 93
4.24 Fitted damped harmonic oscillator parameters for each successfully tracked feature along the first principal component (PC1) of spruce. The upper panel shows the frequency f_0 [Hz] and the lower panel shows the damping coefficient δ on a logarithmic scale. Each marker represents a single pull-and-release test, with marker shape distinguishing tests and colour indicating the feature. The annotated value above each cluster is the mean across tests. The fit was performed on the signal truncated to the usable window T^*. 94
4.25 Fitted damped harmonic oscillator parameters for each successfully tracked feature along the first principal component (PC1) of pine. The upper panel shows the frequency f_0 [Hz] and the lower panel shows the damping coefficient δ on a logarithmic scale. Each marker represents a single pull-and-release test, with marker shape distinguishing tests and colour indicating the feature. The annotated value above each cluster is the mean across tests. The fit was performed on the signal truncated to the usable window T^*. 95

List of Tables

4.1 Intrinsic calibration results for the three cameras. Values are reported as mean ± standard deviation across the five calibration datasets. . . 63
4.2 Sphere fitting results per scan where r_{opt} is the fitted radius and $r_{\text{true}} = 20.0$ mm is the measured radius. 64
4.3 Co-registration error defined as the Euclidean distance between fitted sphere centers from two independent scan positions. 65
4.4 Stereo reprojection errors for the pre- and post-experiment extrinsic calibration sets using intrinsic set 3. Errors are reported per camera pair together with the number of image frames used. 65
4.5 Pairwise differences between endpoint estimates. The upper part of the table reports the Euclidean distance $\|\Delta \boldsymbol{P}\|$ per endpoint and its mean ± standard deviation. The lower part of the table reports the signed component-wise differences, defined as the first sensor minus the second, expressed as $\bar{e} \pm \sigma$ [mm]. 68
4.6 Pairwise inter-point distances and signed differences between sensor estimates. The upper part reports the Euclidean distance $\|\cdot\|$ per pair per method. The lower part reports the signed differences, defined as the first sensor minus the second, expressed as $\bar{e} \pm \sigma$ [mm]. 70
4.7 Sensitivity of the video-robot tracking residual to the choice of intrinsic calibration set, evaluated using the post-experiment extrinsic calibration recording. $\bar{e}$ and σ denote the mean and standard deviation of the per-axis residual over the full recorded trajectory. 71
4.8 Sensitivity of the video-robot tracking residual to the choice of extrinsic calibration recording, evaluated using intrinsic calibration set 3. $\bar{e}$ and σ denote the mean and standard deviation of the per-axis residual over the full recorded trajectory. 71
4.9 Stereo reprojection errors for the tree experiment extrinsic calibration sets. Errors are reported per camera pair together with the number of image frames used. 72
4.10 Tracking success and camera visibility for each feature across the three pull-and-release tests per tree. ✓ indicates that a feature was tracked successfully throughout the full recording and × indicates that a tracking failed due to diverging or being associated with another feature in the last frame. The visibility column reports the average percentage of frames in successful runs where the feature was observed by 3, 2, 1 or 0 cameras respectively. 73

A.1 Detailed intrinsic calibration results for each camera and dataset. N denotes the number of images used after outlier rejection. RPE_{RMS} is the mean reprojection error. II

1
Introduction

Trees exposed to wind loading exhibit complex dynamic behaviour, most visibly in how they sway. Accurate measurement of this motion is necessary for understanding tree dynamics and improving mechanical and aerodynamic computational models. Such models are essential for assessing the risk of wind damage and have applications in urban forestry and forest management. Existing methods are limited by high instrumental cost, sparse spatial coverage, or an inability to resolve movement in all three spatial directions. This thesis presents a video-based method for reconstructing time-resolved three-dimensional tree motion from a calibrated multi-camera system. The method is first validated in a controlled setting, where reconstructed trajectories are compared against ground truth motion from an industrial robot arm and a terrestrial LiDAR laser scanner, before being applied to pull-and-release experiments on trees of the three species, spruce, pine and birch, to evaluate its performance under more realistic motion.

1.1 Background

When trees are subjected to wind, the resulting aerodynamic drag force generates bending moments, with the largest stresses occurring near the tree trunk. To reduce the risk of mechanical failures, trees respond by swaying and streamlining [1][2][3]. Streamlining refers to the reduction of wind load achieved through the reorientation of leaves and branches, which decreases the frontal area presented to the wind [2][3]. Tree sway can be modelled as a damped harmonic oscillator, characterised by a natural frequency and a damping ratio, describing the rate of oscillation and the dissipation of mechanical energy respectively [4][5][1]. While the amplitude of tree sway generally increases with wind speed, the natural frequency remains the same as it is a property inherent to the tree's physical characteristics, including height, architecture, stiffness, and mass distribution [1]. When the dominant frequency of wind excitation coincides with the natural frequency, resonance can occur, amplifying the sway response and the bending load near the stem [6][7][8].

Previous studies have shown a clear distinction in tree motion between conifers and broadleaved trees [5][1]. The motion of conifers is dominated by the main stem, which extends continuously from the base to the tree top, while the motion of broadleaved trees is more strongly governed by the crown structure, where large branches carry a significant portion of the total tree mass. These architectural differences imply that the dynamic response to wind loading differs between species,

motivating the evaluation of measurement methods across multiple species.

The natural frequency of a tree is a global property of its structure, determined by the stiffness, mass distribution and architecture of the whole tree [9][10].[1] However, this single-frequency description is a simplification, since individual branches have their own natural frequencies. It has been shown that primary branches in open-grown trees are tuned to nearly the same frequency as the whole tree [11]. This near-coincidence of frequencies is a prerequisite for the transfer of mechanical energy between the stem and branches, and leads to what has been described as multiple resonance damping, a mechanism in which the tree distributes wind energy across many coupled oscillating components rather than concentrating it at the stem, so that it is dissipated more effectively across the branches and crown [11]. The branches thus act somewhat like the tuned mass dampers used in engineering, where an auxiliary mass tuned to a structure's natural frequency absorbs its vibration, as installed in tall buildings to suppress wind-induced sway [4]. The tree, however, relies not on a single tuned absorber but on a whole hierarchy of branches whose frequencies overlap. The role of branches is further supported by [12], where changes in frequency and damping were observed only once more than 80% of the crown mass had been removed.

Various experimental methods have been used to measure tree motion. For a review of measurement instruments, see [13]. Measurements made in controlled environments typically involve exciting a tree through pull-and-release tests, where the resulting free oscillations are recorded. Measurements made outdoors typically involve monitoring tree motion over longer time periods under dynamic loading caused by natural wind conditions. Contact-based sensors, including strain gauges, inclinometers, accelerometers and load cells, are commonly installed on the tree stem and provide high temporal resolution [13]. However, each sensor measures only a single point, may require invasive installation and can be difficult to mount at height, so capturing motion at many points becomes costly and impractical. High-velocity terrestrial LiDAR offers a non-destructive alternative capable of capturing the full three-dimensional motion of a tree [14], but the cost of the required instrumentation limits its accessibility. Video-based methods are more accessible, non-destructive, and have been shown to produce frequency estimates in close agreement with accelerometers [15], which makes them an attractive route to spatially rich motion data.

1.2 Related Work

Most existing outdoor video-based studies rely on estimating tree motion using a single camera and are limited to estimating the natural frequency and damping, rather than reconstructing the time-resolved three-dimensional motion. Feature- and object-based tracking methods have been applied to estimate tree motion from

[1] Natural frequencies of trees have also been shown to be dependent on crown clashing between neighbouring trees [9][10].

video. The Kanade-Lucas-Tomasi (KLT) algorithm [16][17] tracks image features such as corners and textured regions across consecutive frames and has been used to estimate the natural frequencies of trees in controlled environments [18][19]. The combination of Shi-Tomasi corner detection [20] with KLT tracking has been applied to track a walnut tree under natural wind conditions [21], and the Minimum Output Sum of Squared Error (MOSSE) filter [22] has been used to estimate the natural frequency of a birch tree under natural wind conditions [23].

While these methods have demonstrated good frequency agreement with contact-based sensors, they operate on monocular recordings and do not reconstruct time-resolved three-dimensional motion. Recovering depth from a single camera view is not possible, as the image projection discards depth information. A calibrated stereo- or multi-camera setup is therefore required to accurately reconstruct three-dimensional motion from videos.[2] Feature-based methods rely on assumptions that inherently are difficult to satisfy in tree tracking, i.e. inter-frame displacements are assumed to be small, pixel intensities are assumed to remain constant and features are assumed to remain visible throughout the sequence. Trees present particular challenges in this regard. Their foliage produces highly repetitive textures that make feature identification ambiguous, rapid accelerations under wind gusts can violate the small-displacement assumption and cause tracker drift and partial occlusion from branches is common and unavoidable. Together, these factors make reliable long-term natural feature tracking on trees difficult in practice. Physical markers attached to trees address these issues by providing unambiguous, visually distinct targets with known correspondence across camera views throughout the recorded sequences.

Recovering the full three-dimensional structure of a tree from video would require a large number of simultaneously operating cameras to maintain sufficient coverage across all viewing angles. Studies on static crown reconstruction suggest that at least eight viewpoints are needed to obtain reliable estimates of crown geometry [25]. This motivates a sparse tracking approach, where a number of physically attached markers are tracked, rather than attempting to reconstruct the complete tree geometry.

Physical markers have been widely used in motion capture applications in biomechanics and structural engineering, where reliable point correspondence across views is essential for accurate three-dimensional reconstruction [26][27]. For tree motion, physical installation is a cost that markers and contact sensors share. The difference lies in what is placed on the tree. Each contact sensor is itself a measuring instrument fixed at a single point. A marker is a passive, inexpensive target, leaving the camera as the only measuring instrument. Many markers can therefore be placed and the three-dimensional motion can be recovered at many points from multiple views. In computer graphics, video-based marker-based motion capture has been

[2]Monocular depth estimation methods attempt to resolve this depth ambiguity from single images or video, but their errors in forestry applications exceed the tolerance required here, and they were therefore not considered [24].

used to estimate the dynamic properties of tree branches to generate realistic sway animations [28][29], though that work is animation-driven rather than aimed at providing physically validated displacement measurements.

Kalman filter-based tracking is well established in computer vision. Methods such as SORT [30] and DeepSORT [31] combine object detectors with a Kalman filter operating in two-dimensional image space using a constant velocity motion model, while AB3DMOT [32][33] extends this to three dimensions using objects detected from LiDAR point clouds. Moreover, learning-based methods such as [34], estimate three-dimensional point correspondences across multiple camera views through transformer-based updates but require training on synthetic data and a depth sensor for reliable results.

In contrast, the method presented in this thesis requires no training data and recovers three-dimensional positions from a calibrated multi-camera system. A fixed set of physical markers is attached to the tree and their positions are estimated over time using an Extended Kalman Filter [35] with a Singer acceleration model [36], which captures the oscillatory dynamics of wind-induced tree motion more faithfully than a constant velocity assumption. Kalman smoothing is subsequently applied to obtain trajectory estimates conditioned on the full observation sequence [35]. The resulting method is interpretable, requires no training data, and is designed to operate with a small number of cameras.

1.3 Purpose

The purpose of this project is to develop a cost-effective, non-destructive and reproducible video-based method for measuring wind-induced tree motion. The method aims to provide time-resolved three-dimensional motion data that can serve as input for future work in structural dynamics and computational modelling of wind-tree interaction.

1.4 Objectives

The overall objective is to develop and evaluate a multi-camera video-based method for reconstructing and analysing the three-dimensional motion of trees under controlled excitation. This is addressed through four objectives:

- Develop a motion reconstruction method based on physical marker tracking, camera calibration and state estimation from synchronised multi-camera recordings.
- Validate the method against ground truth motion from an industrial robot arm and independent LiDAR measurements and quantify reconstruction accuracy.
- Apply the method to pull-and-release experiments on three tree species, spruce, pine and birch to extract time-resolved three-dimensional motion of trunk and branch features.

- Estimate and compare dominant oscillation frequencies across marker locations and species and assess consistency across repeated experiments.

1.5 Limitations

The experiments are conducted in a controlled indoor environment using small trees excited through pull-and-release tests. The results are therefore not directly generalisable to mature trees under dynamic wind conditions outdoors, where factors such as greater distances, variable illumination, reduced marker visibility and more complex wind-induced excitation would pose additional challenges.

The method relies on physically attached markers, which must be installed on the tree prior to recording. This limits the spatial coverage of the reconstruction to a sparse set of chosen measurement points and makes deployment on mature trees in outdoor environments more demanding than in the controlled setting considered here.

Each species is represented by a single tree of a particular size and condition. The results are therefore indicative rather than statistically representative of the species.

The method was run entirely in post-processing, and its computational speed was not measured. Real-time performance is therefore neither demonstrated nor claimed. However, nothing in the approach fundamentally prevents an online approach. That would require replacing the fixed-interval smoother with for example a fixed-lag smoother that operates with a fixed shorter delay.

The method assumes calibrated cameras, i.e., cameras remain still once they have been calibrated, this is a condition that can be difficult to satisfy if one were to record in windy weather conditions.
The method assumes calibrated cameras that remain fixed after calibration. In this work, calibration was performed using a checkerboard procedure in the controlled indoor setting, where the fixed-camera assumption holds. Outdoors, however, this assumption is harder to guarantee, since wind can disturb the cameras during recording which would call for a calibration approach that continuously corrects for camera movement.

2

Theory

This chapter presents the theoretical foundation used to reconstruct three-dimensional motion from multi-camera video recordings and to analyse the resulting trajectories. Section 2.1 introduces the geometric framework of the pinhole camera model, lens distortion, the estimation of intrinsic and extrinsic camera parameters from checkerboard calibration and at last how a point in space can be triangulated using cameras. Section 2.2 describes the colour-based segmentation used to extract image-space observations of the tracked markers. Section 2.3 covers the multi-view motion tracking method which combines the Singer acceleration model, extended Kalman filtering and gated data association. Section 2.4 addresses a method used to approximate rigid body transformations from 3D point correspondences. Finally, Section 2.5 presents the methods used to characterise the reconstructed motion. These include the temporal alignment of trajectories recorded at different sampling rates, principal component analysis and two frequency estimation methods, the energy-weighted Welch power spectral density and a fit to a damped harmonic oscillator model.

2.1 Camera Geometry and Calibration

This section establishes the mathematical framework describing how a camera maps three-dimensional points in a scene onto a two-dimensional image plane and how the parameters governing this mapping are estimated in practice. Once these parameters are known, the same geometry can be inverted to recover three-dimensional positions from multiple views, which is the problem that triangulation addresses. The section begins by introducing projective geometry and homogeneous coordinates as the mathematical language needed to express perspective projection as a linear operation, following [37]. Building on this, the pinhole camera model is derived, relating points in the world coordinate system to their pixel coordinates through a sequence of transformations parametrised by the intrinsic and extrinsic camera parameters, following [37]. Lens distortion is then introduced as a correction to the idealised pinhole model, accounting for the deviations introduced by real camera lenses. Camera calibration follows, where the unknown parameters are estimated from observations of a planar checkerboard target, covering corner detection, intrinsic calibration and extrinsic calibration of the multi-view setup. The section concludes with triangulation, which recovers the three-dimensional position of a point from its corresponding image observations in two or more calibrated cameras. Together, these components form the geometric foundation on which the motion reconstruction in the following chapters is built.

2.1.1 Projective Geometry and Homogeneous Coordinates

In the pinhole camera model, to be presented in Section 2.1.4, the projection of a 3D point $(X, Y, Z) \in \mathbb{R}^3$ onto an image involves division by the depth $Z^{\mathcal{C}}$, which is a non-linear operation that cannot be expressed as a matrix multiplication. To express this operation as a linear transformation, we will make use of projective geometry.

The n-dimensional projective space $\mathbb{P}^n$ is defined as the set of equivalent classes of non-zero vectors in $\mathbb{R}^{n+1}$. Two vectors $\boldsymbol{u}, \boldsymbol{v} \in \mathbb{R}^{n+1} \setminus \{\mathbf{0}\}$ are said to be equivalent, which we write $\boldsymbol{u} \sim \boldsymbol{v}$, if there exists a non-zero scalar $\lambda \in \mathbb{R}$ such that,

$$\boldsymbol{u} = \lambda \boldsymbol{v}. \tag{2.1}$$

Geometrically, each equivalence class corresponds to a line through the origin in $\mathbb{R}^{n+1}$, so a point in $\mathbb{P}^n$ can be thought of as a direction in $\mathbb{R}^{n+1}$.

In the context of image formation, the geometric interpretation is that a single pixel observation in the two-dimensional image plane does not correspond to a unique point in three dimensional space, but rather an entire line of points, known as the viewing ray, which all project to the same image coordinate.

A point $\boldsymbol{x} = (x_1, \ldots, x_n) \in \mathbb{R}^n$ can be represented in $\mathbb{P}^n$ by appending 1 as an additional coordinate,

$$\tilde{\boldsymbol{x}} = \begin{bmatrix} x_1 \\ \vdots \\ x_n \\ 1 \end{bmatrix} \in \mathbb{P}^n. \tag{2.2}$$

This representation is known as the point's homogeneous form and will be denoted by $\tilde{\cdot}$. Since $\tilde{\boldsymbol{x}} \sim \lambda \tilde{\boldsymbol{x}}$ for any non-zero $\lambda \in \mathbb{R}$, scaling a homogeneous vector does not change the point it represents in $\mathbb{R}^n$.

Similarly, a vector $\tilde{\boldsymbol{x}} = (x_1, \ldots, x_n, x_{n+1}) \in \mathbb{P}^n$ can be converted back to Euclidean coordinates through dehomogenization,

$$\tilde{\boldsymbol{x}} = \begin{bmatrix} x_1 \\ \vdots \\ x_n \\ x_{n+1} \end{bmatrix} = x_{n+1} \begin{bmatrix} \frac{x_1}{x_{n+1}} \\ \vdots \\ \frac{x_n}{x_{n+1}} \\ 1 \end{bmatrix} \sim \begin{bmatrix} \frac{x_1}{x_{n+1}} \\ \vdots \\ \frac{x_n}{x_{n+1}} \\ 1 \end{bmatrix} \implies \boldsymbol{x} = \begin{bmatrix} \frac{x_1}{x_{n+1}} \\ \vdots \\ \frac{x_n}{x_{n+1}} \end{bmatrix}, \tag{2.3}$$

assuming $x_{n+1} \neq 0$.

2.1.2 World and Camera Coordinate Systems

Two coordinate systems are needed to describe how a three-dimensional scene is related to the camera that observes it. Namely, a world coordinate system in which

the geometry of the scene is defined and a local camera coordinate system attached to each camera in the setup.

Let the world coordinate system be a right-handed Cartesian coordinate system, $\{\boldsymbol{e}_x, \boldsymbol{e}_y, \boldsymbol{e}_z\}$. A point in this system will be denoted by,

$$\boldsymbol{X} = (X, Y, Z) \in \mathbb{R}^3, \tag{2.4}$$

or in homogeneous coordinates,

$$\tilde{\boldsymbol{X}} = (X, Y, Z, 1) \in \mathbb{P}^3. \tag{2.5}$$

Bold capital letters will henceforth be used to denote points in $\mathbb{R}^3$ and $\mathbb{P}^3$. This coordinate system serves as a global reference frame in which the geometry of the scene is described.

A camera in a scene defines its own right-handed Cartesian coordinate system, $\{\boldsymbol{e}_x^{\mathcal{C}}, \boldsymbol{e}_y^{\mathcal{C}}, \boldsymbol{e}_z^{\mathcal{C}}\}$. The origin is located at the camera's optical center, which is known as its camera center, and is denoted by,

$$\boldsymbol{C}^{\mathcal{C}} = (0, 0, 0) \in \mathbb{R}^3. \tag{2.6}$$

The z-axis is aligned with the optical axis of the camera, the direction the camera is facing, while the x- and y-axis are aligned with the horizontal and vertical directions of the image plane respectively. A point expressed in the camera coordinate system is denoted by,

$$\boldsymbol{X}^{\mathcal{C}} = (X^{\mathcal{C}}, Y^{\mathcal{C}}, Z^{\mathcal{C}}) \in \mathbb{R}^3. \tag{2.7}$$

Henceforth, points expressed in the camera coordinate system will be denoted using $\cdot^{\mathcal{C}}$.

2.1.3 Rigid Body Transformation

The transformation from world coordinates $\boldsymbol{X}$ to camera coordinates $\boldsymbol{X}^{\mathcal{C}}$ is a rigid body transformation which consists of a rotation $R \in SO(3)$ and a translation $t \in \mathbb{R}^3$,

$$\boldsymbol{X}^{\mathcal{C}} = R\boldsymbol{X} + \boldsymbol{t}, \tag{2.8}$$

where $SO(3)$ denotes the Special Orthogonal group whose elements satisfy $R^T R = I$ and $\det(R) = 1$. The parameters R, t are known as extrinsic parameters as they define the camera's pose, i.e. position and orientation, relative to the world coordinate system. The rigid body transformation is illustrated in Figure 2.1.

Making use of homogeneous coordinates, we may write (2.8) on matrix form,

$$\boldsymbol{X}^{\mathcal{C}} = \begin{bmatrix} R & \boldsymbol{t} \end{bmatrix} \tilde{\boldsymbol{X}}, \tag{2.9}$$

where $\begin{bmatrix} R & \boldsymbol{t} \end{bmatrix} \in \mathbb{R}^{3\times 4}$. Notice that $\boldsymbol{X}^{\mathcal{C}} \in \mathbb{R}^3$ in (2.9) is expressed in Euclidean coordinates and that the last element of $\tilde{\boldsymbol{X}}$ is 1.

This matrix form will prove convenient when composing multiple transformations to relate several cameras to a common coordinate system, as to be described in Section 2.1.7.3.

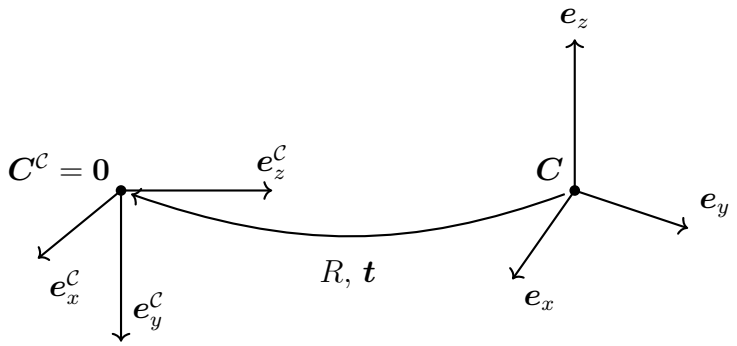

Figure 2.1: The rigid body transformation between the world coordinate system $\{\boldsymbol{e}_x, \boldsymbol{e}_y, \boldsymbol{e}_z\}$ centred at $\boldsymbol{C}$ and the camera coordinate system $\{\boldsymbol{e}_x^{\mathcal{C}}, \boldsymbol{e}_y^{\mathcal{C}}, \boldsymbol{e}_z^{\mathcal{C}}\}$ centred at $\boldsymbol{C}^{\mathcal{C}}$.

2.1.4 The Pinhole Camera Model

The most widely used mathematical model for image formation is the pinhole camera model. It provides an idealised geometric model where light rays are assumed to pass through a single point, the camera center, and then intersect an image plane, where the projection is formed. The model ignores lens distortion and provides a geometric description of perspective projection.

2.1.4.1 Perspective Projection

In the camera coordinate system, with the camera center as the origin, the image plane is the plane $Z^{\mathcal{C}} = f$ where $f \in \mathbb{R}^+$ is known as the focal length. A projection of a 3D point $\boldsymbol{X}^{\mathcal{C}} = (X^{\mathcal{C}}, Y^{\mathcal{C}}, Z^{\mathcal{C}}) \in \mathbb{R}^3$ onto the image plane is formed by intersecting the line passing through the camera center, $\boldsymbol{C}^{\mathcal{C}} = \mathbf{0}$ and $\boldsymbol{X}^{\mathcal{C}}$ with the image plane at $Z^{\mathcal{C}} = f$ as illustrated in Figure 2.2. The line is known as the viewing ray and can be parametrised as,

$$\boldsymbol{r}(s) = s\boldsymbol{X}^{\mathcal{C}}, \quad s \in \mathbb{R}. \tag{2.10}$$

Using that the z-component of (2.10) is f at the image plane we get,

$$sZ^{\mathcal{C}} = f \implies s = \frac{f}{Z^{\mathcal{C}}}, \tag{2.11}$$

assuming $Z^{\mathcal{C}} \neq 0$. Substituting (2.11) into (2.10), the projected image point $\boldsymbol{x} = (x, y) \in \mathbb{R}^2$ becomes,

$$\boldsymbol{x} = \begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} f\dfrac{X^{\mathcal{C}}}{Z^{\mathcal{C}}} \\ f\dfrac{Y^{\mathcal{C}}}{Z^{\mathcal{C}}} \end{bmatrix}. \tag{2.12}$$

Small bold letters will henceforth be used to denote points in $\mathbb{R}^2$ and $\mathbb{P}^2$.

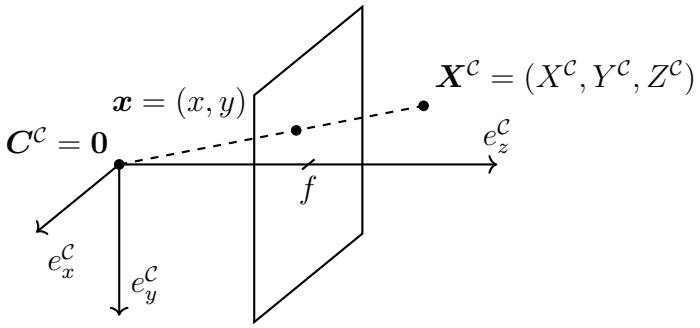

Figure 2.2: Perspective projection in the camera coordinate system. The 3D point $\boldsymbol{X}^{\mathcal{C}}$ is projected along the viewing ray onto the image plane $Z^{\mathcal{C}} = f$ producing the image point $\boldsymbol{x}$.

2.1.4.2 Pixel Coordinate System

Images from real cameras measure points in pixel coordinates. The pixel coordinate system has its origin in the upper left corner of the image and the u- and v-axis pointing horizontally and vertically respectively. A point in pixel coordinates will be denoted,

$$\boldsymbol{x}^{\text{px}} = (u, v) \in \mathbb{R}^2, \tag{2.13}$$

where we treat pixel coordinates as continuous for modelling purposes.

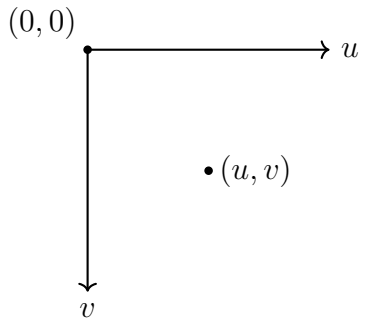

Figure 2.3: Pixel coordinate system with the origin in the upper left corner.

2.1.4.3 Normalised Image Coordinates

The perspective projection derived in Section 2.1.4.1 maps a 3D point in camera coordinates $\boldsymbol{X}^{\mathcal{C}} = (X^{\mathcal{C}}, Y^{\mathcal{C}}, Z^{\mathcal{C}})$ to a point on the image plane at distance f from the camera center. This mapping depends on the focal length and therefore on the specific camera.

To separate the geometric projection from camera-specific parameters, we introduce normalised image coordinates. These are obtained by considering a canonical image plane located at unit distance from the camera center, i.e. by letting $f = 1$ in equation (2.12). The projection in normalised image coordinates becomes,

$$\boldsymbol{x}^{\mathrm{n}} = \begin{bmatrix} x^{\mathrm{n}} \\ y^{\mathrm{n}} \end{bmatrix} = \begin{bmatrix} \frac{X^{\mathcal{C}}}{Z^{\mathcal{C}}} \\ \frac{Y^{\mathcal{C}}}{Z^{\mathcal{C}}} \end{bmatrix} \in \mathbb{R}^2, \tag{2.14}$$

or in homogeneous coordinates,

$$\tilde{\boldsymbol{x}}^{\mathrm{n}} = \begin{bmatrix} \frac{X^{\mathcal{C}}}{Z^{\mathcal{C}}} \\ \frac{Y^{\mathcal{C}}}{Z^{\mathcal{C}}} \\ 1 \end{bmatrix} \sim \begin{bmatrix} X^{\mathcal{C}} \\ Y^{\mathcal{C}} \\ Z^{\mathcal{C}} \end{bmatrix} \in \mathbb{P}^2. \tag{2.15}$$

Thus, normalised image coordinates correspond to the projection of a 3D point onto a unit image plane and is independent of the camera's inner geometry, known as intrinsic parameters.

2.1.4.4 The Intrinsic Calibration Matrix

The pinhole camera model only describes the projection onto a continuous image plane. However, real cameras measure image points in discrete pixel coordinates. The mapping between normalised image coordinates and pixel coordinates is defined by the intrinsic parameters of a camera. The intrinsic parameters encode the focal length and the location of the principal point and are fixed properties of the camera hardware.

The transformation from normalised homogeneous coordinates, $\tilde{\boldsymbol{x}}^{\mathrm{n}} = (x^{\mathrm{n}}, y^{\mathrm{n}}, 1) \in \mathbb{P}^2$, to homogeneous pixel coordinates, $\tilde{\boldsymbol{x}}^{\mathrm{px}} = (u, v, 1) \in \mathbb{P}^2$ is a linear invertible mapping described by the intrinsic calibration matrix $K \in \mathbb{R}^{3\times3}$,

$$\tilde{\boldsymbol{x}}^{\mathrm{px}} = K\tilde{\boldsymbol{x}}^{\mathrm{n}}, \tag{2.16}$$

where,

$$K = \begin{bmatrix} f_x & 0 & c_x \\ 0 & f_y & c_y \\ 0 & 0 & 1 \end{bmatrix}, \tag{2.17}$$

where (c_x, c_y) is the principal point, i.e. the pixel coordinates of the point where the optical axis intersects the image plane, and f_x, f_y are focal lengths expressed in units of pixels. The mapping is illustrated in Figure 2.4. The principal point ideally coincides with the image center but is often slightly offset due to manufacturing imprecision.

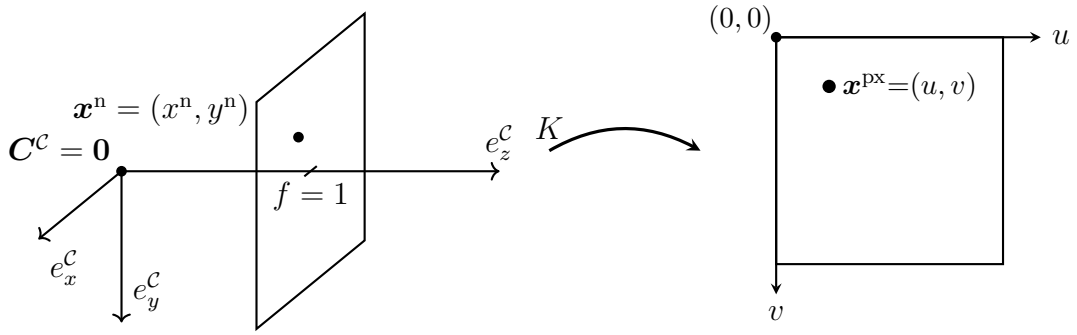

Figure 2.4: The intrinsic matrix K mapping from normalised image coordinates to pixel coordinates.

2.1.4.5 Camera Matrix Formulation

Using homogeneous coordinates, the perspective projection can be expressed as a linear mapping up to scale. Combining the relation,

$$\tilde{\boldsymbol{x}}^{\mathrm{n}} \sim \boldsymbol{X}^{\mathcal{C}}, \tag{2.18}$$

and the rigid body transformation (2.9) we obtain,

$$\lambda \tilde{\boldsymbol{x}}^{\mathrm{n}} = \begin{bmatrix} R & \boldsymbol{t} \end{bmatrix} \tilde{\boldsymbol{X}}, \tag{2.19}$$

where $\lambda = Z^{\mathcal{C}}$ is the depth of the point in the camera coordinate system. The equation expresses the mapping from a 3D point in world coordinates to its normalised image coordinates in homogeneous form.

By incorporating the intrinsic calibration matrix K presented in Section 2.1.4.4, which maps normalised coordinates to pixel coordinates, we obtain the full pinhole camera projection equation,

$$\lambda \tilde{\boldsymbol{x}}^{\mathrm{px}} = K \begin{bmatrix} R & \boldsymbol{t} \end{bmatrix} \tilde{\boldsymbol{X}}. \tag{2.20}$$

2.1.5 Lens Distortion Model

The pinhole camera model presented in Section 2.1.4 assumes linear projections, i.e. straight lines in the world projects as straight lines in the image. Most cameras do not have a pinhole, but a lens, which produces non-linear effects known as lens distortions. These effects cause deviations from the ideal projection and must be accounted for in accurate camera models.

Lens distortion is modelled as a non-linear transformation applied to normalised image coordinates before the mapping to pixel coordinates. We define the distortion function,

$$\boldsymbol{x}^{\mathrm{d}} = f(\boldsymbol{x}^{\mathrm{n}}, \boldsymbol{d}), \tag{2.21}$$

where $\boldsymbol{x}^{\mathrm{n}} \in \mathbb{R}^2$ is a point in normalised image coordinates, $\boldsymbol{x}^{\mathrm{d}} \in \mathbb{R}^2$ is a point in distorted normalised coordinates and $\boldsymbol{d}$ denotes the vector of distortion parameters.

Various mathematical models exist for lens distortion. This work adopts Brown's distortion model [38], which decomposes distortion into radial and tangential components and is implemented in OpenCV [39]. In this work, the five parameter model is used where higher order coefficients are set to zero.

Radial distortion displaces points along the radial direction from the image center, causing the barrel or pincushion effect. Brown's radial distortion model is,

$$x^{\mathrm{rd}} = x^{\mathrm{n}} \left(1 + k_1 r^2 + k_2 r^4 + k_3 r^6\right), \tag{2.22}$$

$$y^{\mathrm{rd}} = y^{\mathrm{n}} \left(1 + k_1 r^2 + k_2 r^4 + k_3 r^6\right), \tag{2.23}$$

where k_1, k_2, k_3 are radial distortion coefficients and,

$$r = \sqrt{\left(x^{\mathrm{n}}\right)^2 + \left(y^{\mathrm{n}}\right)^2}. \tag{2.24}$$

Tangential distortion is caused by the fact that the lens is not parallel to the image plane and shifts points perpendicular to the radial direction. We model it using,

$$x^{\mathrm{td}} = 2p_1 x^{\mathrm{n}} y^{\mathrm{n}} + p_2 \left(r^2 + 2\left(x^{\mathrm{n}}\right)^2\right), \tag{2.25}$$

$$y^{\mathrm{td}} = 2p_2 x^{\mathrm{n}} y^{\mathrm{n}} + p_1 \left(r^2 + 2\left(y^{\mathrm{n}}\right)^2\right), \tag{2.26}$$

where p_1, p_2 are tangential distortion coefficients. The distortion function is then defined by summing the radial and tangential components,

$$\boldsymbol{x}^{\mathrm{d}} = \begin{bmatrix} x^{\mathrm{rd}} + x^{\mathrm{td}} \\ y^{\mathrm{rd}} + y^{\mathrm{td}} \end{bmatrix} = f(\boldsymbol{x}^{\mathrm{n}}, \boldsymbol{d}), \tag{2.27}$$

where the distortion parameter vector is given by,

$$\boldsymbol{d} = (k_1, k_2, k_3, p_1, p_2). \tag{2.28}$$

2.1.6 Camera Projection Model

The complete mapping from a 3D world point to a 2D point in pixel coordinates is described by the sequence of transformations,

$$\boldsymbol{X} \xrightarrow{R,\boldsymbol{t}} \boldsymbol{X}^{\mathcal{C}} \xrightarrow{/Z^{\mathcal{C}}} \boldsymbol{x}^{\mathrm{n}} \xrightarrow{f(\cdot,\boldsymbol{d})} \boldsymbol{x}^{\mathrm{d}} \xrightarrow{K} \boldsymbol{x}^{\mathrm{px}}, \tag{2.29}$$

or explicitly by,

$$\boldsymbol{X}^{\mathcal{C}} = \begin{bmatrix} R & \boldsymbol{t} \end{bmatrix} \tilde{\boldsymbol{X}} \tag{2.30}$$

$$\tilde{\boldsymbol{x}}^{\mathrm{n}} = \frac{\boldsymbol{X}^{\mathcal{C}}}{Z^{\mathcal{C}}} \tag{2.31}$$

$$\boldsymbol{x}^{\mathrm{d}} = f(\boldsymbol{x}^{\mathrm{n}}, \boldsymbol{d}) \tag{2.32}$$

$$\tilde{\boldsymbol{x}}^{\mathrm{px}} = K\tilde{\boldsymbol{x}}^{\mathrm{d}} \tag{2.33}$$

For our convenience, we will denote the full projection model by the function $\pi(\cdot)$, defined as,

$$\boldsymbol{x}^{\mathrm{px}} = \pi(K, \boldsymbol{d}, R, \boldsymbol{t}, \boldsymbol{X}). \tag{2.34}$$

2.1.7 Camera Calibration

The process of estimating the parameters of the camera projection model, summarised in Section 2.1.6, is known as camera calibration. The intrinsic parameters, which consist of the calibration matrix, K, as presented in Section 2.1.4.4 and the distortion coefficients, $\boldsymbol{d}$, as presented in Section 2.1.5 and describe the internal geometry of a camera and remain fixed once they have been calibrated. The extrinsic parameters describe the relative pose between cameras, i.e. positions and orientations in a multi-camera scene.

Calibration methods can broadly be divided into targetless and target-based methods. Target-based calibration, adopted in this work, uses calibration objects with precisely known geometry. The most common choice is a planar checkerboard pattern of alternating black and white squares, as illustrated in Figure 2.5, whose inner corners provide unambiguous 2D-3D point correspondences. This work uses the OpenCV implementation of camera calibration, which is based on the method of [40][41]. The method assumes that the calibration target is planar and observed from multiple unknown orientations and estimates the camera parameters by minimising the reprojection error across all observed corner correspondences.

In this work, intrinsic and extrinsic calibration are performed in two sequential steps. These steps are described in sections 2.1.7.2 and 2.1.7.3 respectively.

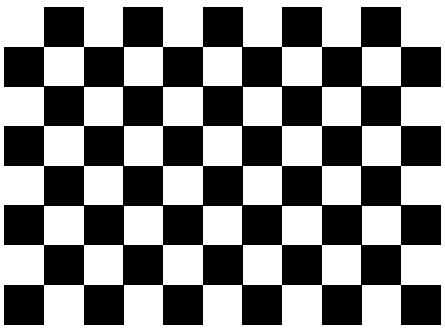

Figure 2.5: Example of a checkerboard pattern with 10×7 inner corners.

2.1.7.1 Corner Detection

To estimate intrinsic or extrinsic parameters, one must be able to establish 2D-3D point correspondences. For a checkerboard calibration, the intersection point of the squares, the inner points, in a single image can be found with a corner detection algorithm.

Because the checkerboard has a known grid size of $N_c \times N_r$ inner corners with a known physical spacing $s \in \mathbb{R}^+$ between adjacent corners, each detected corner $\boldsymbol{x}_{jk}^{\text{px}}$ in image j can be assigned a unique 3D position $\boldsymbol{X}_k$ in the coordinate system of the calibration target. The origin is placed at one corner of the grid, and the target is taken to lie in the plane $Z = 0$, which means,

$$\boldsymbol{X}_k = \begin{bmatrix} sc_k \\ sr_k \\ 0 \end{bmatrix}, \; c_k \in \{0, \ldots, N_c - 1\}, \; r_k \in \{0, \ldots, N_r - 1\}, \tag{2.35}$$

where c_k and r_k are the column and row indices of corner k. To avoid orientation 180° rotation ambiguities, we let the number of squares along one direction be even and along the other direction be odd.

2.1.7.2 Intrinsic Calibration

Intrinsic calibration is the process of estimating the calibration matrix K and distortion coefficients $\boldsymbol{d}$ of a single camera. This is achieved by observing a calibration target across J images and exploiting the known 3D geometry of the target.

Let a calibration target provide a set of K_p known 3D corner positions,

$$\left\{\boldsymbol{X}_k\right\}_{k=1}^{K_p}, \; \boldsymbol{X}_k \in \mathbb{R}^3, \tag{2.36}$$

defined in the target coordinate system. For each of the J calibration images, let the corresponding set of detected pixel coordinates be,

$$\left\{\boldsymbol{x}_{jk}^{\text{px}}\right\}_{k=1}^{K_p}, \; j = 1, \ldots, J, \; \boldsymbol{x}_{jk}^{\text{px}} \in \mathbb{R}^2. \tag{2.37}$$

The intrinsic parameters K and $\boldsymbol{d}$ are shared across all images, while each image j has its own extrinsic parameters $(R_j, \boldsymbol{t}_j)$ describing the pose of the target relative to the camera due to the fact that the checkerboard moves across images.

Using the camera projection model $\pi(\cdot)$ from Section 2.1.6, the predicted pixel coordinate of point k in image j is,

$$\hat{\boldsymbol{x}}_{jk}^{\text{px}} = \pi(K, \, \boldsymbol{d}, \, R_j, \, \boldsymbol{t}_j, \, \boldsymbol{X}_k) \, . \tag{2.38}$$

The reprojection error is then defined as the Euclidean pixel distance between the detection and the prediction,

$$RPE = \left\| \boldsymbol{x}_{jk}^{\text{px}} - \hat{\boldsymbol{x}}_{jk}^{\text{px}} \right\| . \tag{2.39}$$

The intrinsic calibration problem is then formulated as the non-linear least-squares problem,

$$\min_{K,\,\boldsymbol{d},\,\{R_j,\boldsymbol{t}_j\}_{j=1}^{J}} \sum_{j=1}^{J}\sum_{k=1}^{K_p} \left\| \boldsymbol{x}_{jk}^{\text{px}} - \hat{\boldsymbol{x}}_{jk}^{\text{px}} \right\|^2, \tag{2.40}$$

which minimises the total reprojection error across all point correspondences and all images. We define the error of the intrinsic calibration as the root mean squared (RMS) reprojection error,

$$RPE_{RMS} = \sqrt{\frac{1}{JK_p}\sum_{j=1}^{J}\sum_{k=1}^{K_p} \left\| \boldsymbol{x}_{jk}^{\text{px}} - \hat{\boldsymbol{x}}_{jk}^{\text{px}} \right\|^2}. \tag{2.41}$$

2.1.7.3 Extrinsic Calibration

Due to the loss of depth in the projection process, as described in Section 2.1.4 a single camera cannot recover the 3D structure of a scene. To enable 3D reconstruction from multiple views, it is necessary to know the relative pose between cameras, which is estimated through extrinsic calibration. In this work, cameras are assumed to be stationary and their intrinsic parameters K_i, $\boldsymbol{d}_i$, are assumed known from the intrinsic calibration as presented in Section 2.1.7.2.

2.1.7.3.1 Stereo Calibration We first consider the two-camera case, known as stereo calibration, before extending to the general multi-camera setting. Then the goal is to estimate the rigid body transformation from the camera coordinate system of camera 1 to the coordinate system of camera 2,

$$\boldsymbol{X}^{\mathcal{C}_2} = \begin{bmatrix} R_{21} & \boldsymbol{t}_{21} \end{bmatrix} \tilde{\boldsymbol{X}}^{\mathcal{C}_1}, \tag{2.42}$$

where $\tilde{\boldsymbol{X}}^{\mathcal{C}_1} \in \mathbb{P}^3$ and $\boldsymbol{X}^{\mathcal{C}_2} \in \mathbb{R}^3$ is a point in camera 1 and camera 2 local coordinate systems respectively and $R_{21} \in SO(3)$ and $\boldsymbol{t}_{21} \in \mathbb{R}^3$ describe the pose of camera 2 relative to camera 1 as illustrated in Figure 2.6.

Assume a calibration target with K_p known 3D positions is observed simultaneously by both cameras across J images. Let,

$$\left\{\boldsymbol{x}_{ijk}^{\text{px}}\right\}_{k=1}^{K_p},\ i \in \{1,2\},\ j = 1,\ldots,J,\ \boldsymbol{x}_{ijk}^{\text{px}} \in \mathbb{R}^2, \tag{2.43}$$

denote the detected pixel coordinates in camera i, image j, for point k. For each image j, the target has pose $(R_j, \boldsymbol{t}_j)$ relative to camera 1. The predicted projection of point k into camera 1 in image j is,

$$\hat{\boldsymbol{x}}_{1jk}^{\text{px}} = \pi(K_1,\, \boldsymbol{d}_1,\, R_j,\, \boldsymbol{t}_j,\, \boldsymbol{X}_k)\,. \tag{2.44}$$

For camera 2, the same point first undergoes the target to camera 1 transformation and then the camera 1 to camera-2 transformation,

$$\boldsymbol{X}^{\mathcal{C}_1} = \begin{bmatrix} R_j & \boldsymbol{t}_j \end{bmatrix} \tilde{\boldsymbol{X}}_k, \tag{2.45}$$

$$\boldsymbol{X}^{\mathcal{C}_2} = \begin{bmatrix} R_{21} & \boldsymbol{t}_{21} \end{bmatrix} \tilde{\boldsymbol{X}}^{\mathcal{C}_1} = \begin{bmatrix} R_{21}R_j & R_{21}\boldsymbol{t}_j + \boldsymbol{t}_{21} \end{bmatrix} \tilde{\boldsymbol{X}}_k. \tag{2.46}$$

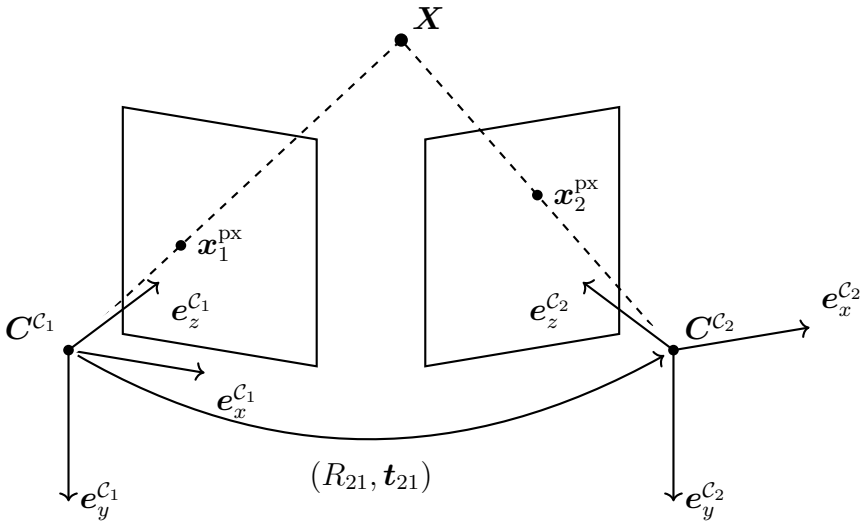

Figure 2.6: Stereo calibration setup. Two cameras with optical centers $\boldsymbol{C}^{\mathcal{C}_1}$ and $\boldsymbol{C}^{\mathcal{C}_2}$ simultaneously observe a 3D point $\boldsymbol{X}$, which projects to image coordinates $\boldsymbol{x}_1^{\text{px}}$ and $\boldsymbol{x}_2^{\text{px}}$ respectively. The rigid body transformation $(R_{21}, \boldsymbol{t}_{21})$ describes the pose of camera 2 relative to camera 1 and is the quantity estimated during stereo calibration.

The predicted projection to camera 2 is then given by,

$$\hat{\boldsymbol{x}}_{2jk}^{\text{px}} = \pi(K_2,\, \boldsymbol{d}_2,\, R_{21}R_j,\, R_{21}\boldsymbol{t}_j + \boldsymbol{t}_{21},\, \boldsymbol{X}_k)\,. \tag{2.47}$$

The stereo calibration problem is then formulated as a non-linear least squares problem,

$$\min_{R_{21},\, \boldsymbol{t}_{21},\, \{R_j, \boldsymbol{t}_j\}_{j=1}^{J}} \sum_{i=1}^{2} \sum_{j=1}^{J} \sum_{k=1}^{K_p} \left\| \boldsymbol{x}_{ijk}^{\text{px}} - \hat{\boldsymbol{x}}_{ijk}^{\text{px}} \right\|^2, \tag{2.48}$$

with the objective of minimising the reprojection error between all point correspondences across all images. The stereo calibration error is quantified by the root mean squared (RMS) reprojection error,

$$RPE_{RMS} = \sqrt{\frac{1}{2JK_p} \sum_{i=1}^{2} \sum_{j=1}^{J} \sum_{k=1}^{K_p} \left\| \boldsymbol{x}_{ijk}^{\text{px}} - \hat{\boldsymbol{x}}_{ijk}^{\text{px}} \right\|^2}. \tag{2.49}$$

2.1.7.3.2 Multi-Camera Composition When calibrating a scene with $N > 2$ cameras, stereo calibration can be applied to each adjacent camera pair $(i, i+1)$, $i = 1, \ldots, N-1$, yielding the pairwise relative pose $(R_{i+1,i},\, \boldsymbol{t}_{i+1,i})$ such that,

$$\boldsymbol{X}^{\mathcal{C}_{i+1}} = \begin{bmatrix} R_{i+1,i} & \boldsymbol{t}_{i+1,i} \end{bmatrix} \tilde{\boldsymbol{X}}^{\mathcal{C}_i}. \tag{2.50}$$

Camera 1 is chosen as the reference frame, defining the world camera coordinate system,

$$R_1 = I_3,\ \boldsymbol{t}_1 = \boldsymbol{0}. \tag{2.51}$$

The pose of camera $i+1$ relative to this world frame is then obtained recursively by composition as previously seen in equations (2.45)-(2.46),

$$R_{i+1} = R_{i+1,i} R_i, \tag{2.52}$$
$$\boldsymbol{t}_{i+1} = R_{i+1,i} \boldsymbol{t}_i + \boldsymbol{t}_{i+1,i}, \tag{2.53}$$

for $i = 1, ..., N-1$. The resulting set $\{(R_i, \boldsymbol{t}_i)\}_{i=1}^{N}$ is then all camera poses in the common camera world coordinate system.

2.1.8 Triangulation

Given multiple calibrated cameras observing the same scene, triangulation is the process of recovering the three-dimensional position of a feature from its corresponding image observations.

Following [42], assume that a 3D point $\boldsymbol{X} \in \mathbb{R}^3$ is observed in N cameras with the pixel locations $\boldsymbol{x}_i^{\text{px}} \in \mathbb{R}^2$ and with known intrinsic and extrinsic parameters, K_i, $\boldsymbol{d}_i$, R_i, $\boldsymbol{t}_i$, $i = 1, ..., N$. Since the complete camera model includes intrinsic scaling and lens distortion, the pixel observations are first transformed into normalised image coordinates,

$$\boldsymbol{x}_i^{\text{n}} = f^{-1}\left(K_i^{-1} \tilde{\boldsymbol{x}}_i^{\text{px}}, \boldsymbol{d}_i\right), \tag{2.54}$$

where f^{-1} is the inverse distortion mapping as presented in Section 2.1.5. In normalised image coordinates, the projection equation reduces to

$$\lambda_i \tilde{\boldsymbol{x}}_i^{\text{n}} = \begin{bmatrix} R_i & \boldsymbol{t}_i \end{bmatrix} \tilde{\boldsymbol{X}}, \; i = 1, \ldots, N, \tag{2.55}$$

where $\tilde{\boldsymbol{X}} \in \mathbb{P}^3$ denotes the homogeneous representation of the 3D point and $\lambda_i = Z_i^{\mathcal{C}} \in \mathbb{R}^+$ is the depth of the point in camera i.

Since both sides of (2.55) represent the same point in $\mathbb{P}^2$, their cross product is zero,

$$\tilde{\boldsymbol{x}}_i^{\text{n}} \times \left(\begin{bmatrix} R_i & \boldsymbol{t}_i \end{bmatrix} \tilde{\boldsymbol{X}} \right) = \boldsymbol{0}. \tag{2.56}$$

This eliminates the unknown scale λ_i and yields a linear equation in $\tilde{\boldsymbol{X}}$. Denoting the j-th row of $\begin{bmatrix} R_i & \boldsymbol{t}_i \end{bmatrix}$ by $m_{i,j}^T \in \mathbb{R}^{1\times 4}$ and writing $\tilde{\boldsymbol{x}}_i^{\text{n}} = (x_i^{\text{n}}, y_i^{\text{n}}, 1) \in \mathbb{P}^2$, expanding (2.56) yields two linearly independent equations per camera,

$$\begin{bmatrix} x_i^{\text{n}} m_{i,3}^T - m_{i,1}^T \\ y_i^{\text{n}} m_{i,3}^T - m_{i,2}^T \end{bmatrix} \tilde{\boldsymbol{X}} = \boldsymbol{0}. \tag{2.57}$$

Stacking the contributions from all N cameras gives the homogeneous linear system,

$$A\tilde{\boldsymbol{X}} = \boldsymbol{0}, \; A = \begin{bmatrix} x_1^{\text{n}} m_{1,3}^T - m_{1,1}^T \\ y_1^{\text{n}} m_{1,3}^T - m_{1,2}^T \\ \vdots \\ x_n^{\text{n}} m_{n,3}^T - m_{n,1}^T \\ y_n^{\text{n}} m_{n,3}^T - m_{n,2}^T \end{bmatrix} \in \mathbb{R}^{2N\times 4}. \tag{2.58}$$

Since $\tilde{\boldsymbol{X}} \in \mathbb{P}^3$ has 3 degrees of freedom and each camera contributes 2 independent equations, the system is determined for $N \geq 2$ cameras. In absence of measurement noise, the viewing rays intersect and a solution exists to (2.58). However, in real applications, the triangulated point can be estimated by solving [42],

$$\min_{\tilde{\boldsymbol{X}}} \left\| A\tilde{\boldsymbol{X}} \right\|^2 \text{ s.t } \left\| \tilde{\boldsymbol{X}} \right\| = 1. \tag{2.59}$$

The constraint avoids the trivial solution $\tilde{\boldsymbol{X}} = \mathbf{0}$ and reflects the fact that homogeneous coordinates are defined up to scale. The solution is given by the right singular vector of A corresponding to the smallest singular value, obtained through singular value decomposition (SVD). The homogeneous point $\tilde{\boldsymbol{X}} \in \mathbb{P}^3$ is at last converted to Euclidean coordinates $\boldsymbol{X} \in \mathbb{R}^3$ through dehomogenisation as described in Section 2.1.1.

2.2 Image Segmentation

In this work, the objects that will be tracked will have distinct colours and will serve as visual features. A colour-based image segmentation method will be used to isolate these features in each video-frame, thus producing pixel observations from segmentation centers. The segmentation pipeline consists of colour-based thresholding followed by refinements using morphological operations.

2.2.1 Colour Space Representation

The HSV (Hue, Saturation, Value) colour space represents colours using three channels. Hue encodes the colour wavelength, saturation its intensity and value its brightness [37]. Compared to the RGB colour space, HSV separates chromatic information from illumination, making it more robust to variations in lighting conditions and therefore better suited for colour-based segmentation. In particular, a colour of interest occupies a compact and predictable region in hue, regardless of changes in scene illumination that primarily affect the V channel.

Following the OpenCV [39] convention used throughout this work, the channels take values in the ranges,

$$H \in [0, 179],\ S \in [0, 255],\ V \in [0, 255], \tag{2.60}$$

where hue is expressed on a half-degree scale covering the full 360° colour wheel, as illustrated in figure 2.7. Note that red hues wrap around the hue axis and therefore occupy two disjoint intervals at opposite ends of the range.

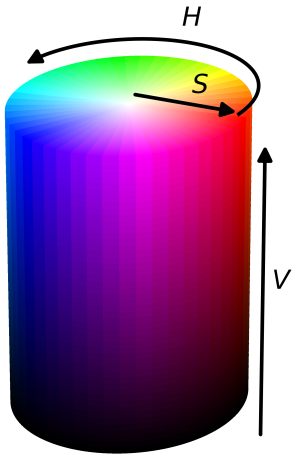

Figure 2.7: The HSV colour space represented as a cylinder. Hue encodes the colour type along the angular axis, saturation its intensity along the radial axis, and value its brightness along the vertical axis.

2.2.2 Colour-Based Thresholding

Let,

$$f : \Omega \subset \mathbb{Z}^2 \to \mathbb{R}^3, \tag{2.61}$$

denote an HSV image where each pixel $\boldsymbol{x}^{\text{px}} \in \Omega$ is associated with the three-channel value,

$$f(\boldsymbol{x}^{\text{px}}) = (f_H(\boldsymbol{x}^{\text{px}}), f_S(\boldsymbol{x}^{\text{px}}), f_V(\boldsymbol{x}^{\text{px}}))\,, \tag{2.62}$$

where,

$$f_H : \Omega \to \mathbb{R}, \tag{2.63}$$
$$f_S : \Omega \to \mathbb{R}, \tag{2.64}$$
$$f_V : \Omega \to \mathbb{R}, \tag{2.65}$$

denotes the hue, saturation and value channels respectively as described in Section 2.2.1. Note that an image is denoted in discrete pixel coordinates in contrast with Section 2.1. Given some lower and upper HSV bound,

$$l = (l_H, l_S, l_V), \tag{2.66}$$
$$u = (u_H, u_S, u_V), \tag{2.67}$$

a binary image,

$$f_b : \Omega \subset \mathbb{Z}^2 \to \{0, 1\}, \tag{2.68}$$

can be obtained by thresholding each HSV channel independently,

$$f_b(\boldsymbol{x}^{\text{px}}) = \begin{cases} 1, & l_c \leq f_c(\boldsymbol{x}^{\text{px}}) \leq u_c, \quad \forall c \in \{H, S, V\}, \\ 0, & \text{otherwise.} \end{cases} \tag{2.69}$$

Pixels with value 1 are said to belong to the foreground and pixels with value 0 to the background. For the colour red, that wrap around the hue axis, the condition on H can be replaced by a union of two intervals.

2.2.3 Morphological Operations

To transform the binary masks obtained from the colour-based thresholding procedure introduced in Section 2.2.2 into refined segmentations, morphological operations are applied to remove noise and fill small holes within segmented regions. Morphological operations operate on binary images and are based on the local neighbourhood structure of foreground pixels [37].

A structuring element $s : \mathbb{Z}^2 \to \{0, 1\}$ is a small binary pattern that defines a local neighbourhood around each pixel, such as a $n \times n$ square. Its size is the number of pixels it contains,

$$|s| = \sum_{\boldsymbol{x}^{\mathrm{px}}} s(\boldsymbol{x}^{\mathrm{px}}). \tag{2.70}$$

For each pixel $\boldsymbol{x}^{\mathrm{px}} \in \Omega$, define the local foreground count image $c : \Omega \to \{0, 1, \ldots, |s|\}$ as,

$$c = f_b * s, \tag{2.71}$$

where $*$ denotes convolution. At each pixel location, $c(\boldsymbol{x}^{\mathrm{px}})$ gives the number of foreground pixels in the neighbourhood of $\boldsymbol{x}^{\mathrm{px}}$ defined by s. The morphological operations are then defined pointwise as,

$$\mathrm{dilate}(f_b, s)(\boldsymbol{x}^{\mathrm{px}}) = \begin{cases} 1 & \text{if } c(\boldsymbol{x}^{\mathrm{px}}) \geq 1, \\ 0 & \text{otherwise,} \end{cases} \tag{2.72}$$

$$\mathrm{erode}(f_b, s)(\boldsymbol{x}^{\mathrm{px}}) = \begin{cases} 1 & \text{if } c(\boldsymbol{x}^{\mathrm{px}}) = |s|, \\ 0 & \text{otherwise,} \end{cases} \tag{2.73}$$

with the composite operations,

$$\mathrm{open}(f_b, s) = \mathrm{dilate}(\mathrm{erode}(f_b, s),\ s)\,, \tag{2.74}$$

$$\mathrm{close}(f_b, s) = \mathrm{erode}(\mathrm{dilate}(f_b, s),\ s)\,. \tag{2.75}$$

Dilation sets a pixel to 1 if at least one pixel in its neighbourhood is foreground, thereby expanding foreground regions. Erosion sets a pixel to 1 only if every pixel in its neighbourhood is foreground, thereby shrinking foreground regions. Opening, which applies erosion followed by dilation, removes small foreground objects and smooths boundaries. Closing, which applies dilation followed by erosion, fills small holes and gaps within foreground regions.

2.2.4 Connected Components and Centroid Computation

Given a binary image, $f_b : \Omega \to \{0, 1\}$, foreground pixels form connected regions corresponding to segmented objects. Connected-component labelling (CCL) is the process of assigning a unique integer label to each connected foreground region [43], meaning that two foreground pixels are considered part of the same region if they are horizontally, vertically or diagonally adjacent in discrete pixel space.

Assume that the connected-component labelling problem has been solved, yielding a label image,

$$L : \Omega \to \{0, 1, \ldots, N_{\text{seg}}\}, \tag{2.76}$$

where $L(\boldsymbol{x}^{\text{px}}) = 0$, denotes background pixels and $L(\boldsymbol{x}^{\text{px}}) = k$ denotes that pixel $\boldsymbol{x}^{\text{px}} \in \Omega$ belongs to connected region k. The segmented region Ω_k is therefore defined as,

$$\Omega_k = \left\{\boldsymbol{x}^{\text{px}} \in \Omega : L(\boldsymbol{x}^{\text{px}}) = k\right\}, \; k = 1, \ldots, N_{\text{seg}}, \tag{2.77}$$

where N_{seg} denotes the number of connected foreground regions. The segmentation center associated with region Ω_k can then be computed as the centroid of its foreground pixels,

$$\boldsymbol{x}_k^{\text{px}} = \frac{1}{|\Omega_k|} \sum_{\boldsymbol{x}^{\text{px}} \in \Omega_k} \boldsymbol{x}^{\text{px}}, \tag{2.78}$$

where $|\Omega_k|$ denotes the number of pixels contained in the segmented region. The centroid $\boldsymbol{x}_k^{\text{px}}$ provides a 2D image observation associated with the segmented object.

2.3 Multi-View Motion Tracking

Multi-view motion tracking concerns the estimation of time-varying three-dimensional feature trajectories from multiple synchronised camera observations. Given calibrated cameras and image measurements of the same scene, the problem combines multi-view geometry with recursive state estimation to deal with noisy measurements and missing data.

The framework considered in this work models the temporal evolution of a feature using a stochastic dynamical system, while image observations are related to the hidden state through a non-linear camera projection model. State estimation is performed using an Extended Kalman Filter (EKF) to handle the non-linearity of the camera projection model.

The following sections present the theoretical components underlying the tracking framework. The state-space model and Singer acceleration model together define the process model, describing how the hidden state over a track is assumed to evolve. The camera measurement model then relates the hidden state to image observations while the Extended Kalman Filter (EKF) provides the recursive state estimation framework. Since multiple features are tracked simultaneously, data association handles the correspondence between observations and tracks. Finally, as tracking is performed offline on recorded sequences, fixed-interval smoothing is introduced to incorporate future measurements and refine past state estimates. The integration of these components into a complete tracking method is described in Section 3.2.

2.3.1 State-Space Models

The temporal evolution of a moving feature can be modelled as a discrete-time stochastic dynamical system. Without such a model, each feature's position would be estimated independently at every frame by triangulation alone, with no link between consecutive frames. This leaves the estimate sensitive to measurement noise and undefined whenever a feature is occluded or fails to be detected. A state-space model instead describes how the feature is expected to move over time, which allows past estimates to be propagated through missing measurements and combined with new observations to suppress noise.

A temporal model is also what makes it possible to track several features at once. When multiple features are present, the observations in a given frame carry no inherent label indicating which feature produced which detection. By predicting where each feature is expected to appear, the model provides a basis for deciding which observation belongs to which feature, the problem of data association addressed in Section 2.3.3. Estimating each frame in isolation offers no such prediction and therefore no principled way to maintain the identity of a feature over time.

Let $t \in \mathbb{R}$ denote continuous time and let $T > 0$ denote the sampling interval between two consecutive observations. Equivalently,the samples may be indexed by an integer $k = t/T$, as is standard in discrete-time filtering. We retain continuous time t with sampling interval T throughout. The hidden state at time t is represented by the state vector,

$$x(t) \in \mathbb{R}^n, \tag{2.79}$$

while the corresponding measurement vector is denoted by,

$$y(t) \in \mathbb{R}^m. \tag{2.80}$$

For the tracking framework considered in this work, the state process is assumed to be linear, while the measurement process is non-linear. The dynamical system is therefore described by,

$$x(t+T) = Fx(t) + \bar{w}(t), \tag{2.81}$$
$$y(t) = h(x(t)) + \bar{v}(t), \tag{2.82}$$

where $F \in \mathbb{R}^{n\times n}$ is the state transition matrix, $h : \mathbb{R}^n \to \mathbb{R}^m$ is a non-linear measurement function, $\bar{w}(t) \in \mathbb{R}^n$ is the process noise and $\bar{v}(t) \in \mathbb{R}^m$ is the measurement noise. The process and measurement noise are assumed to be mutually independent and uncorrelated across time steps.

The process noise represents uncertainty in the assumed motion model and is modelled as a zero-mean Gaussian random vector,

$$\bar{w}(t) \sim \mathcal{N}(0, Q), \tag{2.83}$$

where $Q \in \mathbb{R}^{n \times n}$ denotes the process noise covariance matrix. Similarly, the measurement noise is modelled as a zero-mean Gaussian random vector,

$$\bar{v}(t) \sim \mathcal{N}(0, R), \tag{2.84}$$

where $R \in \mathbb{R}^{m \times m}$ denotes the measurement covariance matrix.

2.3.1.1 Singer Acceleration Model

The Singer acceleration model was originally developed for tracking manoeuvring targets [36]. Unlike the constant acceleration model, which treats deviations from constant acceleration as independent random impulses at each time step, the Singer model explicitly accounts for the temporal correlation of acceleration by modelling it as a continuous stochastic process. Specifically, the acceleration is assumed to follow a zero-mean Gauss-Markov process, meaning it evolves smoothly over time rather than changing instantaneously.

For a three-dimensional motion model, the state vector is defined as,

$$x(t) = (x_1(t), x_2(t), x_3(t), \dot{x}_1(t), \dot{x}_2(t), \dot{x}_3(t), \ddot{x}_1(t), \ddot{x}_2(t), \ddot{x}_3(t)) \in \mathbb{R}^9. \tag{2.85}$$

Under the assumption that the acceleration is a Gauss-Markov process, the acceleration evolves according to the continuous-time stochastic differential equation,

$$\frac{d}{dt}\ddot{x}_i = -\alpha \ddot{x}_i + w_i(t), \;\; i = 1, 2, 3, \tag{2.86}$$

where $\alpha \in \mathbb{R}^+$ is the inverse correlation time and $w_i(t)$, $i = 1, 2, 3$, is zero-mean white noise with spectral density $2\alpha\sigma_m^2$. The parameter σ_m^2 denotes the steady-state acceleration variance. The discrete-time state transition matrix is then,

$$F = \begin{bmatrix} I_3 & TI_3 & \frac{1}{\alpha^2}\left(e^{-\alpha T} - 1 + \alpha T\right) I_3 \\ 0_3 & I_3 & \frac{1}{\alpha}\left(1 - e^{-\alpha T}\right) I_3 \\ 0_3 & 0_3 & e^{-\alpha T} I_3 \end{bmatrix} \in \mathbb{R}^{9 \times 9}, \tag{2.87}$$

where $I_3, 0_3 \in \mathbb{R}^{3 \times 3}$ denotes the identity and null matrices. The process noise covariance matrix is then,

$$Q = 2\alpha\sigma_m^2 \begin{bmatrix} q_{11} I_3 & q_{12} I_3 & q_{13} I_3 \\ q_{12} I_3 & q_{22} I_3 & q_{23} I_3 \\ q_{13} I_3 & q_{23} I_3 & q_{33} I_3 \end{bmatrix} \in \mathbb{R}^{9 \times 9}, \tag{2.88}$$

where,

$$q_{11} = \frac{1}{2\alpha^5}\left(1 - e^{-2\alpha T} + 2\alpha T + \frac{2\alpha^3 T^3}{3} - 2\alpha^2 T^2 - 4\alpha T e^{-\alpha T}\right), \tag{2.89}$$

$$q_{12} = \frac{1}{2\alpha^4}\left(e^{-2\alpha T} + 1 - 2e^{-\alpha T} + 2\alpha T e^{-\alpha T} - 2\alpha T + \alpha^2 T^2\right), \tag{2.90}$$

$$q_{13} = \frac{1}{2\alpha^3}\left(1 - e^{-2\alpha T} - 2\alpha T e^{-\alpha T}\right), \tag{2.91}$$

$$q_{22} = \frac{1}{2\alpha^3}\left(4e^{-\alpha T} - 3 - e^{-2\alpha T} + 2\alpha T\right), \tag{2.92}$$

$$q_{23} = \frac{1}{2\alpha^2}\left(e^{-2\alpha T} + 1 - 2e^{-\alpha T}\right), \tag{2.93}$$

$$q_{33} = \frac{1}{2\alpha}\left(1 - e^{-2\alpha T}\right), \tag{2.94}$$

as derived by [36].

The parameter $\alpha > 0$ is the inverse correlation time of the acceleration process, with $1/\alpha$ defining the timescale over which the acceleration remains correlated. Small values of α correspond to slowly varying, persistent acceleration, while large values cause the acceleration to decorrelate rapidly, approaching uncorrelated noise. In the limiting case $\alpha \to 0$, the state transition matrix F reduces to that of the constant acceleration model.

2.3.1.2 Camera Measurement Model

In the multi-view motion tracking framework, the hidden state describes the three-dimensional motion of a feature, while the measurements are two-dimensional pixel observations from the cameras. The measurement model therefore maps the 3D position component of the state to image coordinates using the camera projection model.

Given a scene with N cameras with known intrinsic and extrinsic parameters, K_i, $\boldsymbol{d}_i$, R_i, $\boldsymbol{t}_i$, $i = 1, .., N$. Assuming each camera tracks some point in pixel space, which we denote by $y_i(t) = (u_i(t), v_i(t)) \in \mathbb{R}^2$ for cameras $i = 1, .., N$, we have that the entire measurement vector is given by,

$$y(t) = \begin{bmatrix} y_1(t) \\ \vdots \\ y_n(t) \end{bmatrix} \in \mathbb{R}^{2N}. \tag{2.95}$$

Let $\pi_i(K_i, \boldsymbol{d}_i, R_i, \boldsymbol{t}_i, \boldsymbol{X})$ denote the full projection model from a world 3D point $\boldsymbol{X}$ to the pixel point $\boldsymbol{x}^{\text{px}} \in \mathbb{R}^2$ for each camera $i = 1, .., N$ as summarised in Section 2.1.6. Given that,

$$\boldsymbol{X} = \begin{bmatrix} I_3 & 0_3 & 0_3 \end{bmatrix} x(t), \tag{2.96}$$

we define the camera measurement function by stacking the camera projection mod-

els,

$$h(x(t)) = \begin{bmatrix} \pi_1(K_i, \boldsymbol{d}_i, R_i, \boldsymbol{t}_i, \begin{bmatrix} I_3 & 0_3 & 0_3 \end{bmatrix} x(t)) \\ \vdots \\ \pi_n(K_n, \boldsymbol{d}_n, R_n, \boldsymbol{t}_n, \begin{bmatrix} I_3 & 0_3 & 0_3 \end{bmatrix} x(t)) \end{bmatrix} \in \mathbb{R}^{2N}. \tag{2.97}$$

It follows for the measurement covariance matrix that $R \in \mathbb{R}^{2N \times 2N}$.

Since the projection model is non-linear, the measurement function is linearised for use in the Extended Kalman Filter to be introduced in Section 2.3.2. The Jacobian of the measurement function is,

$$H(t) = \begin{bmatrix} H_1(t) \\ \vdots \\ H_N(t) \end{bmatrix} = \begin{bmatrix} \frac{\partial h_1}{\partial x} \\ \vdots \\ \frac{\partial h_N}{\partial x} \end{bmatrix} \in \mathbb{R}^{2N \times 9}, \tag{2.98}$$

where,

$$H_i(t) = \frac{\partial h_i}{\partial x} \in \mathbb{R}^{2 \times 9}. \tag{2.99}$$

Since the projection model only depends on the position, we get,

$$H_i(t) = \begin{bmatrix} \frac{\partial h_i}{\partial \boldsymbol{X}} & 0_{2\times 6} \end{bmatrix}, \tag{2.100}$$

with,

$$\frac{\partial h_i}{\partial \boldsymbol{X}} = \begin{bmatrix} \frac{\partial u_i}{\partial x_1} & \frac{\partial u_i}{\partial x_2} & \frac{\partial u_i}{\partial x_3} \\ \frac{\partial v_i}{\partial x_1} & \frac{\partial v_i}{\partial x_2} & \frac{\partial v_i}{\partial x_3} \end{bmatrix}. \tag{2.101}$$

2.3.2 Extended Kalman Filter

The Kalman filter (KF) provides a recursive framework for estimating the hidden state of a dynamical system from noisy measurements [44]. For linear systems with Gaussian process and measurement noise, the Kalman filter yields the minimum mean-square error estimate of the hidden state.

In the tracking framework, to be presented in Section 3.2, the state dynamics are linear, while the camera measurement model is non-linear due to the perspective projection and lens distortion model. The Extended Kalman Filter (EKF) extends the Kalman filter to non-linear systems by linearising the measurement function around the current predicted state estimate [35].

Consider the discrete-time dynamical system introduced in Section 2.3.1, defined by the equations (2.81)-(2.82). Let,

$$\hat{x}(t \mid t) \in \mathbb{R}^n, \tag{2.102}$$

denote the filtered state estimate at time $t \in \mathbb{R}$ conditioned on all measurements up to and including time step t. Similarly,

$$\hat{x}(t \mid t-T) \in \mathbb{R}^n, \tag{2.103}$$

denotes the predicted state estimate at time t prior to incorporating the measurement at time step t. The corresponding covariance matrices are denoted by,

$$P(t \mid t) \in \mathbb{R}^{n \times n}, \tag{2.104}$$

and,

$$P(t \mid t-T) \in \mathbb{R}^{n \times n}, \tag{2.105}$$

which quantify the uncertainty of the filtered and predicted state estimates, respectively.

2.3.2.1 Prediction

The prediction step propagates the state estimate and covariance forward in time using the process model,

$$\hat{x}(t \mid t-T) = F\hat{x}(t-T \mid t-T), \tag{2.106}$$

and,

$$P(t \mid t-T) = FP(t-T \mid t-T)F^T + Q. \tag{2.107}$$

where $F \in \mathbb{R}^{n \times n}$ is the state transition matrix and $Q \in \mathbb{R}^{n \times n}$ is the process noise covariance matrix introduced in Section 2.3.1.

2.3.2.2 Linearisation

Since the measurement function h is non-linear, it is linearised around the predicted state estimate,

$$H(t) = \left.\frac{\partial h}{\partial x}\right|_{x=\hat{x}(t|t-T)} \in \mathbb{R}^{m \times n}. \tag{2.108}$$

2.3.2.3 Update

Given a measurement vector $y(t) \in \mathbb{R}^m$, the innovation $r(t) \in \mathbb{R}^m$ is defined as,

$$r(t) = y(t) - h(\hat{x}(t \mid t-T)). \tag{2.109}$$

The innovation covariance $S(t) \in \mathbb{R}^{m \times m}$ is then,

$$S(t) = H(t)P(t \mid t-T)H^T(t) + R, \tag{2.110}$$

where $R \in \mathbb{R}^{m \times m}$ is the measurement noise covariance matrix as introduced in Section 2.3.1. The Kalman gain $K(t) \in \mathbb{R}^{n \times m}$ is then computed according to,

$$K(t) = P(t \mid t-T)H(t)^T S^{-1}(t). \tag{2.111}$$

The filtered state estimate becomes,

$$\hat{x}(t \mid t) = \hat{x}(t \mid t-T) + K(t)r(t), \tag{2.112}$$

and the covariance matrix is updated using the Joseph-form covariance update,

$$P(t \mid t) = (I_n - K(t)H(t))\, P(t \mid t-T)\, (I_n - K(t)H(t))^T + K(t)RK^T(t). \tag{2.113}$$

2.3.3 Data Association

The Extended Kalman Filter introduced in Section 2.3.2 assumes that incoming measurements can be associated with the corresponding predicted object states. In practical multi-object tracking problems, however, multiple observations may be present simultaneously, while some objects may become temporarily occluded or fail to generate measurements. Therefore, the correspondence between predicted states and observations must be estimated simultaneously. This problem is generally referred to as the data association problem [35].

Let,

$$\hat{x}_j(t \mid t-T) \in \mathbb{R}^n, \tag{2.114}$$

denote the predicted state estimate of some track j at time $t \in \mathbb{R}$ and let,

$$y_i(t) \in \mathbb{R}^m, \tag{2.115}$$

denote an observation where $i = 1, .., M$ and M is the number of observations. In this work, data association is performed independently at each time step, using only the observations available in the current frame. Each step therefore reduces to a single assignment problem matching the M observations of one frame to the existing tracks, rather than a search over association hypotheses spanning several frames.

Consider some measurement model as described in Section 2.3.1 equation (2.82), each predicted state induces a predicted measurement,

$$\hat{y}_j(t) = h(\hat{x}_j(t \mid t-T)) \in \mathbb{R}^m. \tag{2.116}$$

The objective of data association is to determine which observations are most consistent with the predicted measurements. This can be formulated as an optimisation problem based on a distance metric between observations and predicted measurements as described in the following subsections.

2.3.3.1 Measurement Gating

Before solving the assignment problem, introduced in Section 2.3.3.2, unlikely observation track pairings can be rejected using statistical gating. The purpose of gating is to reduce the number of candidate associations and suppress measurements that are inconsistent with the predicted state estimate.

Given a predicted measurement $\hat{y}_j(t) \in \mathbb{R}^m$ and an observation $y_i(t) \in \mathbb{R}^m$, the innovation $r_{ij}(t) \in \mathbb{R}^m$ is defined as,

$$r_{ij}(t) = y_i(t) - \hat{y}_j(t). \tag{2.117}$$

The corresponding innovation covariance $S_j(t) \in \mathbb{R}^{m \times m}$ is,

$$S_j(t) = H_j(t) P_j(t \mid t - T) H_j^T(t) + R, \tag{2.118}$$

where $H_j(t)$ is the measurement Jacobian evaluated at the predicted state estimate.

The statistical consistency between the observation and predicted measurement can then be quantified using the squared Mahalanobis distance,

$$d_{ij}^2(t) = r_{ij}^T(t) S_j^{-1}(t) r_{ij}(t). \tag{2.119}$$

Under the assumption that the innovation $r_{ij}(t)$ is Gaussian distributed, the squared Mahalanobis distance follows a chi-squared distribution with m degrees of freedom,

$$d_{ij}^2(t) \sim \chi_m^2. \tag{2.120}$$

This allows the gating threshold γ to be chosen by selecting a significance level $p \in (0, 1)$ and setting,

$$\gamma = F_{\chi_m^2}^{-1}(p), \tag{2.121}$$

where $F_{\chi_m^2}^{-1}$ denotes the inverse cumulative distribution function of the χ_m^2 distribution. This presumes a correctly specified filter, so that $S_j(t)$ is the true innovation covariance. As the model adopted here is only approximate, γ is best treated as a design parameter for which $F_{\chi_m^2}^{-1}(p)$ provides a principled starting value. Candidate associations satisfying,

$$d_{ij}^2(t) \leq \gamma, \tag{2.122}$$

are then considered valid candidates for assignment while pairs exceeding the threshold are rejected.

2.3.3.2 Assignment Problem

After gating, the remaining candidate associations are used to construct a cost matrix,

$$C(t) \in \mathbb{R}^{M \times N}, \tag{2.123}$$

where N denotes the number of predicted object states and M the number of observations. Each matrix element, $C_{ij}(t)$, represents the assignment cost between predicted state j and observation i. A common choice is to use the squared Mahalanobis distance,

$$C_{ij}(t) = d_{ij}^2(t), \tag{2.124}$$

as introduced in equation (2.119), while invalid associations outside the gating region are assigned a cost of ∞.

The data association problem can then be formulated as a linear assignment problem, where the objective is to minimise the total assignment cost under the constraint that each observation is assigned to at most one object state and each object state is assigned to at most one observation. The resulting optimisation problem can be solved using the modified Jonker-Volgenant algorithm as presented in [45].

2.3.4 Fixed-Interval Smoothing

The Extended Kalman Filter presented in Section 2.3.2 estimates the hidden state recursively forward in time. The filtered estimate at time t therefore depends only on measurements up to and including time t, while later measurements, which also contains information about the state at t, are left unused. When the measurements following time t are available, they can be incorporated to refine the estimate at t, a process known as smoothing. The variants of smoothing differ in how much of the future they use. A fixed-lag smoother delays each estimate by a fixed interval and uses only the measurements within that window, allowing it to operate in near-real-time [35]. Fixed-interval smoothing instead conditions each estimate on the entire measurement sequence and therefore requires the complete recording [35].

As tracking in this work is performed offline on recorded sequences, the full measurement sequence is available, and fixed-interval smoothing is adopted. Specifically, a forward-backward smoother is used, which combines a forward filtering pass with a backward filtering pass to obtain state estimates conditioned on the complete sequence [35].

Let $\hat{x}^F(t \mid t) \in \mathbb{R}^n$ and $P^F(t \mid t) \in \mathbb{R}^{n \times n}$ denote the forward filtered state estimate and covariance obtained from the EKF conditioned on measurements up to time t.

The backward filter is constructed by defining the prior covariance $\Pi(t) \in \mathbb{R}^{n \times n}$, which satisfies,

$$\Pi(t+T) = F\Pi(t)F^T + Q. \tag{2.125}$$

The backward transition matrix is then,

$$F^B(t) = \Pi(t)F^T\Pi(t+T)^{-1}, \tag{2.126}$$

and the corresponding backward process noise covariance is,

$$Q^B(t) = \Pi(t) - \Pi(t)F^T\Pi(t+T)^{-1}F\Pi(t). \tag{2.127}$$

The backward dynamical model is defined as,

$$x(t) = F^B(t)x(t+T) + \bar{w}^B(t+T), \tag{2.128}$$
$$y(t) = H(t)x(t) + \bar{v}(t), \tag{2.129}$$

where $\bar{w}^B(t+T) \sim \mathcal{N}(0, Q^B(t))$ and $H(t) \in \mathbb{R}^{m\times n}$ is the measurement Jacobian from equation (2.108), evaluated at the forward predicted state estimate. Running a Kalman filter backwards in time on this model produces the backward state estimate $\hat{x}^B(t \mid t+T) \in \mathbb{R}^n$ and covariance $P^B(t \mid t+T) \in \mathbb{R}^{n\times n}$, conditioned on measurements from future time steps.

The forward and backward estimates are combined to obtain the smoothed covariance,

$$P(t \mid t_f) = \left(\left(P^F(t \mid t)\right)^{-1} + \left(P^B(t \mid t+T)\right)^{-1}\right)^{-1}, \tag{2.130}$$

and the smoothed state estimate,

$$\hat{x}(t \mid t_f) = P(t \mid t_f)\left(\left(P^F(t \mid t)\right)^{-1}\hat{x}^F(t \mid t) + \left(P^B(t \mid t+T)\right)^{-1}\hat{x}^B(t \mid t+T)\right), \tag{2.131}$$

where t_f denotes the final time step of the measurement sequence. The smoothed estimate therefore incorporates information from both the forward and backward passes, yielding improved state estimates compared to the forward filter alone.

The forward-backward smoother is exact only for linear-Gaussian systems, so its application to the EKF is approximate. Here the approximation is confined to the measurement model, since the process model is linear and the backward dynamics are therefore exact. The backward measurement update is linearised around the forward predicted state estimate, so the smoother inherits the same linearisation as the forward filter rather than introducing a new one.

2.4 Coordinate Frame Alignment

When comparing spatial measurements from sensors of different kinds, it is necessary to express each sensor's data in a common coordinate frame. In the experiment presented in Section 3.3, a world coordinate system is established to which the data from three sensors are registered, namely a set of calibrated cameras, a robot arm and a terrestrial LiDAR scanner. A dedicated registration method is used for each sensor, with the camera and LiDAR registrations presented in sections 3.3.4.2 and 3.3.4.3 respectively.

The following section presents a least-squares method for estimating the rigid body transformation between two coordinate systems given a set of corresponding points. In this work, it is applied specifically to register the robot coordinate system to the world frame as described in Section 3.3.4.1.

2.4.1 Point Set Registration

Given two sets of corresponding points expressed in different coordinate frames, the goal is to estimate the rigid body transformation that best aligns one set to the other in the least-squares sense. Let,

$$\left\{\boldsymbol{X}_k^{\mathcal{A}}\right\}_{k=1}^{M}, \tag{2.132}$$

and,

$$\left\{\boldsymbol{X}_k^{\mathcal{B}}\right\}_{k=1}^{M}, \tag{2.133}$$

denote M corresponding points expressed in frames $\mathcal{A}$ and $\mathcal{B}$ respectively. The transformation from $\mathcal{B}$ to $\mathcal{A}$ is estimated by solving,

$$\min_{R,\,\boldsymbol{t}} \sum_{k=1}^{M} \left\| \boldsymbol{X}_k^{\mathcal{A}} - \begin{bmatrix} R & \boldsymbol{t} \end{bmatrix} \tilde{\boldsymbol{X}}_k^{\mathcal{B}} \right\|^2 \text{ s.t. } R \in SO(3). \tag{2.134}$$

Following [46], the rotation and translation can be decoupled by working in mean-centred coordinates. Let,

$$\bar{\boldsymbol{X}}^{\mathcal{A}} = \frac{1}{M} \sum_{k=1}^{M} \boldsymbol{X}_k^{\mathcal{A}}, \tag{2.135}$$

$$\bar{\boldsymbol{X}}^{\mathcal{B}} = \frac{1}{M} \sum_{k=1}^{M} \boldsymbol{X}_k^{\mathcal{B}}, \tag{2.136}$$

be the centroids of the two point sets, and define the mean-centred residuals,

$$\boldsymbol{Q}_k^{\mathcal{A}} = \boldsymbol{X}_k^{\mathcal{A}} - \bar{\boldsymbol{X}}^{\mathcal{A}}, \tag{2.137}$$

$$\boldsymbol{Q}_k^{\mathcal{B}} = \boldsymbol{X}_k^{\mathcal{B}} - \bar{\boldsymbol{X}}^{\mathcal{B}}. \tag{2.138}$$

This reduces the problem to the Orthogonal Procrustes problem,

$$\min_{R \in SO(3)} \sum_{k=1}^{M} \left\| \boldsymbol{Q}_k^{\mathcal{A}} - R\,\boldsymbol{Q}_k^{\mathcal{B}} \right\|^2. \tag{2.139}$$

The optimal rotation is found by maximising $\operatorname{tr}(RH)$, where $H \in \mathbb{R}^{3\times 3}$ is the cross-covariance matrix,

$$H = \sum_{k=1}^{M} \boldsymbol{Q}_k^{\mathcal{B}} \left(\boldsymbol{Q}_k^{\mathcal{A}}\right)^T. \tag{2.140}$$

Computing the singular value decomposition $H = U\Lambda V^T$, The optimal rotation is given by [46],

$$R = VU^T, \tag{2.141}$$

provided $\det(VU^T) = +1$, which ensures a proper rotation. The optimal translation is then recovered as,

$$t = \bar{\boldsymbol{X}}^{\mathcal{A}} - R\,\bar{\boldsymbol{X}}^{\mathcal{B}}. \tag{2.142}$$

2.5 Post-processing of Reconstructed Trajectories

This section presents the methods used to analyse the reconstructed trajectories. Temporal alignment is introduced first, addressing the synchronisation of signals recorded asynchronously at different sampling rates, which is required for quantitative sensor comparison. Principal component analysis is then presented as a means of extracting the dominant direction of motion from three-dimensional trajectory data. Finally, two methods for estimating the frequency content of a signal are described, the energy-weighted Welch power spectral density and a fit to a damped harmonic oscillator model.

2.5.1 Temporal Alignment

When comparing spatial trajectories recorded by different sensors, the signals are generally sampled asynchronously and at different rates. Before any point-wise comparison can be made, the trajectories must therefore be brought to a common time. In this work, this is achieved in two steps. Linear interpolation is used to evaluate a trajectory at arbitrary query times and time-shift estimation is used to estimate the unknown temporal offset between the two data recordings.

2.5.1.1 Linear Interpolation

Consider a sequence of 3D positions,

$$\{\boldsymbol{X}(t_j)\}_{j=1}^{J}, \tag{2.143}$$

sampled at times $\{t_j\}_{j=1}^{J}$ and a reference sequence,

$$\left\{\boldsymbol{X}^{ref}(t_k^{ref})\right\}_{k=1}^{K}, \tag{2.144}$$

sampled at times $\{t_k^{ref}\}_{k=1}^{K}$. To evaluate the reference trajectory at an arbitrary query time $t \in \{t_j\}_{j=1}^{J}$, linear interpolation can be used. Let k be the index such that $t_k^{ref} \leq t < t_{k+1}^{ref}$. Let the normalised weight be defined as,

$$\beta(t) = \frac{t - t_k^{ref}}{t_{k+1}^{ref} - t_k^{ref}} \in [0, 1]. \tag{2.145}$$

The interpolated position is then the convex combination,

$$\boldsymbol{X}^{ref}(t) = (1 - \beta(t))\, \boldsymbol{X}^{ref}(t_k^{ref}) + \beta(t)\, \boldsymbol{X}^{ref}(t_{k+1}^{ref}). \tag{2.146}$$

When $\beta = 0$, the left neighbour is recovered and when $\beta = 1$, the right neighbour is recovered. Interpolation is only considered valid within the recorded time range of the reference trajectory, i.e. for $t \in [t_1^{ref}, t_K^{ref}]$.

2.5.1.2 Time-Shift Estimation

Consider the two trajectory sequences defined in equations (2.143) and (2.144). Due to asynchronous recording, the two trajectories may have an unknown constant time offset $\Delta t \in \mathbb{R}$. The valid index set at a candidate shift Δt is defined as,

$$\mathcal{J}(\Delta t) = \left\{ j \in \{1, \dots, J\} \;\middle|\; t_j + \Delta t \in [t_1^{ref}, t_K^{ref}] \right\}, \tag{2.147}$$

i.e. the trajectory set indices for which the shifted time falls within the valid range of the reference trajectory.

If the two trajectories are expressed in the same coordinate system, the optimal time shift, Δt^*, can be estimated by minimising the root mean squared Euclidean distance,

$$\Delta t^* = \underset{\Delta t}{\operatorname{argmin}} \sqrt{\frac{1}{|\mathcal{J}(\Delta t)|} \sum_{j \in \mathcal{J}(\Delta t)} \|\boldsymbol{X}(t_j) - \boldsymbol{X}^{ref}(t_j + \Delta t)\|^2}. \tag{2.148}$$

where $\boldsymbol{X}^{ref}$ is evaluated using linear interpolation as defined in equation (2.146).

2.5.2 Principal Component Analysis

Principal component analysis (PCA) is a technique for identifying the directions of maximum variance in a multivariate dataset. In the context of this work, PCA is applied to three-dimensional trajectory data to extract the dominant direction of motion, producing a scalar time series that concentrates signal energy before spectral analysis.

Consider a sequence of 3D positions,

$$\{\boldsymbol{X}(t_j)\}_{j=0}^{J-1}, \tag{2.149}$$

sampled at times $\{t_j\}_{j=0}^{J-1}$. The mean-centred observations are,

$$\boldsymbol{Q}(t_j) = \boldsymbol{X}(t_j) - \bar{\boldsymbol{X}}, \tag{2.150}$$

where,

$$\bar{\boldsymbol{X}} = \frac{1}{J} \sum_{j=0}^{J-1} \boldsymbol{X}(t_j). \tag{2.151}$$

The sample covariance matrix $\Sigma \in \mathbb{R}^{3\times3}$ is then,

$$\Sigma = \frac{1}{J-1} \sum_{j=0}^{J-1} \boldsymbol{Q}(t_j) \boldsymbol{Q}^T(t_j). \tag{2.152}$$

Since Σ is symmetric positive semi-definite, it admits the eigendecomposition,

$$\Sigma = V \Lambda V^T, \tag{2.153}$$

where $V = [\boldsymbol{v}_1, \boldsymbol{v}_2, \boldsymbol{v}_3] \in \mathbb{R}^{3\times3}$ contains the eigenvectors and $\Lambda = \text{diag}(\lambda_1, \lambda_2, \lambda_3)$ the corresponding eigenvalues, ordered such that $\lambda_1 \geq \lambda_2 \geq \lambda_3 \geq 0$. Since Σ is symmetric, the eigenvectors are mutually orthogonal and form an orthonormal basis in $\mathbb{R}^3$, i.e. $V^TV = I_3$. The eigenvectors $\boldsymbol{v}_1, \boldsymbol{v}_2, \boldsymbol{v}_3$ are the principal components with $\boldsymbol{v}_1$ pointing in the direction of maximum variance.

The projection of the trajectory onto the i-th principal component yields a scalar time series,

$$s_i(t_j) = \boldsymbol{v}_i^T \boldsymbol{Q}(t_j),\ j = 0, \ldots, J-1, \tag{2.154}$$

whose variance equals the corresponding eigenvalue λ_i. The first principal component $s_1(t_j)$ therefore captures the dominant direction of motion.

2.5.3 Frequency Estimation

The natural oscillation frequencies of the tracked features are estimated from the projected scalar time series obtained via PCA. Spectral analysis is used to decompose the signal into its frequency components and identify the dominant peaks.

2.5.3.1 Discrete Fourier Transform

Let $s(t_j),\ j = 0, ..., J-1$, denote some scalar signal sampled at interval T. The discrete Fourier transform (DFT) is then defined as,

$$S(k) = \sum_{j=0}^{J-1} s(t_j) e^{-i2\pi \frac{k}{J} j},\ k = 0, \ldots, J-1, \tag{2.155}$$

where k indexes the discrete frequency $f_k = k/(JT)$ in Hz. The magnitude $|S(k)|^2$ is known as the periodogram which for a random signal is an estimate of the power spectral density (PSD) at frequency f_k. For a real signal, only the first $\lfloor J/2 \rfloor + 1$ components are unique due to conjugate symmetry. However, the periodogram is a high-variance estimator of the PSD, motivating the use of the Welch method described below.

2.5.3.2 Energy-Weighted Welch PSD

The periodogram defined in the previous section is a high-variance estimator of the PSD. Applied to a short or noisy signal, it produces a spectral estimate that is itself noisy, making dominant frequency peaks difficult to identify reliably. The Welch method [47] reduces this variance by dividing the signal into overlapping segments, computing a periodogram for each segment independently and then averaging the results. The averaged estimate is smoother and more stable than the single periodogram but at the cost of frequency resolution. This averaging assumes the signal is stationary, which means its statistical properties, such as variance and frequency content, do not change over time. It means that each segment reflects the same underlying spectrum.

Let $s(t_j)$, $j = 0, \ldots, J-1$, denote a scalar signal sampled at interval T. The signal is divided into N_w overlapping segments of length $L < J$, where segment l consists of the samples,

$$s_l(t_j) = s(t_{lh+j}),\ j = 0, \ldots, L-1, \tag{2.156}$$

and $h < L$ is the step size between consecutive segment starts. The overlap between adjacent segments is $L - h$ samples.

Before computing the DFT of each segment, a window function $w(j)$ is applied. Windowing reduces spectral leakage, which refers to the artificial spreading of energy from one frequency bin into neighbouring bins caused by treating a finite segment as if it were periodic. A Hann window,

$$w(j) = \frac{1}{2}\left(1 - \cos\left(\frac{2\pi j}{L-1}\right)\right),\ j = 0, \ldots, L-1, \tag{2.157}$$

is used here, which tapers the segment smoothly to zero at both ends. The windowed segment is,

$$s_l^w(t_j) = w(j)s_l(t_j). \tag{2.158}$$

The periodogram of segment l is then computed as,

$$P_l(f_k) = \frac{1}{W_{ss}}\left|\sum_{j=0}^{L-1} s_l^w(t_j)\, e^{-i2\pi\frac{k}{L}j}\right|^2,\ f_k = \frac{k}{LT}, \tag{2.159}$$

where $W_{ss} = \sum_{j=0}^{L-1} w(j)^2$ normalises for the energy reduction introduced by the window. For a real signal, only the first $\lfloor L/2 \rfloor + 1$ frequency bins are unique. The standard Welch estimate is the simple average of all segment periodograms,

$$\hat{S}(f_k) = \frac{1}{N_w}\sum_{l=0}^{N_w-1} P_l(f_k). \tag{2.160}$$

For a signal whose amplitude decays over time, such as the free response following a pull-and-release excitation, the stationarity assumption does not hold as early segments carry strong oscillation while later segments consist largely of low-amplitude noise. Treating all segments equally in the standard Welch estimate then lets the low-energy late segments contribute as much to the spectral average as the high-energy early ones which can suppress the dominant frequency peaks. To address this, each segment is assigned a weight proportional to its windowed signal energy,

$$E_l = \sum_{j=0}^{L-1} s_l^w(t_j)^2, \tag{2.161}$$

and the energy-weighted PSD estimate is formed as,

$$\hat{S}(f_k) = \frac{\sum_{l=0}^{N_w-1} E_l\, P_l(f_k)}{\sum_{l=0}^{N_w-1} E_l}. \tag{2.162}$$

Segments with high energy, corresponding to time intervals with strong oscillation, contribute more to the spectral estimate, while low-energy segments dominated by noise contribute less. The resulting estimate emphasises the spectral content of the oscillatory phase of the motion, improving the identifiability of dominant frequency peaks in decaying signals.

2.5.3.3 Damped Harmonic Oscillator

A pull-and-release excitation produces a free oscillation that decays over time as mechanical energy is dissipated through damping. Modelling this behaviour as underdamped harmonic oscillator gives the mathematical representation,

$$s(t) = A\, e^{-\delta t} \cos\Big(2\pi f_0\, t + \phi\Big), \tag{2.163}$$

where $A \in \mathbb{R}$ is the initial amplitude, $\delta > 0$ is the decay rate in s^{-1}, $f_0 > 0$ is the frequency in Hz and $\phi \in [-\pi, \pi]$ is the initial phase. The four parameters A, δ, f_0 and ϕ are estimated by fitting equation (2.163) to the observed signal $s(t_j)$ by solving the non-linear least-squares problem,

$$\min_{A,\,\delta,\,f_0,\,\phi} \sum_{j=1}^{J} \Big(s(t_j) - A\, e^{-\delta t_j} \cos\Big(2\pi f_0\, t_j + \phi\Big) \Big)^2. \tag{2.164}$$

Unlike the Welch PSD, which identifies all frequency components present in the signal, the damped harmonic oscillator fit assumes the signal is dominated by a single oscillation mode. When this assumption holds, the fit provides both the dominant frequency f_0 and the amplitude decay rate δ, whereas estimating δ from the PSD would require resolving the width of the spectral peak, which demands higher frequency resolution than is practical for short transient signals.

3
Methods

This chapter describes the complete methodology used to reconstruct and evaluate three-dimensional feature trajectories from synchronised multi-camera video recordings. The video pipeline combines camera calibration, colour-based image segmentation, multi-view triangulation and recursive state estimation to produce 3D trajectories over time. All computations were implemented in Python using the open-source libraries OpenCV [39], NumPy [48] and SciPy [49].

The chapter is structured as follows. Section 3.1 describes the camera calibration procedure used to estimate the intrinsic and extrinsic parameters of the multi-camera setup. Section 3.2 presents the motion tracking pipeline. The methodology is then evaluated in two experimental settings. Section 3.3 describes the controlled base-case experiment and Section 3.4 describes the tree motion experiments. The two experiments differ fundamentally in their tracking complexity. In the base case, a single spherical marker was tracked under controlled conditions, remaining visible in all cameras throughout the entire recording. This simplified setting removed occlusion and data association ambiguity from the problem which allows the focus to be placed on validating the metric accuracy of the reconstructed 3D trajectories against independent measurements from an industrial robot arm and a terrestrial LiDAR scanner. The tree motion experiments introduced the full tracking challenge, multiple features distributed across the stem and branches, subject to partial and complete occlusion due to varying visibility across cameras. Together, the two experiments therefore assess both the quantitative accuracy and the practical robustness of the pipeline.

3.1 Camera Calibration

Camera calibration was performed to estimate the intrinsic and extrinsic parameters of the three-camera setup, following the checkerboard-based procedure described in Section 2.1.7. The calibration was divided into two sequential stages. Intrinsic calibration was performed first and independently for each camera, since the intrinsic parameters depend only on the internal imaging geometry of the camera and remain fixed regardless of its physical placement. Extrinsic calibration was then performed separately for each experimental camera configuration, since the cameras were repositioned between experiments and the relative poses therefore had to be re-estimated each time. Performing the two stages sequentially, with intrinsic parameters fixed during extrinsic calibration, reduces the degrees of freedom in

the extrinsic optimisation and avoids perturbing well-estimated intrinsic parameters when the stereo calibration sequence is shorter and less diverse than the dedicated intrinsic calibration recording.

3.1.1 Calibration Setup

The calibration setup consisted of three Sony DSC-RX0 cameras mounted on tripods and a planar checkerboard calibration target. The checkerboard pattern contained 10×7 inner corners, as seen in Figure 2.5, and was printed on A3 paper before being attached to a rigid clipboard to minimise surface deformation. No visible air bubbles between the paper and the clipboard could be seen. The checkerboard square side length was measured to be 34.67 mm.

3.1.2 Corner Detection

Checkerboard inner corners were detected in each calibration image to establish 2D-3D point correspondences between image coordinates and the known checkerboard geometry as described in Section 2.1.7.1. Initial corner estimates were obtained using the OpenCV routine `findChessboardCorners`, which detects the ordered grid of inner checkerboard intersections. The detected positions were subsequently refined to sub-pixel accuracy using `cornerSubPix`, with a search window of 11×11 pixels and a termination criterion of either 1000 iterations or a positional change below 0.0001 pixels [39].

3.1.3 Intrinsic Calibration

Intrinsic calibration was performed independently for each camera to estimate the intrinsic calibration matrix K and lens distortion coefficients d, as described in Section 2.1.7.2. The calibration was based on multiple checkerboard observations recorded at different positions and orientations. To assess the stability of the estimated parameters, the calibration was repeated across five independent image datasets per camera, as described below.

3.1.3.1 Data Acquisition

For each of the three cameras, a calibration video was recorded in which the checkerboard target was moved throughout the scene at varying distances and orientations. Diverse poses were used to ensure the checkerboard covered a large portion of the image plane, which improves the stability of the estimated parameters [40][41]. The videos were recorded at 50 frames per second with a resolution of 1920×1080 pixels. The recording durations were approximately 4, 6 and 6 minutes for camera 1, 2 and 3 respectively. From each video, 337, 369 and 384 frames in which the checkerboard was fully visible were manually extracted. The extracted images for each camera were divided into five datasets of approximately equal size.

3.1.3.2 Parameter Estimation

For each camera and dataset, checkerboard inner corners were detected and refined as described in Section 3.1.2. The intrinsic parameters were estimated using the OpenCV routine `calibrateCamera` [39], based on the works of [40][41], which numerically solves the reprojection minimisation problem (2.40). The estimated parameters consisted of the intrinsic matrix K and the five-parameter distortion model $(k_1, k_2, k_3, p_1, p_2)$.

To reduce the influence of motion blur and occasional corner detection failures, an outlier rejection step was applied. For each dataset, the 15% of images with the largest reprojection errors were removed and the parameters re-estimated. While this improved the overall reprojection accuracy, it introduces a bias by preferentially removing images with more extreme checkerboard poses. The quality of the intrinsic calibration was assessed using the RMS reprojection error (2.41).

3.1.3.3 Uncertainty Estimation

To evaluate the stability of the intrinsic calibration, the mean and standard deviation of the estimated parameters f_x, f_y, c_x, c_y, the distortion coefficients and the reprojection errors were computed across the five datasets for each camera. These statistics are reported in the results and used in the base-case sensitivity analysis to assess how variations in the intrinsic parameters propagate through the full tracking pipeline.

3.1.4 Extrinsic Calibration

Extrinsic calibration was performed to estimate the relative poses between the three cameras for each experimental configuration, as described in Section 2.1.7.3. Since the camera positions were adjusted between experiments to accommodate differences in scene geometry, the extrinsic calibration was repeated for each unique configuration.

3.1.4.1 Data Acquisition

The three cameras were mounted on tripods and positioned along an arc facing the measurement volume. The placement was chosen to balance two competing requirements, sufficient angular separation between cameras to ensure robust triangulation and sufficient overlap in the field of view to guarantee that tracked features remained visible in multiple cameras throughout the recording.

For each experimental setup, a dedicated extrinsic calibration sequence was recorded prior to and/or after the motion experiments. During the calibration sequence, the checkerboard target was moved throughout the shared field of view of the cameras at varying distances and orientations while ensuring that the checkerboard remained simultaneously visible in at least two neighbouring cameras.

The app `Imaging Edge`, developed by Sony, was used to start and stop the video streams. However, due to observed frame offsets between the videos in each video recording instance, temporal alignment between videos was performed manually. A flashlight was turned on and off repeatedly in front of all cameras at the start of each recording, producing a clearly identifiable transient in all video streams that served as a common synchronisation reference. The videos were recorded at 50 frames per second at a resolution of 1920×1080 pixels.

3.1.4.2 Parameter Estimation

Frames containing visible checkerboard observations were manually extracted from the synchronised calibration videos. Checkerboard corners were detected and refined using the procedure described in section 3.1.2. Stereo calibration was then performed for the camera pairs $(1, 2)$ and $(2, 3)$ using the OpenCV routine `stereoCalibrate` [39], which is based on the works of [40][41], with the intrinsic parameters held fixed at the values estimated during the intrinsic calibration stage. The relative rotation and translation between each camera pair were estimated by minimising the total reprojection error as defined in (2.48). As in the intrinsic calibration, the 15% of image pairs with the largest reprojection errors were removed and the parameters re-estimated, for the same reasons discussed in Section 3.1.3.

Camera 1 was chosen as the reference frame, and the poses of cameras 2 and 3 relative to camera 1 were obtained by composing the pairwise stereo calibration transformations, as described in Section 2.1.7.3.2. The resulting camera poses $\{(R_i, \boldsymbol{t}_i)\}_{i=1}^{3}$ define the rigid body transformations from each camera coordinate system to the global camera reference frame, which is used throughout the subsequent triangulation and tracking procedures. The quality of the extrinsic calibration was assessed using the RMS reprojection error (2.49) for each camera pair.

3.2 Multi-View Motion Tracking

This section describes the implemented method used to estimate three-dimensional trajectories of red markers from multi-camera video recordings. The method assumes that the video streams have been temporally synchronised using the flashlight procedure described in Section 3.1.4.1 and that calibrated camera parameters are available from Section 3.1. An overview of the full method is shown in Figure 3.1.

The method consists of the following stages. Colour-based segmentation extracts image-space centroid observations from each camera view at every frame. Tracks are initialised manually in the first frame by triangulating corresponding detections across camera views. At each subsequent frame, each track is propagated forward using the Singer acceleration model and the predicted position is projected into each camera view. Segmented detections are then associated with tracks independently per camera using gated assignment. When valid observations are available in one or more camera views, an EKF measurement update is performed. Otherwise, the prediction is used as the next state. After the full forward pass, a fixed-interval

smoothing step is applied to incorporate future measurements and refine the estimates. Each of these stages is described in the following subsections.

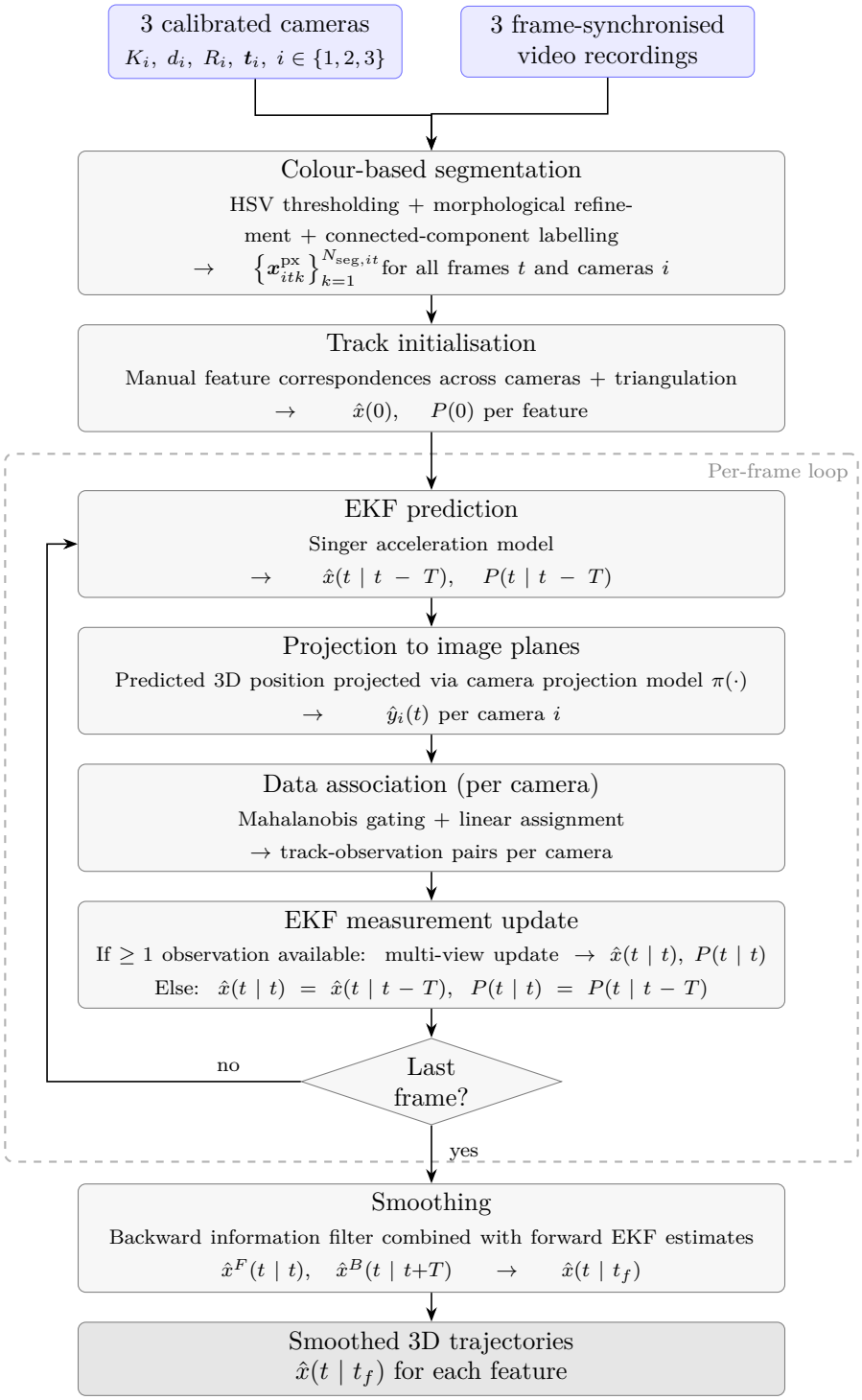

Figure 3.1: Overview of the motion tracking method. Blue boxes denote inputs, grey boxes denote processing steps and the dark-shaded box denotes the final output. Segmentation is performed as a pre-processing step over all frames before tracking begins. The dashed region marks the per-frame loop executed over all synchronised video frames.

3.2.1 Colour-Based Segmentation

Image observations were extracted independently in each camera view using colour-based segmentation. Since the tracked features were marked with red tape or paint, HSV thresholding was used to isolate image regions with the expected colour distribution. This approach was chosen for its low computational cost, interpretability and sufficiency for the controlled indoor recording conditions of this work. The procedure is summarised in Algorithm 1.

For each frame, the RGB image was converted to HSV colour space and thresholded using fixed lower and upper HSV bounds. The resulting binary mask was refined by morphological opening followed by morphological closing using OpenCV's `morphologyEx` routine [39] with a rectangular 5×5 structuring element. Opening removed small isolated foreground regions due to background noise, while closing filled small gaps inside segmented marker regions caused by specular highlights on the marker surface. Connected-component labelling (CCL) with 8-connectivity was then applied to the refined binary mask using OpenCV's `connectedComponentsWithStats` routine [39], based on the algorithm of [43]. The centroid of each connected region was computed according to equation (2.78) and treated as an image-space observation. The theoretical basis for each step is described in Sections 2.2.1, 2.2.2, 2.2.3 and 2.2.4.

The segmentation stage produced a set of image-space observations,

$$\left\{ \boldsymbol{x}_{itk}^{\mathrm{px}} \right\}_{k=1}^{N_{\mathrm{seg},it}}, \ \boldsymbol{x}_{itk}^{\mathrm{px}} \in \mathbb{R}^2, \tag{3.1}$$

for each camera i and time step t, where $N_{\mathrm{seg},it}$ denotes the number of detected regions in camera i at frame t.

Algorithm 1 Colour-Based Segmentation

Input: RGB video frames for cameras $i = 1, 2, 3$ and frames $t = 0, T, \ldots, t_f$, defined on $\Omega \subset \mathbb{Z}^2$, HSV bounds l, u, structuring element s

Output: Image-space observations $\{\boldsymbol{x}_{itk}^{\mathrm{px}}\}_{k=1}^{N_{\mathrm{seg},it}}$

for each camera $i = 1, 2, 3$ **do**
  for each frame $t = 0, T \ldots, t_f$ **do**
    Convert the image to HSV colour space, yielding f_{it}
    Compute the binary threshold image,

$$f_{b,it}(\boldsymbol{x}^{\mathrm{px}}) = \begin{cases} 1, & l_c \leq f_{c,it}(\boldsymbol{x}^{\mathrm{px}}) \leq u_c, \quad \forall c \in \{H, S, V\}, \\ 0, & \text{otherwise} \end{cases}$$

    Refine the binary mask using morphological opening followed by closing,

$$M_{it} = \mathrm{close}\left(\mathrm{open}\left(f_{b,it}, s\right), s\right)$$

    Apply 8-connected CCL to obtain,

$$L_{it} : \Omega \to \{0, 1, \ldots, N_{\mathrm{seg},it}\}$$

    for each connected component $k = 1, \ldots, N_{\mathrm{seg},it}$ **do**
      Define the segmented region,

$$\Omega_{itk} = \{\boldsymbol{x}^{\mathrm{px}} \in \Omega : L_{it}(\boldsymbol{x}^{\mathrm{px}}) = k\}$$

      Compute the segmentation center,

$$\boldsymbol{x}_{itk}^{\mathrm{px}} = \frac{1}{|\Omega_{itk}|} \sum_{\boldsymbol{x}^{\mathrm{px}} \in \Omega_{itk}} \boldsymbol{x}^{\mathrm{px}}$$

    end for
  end for
end for

3.2.2 Track Initialisation

Tracks were initialised manually in the first frame of each synchronised sequence. For each physical marker, the corresponding image-space centroid was selected manually in at least two camera views as required for triangulation. All three camera views were used where the marker was visible. The selected correspondences were triangulated using the method described in Section 2.1.8, yielding an initial three-dimensional position estimate $\boldsymbol{X}_0 \in \mathbb{R}^3$.

The triangulated position was used to initialise the Singer model state vector. The initial velocity and acceleration were set to zero since no prior motion estimate was

available,

$$\hat{x}(0) = \left(\boldsymbol{X}_0^T,\ \mathbf{0}^T,\ \mathbf{0}^T\right) \in \mathbb{R}^9. \tag{3.2}$$

The initial covariance matrix was set to,

$$P(0) = \operatorname{diag}\left(\sigma_p^2 I_3,\ \sigma_v^2 I_3,\ \sigma_a^2 I_3\right) \in \mathbb{R}^{9\times 9} \tag{3.3}$$

where σ_p^2, σ_v^2 and σ_a^2 reflect the higher uncertainty in the unobserved velocity and acceleration components relative to the triangulated position. The values used were $\sigma_p^2 = 1e4$ mm^2, $\sigma_v^2 = 1e6$ mm^2/s^2 and $\sigma_a^2 = 1e8$ mm^2/s^4 throughout all experiments.

3.2.3 Process Model

Each tracked feature was modelled independently using the Singer acceleration model described in Section 2.3.1.1. The state vector $\hat{x}(t) \in \mathbb{R}^9$ consisted of position, velocity and acceleration in all three spatial directions, and the sampling interval $T = 1/50$ s corresponded to the frame rate of the synchronised video recordings.

The Singer parameters α and σ_m^2 together with the measurement noise standard deviation r were treated as fixed tuning parameters. The values $\alpha = 20$ s^{-1}, $\sigma_m^2 = 1e7$ mm^2 s^{-4} and $r = 5.0$ px were used throughout both motion experiments. They were chosen empirically, by adjusting them until the tracking produced satisfactory trajectories for the tree motion experiments, rather than being derived from the expected dynamics of the individual markers throughout the experiments. The same parameter set was applied to every feature, even though markers at different experiments, trees and locations moved with differing acceleration and displacement.

3.2.4 Prediction and Projection

At each time step, every track was propagated independently using the EKF prediction step described in Section 2.3.2, yielding the predicted state estimate,

$$\hat{x}(t \mid t-T) \in \mathbb{R}^9, \tag{3.4}$$

and predicted covariance,

$$P(t \mid t-T) \in \mathbb{R}^{9\times 9}. \tag{3.5}$$

The predicted three-dimensional position was subsequently projected into each camera view using the camera projection model described in Section 2.3.1.2, producing the predicted image-space observation $\hat{y}_i(t) \in \mathbb{R}^2$ for each camera $i \in \{1, 2, 3\}$. These predicted observations were used in the following data association step to define the search region for matching segmented detections to tracks.

3.2.5 Data Association

For each camera $i \in \{1, 2, 3\}$, the predicted image observations $\hat{y}_i(t)$ were compared with the segmented centroid detections,

$$\left\{\boldsymbol{x}_{itk}^{\mathrm{px}}\right\}_{k=1}^{N_{\mathrm{seg},it}}. \tag{3.6}$$

Candidate associations were first filtered using Mahalanobis gating as described in Section 2.3.3.1, with the threshold $\gamma = 5.991$ chosen from the χ^2_2 distribution at a significance level of $p = 95\%$. Pairs outside the gating region were assigned a cost of ∞ and excluded from the subsequent assignment step.

For the remaining candidate associations, the assignment cost was defined as the squared Mahalanobis distance. Since multiple tracks could potentially satisfy the gating criterion for the same detection, the correspondence problem was formulated as a linear assignment problem, as described in Section 2.3.3.2, solved independently for each camera. The assignment was solved using the `optimize.linear_sum_assignment` routine, based on the works of [45], in SciPy [49].

3.2.6 Measurement Update

After per-camera data association, the image observations available across all cameras were combined into a stacked measurement vector $y(t) \in \mathbb{R}^{2n}$, where n denotes the total number of associated detections across all cameras at time t. The corresponding EKF measurement update was then performed as described in Section 2.3.2. The non-linear measurement function consisted of the stacked camera projection models and the measurement Jacobian $H(t)$ was evaluated numerically at the predicted state estimate.

A measurement update was performed whenever at least one valid observation was available across any camera view. If no observation was associated with a track in any camera at the current frame, no update was performed and the filtered estimate and covariance were set equal to the prediction,

$$\hat{x}(t \mid t) = \hat{x}(t \mid t - T) \in \mathbb{R}^9, \tag{3.7}$$

$$P(t \mid t) = P(t \mid t - T) \in \mathbb{R}^{9\times 9}. \tag{3.8}$$

3.2.7 Smoothing

After the complete forward filtering pass, a smoothing step was applied to improve the trajectory estimates by incorporating information from future observations. The fixed-interval smoothing approach described in Section 2.3.4 was used. The backward information filter was initialised with $\Pi(0) = 1e10 I_9$, reflecting the absence of any prior information about the state at the final time step, and operated on the same track-observation associations produced during the forward pass, without re-running data association. The resulting backward estimates $\hat{x}^B(t \mid t + T)$ were combined with the forward-filtered estimates $\hat{x}^F(t \mid t) \in \mathbb{R}^9$ to produce the smoothed trajectories $\hat{x}(t \mid t_f) \in \mathbb{R}^9$ and associated covariance estimates $P(t \mid t_f) \in \mathbb{R}^{9\times 9}$. These smoothed trajectories formed the final output and were used in all subsequent analysis and evaluation.

3.3 Controlled Motion Experiment

This experiment was designed to validate the motion estimation against two independent reference sensors, an industrial robot arm and a terrestrial LiDAR, under controlled conditions. A single spherical marker was tracked with no occlusion and full camera visibility throughout, isolating the metric accuracy of the pipeline from the additional challenges of occlusion and multi-feature tracking introduced in the tree experiments. This enabled a quantitative, multi-sensor assessment of the three-dimensional position accuracy before applying the pipeline to the more complex tree motion experiments.

3.3.1 Experimental Scene and Equipment

The experimental scene consisted of a Universal Robots UR5e robot arm, three Sony DSC-RX0 cameras with calibrated intrinsics and a Faro Focus3D S20 terrestrial laser scanner. The overall scene is shown in Figure 3.2.

The UR5e is a six degree-of-freedom robot arm with a positional repeatability of ± 0.03 mm which provides a precise and reproducible ground truth reference for the tracked feature position. The robot state was recorded at 500 Hz throughout the experiment. The tracked feature was mounted at a fixed distance from the robot's so called tool centre point (TCP) along its tool axis.

The three Sony DSC-RX0 cameras were mounted on tripods and positioned on an arc facing the measurement volume at distances around 1 m from the robot. The cameras recorded at 50 frames per second at a resolution of 1920×1080 pixels. The app `Imaging Edge`, developed by Sony, was used to start and stop the video recordings. Temporal synchronisation and extrinsic calibration were performed as described in Section 3.1.4. For temporal synchronisation, a flashlight was turned on and off multiple times in front of all cameras at the start of each recording. A dedicated extrinsic calibration sequence was recorded both prior to and following the motion experiment, as described in Section 3.1.4.

A terrestrial LiDAR emits laser pulses in a sweeping pattern and records the return time of each pulse to produce a dense three-dimensional point cloud. In addition to geometry, the Faro Focus3D S20 takes a panoramic image after each scan and projects the colour values onto the corresponding points yielding a colourised point cloud. Since each LiDAR scan required approximately 10 minutes, the robot was held stationary at each of the four target positions, illustrated in Figure 3.2, while scans were acquired, rather than scanning during continuous motion. Three scan positions were used, denoted L_1, L_2 and L_3 in Figure 3.2. All scans were acquired with a resolution setting of 1/4 and a quality setting of $4\times$, corresponding to a point spacing of 6.136 mm at a range of 10 m. The Faro Focus3D S20 has a specified ranging error of ± 2 mm and a ranging noise of 0.6 mm and 1.2 mm for surfaces with 90% and 10% reflectivity at 10 m, respectively.

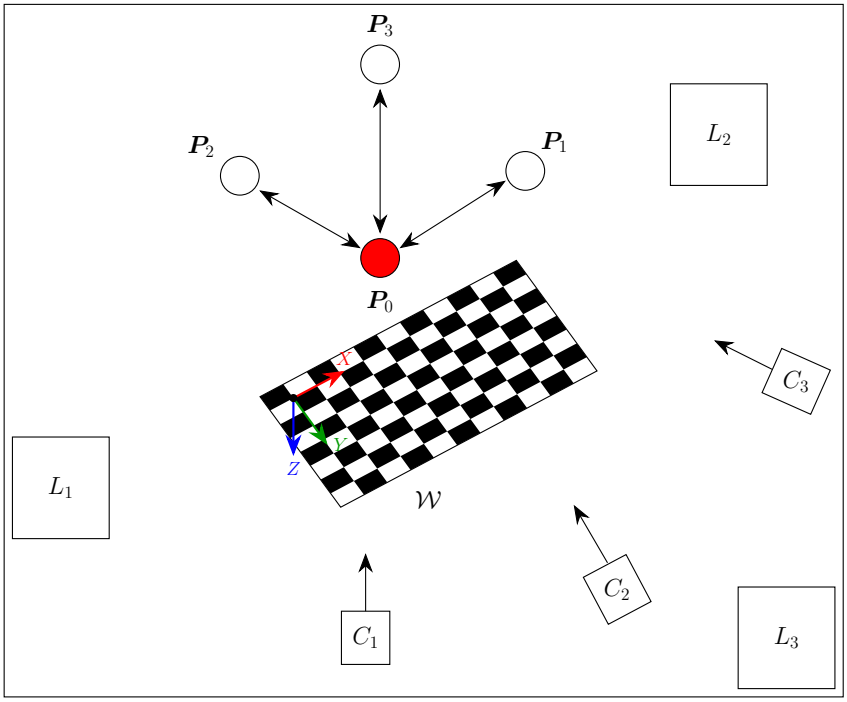

Figure 3.2: Controlled motion experimental setup. Camera positions are denoted C_1, C_2 and C_3. LiDAR scanner positions are denoted L_1, L_2 and L_3. Motion target endpoint positions are denoted $\boldsymbol{P}_0$, $\boldsymbol{P}_1$, $\boldsymbol{P}_2$ and $\boldsymbol{P}_3$. The checkerboard calibration target, labelled $\mathcal{W}$, was placed flat on the table beneath the robot workspace and defines the common world coordinate frame to which all sensor data were registered.

3.3.2 Tracked Feature

The tracked feature consisted of a standard ping-pong ball spray-painted red and mounted to the robot gripper using a rigid stick inserted through a drilled hole at a measured TCP offset. A spherical geometry was chosen because the projection of a sphere's centroid onto any image plane closely coincides with the projection of its three-dimensional centre, making the image-space centroid estimate from colour-based segmentation geometrically consistent across all camera viewing angles. The colour red was chosen since its hue is well-separated from the background colours present in the scene, making it straightforward to isolate using HSV thresholding as described in Section 3.2.1, and because the painted surface produced sufficient reflectance to register reliable returns in the LiDAR point cloud. After painting, the outer diameter of the ball was measured to be 40.0 mm which was large enough to ensure reliable sphere fits to the LiDAR point cloud at the scan distances used.

3.3.3 Motion Protocol

The robot arm executed a prescribed constant-velocity motion between four discrete positions, $\boldsymbol{P}_0$, $\boldsymbol{P}_1$, $\boldsymbol{P}_2$ and $\boldsymbol{P}_3$, illustrated in Figure 3.2. $\boldsymbol{P}_0$ served as the home position to which the robot returned between each displacement, while $\boldsymbol{P}_1$, $\boldsymbol{P}_2$ and $\boldsymbol{P}_3$ were target positions along distinct spatial directions, designed to test the tracking pipeline across all three spatial dimensions. The dynamic recording consisted of

three full cycles of the sequence,

$$\boldsymbol{P}_0 \rightarrow \boldsymbol{P}_1 \rightarrow \boldsymbol{P}_0 \rightarrow \boldsymbol{P}_2 \rightarrow \boldsymbol{P}_0 \rightarrow \boldsymbol{P}_3, \tag{3.9}$$

recorded simultaneously by all three cameras and the robot at their respective sampling rates.

Following the dynamic recording, the robot was held stationary at each of the four endpoint positions while LiDAR scans were acquired, as described in Section 3.3.1. At $\boldsymbol{P}_0$, scans were acquired from positions L_1 and L_3, while at $\boldsymbol{P}_1$, $\boldsymbol{P}_2$ and $\boldsymbol{P}_3$, scans were acquired from positions L_1 and L_2. Using two scanner positions per endpoint allowed the co-registration error of the LiDAR data to be estimated, as described in Section 3.3.5.3.

A scan in L_2 at $\boldsymbol{P}_0$ was also made, but due to an accidental displacement of the robot arm during the acquisition, it was discarded. The additional scan from L_3 had been acquired at $\boldsymbol{P}_0$ and was used in its place, preserving two independent scan positions for that endpoint.

3.3.4 Coordinate Frame Registration

Since each sensor records data in its own local coordinate system, a common global reference frame $\mathcal{W}$ was established to enable direct comparison across all three sensors. The frame was defined by the planar checkerboard calibration target, with its origin at one corner of the inner grid, the X- and Y-axes aligned with the grid directions and the Z-axis pointing perpendicularly downward from the board surface into the table, as illustrated in Figure 3.2. The checkerboard was placed flat on the table beneath the robot's workspace and remained stationary throughout all measurements, with the exception of the extrinsic calibration sequences recorded before and after the experiment. Its position was chosen such that it was simultaneously visible from all three cameras and all LiDAR scan positions and reachable by the robot.

3.3.4.1 Robot to World

The robot coordinate system was registered to $\mathcal{W}$ by solving the point set registration problem described in Section 2.4.1, using $M = 4$ point correspondences between the two frames.

The four corner-most inner corners of the checkerboard grid served as the correspondence points. Their positions in the world frame $\{\boldsymbol{X}_k^{\mathcal{W}}\}_{k=1}^4$ were computed directly from the known checkerboard geometry, using the measured square side length of 34.67 mm. The corresponding robot-frame positions $\{\boldsymbol{X}_k^{\mathcal{R}}\}_{k=1}^4$ were obtained by moving the ball to each corner in turn, positioning it as close to the surface as possible without physical contact. Approximately 5000 TCP position measurements were recorded at each corner and averaged to obtain a single position estimate, to which an offset of 20 mm was applied, corresponding to the distance from the board

to the center of the sphere. The optimal rigid body transformation $(R_{\mathcal{WR}}, \boldsymbol{t}_{\mathcal{WR}})$ was then estimated by solving equation (2.139) as described in Section 2.4.1.

3.3.4.2 Video to World

The camera coordinate system was registered to $\mathcal{W}$ by estimating the pose of the stationary checkerboard target relative to the camera system. Checkerboard corners were detected in a single dedicated static frame from each camera using the procedure described in Section 3.1.2.

The three cameras share a common reference frame $\mathcal{C}$, which is of camera 1, where the calibrated poses $\{(R_i, \boldsymbol{t}_i)\}_{i=1}^{3}$ map the reference frame to each camera i, as established in Section 3.1.4. The checkerboard defines the world frame $\mathcal{W}$, so its inner corners have known positions $\{\boldsymbol{X}_k^{\mathcal{W}}\}_{k=1}^{K_p}$. Let $\{\boldsymbol{x}_{ik}^{\text{px}}\}_{k=1}^{K_p}$ denote the corresponding detected pixel coordinates in camera i.

The unknown is the rigid transformation $(R_{\mathcal{CW}}, \boldsymbol{t}_{\mathcal{CW}})$ from the world frame to the camera reference frame. Composing it with the fixed camera poses gives the predicted projection of corner k into camera i,

$$\hat{\boldsymbol{x}}_{ik}^{\text{px}} = \pi\Big(K_i,\, \boldsymbol{d}_i,\, R_i R_{\mathcal{CW}},\, R_i \boldsymbol{t}_{\mathcal{CW}} + \boldsymbol{t}_i,\, \boldsymbol{X}_k^{\mathcal{W}}\Big)\,. \tag{3.10}$$

The transformation was estimated by minimising the joint reprojection error across all three cameras simultaneously,

$$\min_{R_{\mathcal{CW}},\, \boldsymbol{t}_{\mathcal{CW}}} \sum_{i=1}^{3} \sum_{k=1}^{K_p} \left\| \boldsymbol{x}_{ik}^{\text{px}} - \hat{\boldsymbol{x}}_{ik}^{\text{px}} \right\|^2, \tag{3.11}$$

with the intrinsics and the camera poses $\{(R_i, \boldsymbol{t}_i)\}_{i=1}^{3}$ held fixed. The optimisation was solved using SciPy's `optimize.least_squares` routine [49]. The reconstructed trajectories, obtained in the camera reference frame $\mathcal{C}$, were mapped into $\mathcal{W}$ by the inverse transformation $(R_{\mathcal{WC}}, \boldsymbol{t}_{\mathcal{WC}})$.

3.3.4.3 LiDAR to World

The LiDAR coordinate system was registered to $\mathcal{W}$ using observations of the checkerboard target in the merged point cloud. Since black checkerboard squares exhibited significantly higher noise due to their lower surface reflectivity, only the white square points were used. These were extracted from the colourised point cloud using HSV thresholding as described in Section 2.2.2, with bounds $H \in [0, 179]$, $S \in [0, 30]$ and $V \in [200, 255]$. The registration approach is adapted from [50], in which a checkerboard point cloud is aligned to the known checkerboard geometry for LiDAR-to-camera calibration. The shared idea is to fit a plane to the checkerboard points, project them onto that plane, recover a two-dimensional rigid transformation that aligns the projected cloud to the known grid and at last lift this transformation back to three dimensions. The present method differs in several ways. The transformation is estimated between the LiDAR frame and the world frame rather than

between the LiDAR and a camera. The plane is fitted by principal component analysis of the extracted points rather than by iterative RANSAC. White squares are used instead of black squares. The alignment is driven by a cell-occupancy loss that penalises projected points falling outside the board or on black cells, rather than by the distance from each point to its nearest ideal corner. The two-dimensional transformation is obtained by a grid search over the rotation angle followed by local refinement, rather than by the adaptive optimisation in the original work.

Let $\{\boldsymbol{X}_k^{\mathcal{L}}\}_{k=1}^{K_p}$ denote the extracted white-square point cloud in the LiDAR frame. The centroid is computed as,

$$\bar{\boldsymbol{X}}^{\mathcal{L}} = \frac{1}{K_p} \sum_{k=1}^{K_p} \boldsymbol{X}_k^{\mathcal{L}}, \tag{3.12}$$

and the mean-centred residual matrix is formed as,

$$D = \begin{bmatrix} \left(\boldsymbol{X}_1^{\mathcal{L}} - \bar{\boldsymbol{X}}^{\mathcal{L}}\right)^T \\ \vdots \\ \left(\boldsymbol{X}_K^{\mathcal{L}} - \bar{\boldsymbol{X}}^{\mathcal{L}}\right)^T \end{bmatrix} \in \mathbb{R}^{K_p \times 3}. \tag{3.13}$$

The singular value decomposition $D = U\Sigma V^T$ is computed where the rows of V^T are the principal directions of the point cloud as described in Section 2.5.2. The plane normal $\hat{\boldsymbol{n}}$ is taken as the row of V^T corresponding to the smallest singular value, while the two remaining rows $\hat{\boldsymbol{u}}$ and $\hat{\boldsymbol{v}}$ form an orthonormal basis spanning the fitted plane.

The orthonormal basis matrix $B = [\hat{\boldsymbol{u}},\, \hat{\boldsymbol{v}}] \in \mathbb{R}^{3\times 2}$ is used to project the white-square points onto the fitted plane,

$$\boldsymbol{q}_k = B^T\left(\boldsymbol{X}_k^{\mathcal{L}} - \bar{\boldsymbol{X}}^{\mathcal{L}}\right) \in \mathbb{R}^2,\ k = 1, \ldots, K_p. \tag{3.14}$$

The projected 2D point cloud is aligned to the known checkerboard grid by estimating a 2D rigid transformation parametrised by a rotation angle $\theta \in [0, 2\pi]$ and a translation $(t_x, t_y) \in \mathbb{R}^2$. Each transformed point,

$$\boldsymbol{p}_k = R_{2D}(\theta) \begin{bmatrix} q_{k,x} \\ q_{k,y} \end{bmatrix} + \begin{bmatrix} t_x \\ t_y \end{bmatrix}, \tag{3.15}$$

is assigned to a cell (c_x, c_y) of the checkerboard grid by,

$$c_x = \lfloor p_{k,x}/s \rfloor,\ c_y = \lfloor p_{k,y}/s \rfloor, \tag{3.16}$$

where s is the square side length. The cell colour is determined by the parity $(c_x + c_y) \bmod 2$. Since only white-square points were retained, the alignment is found by minimising the white-occupancy loss,

$$\min_{\theta, t_x, t_y} \ \mathcal{L}(\theta, t_x, t_y) = \frac{1}{K_p} \sum_{k=1}^{K_p} \ell\left(\boldsymbol{p}_k\right), \tag{3.17}$$

where,

$$\ell(\boldsymbol{p}) = \begin{cases} \lambda_{\text{out}} & \text{if } \boldsymbol{p} \text{ lies outside the board,} \\ \lambda_{\text{black}} & \text{if } \boldsymbol{p} \text{ lies on a black square,} \\ 0 & \text{if } \boldsymbol{p} \text{ lies on a white square.} \end{cases} \tag{3.18}$$

The penalty values $\lambda_{\text{out}} = 2.0$ and $\lambda_{\text{black}} = 1.0$ were set to penalise points falling outside the board or on black squares respectively, with no penalty assigned to correctly placed white-square points. The optimisation was solved over a grid search of θ followed by local refinement using `optimize.minimize`, based on the works of [51], in SciPy [49].

The optimal parameters (θ^*, t_x^*, t_y^*) are combined with the plane basis and centroid to recover the full 3D transformation. The rotation matrix is,

$$R_{\mathcal{WL}} = \begin{bmatrix} \cos\theta^* \, \hat{\boldsymbol{u}}^T + \sin\theta^* \, \hat{\boldsymbol{v}}^T \\ -\sin\theta^* \, \hat{\boldsymbol{u}}^T + \cos\theta^* \, \hat{\boldsymbol{v}}^T \\ \hat{\boldsymbol{n}}^T \end{bmatrix} \in \mathbb{R}^{3\times 3}, \tag{3.19}$$

and the translation is,

$$\boldsymbol{t}_{\mathcal{WL}} = \begin{bmatrix} t_x^* \\ t_y^* \\ 0 \end{bmatrix} - R_{\mathcal{WL}} \, \bar{\boldsymbol{X}}^{\mathcal{L}}. \tag{3.20}$$

The initial transformation places the origin at the outer bottom-left corner of the board. To match the world frame convention defined in Section 3.3.4, a fixed offset of $((N_c - 1)s,\ s,\ 0)$ was applied to shift the origin to the correct inner corner, followed by a sign flip of the X and Z axes to align the axis directions. The resulting transformation $(R_{\mathcal{WL}},\ \boldsymbol{t}_{\mathcal{WL}})$ maps LiDAR frame coordinates directly to $\mathcal{W}$.

3.3.5 Data Processing

Following data collection, each sensor's data were processed independently, with the exception of the temporal alignment between the robot and video data.

3.3.5.1 Video

Extrinsic parameters were estimated from both the pre- and post-experiment calibration sequences, as described in Section 3.1.4. The three video streams were manually temporally aligned using the flashlight, after which the ball was tracked using the pipeline described in Section 3.2. The HSV threshold with values $H \in [0, 10] \cup [170, 179]$ and $S \in [60, 255]$, $V \in [60, 255]$ was used for the segmentation, the Singer tracking parameters $\alpha = 20\ \text{s}^{-1}$, $\sigma_m^2 = 1e7\ \text{mm}^2\text{s}^{-4}$ and the measurement noise parameters $r = 5.0$ px was used for the motion tracking as described in Section 3.2. These parameters were chosen to be consistent with the tree motion experiments, described in Section 3.4, and were not chosen to best fit this particular motion case. The resulting trajectories were transformed into $\mathcal{W}$ as described in

Section 3.3.4.2.

This procedure was repeated for each of the five intrinsic calibration sets and for each of the two extrinsic calibration sets with the intrinsics fixed to set 3. For the trajectory obtained using intrinsic set 3 and the post-experiment extrinsic calibration, 5 measurements were extracted and averaged to obtain the endpoint estimates $\boldsymbol{P}_{\text{video},0}$, $\boldsymbol{P}_{\text{video},1}$, $\boldsymbol{P}_{\text{video},2}$ and $\boldsymbol{P}_{\text{video},3}$, for comparison against the robot and LiDAR data.

3.3.5.2 Robot

The robot coordinate system was transformed to $\mathcal{W}$ as described in 3.3.4.1. The robot data were then temporally aligned with the video data by minimising the point-wise Euclidean distance between the two trajectories over a grid search of time offsets, after interpolating the 500 Hz robot signal to the 50 fps video timeline as described in Section 2.5.1. This step was necessary since the robot and cameras operated on independent clocks. For each target position, 10 measurements were extracted and averaged to obtain the endpoint estimates $\boldsymbol{P}_{\text{robot},0}$, $\boldsymbol{P}_{\text{robot},1}$, $\boldsymbol{P}_{\text{robot},2}$ and $\boldsymbol{P}_{\text{robot},3}$.

3.3.5.3 LiDAR

All eight individual scans were co-registered to a common coordinate system using FARO SCENE software by aligning scene reference objects across scan positions. The merged point cloud was then transformed to $\mathcal{W}$ using the procedure described in Section 3.3.4.3 where the same transformation was applied to all scans.

Each scan is identified by the pair $(\boldsymbol{P}_j, L_k)$, where $j \in \{0,1,2,3\}$ denotes the endpoint ball position and $k \in \{1,2,3\}$ denotes the scanner position, giving the set of acquired scans,

$$\mathcal{S} = \{(\boldsymbol{P}_0, L_1),\ (\boldsymbol{P}_0, L_3),\ (\boldsymbol{P}_1, L_1),\ (\boldsymbol{P}_1, L_2),\ (\boldsymbol{P}_2, L_1),\ (\boldsymbol{P}_2, L_2),\ (\boldsymbol{P}_3, L_1),\ (\boldsymbol{P}_3, L_2)\}. \tag{3.21}$$

For each scan $(\boldsymbol{P}_j, L_k) \in \mathcal{S}$, the ball surface points $\{\boldsymbol{S}_i\}_{i=1}^{N_{j,k}} \subset \mathbb{R}^3$ were manually isolated in CloudCompare [52]. The 10% of points with the largest radial residuals from an initial sphere fit were removed as outliers, after which the sphere was refitted to the remaining points by solving,

$$\min_{\boldsymbol{C}_{jk},\, r_{jk}} \sum_{i=1}^{N_{j,k}} \left(\|\boldsymbol{S}_i - \boldsymbol{C}_{jk}\|^2 - r_{jk}^2 \right)^2, \tag{3.22}$$

where $\boldsymbol{C}_{jk} \in \mathbb{R}^3$ is the estimated sphere centre, r_{jk} is the estimated radius and N_{jk} is the number of points in each scan after outlier rejection. The optimisation was solved using SciPy's `optimize.least_squares` routine [49]. The deviation of r_{jk} from the true measured radius of 20.0 mm was used as an indicator of individual scan quality.

For each endpoint $\boldsymbol{P}_j$, the co-registration error was estimated by computing the Euclidean distance between the two independently fitted sphere centres,

$$e_j = \|\boldsymbol{C}_{jk_1} - \boldsymbol{C}_{jk_2}\|, \tag{3.23}$$

where k_1 and k_2 denote the two scanner positions used at $\boldsymbol{P}_j$. The final endpoint position was then estimated as,

$$\boldsymbol{P}_{\mathrm{LiDAR},j} = \frac{1}{2}\left(\boldsymbol{C}_{jk_1} + \boldsymbol{C}_{jk_2}\right), \; j \in \{0,1,2,3\}. \tag{3.24}$$

3.3.6 Sensor Comparison

The accuracy of the video-based tracking was assessed through three comparisons.

First, the three-dimensional endpoint positions $\boldsymbol{P}_{\mathrm{video},j}$, $\boldsymbol{P}_{\mathrm{robot},j}$ and $\boldsymbol{P}_{\mathrm{LiDAR},j}$ were compared across all three sensors for each endpoint $j \in \{0,1,2,3\}$ by computing pairwise Euclidean distances in $\mathcal{W}$ providing an overall assessment of spatial agreement after registration.

Second, the displacement from the home position $\boldsymbol{P}_0$ to each target position was computed independently within each sensor's own coordinate frame,

$$d_{\mathrm{sensor},j} = \|\boldsymbol{P}_{\mathrm{sensor},j} - \boldsymbol{P}_{\mathrm{sensor},0}\|, \quad j \in \{1,2,3\}, \tag{3.25}$$

and compared across sensors. Since these distances are frame-internal quantities, they are independent of the registration transformation and provide a direct assessment of each sensor's intrinsic metric accuracy.

Third, the full video trajectory was compared against the robot ground truth on a frame-by-frame basis. The coordinate-wise residuals between the two trajectories were computed after temporal alignment and the bias and standard deviation in each spatial direction were reported.

3.4 Tree Motion Experiments

The tree motion experiments extend the controlled motion experiment to a more realistic setting, where multiple features are tracked simultaneously across stem and branches, occlusion is frequent and the motion is governed by the physical dynamics of the tree rather than a prescribed robot path.

Three tree species were selected, birch, spruce and pine, representing two structurally distinct groups. Broadleaved trees such as birch have a distributed crown mass that is known to produce different oscillatory behaviour from conifers such as spruce and pine, whose motion is dominated by the main stem. Testing across species therefore assesses whether the pipeline can resolve these structural differences through their dynamic signatures.

Pull-and-release tests were used as the excitation method. By displacing the tree top with a rope and releasing it, a free oscillation is induced that approximates the fundamental sway response of the tree under wind loading. Each test was repeated three times per species to assess the reproducibility of both the reconstructed trajectories and the estimated frequencies. Features were attached at multiple locations along the stem and on lateral branches allowing dynamics of structurally distinct tree components to be compared within a single experiment. The natural frequencies estimated from the reconstructed trajectories are compared to evaluate whether the method produces physically meaningful and scientifically useful results.

3.4.1 Experimental Scene and Equipment

The experimental scene consisted of three Sony DSC-RX0 cameras with calibrated intrinsic parameters and a Universal Robots UR5e robot arm. Each tree was mounted vertically and secured to a table using multiple straps fastened at several heights along the lower portion of the stem. The overall scene is shown in Figure 3.3.

The UR5e is a six degree-of-freedom robot arm that was used to pull the tree and was positioned on the same table as the tree was attached to. This was achieved by attaching a rope between the robot's gripper and near the top of the tree stem.

Similarly as in the controlled motion experiment 3.3, the three Sony DSC-RX0 cameras were mounted on tripods and positioned along an arc facing the trees at a distance of about 1.5 m from the trees. The middle camera was positioned such that its principal viewing direction was orientated perpendicular to the pulling direction. The cameras recorded at 50 frames per second at a resolution of 1920×1080 pixels. The app `Imaging Edge`, developed by Sony, was used to start and stop the video recordings. Temporal synchronisation was performed using the flashlight procedure described in Section 3.1.4.1. Since the camera positions had to be adjusted between species to accommodate differences in tree size and geometry, the extrinsic calibration procedure described in Section 3.1.4 was repeated for each unique camera configuration. Two distinct configurations were used, one shared by the spruce and pine experiments and a separate configuration for the birch experiment.

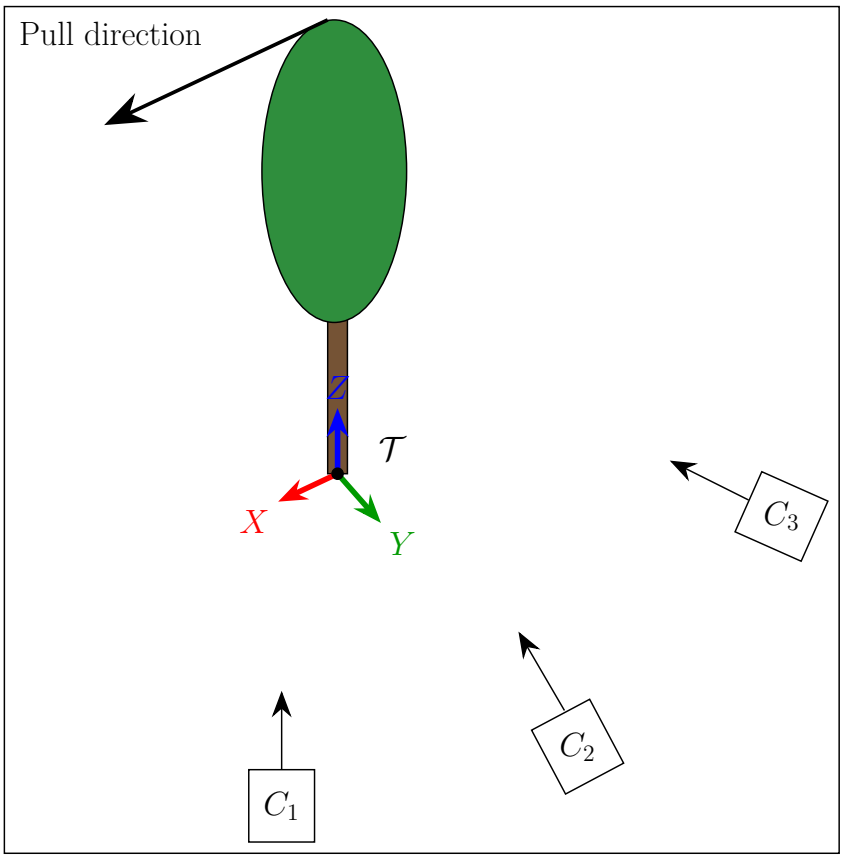

Figure 3.3: Tree experimental setup. Camera positions shown as C_1, C_2 and C_3 and the pull direction from where the rope was attached on the trees. The coordinate frame $\mathcal{T}$ is centred at the tree base, i.e., the point at which the stem was fixed to the table, with X aligned with the pull direction, Z aligned with the stem axis and Y aligned with the direction opposing the viewing ray from C_2.

3.4.2 Tracked Features

Red tape strips were attached at multiple locations along each tree to serve as visually distinguishable tracking features. Red was chosen because its hue is well-separated from the green and brown tones of the tree, making it straightforward to isolate using HSV thresholding as described in Section 3.2.1. Features were placed at several heights along the stem and on lateral branches, enabling the dynamics of structurally distinct tree components to be compared within a single experiment.

The initial feature correspondences established across camera views for each tree are shown in figures 3.4, 3.5 and 3.6. For the spruce and pine trees, feature F1 corresponds to the rope attachment point near the tree top and therefore represents the location of maximum excitation.

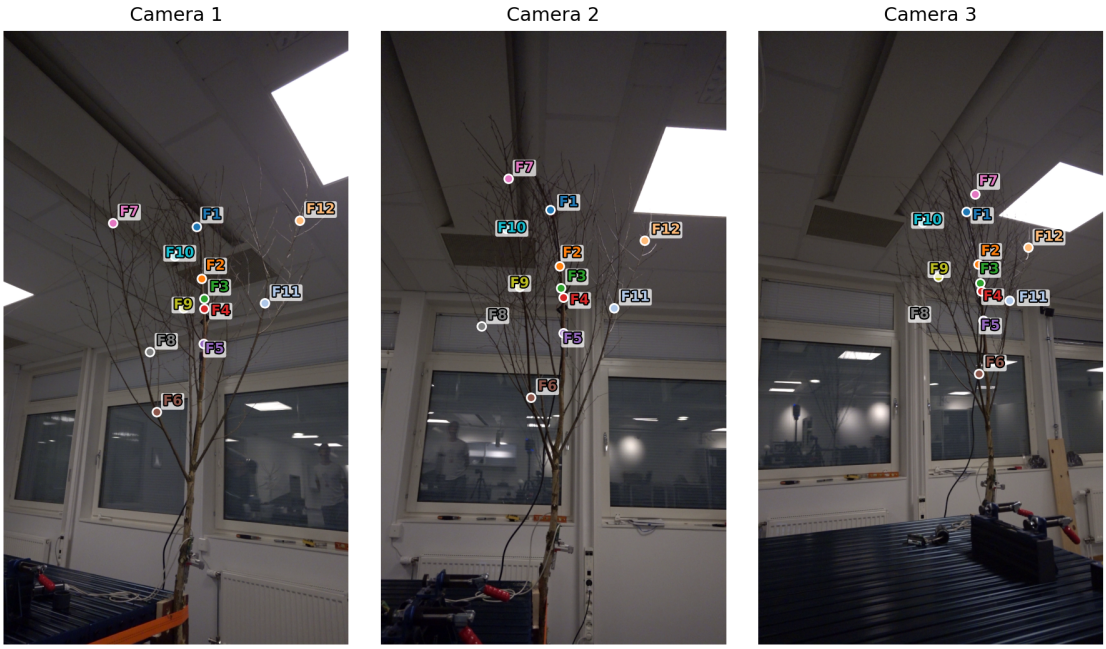

Figure 3.4: Initial feature correspondences across camera views for birch.

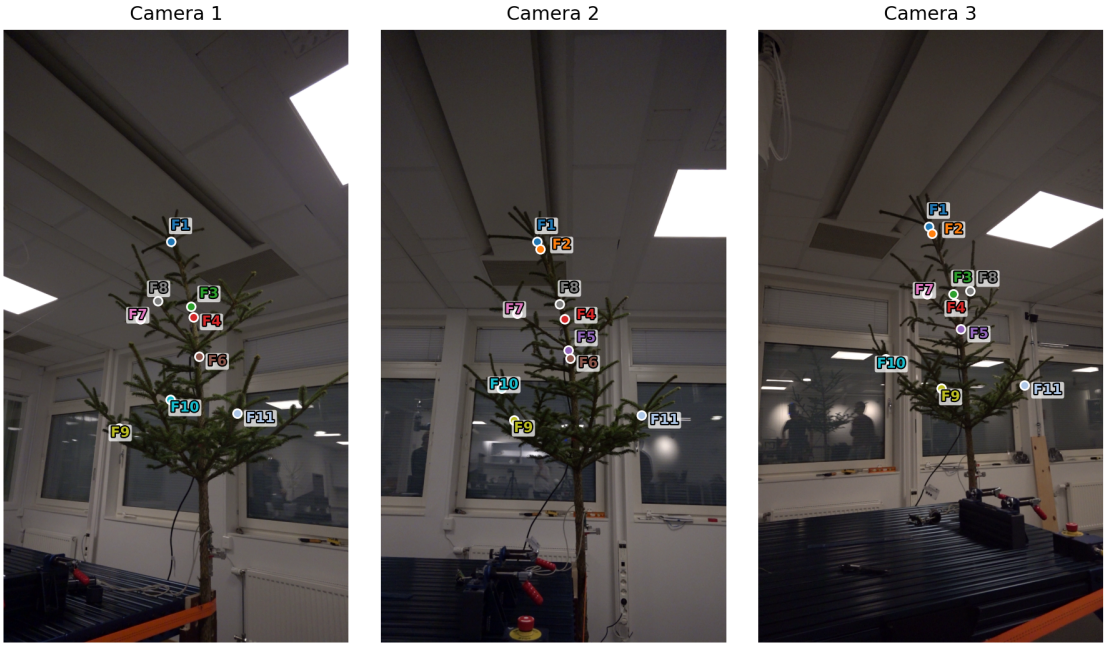

Figure 3.5: Initial feature correspondences across camera views for spruce.

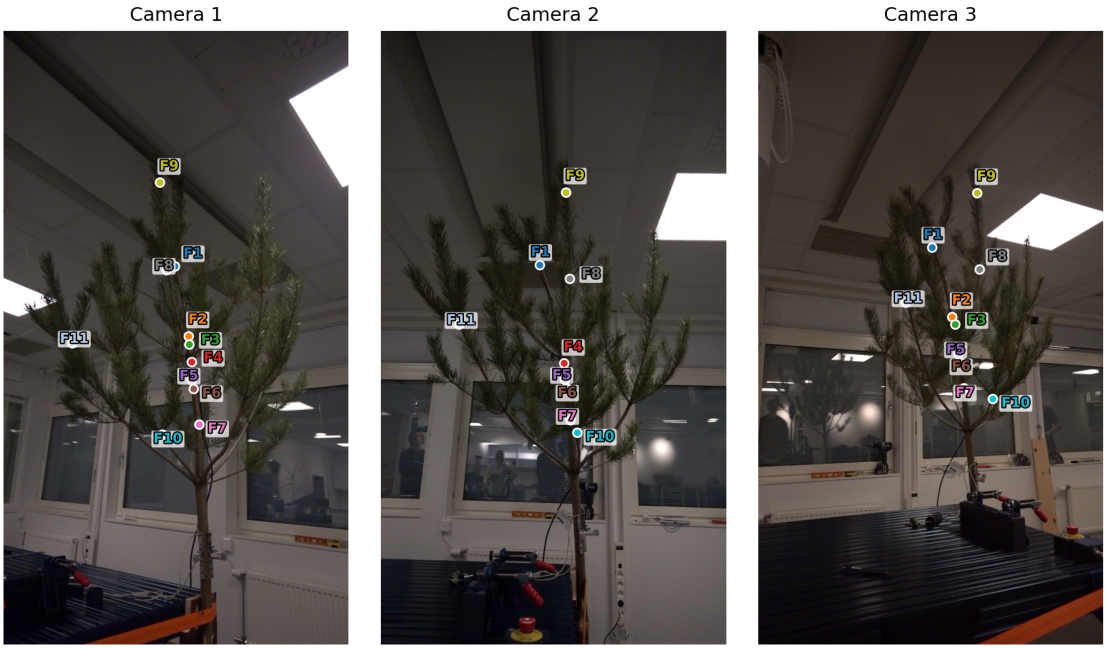

Figure 3.6: Initial feature correspondences across camera views for pine.

3.4.3 Pull-and-Release Protocol

Three pull-and-release tests were performed for each tree. In each test, a rope was attached between the robot gripper and a point near the top of the stem. The robot moved to a fixed target position, displacing the tree top from its equilibrium and held the position briefly before opening the gripper to release the rope. This allowed the tree to oscillate freely. The rope attachment point and the robot target position were kept consistent across the three repetitions for each species to ensure comparable excitation conditions.

3.4.4 Tree Coordinate Frame

Reconstructed trajectories are initially expressed in the camera coordinate frame established during extrinsic calibration, as described in Section 3.1.4. To express the motion in a frame aligned with the tree geometry, a tree-specific coordinate frame $\mathcal{T}$ was defined for each tree, with the Z-axis aligned with the stem, the X-axis aligned with the pull direction and the origin at the base of the stem as illustrated in Figure 3.3.

The frame was constructed from the triangulated three-dimensional positions of two stem points in the camera coordinate system, a base position $\boldsymbol{p}_{\text{base}} \in \mathbb{R}^3$ near the bottom of the stem and a top position $\boldsymbol{p}_{\text{top}} \in \mathbb{R}^3$ near the tree top. The Z-axis was defined along the stem directed upward,

$$\hat{\boldsymbol{e}}_Z = \frac{\boldsymbol{p}_{\text{top}} - \boldsymbol{p}_{\text{base}}}{\|\boldsymbol{p}_{\text{top}} - \boldsymbol{p}_{\text{base}}\|}. \tag{3.26}$$

The X-axis was defined perpendicular to both the stem and the principal viewing direction $\hat{\boldsymbol{d}}_2$ of camera 2,

$$\hat{\boldsymbol{e}}_X = \frac{\hat{\boldsymbol{e}}_Z \times \hat{\boldsymbol{d}}_2}{\|\hat{\boldsymbol{e}}_Z \times \hat{\boldsymbol{d}}_2\|}. \tag{3.27}$$

The Y-axis was then obtained as,

$$\hat{\boldsymbol{e}}_Y = \hat{\boldsymbol{e}}_Z \times \hat{\boldsymbol{e}}_X, \tag{3.28}$$

completing a right-handed coordinate system in which the X-axis is aligned with the pulling direction and the Y-axis being approximately in the opposite direction of camera 2's principal direction. The rotation matrix expressing the camera coordinate system in the tree frame is then,

$$R_{\mathcal{TC}} = \begin{bmatrix} \hat{\boldsymbol{e}}_X^T \\ \hat{\boldsymbol{e}}_Y^T \\ \hat{\boldsymbol{e}}_Z^T \end{bmatrix} \in SO(3), \tag{3.29}$$

with the translation vector placing the origin at $\boldsymbol{p}_{\text{base}}$,

$$\boldsymbol{t}_{\mathcal{TC}} = -R_{\mathcal{TC}}\, \boldsymbol{p}_{\text{base}} \in \mathbb{R}^3. \tag{3.30}$$

The full rigid body transformation from the camera coordinate system to the tree frame is then,

$$\boldsymbol{X}^{\mathcal{T}} = \begin{bmatrix} R_{\mathcal{TC}} & \boldsymbol{t}_{\mathcal{TC}} \end{bmatrix} \tilde{\boldsymbol{X}}^{\mathcal{C}}, \tag{3.31}$$

where $\boldsymbol{X}^{\mathcal{T}} \in \mathbb{R}^3$ is a point expressed in the tree frame and $\tilde{\boldsymbol{X}}^{\mathcal{C}} \in \mathbb{R}^4$ denotes the corresponding homogeneous coordinates in the camera frame. This procedure was repeated independently for each tree.

3.4.5 Data Processing

The extrinsic parameters were estimated from the spruce and pine extrinsic recording and the birch extrinsic recording, as described in Section 3.1.4. For all video recordings, the three video streams were manually synchronised using the flashlight reference, after which the features shown in figures 3.4, 3.5 and 3.6 were tracked using the pipeline described in Section 3.2. The HSV thresholds $H \in [0, 10] \cup [170, 179]$, $S \in [60, 255]$ and $V \in [60, 255]$, Singer model parameters $\alpha = 20\ \text{s}^{-1}$ and $\sigma_m^2 = 1e7\ \text{mm}^2\,\text{s}^{-4}$ and measurement noise standard deviation $r = 5.0$ px were used throughout. A track was considered successful if manual visual inspection confirmed that the tracked feature at the end of the recording corresponded to the same physical marker as at initialisation, i.e. that the tracker had not diverged or been associated by a neighbouring marker or background region. The resulting trajectories were transformed to the tree coordinate frame $\mathcal{T}$ as described in Section 3.4.4.

3.4.5.1 Spectral Analysis and Oscillator Fitting

For each tracked feature, the three-dimensional trajectory $\{\boldsymbol{X}(t_j)\}_{j=0}^{J-1}$ was first projected onto the basis defined by principal component analysis (PCA) as described in Section 2.5.2. PCA was applied to the full trajectory, yielding an orthonormal basis $\{\boldsymbol{v}_1, \boldsymbol{v}_2, \boldsymbol{v}_3\}$ whose first vector $\boldsymbol{v}_1$ points in the direction of maximum displacement variance. The scalar first principal component (PC1) signal,

$$s(t_j) = \boldsymbol{v}_1^T \Big(\boldsymbol{X}(t_j) - \boldsymbol{X}_{\text{med}} \Big), \tag{3.32}$$

where $\boldsymbol{X}_{\text{med}}$ is the sample median of the trajectory, captures the dominant oscillation direction. The median rather than the mean is used as the reference position to reduce sensitivity to the large initial displacement at the moment of release. All subsequent spectral analysis was performed on $s(t_j)$.

3.4.5.1.1 Signal Truncation Pull-and-release oscillations decay over time, so the later portion of each PC1 signal consists primarily of low-amplitude noise rather than meaningful oscillatory motion. To prevent noise-dominated frames from biasing the spectral estimate, the usable length of each PC1 signal was determined automatically prior to spectral analysis using a rolling RMS criterion, with the energy-weighted Welch estimator providing additional robustness against residual low-energy frames within the retained window.

The rolling RMS was computed over non-overlapping windows of $W_r = \lfloor 0.2 \cdot f_s \rfloor$ samples, where $f_s = 1/T = 50$ Hz is the sampling rate. The noise floor σ_η was estimated as the standard deviation of the final 20% of the signal. The usable signal length was defined as,

$$T^* = \max\Big\{ t_j : \ \text{RMS}(t_j) > k\, \sigma_\eta \Big\}, \tag{3.33}$$

where $k = 2.5$ is a fixed threshold multiplier. Each PC1 signal was truncated to the interval $[t_0, T^*]$ before spectral estimation.

3.4.5.1.2 Energy-Weighted Welch PSD The truncated PC1 signal was analysed using the energy-weighted Welch estimator described in Section 2.5.3 and equation (2.162). The estimator was configured with a segment length of $L = 250$ samples (5 s), giving a frequency resolution of $\Delta f = 1/(LT) = 0.2$ Hz, a step size of $h = 63$ samples (75% overlap) and a Hann window. Spectra were computed independently for each feature and each pull-and-release test.

3.4.5.1.3 Damped Harmonic Oscillator In addition to the spectral estimate, an underdamped harmonic oscillator model was fitted to each truncated PC1 signal $s(t_j)$, $t_j \in [t_0, T^*]$, as described in Section 2.5.3.3. The model parameters A, δ, f_0 and ϕ were estimated by solving the non-linear least-squares problem in equation (2.164) using SciPy's `optimize.curve_fit` routine [49]. The optimisation was initialised with the amplitude to the signal value of largest magnitude, retaining its sign, so that the initial guess captures the starting direction of the oscillation, the frequency to

the dominant peak of the energy-weighted Welch PSD, the damping rate to a small fixed value $\delta = 0.05$ and the phase to $\phi = 0$. The fit was performed independently for each feature, pull-and-release test and tree.

4

Results

This chapter presents the results of the experiments described in chapter 3. Section 4.1 reports the intrinsic calibration results, which are shared across both experiments. Section 4.2 presents the results of the controlled motion experiment and Section 4.3 presents the results of the tree motion experiments.

4.1 Intrinsic Calibration

Table 4.1 summarises the intrinsic calibration results for each camera, reported as mean and standard deviation across the five calibration datasets. Detailed per-dataset results including distortion coefficients are provided in Table A.1 in the appendix.

All three cameras achieve sub-pixel reprojection errors with means ranging from 0.2840 to 0.3493 px. The estimated focal lengths are consistent across cameras, with f_x and f_y in the range mean 1291-1298 px as expected for cameras of the same model. Camera 2 achieves the lowest mean reprojection error and the smallest parameter variation across datasets. Camera 3 yields the highest reprojection error at 0.3493 px, though still well below one pixel. Camera 1 exhibits a notably larger standard deviation in the principal point c_y ($\pm$14.2 px) compared to cameras 2 and 3 ($\pm$3.9 and $\pm$9.3 px respectively). This is driven by dataset 1, in which $c_y = 1005.4$ px deviates substantially from the 971-979 px range observed across the remaining four datasets, as detailed in the appendix.

Table 4.1: Intrinsic calibration results for the three cameras. Values are reported as mean $\pm$ standard deviation across the five calibration datasets.

Camera	RPE_{RMS} [px]	f_x [px]	f_y [px]	c_x [px]	c_y [px]
1	0.3024 ± 0.0230	1298.3 ± 5.5	1295.3 ± 4.3	538.3 ± 5.4	980.9 ± 14.2
2	0.2840 ± 0.0156	1296.7 ± 3.2	1293.7 ± 2.2	522.2 ± 4.0	951.9 ± 3.9
3	0.3493 ± 0.0157	1293.0 ± 3.8	1291.0 ± 3.8	570.7 ± 8.9	979.8 ± 9.3

4.2 Controlled Motion Experiment

This section reports the results of the controlled motion experiment. The quality of the individual sensor measurements is assessed first, covering LiDAR scan quality in Section 4.2.1 and extrinsic calibration in Section 4.2.2. The coordinate frame registration results are then reported in Section 4.2.3, followed by the sensor comparisons in Section 4.2.4 and camera calibration sensitivity in Section 4.2.5.

4.2.1 LiDAR Scan Quality

The LiDAR accuracy was characterised by two measures, the single-scan radius error, which reflects the precision of an individual scan, and the co-registration error, which quantifies the alignment between scans.

Table 4.2 reports the radius error from free-radius sphere fitting for each scan in $\mathcal{S}$, used as an indicator of individual scan quality. The mean radius error across all scans is 0.10 mm with a standard deviation of 0.75 mm. The majority of scans yield errors well within the scanner's specified ranging error of ±2 mm at 10 m. The two largest errors occur at $\boldsymbol{P}_0$, L_3 (−1.52 mm) and $\boldsymbol{P}_3$, L_1 (+1.33 mm), the former from scan position L_3, which was noticeably further from the sphere than positions L_1 and L_2. Both values nonetheless remain within the specified ranging error.

Table 4.3 reports the co-registration error for each endpoint, defined as the Euclidean distance between sphere centres fitted independently from two scan positions. The errors range from 5.32 mm at $\boldsymbol{P}_0$ to 11.61 mm at $\boldsymbol{P}_3$, with a mean of 9.15 mm and a standard deviation of 2.73 mm. Since each endpoint is estimated as the midpoint of the two independently fitted sphere centres, this corresponds to an effective endpoint uncertainty of 4.58 mm. These values are roughly an order of magnitude larger than the single-scan radius errors reported above.

Table 4.2: Sphere fitting results per scan where r_{opt} is the fitted radius and $r_{\text{true}} = 20.0$ mm is the measured radius.

Endpoint	Scan position	$r_{\text{opt}} - r_{\text{true}}$ [mm]
$\boldsymbol{P}_0$	L_1	0.12
$\boldsymbol{P}_0$	L_3	−1.52
$\boldsymbol{P}_1$	L_1	0.17
$\boldsymbol{P}_1$	L_2	0.03
$\boldsymbol{P}_2$	L_1	0.09
$\boldsymbol{P}_2$	L_2	0.01
$\boldsymbol{P}_3$	L_1	1.33
$\boldsymbol{P}_3$	L_2	0.57
Mean		0.10
Std		0.75

Table 4.3 reports the co-registration error for each endpoint, defined as the Euclidean

distance between sphere centres fitted independently from two scan positions. The errors range from 5.32 mm at $\boldsymbol{P}_0$ to 11.61 mm at $\boldsymbol{P}_3$, with a mean of 9.15 mm and a standard deviation of 2.73 mm. These values are considerably larger than the individual scan quality errors reported above, indicating that the co-registration step is the dominant source of LiDAR uncertainty rather than the scanner's intrinsic ranging precision.

Table 4.3: Co-registration error defined as the Euclidean distance between fitted sphere centers from two independent scan positions.

Endpoint	Scan 1	Scan 2	Distance [mm]
$\boldsymbol{P}_0$	L_1	L_3	5.32
$\boldsymbol{P}_1$	L_1	L_2	10.44
$\boldsymbol{P}_2$	L_1	L_2	9.24
$\boldsymbol{P}_3$	L_1	L_2	11.61
Mean			9.15
Std			2.73

4.2.2 Extrinsic Calibration

Table 4.4 reports the stereo reprojection errors for the pre- and post-experiment extrinsic calibration sets. The pre-experiment set achieves 0.6658 px for the $(1, 2)$ pair and 1.2417 px for the $(2, 3)$ pair. The post-experiment set is more consistent across pairs, with errors of 0.7697 px and 0.7283 px for $(1, 2)$ and $(2, 3)$ respectively, and was estimated from a larger number of frames. The effect of the extrinsic calibration choice on tracking accuracy is reported in Section 4.2.5.

Table 4.4: Stereo reprojection errors for the pre- and post-experiment extrinsic calibration sets using intrinsic set 3. Errors are reported per camera pair together with the number of image frames used.

Extrinsic set	RPE_{RMS} [px]		Images	
	$(1, 2)$	$(2, 3)$	$(1, 2)$	$(2, 3)$
Pre-experiment	0.6658	1.2417	37	41
Post-experiment	0.7697	0.7283	59	53

4.2.3 Coordinate Frame Registration

The three sensors were registered to $\mathcal{W}$ by different procedures, each assessed by a different measure. The video registration is assessed by a reprojection error in pixels and the robot by a point-correspondence residual in millimetres, while the LiDAR registration involves two stages, a plane fit quantified by its RMS residual in millimetres and a 2D grid alignment assessed qualitatively from the point-occupancy

figure. These measures are therefore not directly comparable.

The video-to-world transformation was estimated by jointly minimising the reprojection error of the checkerboard corners across all three cameras with the intrinsic and extrinsic parameters held fixed, as described in Section 3.3.4.2 and illustrated in Figure 4.1. The resulting per-camera reprojection errors were 0.83 px, 1.10 px and 0.89 px for cameras 1, 2 and 3 respectively, camera 2 being the only one to exceed one pixel. These are noticeably larger than the intrinsic calibration errors (Table 4.1) and comparable to the stereo calibration errors (Table 4.4), as expected since the registration reuses the fixed calibration parameters rather than re-estimating them.

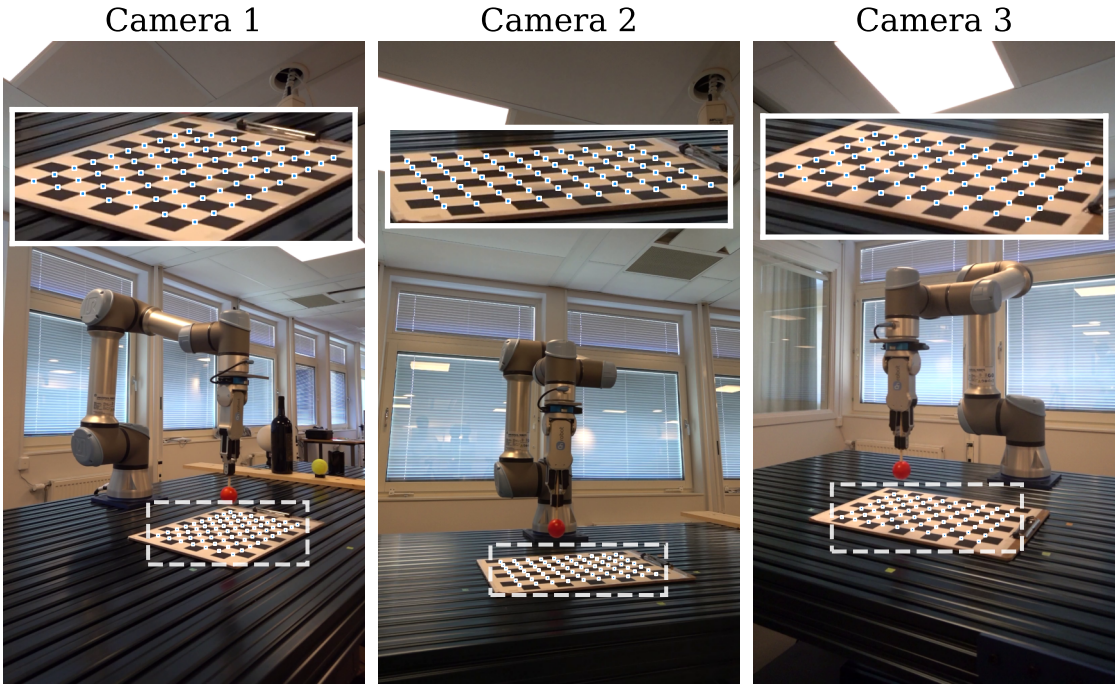

Figure 4.1: Reprojection of the 3D checkerboard inner corners into each camera view using the jointly optimised board pose. The board pose was estimated by minimising the reprojection error across all three cameras simultaneously.

The robot-to-world transformation was estimated by solving the absolute orientation problem from four checkerboard corner correspondences, as described in Section 3.3.4.1. The resulting RMS residual was 1.29 mm.

The LiDAR-to-world transformation was estimated from 69 487 extracted white checkerboard points, as described in Section 3.3.4.3. The plane-fit RMS residual was 0.47 mm, indicating that the extracted points lie close to a common plane. The subsequent 2D alignment to the checkerboard grid is shown in Figure 4.2, where the large majority of points fall on white squares after alignment, with a smaller number on black squares or outside of the board.

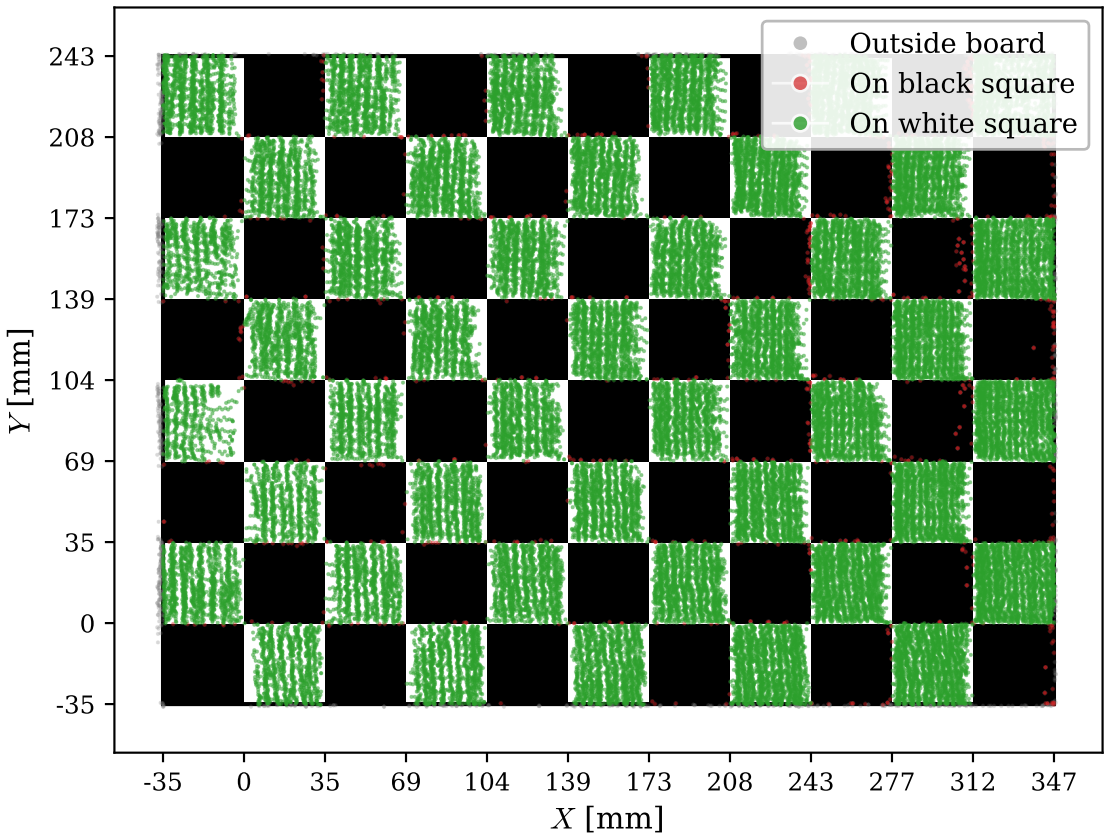

Figure 4.2: Filtered white LiDAR points projected onto the fitted checkerboard plane and aligned to board coordinates via a white-square occupancy optimisation, used to estimate the LiDAR-to-world coordinate transformation. Green and red points indicate white LiDAR points on white and black squares respectively, while grey points fall outside the board boundary.

4.2.4 Sensor Comparisons

Table 4.5 reports the pairwise Euclidean distances between the endpoint estimates $\boldsymbol{P}_0$, $\boldsymbol{P}_1$, $\boldsymbol{P}_2$ and $\boldsymbol{P}_3$ across all three sensors, together with the signed component-wise differences. The Robot-LiDAR distances are consistently the smallest across all four endpoints, with a mean of 2.80 mm and a standard deviation of 0.81 mm. The Video-LiDAR and Video-Robot comparisons yield larger mean distances of 4.40 mm and 6.16 mm respectively. The standard deviation across endpoints is small relative to the mean in all three pairs, indicating that the pairwise disagreement is approximately consistent across the four measured positions.

The component-wise breakdown shows that the Euclidean error is dominated by a systematic bias in the Z-axis, which is the direction normal to the checkerboard plane, across all three sensor pairs. For the Video-Robot pair, $\bar{e}_Z = -5.82$ mm accounts for the majority of the 6.16 mm mean Euclidean distance, while $\bar{e}_X = 1.87$ mm and $\bar{e}_Y = -0.03$ mm are considerably smaller. The Y bias is near zero in all three pairs, with $|\bar{e}_Y| \leq 0.39$ mm. A Z bias is present in all pairs, $\bar{e}_Z = 2.29$

mm for Robot-LiDAR and $\bar{e}_Z = -3.53$ mm for Video-LiDAR. The standard deviation of the Z component is small relative to its mean in all cases, with $\sigma_Z \leq 1.28$ mm.

Table 4.5: Pairwise differences between endpoint estimates. The upper part of the table reports the Euclidean distance $\|\Delta \boldsymbol{P}\|$ per endpoint and its mean ± standard deviation. The lower part of the table reports the signed component-wise differences, defined as the first sensor minus the second, expressed as $\bar{e} \pm \sigma$ [mm].

	Robot-LiDAR [mm]	Video-LiDAR [mm]	Video-Robot [mm]
$\boldsymbol{P}_0$	3.79	3.46	5.04
$\boldsymbol{P}_1$	1.80	4.40	5.38
$\boldsymbol{P}_2$	2.86	4.54	6.35
$\boldsymbol{P}_3$	2.74	5.19	7.87
$\overline{\|\Delta \boldsymbol{P}\|} \pm \sigma_{\|\Delta \boldsymbol{P}\|}$	2.80 ± 0.81	4.40 ± 0.71	6.16 ± 1.27
$\bar{e}_X \pm \sigma_X$	-0.47 ± 1.77	1.40 ± 2.43	1.87 ± 0.71
$\bar{e}_Y \pm \sigma_Y$	0.39 ± 0.55	0.36 ± 0.31	-0.03 ± 0.49
$\bar{e}_Z \pm \sigma_Z$	2.29 ± 0.44	-3.53 ± 0.93	-5.82 ± 1.28

Figure 4.3 shows the per-axis video-robot residual over the full recorded trajectory, evaluated using intrinsic calibration set 3 and the post-experiment extrinsic calibration. The residuals in x and y have biases of $\bar{e}_x = 1.60$ mm and $\bar{e}_y = 0.11$ mm with standard deviations of $\sigma_x = 0.58$ mm and $\sigma_y = 0.78$ mm. The z residual exhibits a larger systematic bias of $\bar{e}_z = -5.64$ mm with a standard deviation of $\sigma_z = 0.79$ mm. Note that the z bias reported here differs slightly from the -5.82 mm in table 4.5, as the former is computed over the full trajectory while the latter is derived from four averaged endpoint positions. In all three axes, the residual magnitude increases near the turning points of the trajectory.

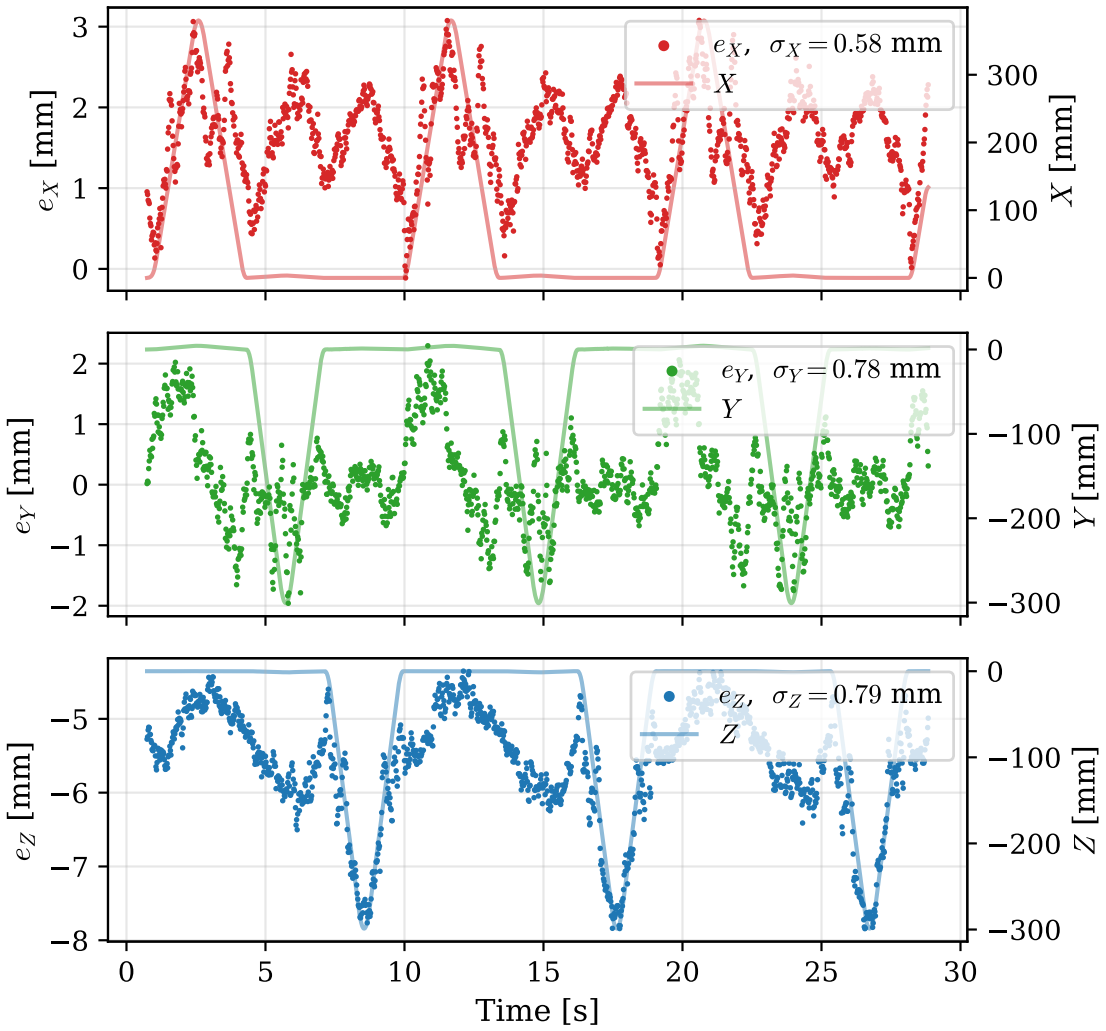

Figure 4.3: Per-axis video-robot residuals e_X, e_Y and e_Z over time, evaluated using intrinsic calibration set 3 and the post-experiment extrinsic calibration recording. The robot trajectory (right axis) is overlaid in the background to illustrate the correlation between residual magnitude and trajectory position.

The segmentation size of the sphere marker, measured as the number of foreground pixels across all cameras and time steps, reached a maximum of approximately 3500 pixels. Assuming the sphere projects as a circle of radius $r = 20.0$ mm, the physical projected area is $A = \pi r^2 \approx 1257$ mm^2, yielding an image-to-scene scale of $\sqrt{A/A_{\text{px}}} = \sqrt{1257/3500} \approx 0.60$ mm/px. The coordinate-wise standard deviations of the video-robot trajectory residuals are consistent with this scale.

Finally, Table 4.6 reports the inter-endpoint distances $d_j = \|\boldsymbol{P}_j - \boldsymbol{P}_0\|$, $j = 1, 2, 3$, computed independently within each sensor's own coordinate frame, together with the signed differences between sensors. The distances are broadly consistent across sensors, $\boldsymbol{P}_0$-$\boldsymbol{P}_1$ ranges from 376.45 mm (LiDAR) to 381.69 mm (Video), while $\boldsymbol{P}_0$-$\boldsymbol{P}_2$ and $\boldsymbol{P}_0$-$\boldsymbol{P}_3$ show closer agreement across sensors. The mean signed differences are 0.71 mm, 2.38 mm and 1.67 mm for the Robot-LiDAR, Video-LiDAR and Video-

Robot pairs respectively. These errors are smaller than the absolute endpoint position errors reported in Table 4.5.

Table 4.6: Pairwise inter-point distances and signed differences between sensor estimates. The upper part reports the Euclidean distance $\|\cdot\|$ per pair per method. The lower part reports the signed differences, defined as the first sensor minus the second, expressed as $\bar{e} \pm \sigma$ [mm].

	LiDAR [mm]	Robot [mm]	Video [mm]
$\boldsymbol{P}_0$–$\boldsymbol{P}_1$	376.45	380.08	381.69
$\boldsymbol{P}_0$–$\boldsymbol{P}_2$	301.60	300.56	301.36
$\boldsymbol{P}_0$–$\boldsymbol{P}_3$	299.25	298.79	301.38
	Robot-LiDAR [mm]	Video-LiDAR [mm]	Video-Robot [mm]
$\boldsymbol{P}_0$–$\boldsymbol{P}_1$	3.63	5.24	1.61
$\boldsymbol{P}_0$–$\boldsymbol{P}_2$	-1.04	-0.24	0.79
$\boldsymbol{P}_0$–$\boldsymbol{P}_3$	-0.47	2.13	2.60
$\bar{e} \pm \sigma$	0.71 ± 2.54	2.38 ± 2.75	1.67 ± 0.90

4.2.5 Camera Calibration Sensitivity

To assess whether the choice of calibration set affects tracking, the video-robot residual was recomputed for each intrinsic calibration dataset and for both extrinsic recordings.

Table 4.7 reports the video-robot tracking residual for each of the five intrinsic calibration datasets evaluated using the post-experiment extrinsic calibration. The mean Euclidean residual $\overline{\|\boldsymbol{e}\|}$ ranges from 5.63 mm to 5.94 mm across the five sets, a spread of 0.31 mm. The per-axis biases show comparable stability, $\bar{e}_X$ varies between 1.47 and 1.73 mm, $\bar{e}_Y$ between -0.05 and 0.37 mm and $\bar{e}_Z$ between -5.32 and -5.64 mm. The per-axis standard deviations are consistent across all five sets, ranging from 0.56-0.64 mm in X, 0.69-0.98 mm in Y and 0.67-0.79 mm in Z. Dataset 1, which carries the anomalous camera 1 c_y estimate noted in Section 4.1, produces a residual within this same spread.

Table 4.8 reports the video-robot tracking residual for the pre- and post-experiment extrinsic calibration recordings evaluated using intrinsic dataset 3. The mean Euclidean residual is 5.28 mm for the pre-experiment recording and 5.94 mm for the post-experiment recording, a difference of 0.66 mm. The Z component changes from $\bar{e}_Z = -5.00$ mm to $\bar{e}_Z = -5.64$ mm between the two recordings, while $\bar{e}_X$ shifts from 1.31 mm to 1.60 mm and $\bar{e}_Y$ from 0.23 mm to 0.11 mm. The per-axis standard deviations are similar across the two sets. The elevated $(2, 3)$ reprojection error of the pre-experiment set, reported in Section 4.2.2, is therefore not reflected in a larger tracking residual for that recording.

Table 4.7: Sensitivity of the video-robot tracking residual to the choice of intrinsic calibration set, evaluated using the post-experiment extrinsic calibration recording. $\bar{e}$ and σ denote the mean and standard deviation of the per-axis residual over the full recorded trajectory.

Intrinsic set	X [mm]		Y [mm]		Z [mm]		$\overline{\|\boldsymbol{e}\|}$ [mm]
	$\bar{e}_X$	σ_X	$\bar{e}_Y$	σ_Y	$\bar{e}_Z$	σ_Z	
1	1.73	0.64	−0.05	0.98	−5.51	0.77	5.90
2	1.47	0.65	0.37	0.74	−5.44	0.67	5.73
3	1.60	0.58	0.11	0.78	−5.64	0.79	5.94
4	1.48	0.65	0.22	0.69	−5.43	0.68	5.71
5	1.62	0.56	0.21	0.74	−5.32	0.73	5.63
Range							0.31

Extrinsic set	X [mm]		Y [mm]		Z [mm]		$\overline{\|\boldsymbol{e}\|}$ [mm]
	$\bar{e}_X$	σ_X	$\bar{e}_Y$	σ_Y	$\bar{e}_Z$	σ_Z	
Pre-experiment	1.31	0.58	0.23	0.75	−5.00	0.49	5.28
Post-experiment	1.60	0.59	0.11	0.78	−5.64	0.79	5.94
Difference							0.66

Table 4.8: Sensitivity of the video-robot tracking residual to the choice of extrinsic calibration recording, evaluated using intrinsic calibration set 3. $\bar{e}$ and σ denote the mean and standard deviation of the per-axis residual over the full recorded trajectory.

4.3 Tree Motion Experiments

This section presents the results of the pull-and-release experiments conducted on the three tree species, birch, spruce and pine. Each tree was subjected to three repeated pull-and-release tests from which three-dimensional feature trajectories were reconstructed using the multi-camera pipeline described in 3.4. The section is structured as follows. The quality of the extrinsic calibration used for each tree is reported in Section 4.3.1. Tracking performance across features and tests is summarised in Section 4.3.2. The reconstructed three-dimensional trajectories are shown in Section 4.3.3, followed by a look at the oscillatory response in the time domain in Section 4.3.4, covering stem feature behaviour, stem-branch differences and test reproducibility. Finally, Section 4.3.5 presents the frequency and damping estimates obtained from the energy-weighted Welch method and the damped harmonic oscillator fit.

4.3.1 Extrinsic Calibration

Table 4.9 reports the stereo reprojection errors for the two extrinsic calibration sets used in the tree experiments. The spruce and pine set achieves sub-pixel accuracy

across both camera pairs, with errors of 0.5086 px and 0.5791 px for pairs $(1, 2)$ and $(2, 3)$ respectively. The birch set yields a sub-pixel error of 0.7754 px for pair $(1, 2)$, while pair $(2, 3)$ produces a noticeably elevated error of 1.3290 px. A similarly elevated single-pair error in the controlled motion experiment did not worsen the tracking residual, so the birch $(2, 3)$ error is unlikely to degrade the reconstruction. The reprojection errors are of the same order as those of the controlled experiment.

Table 4.9: Stereo reprojection errors for the tree experiment extrinsic calibration sets. Errors are reported per camera pair together with the number of image frames used.

Extrinsic set	RPE_{RMS} [px]		Images	
	$(1, 2)$	$(2, 3)$	$(1, 2)$	$(2, 3)$
Spruce & Pine	0.5086	0.5791	47	51
Birch	0.7754	1.3290	55	57

4.3.2 Tracking Success

Table 4.10 summarises the tracking outcome for each initialised feature across the three pull-and-release tests for birch, spruce and pine, with the feature locations shown in figures 3.4, 3.5 and 3.6. In total, 12, 11 and 11 features were initialised for the birch, spruce and pine respectively, of which 7, 10 and 3 were tracked successfully across all three tests. Tracking was therefore most reliable for the spruce, followed by the birch, while the majority of pine features could not be tracked in any test. The pine differed from the other two in having denser foliage and longer needles, and its stem markers were more closely spaced. The failed tracks either diverged or became associated with a neighbouring marker.

For the successfully tracked features, the visibility statistics in Table 4.10 show that most frames were supported by two- or three-camera observations and that complete loss of visibility was rare. The trajectories therefore seldom relied on prediction alone.

Table 4.10: Tracking success and camera visibility for each feature across the three pull-and-release tests per tree. ✓ indicates that a feature was tracked successfully throughout the full recording and × indicates that a tracking failed due to diverging or being associated with another feature in the last frame. The visibility column reports the average percentage of frames in successful runs where the feature was observed by 3, 2, 1 or 0 cameras respectively.

Feature	Location	Test 1	Test 2	Test 3	Tracked	Visibility [3/2/1/0 cams] (%)
Birch						
F1	Stem	✓	✓	✓	3/3	88.9/ 11.1/ 0.0/ 0.0
F2	Stem	✓	✓	✓	3/3	92.6/ 7.4/ 0.0/ 0.0
F3	Stem	×	✓	✓	2/3	78.5/ 19.4/ 1.7/ 0.3
F4	Stem	×	×	×	0/3	–
F5	Stem	✓	×	✓	2/3	99.9/ 0.0/ 0.1/ 0.0
F6	Branch 1	✓	✓	✓	3/3	100.0/ 0.0/ 0.0/ 0.0
F7	Branch 1	✓	✓	✓	3/3	96.9/ 2.9/ 0.2/ 0.0
F8	Branch 2	✓	✓	✓	3/3	99.9/ 0.1/ 0.0/ 0.0
F9	Branch 3	✓	×	×	1/3	3.0/ 96.0/ 0.7/ 0.3
F10	Branch 4	×	✓	✓	2/3	63.0/ 30.7/ 5.7/ 0.6
F11	Branch 5	✓	✓	✓	3/3	99.3/ 0.7/ 0.0/ 0.0
F12	Branch 5	✓	✓	✓	3/3	97.9/ 2.1/ 0.0/ 0.0
Spruce						
F1	Stem	✓	✓	✓	3/3	21.2/ 45.9/ 24.6/ 8.3
F2	Stem	×	✓	✓	2/3	35.1/ 32.9/ 26.1/ 5.8
F3	Stem	✓	✓	✓	3/3	53.7/ 46.3/ 0.0/ 0.0
F4	Stem	✓	✓	✓	3/3	92.1/ 7.9/ 0.0/ 0.0
F5	Stem	✓	✓	✓	3/3	2.5/ 97.1/ 0.4/ 0.0
F6	Stem	✓	✓	✓	3/3	78.1/ 18.9/ 3.0/ 0.0
F7	Branch 1	✓	✓	✓	3/3	100.0/ 0.0/ 0.0/ 0.0
F8	Branch 2	✓	✓	✓	3/3	98.8/ 1.2/ 0.0/ 0.0
F9	Branch 3	✓	✓	✓	3/3	33.8/ 66.2/ 0.0/ 0.0
F10	Branch 4	✓	✓	✓	3/3	20.9/ 64.8/ 13.3/ 1.1
F11	Branch 5	✓	✓	✓	3/3	89.5/ 8.7/ 1.8/ 0.0
Pine						
F1	Stem	✓	✓	✓	3/3	91.5/ 8.5/ 0.0/ 0.0
F2	Stem	×	×	×	0/3	–
F3	Stem	×	×	×	0/3	–
F4	Stem	×	×	×	0/3	–
F5	Stem	×	×	×	0/3	–
F6	Stem	×	×	×	0/3	–
F7	Stem	×	×	×	0/3	–
F8	Branch 1	✓	✓	✓	3/3	94.8/ 5.2/ 0.0/ 0.0
F9	Branch 1	×	×	×	0/3	–
F10	Branch 2	×	×	×	0/3	–
F11	Branch 3	✓	✓	✓	3/3	0.5/ 98.6/ 1.0/ 0.0

4.3.3 Motion Trajectories

Figures 4.4, 4.5 and 4.6 show the coordinate-plane projections of all successfully tracked features during a representative pull-and-release test for each tree. All axes within a tree share the same scale in millimetres, so each figure directly reflects the spatial extent of the motion and the physical separation between features. The trees were pulled along the X direction, and the Z-axis is aligned with the stem pointing upward. Stem features cluster near the origin in the XY-plane, reflecting their position along the stem axis, while branch features lie further out.

The oscillations are largest in the pull direction but not confined to it, nor aligned

with the world axes. Because a branch or stem bends far more easily than it stretches, each feature oscillates in the plane transverse to it. As a result, the motion appears in several coordinate directions even though the excitation was applied along one. The spruce branch feature F7, for instance, shows a clear oscillation in the Z direction in Figure 4.5.

The spruce figure also shows the trajectories of the two uppermost stem features, F1 and F2, becoming intermixed (Figure 4.5). These features were positioned close together on the stem, which led to data association issues during tracking.

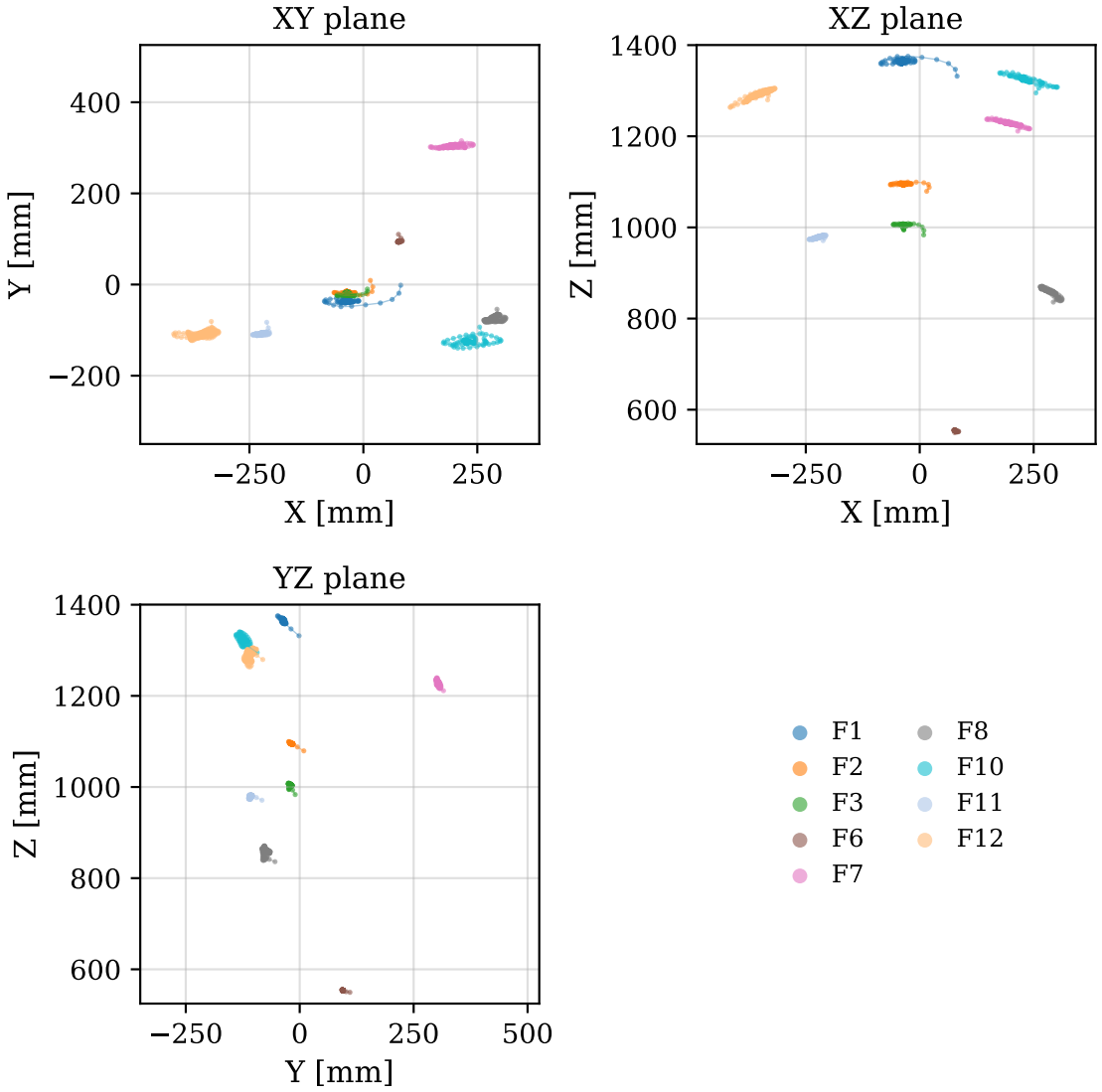

Figure 4.4: Successfully tracked features from the second pull and release test for birch projected on to the coordinate planes. The tree was pulled along the X-axis. Features F1-F3 are along the stem, while features F6-F12 are on branches.

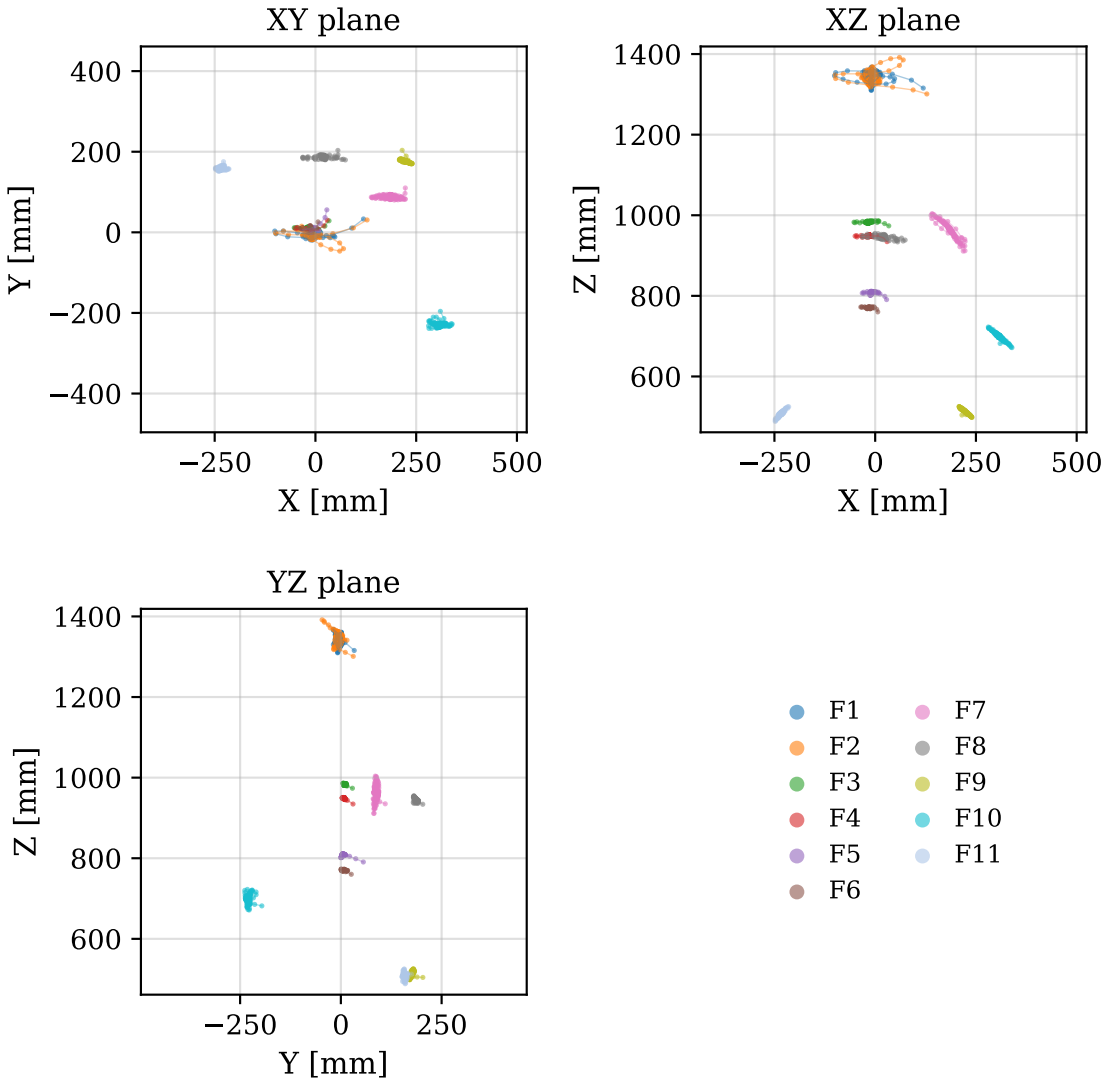

Figure 4.5: Successfully tracked features from the second pull-and-release test for spruce projected on to the coordinate planes. The tree was pulled along the X-axis. Features F1-F6 are along the stem, while features F7-F11 are on branches.

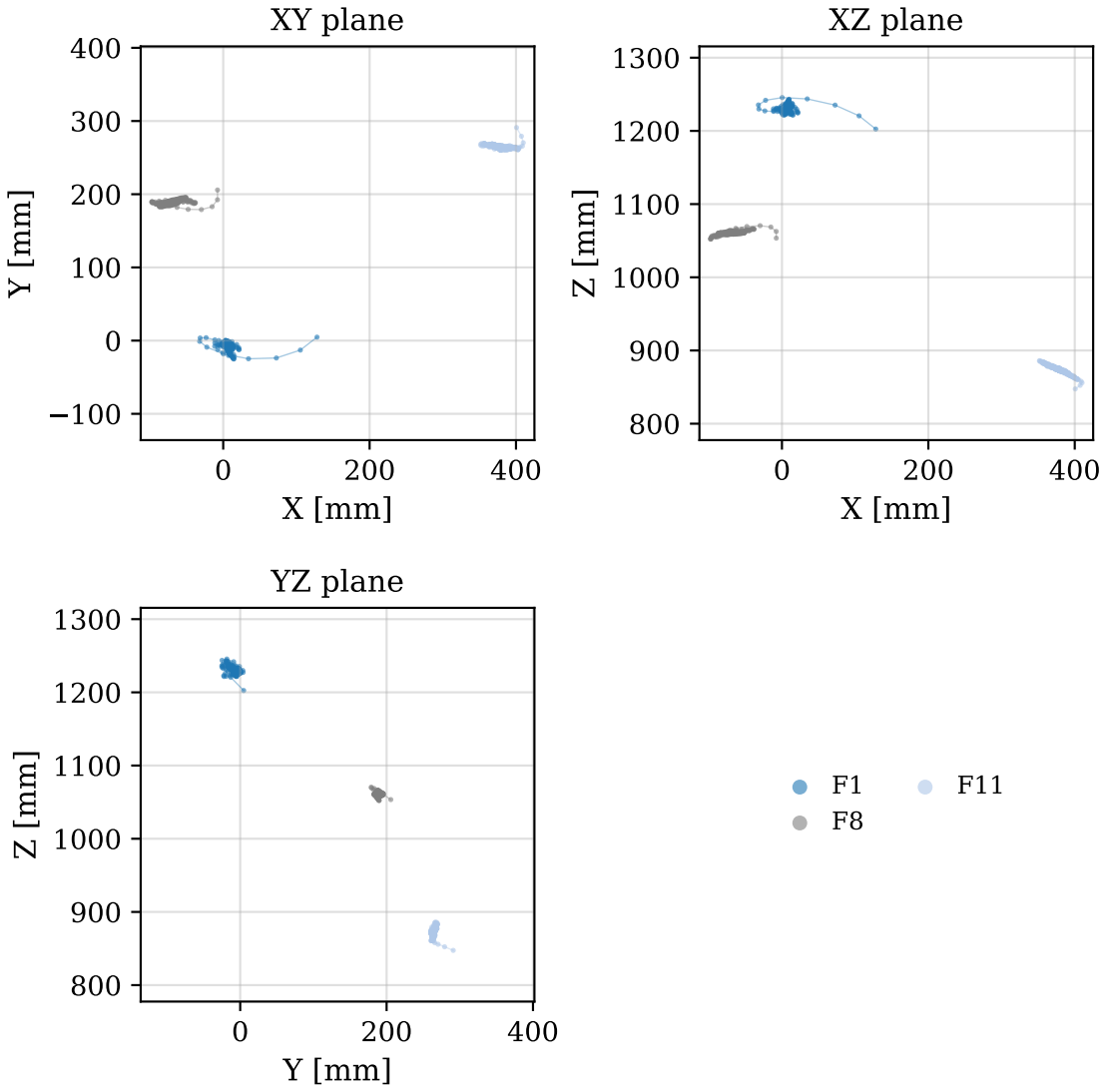

Figure 4.6: Successfully tracked features from the second pull-and-release test for pine projected on to the coordinate planes. The tree was pulled along the X-axis. Feature F1 is on the stem, while features F8 and F11 are on branches.

Figures 4.7, 4.8 and 4.9 show the same projections for individual features F10, F7 and F1 from the birch, spruce and pine respectively, with the PCA basis also shown with the variance explained by PC1, PC2 and PC3 given in each figure. The basis is computed separately for each feature from its own trajectory, so the PC1 direction is feature-specific and points along that feature's dominant motion rather than a shared world axis, which is why the PC1 line differs across the three. PC1 captures the large majority of the variance, while PC2 and PC3 capture progressively smaller components. These plots motivate the scalar PC1 signal used in the subsequent frequency analysis, since it captures the dominant oscillation while the smaller PC2 and PC3 components confirm that the motion is not strictly one-dimensional. The pine feature F1 (Figure 4.9) shows a noticeably larger initial displacement than the other two, since it was at the pull point and underwent the greatest excitation.

Figures 4.7, 4.8 and 4.9 show the same projections for the individual features F10, F7 and F1 from the birch, spruce and pine respectively. The PCA used in the transformation is also shown with the fraction of variance explained by PC1, PC2 and PC3 in each figure. The PCA basis is computed separately for each feature from its own trajectory, so the PC1 direction is feature- and test-specific and points along that feature's dominant motion rather than along a shared world axis. Notice that the PC1 line differs for the three features. PC1 captures the large majority of the variance and aligns with the dominant motion direction, while PC2 and PC3 capture progressively smaller components. These plots motivate the use of the scalar PC1 signal in the subsequent frequency analysis, since it captures the dominant oscillation while still revealing, through the smaller PC2 and PC3 components, that the motion is not strictly one-dimensional. The pine feature F1 (Figure 4.9) shows a noticeably larger initial displacement than the other two, since it was located at the pull point and therefore underwent the greatest initial excitation.

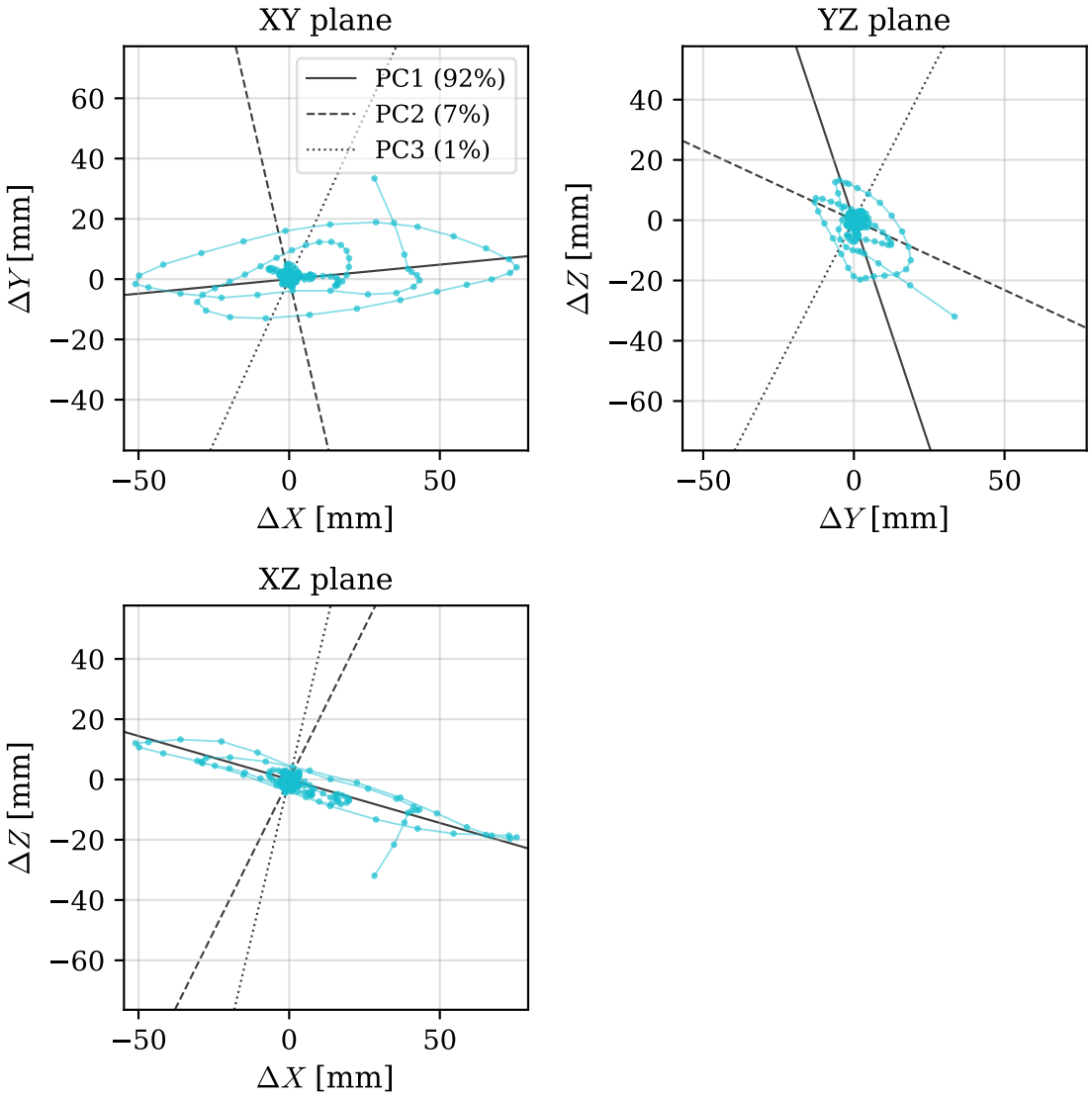

Figure 4.7: Tracked feature F10 from the second pull and release test for birch projected on to the coordinate planes with PCA directions and their explained variance.

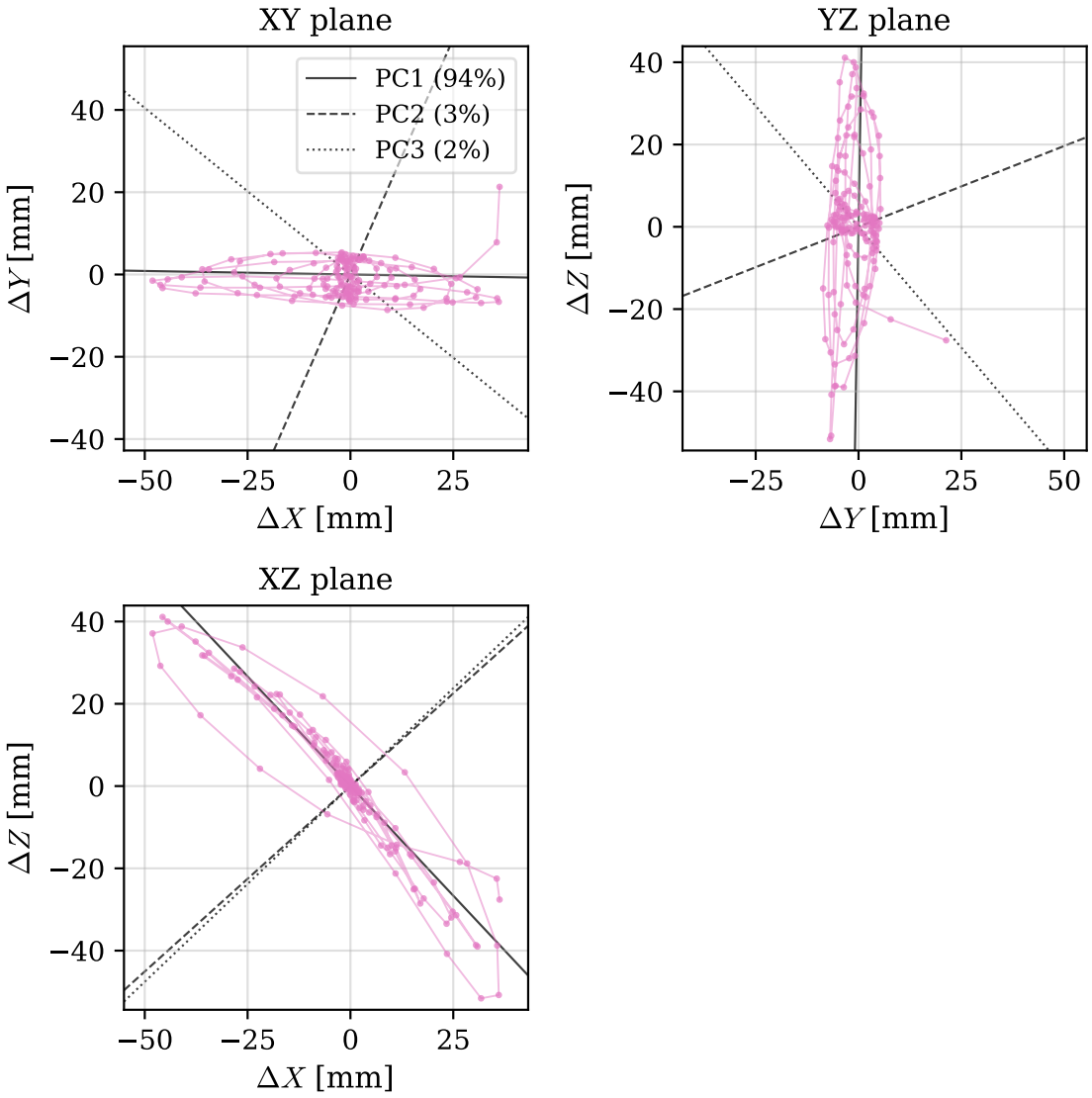

Figure 4.8: Tracked feature F7 from the second pull and release test for spruce projected on to the coordinate planes with PCA directions and their explained variance.

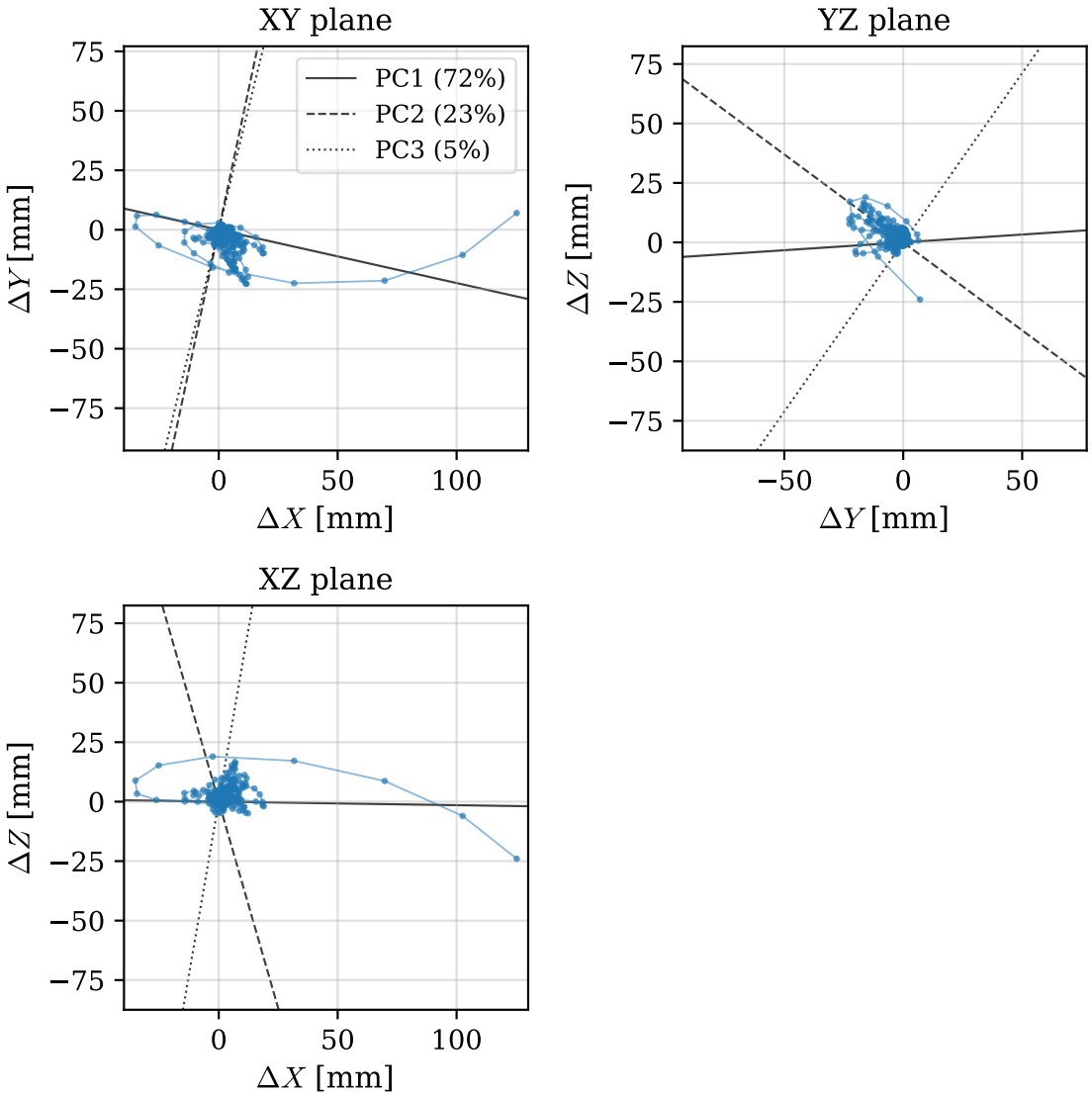

Figure 4.9: Tracked feature F1 from the second pull and release test for pine projected on to the coordinate planes with PCA directions and their explained variance.

4.3.4 Time Domain Response

This section examines the reconstructed displacement signals in the time domain. The motion in the first two principal component directions is first presented for a representative feature for each tree, followed by close look at the response of the stem features, a comparison between stem and branch features and finally the reproducibility of the response across the three repeated tests.

4.3.4.1 Motion in the Principal Directions

Figures 4.10, 4.11 and 4.12 show the displacement of a representative branch feature for each tree in both the PC1 and PC2 directions over time. While the PC1 direction captures the oscillations with largest amplitude, oscillatory signals are also present in PC2.

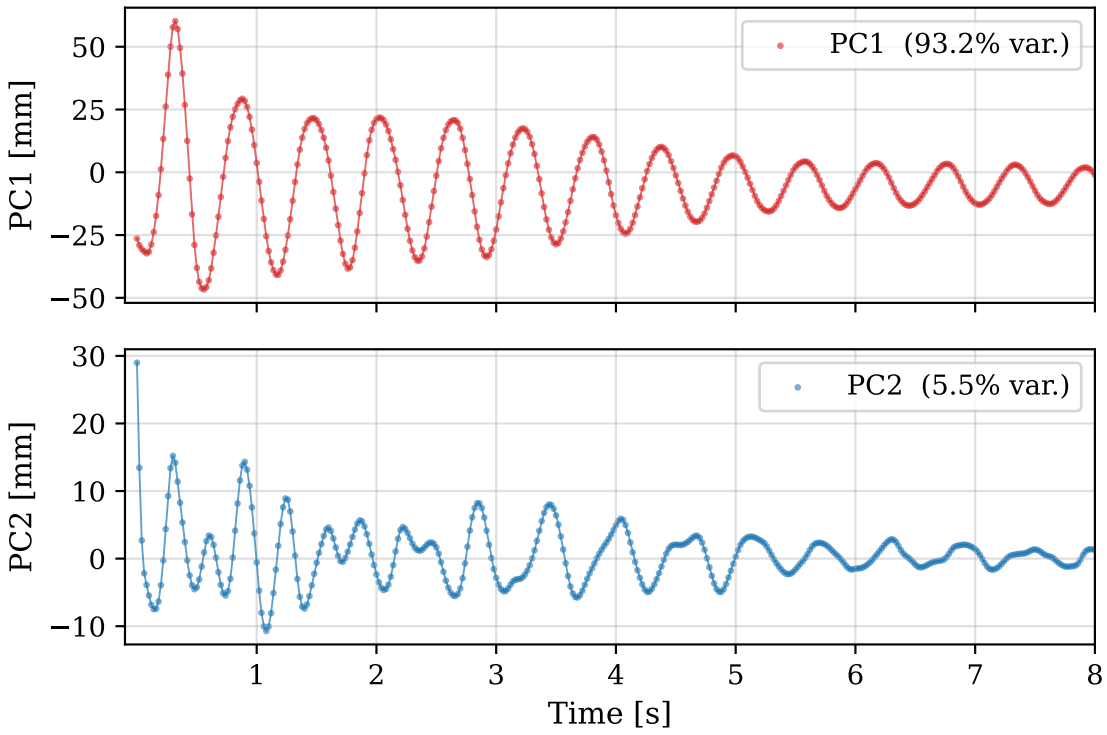

Figure 4.10: Displacement of F12 in the PC1 and PC2 directions over time for birch pull-and-release tests 3.

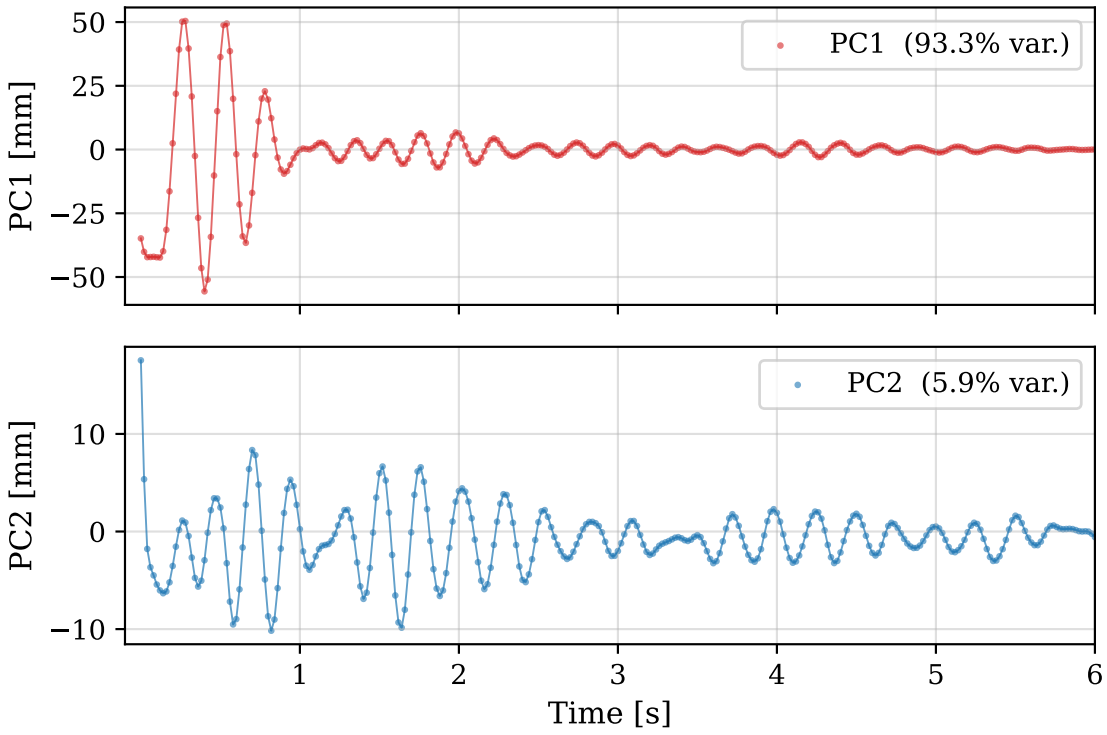

Figure 4.11: Displacement of F8 in the PC1 and PC2 directions over time for spruce pull-and-release tests 1.

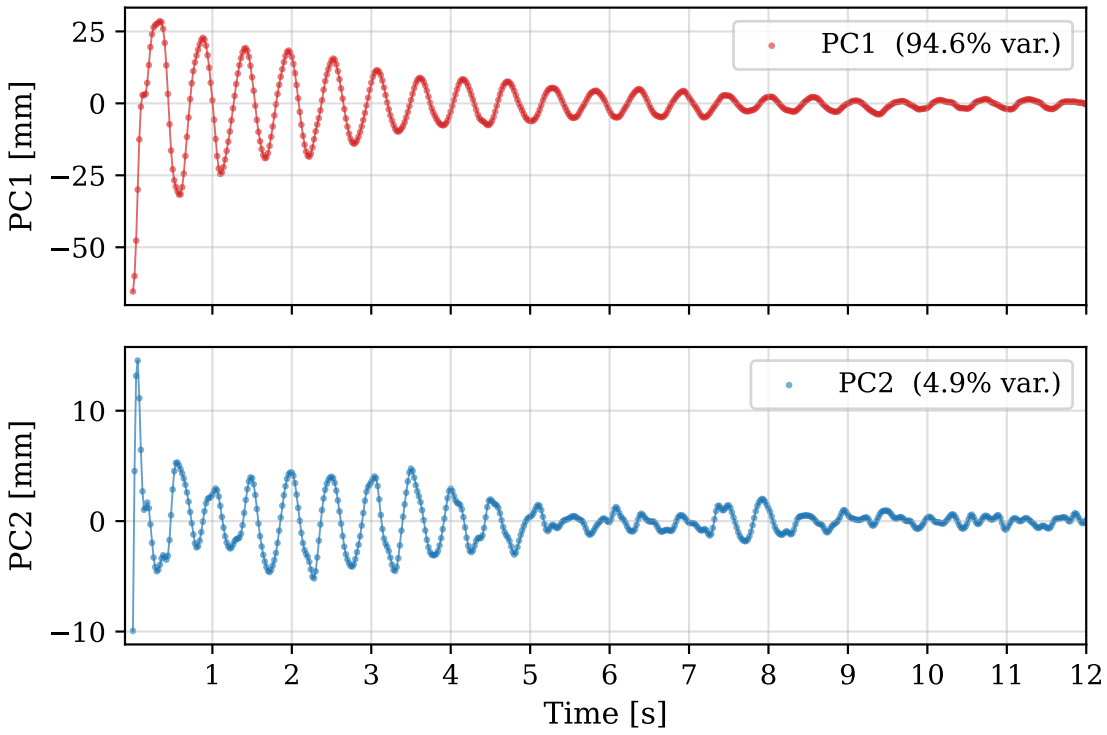

Figure 4.12: Displacement of F8 in the PC1 and PC2 directions over time for pine pull-and-release tests 1.

4.3.4.2 Stem Features

Figure 4.13 shows the PC1 stem features plotted over each other for the birch and spruce respectively. In both trees the oscillation amplitude decreases for features located further down the stem, with the uppermost feature showing the largest displacement and the lowest the smallest. This is the expected pattern for a stem displaced near its top and clamped at its base, where the bending deflection grows with height and the points nearest the base undergo the smallest translation. The stem features within each tree oscillate approximately in phase, consistent with the stem moving as a single bending mode rather than the features responding independently. For the pine, only one stem feature was tracked successfully, so no such comparison is shown.

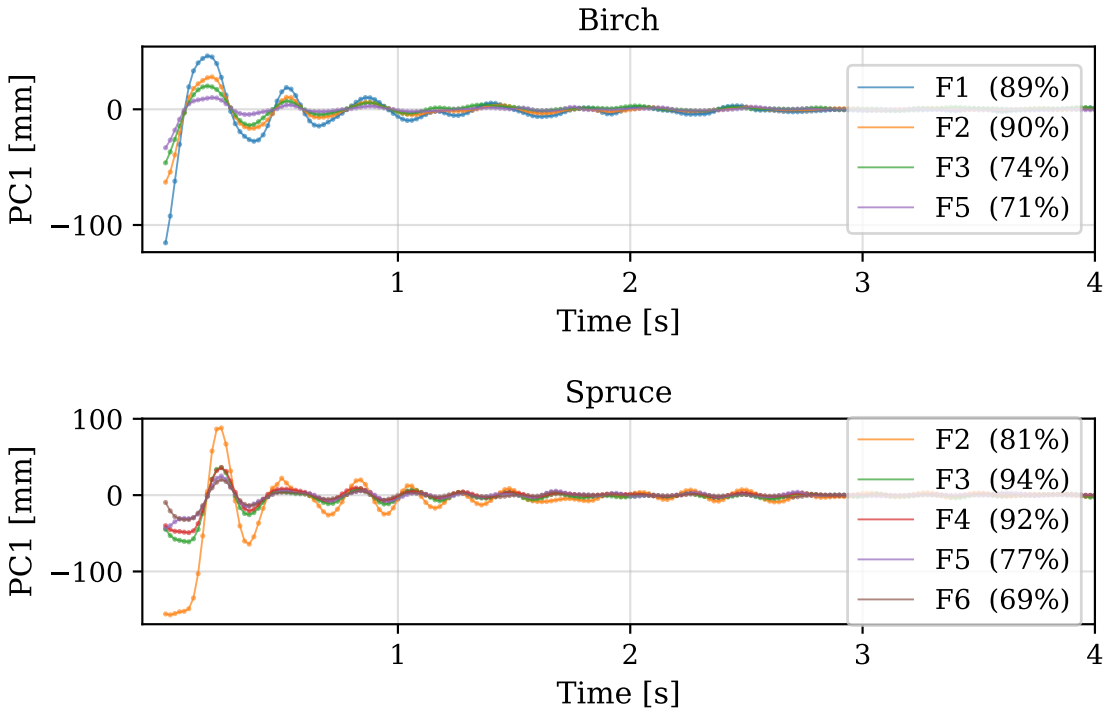

Figure 4.13: PC1 displacement of stem features during pull-and-release test 3 and 1 for the birch and spruce respectively. The explained variance of the PC1 component for each feature is shown in parentheses in the legend.

4.3.4.3 Stem and Branch Comparison

Figure 4.14 shows for each tree, representative the motion in PC1 trunk and branch features from representative tests. For all trees, it can be visibly observed that the stem feature exhibits stronger damping than the branch feature, whose oscillations persist longer after release. In the spruce and pine case, the stem feature also oscillates at a higher frequency than the branch feature, completing approximately two oscillations in the time the branch feature completes one.

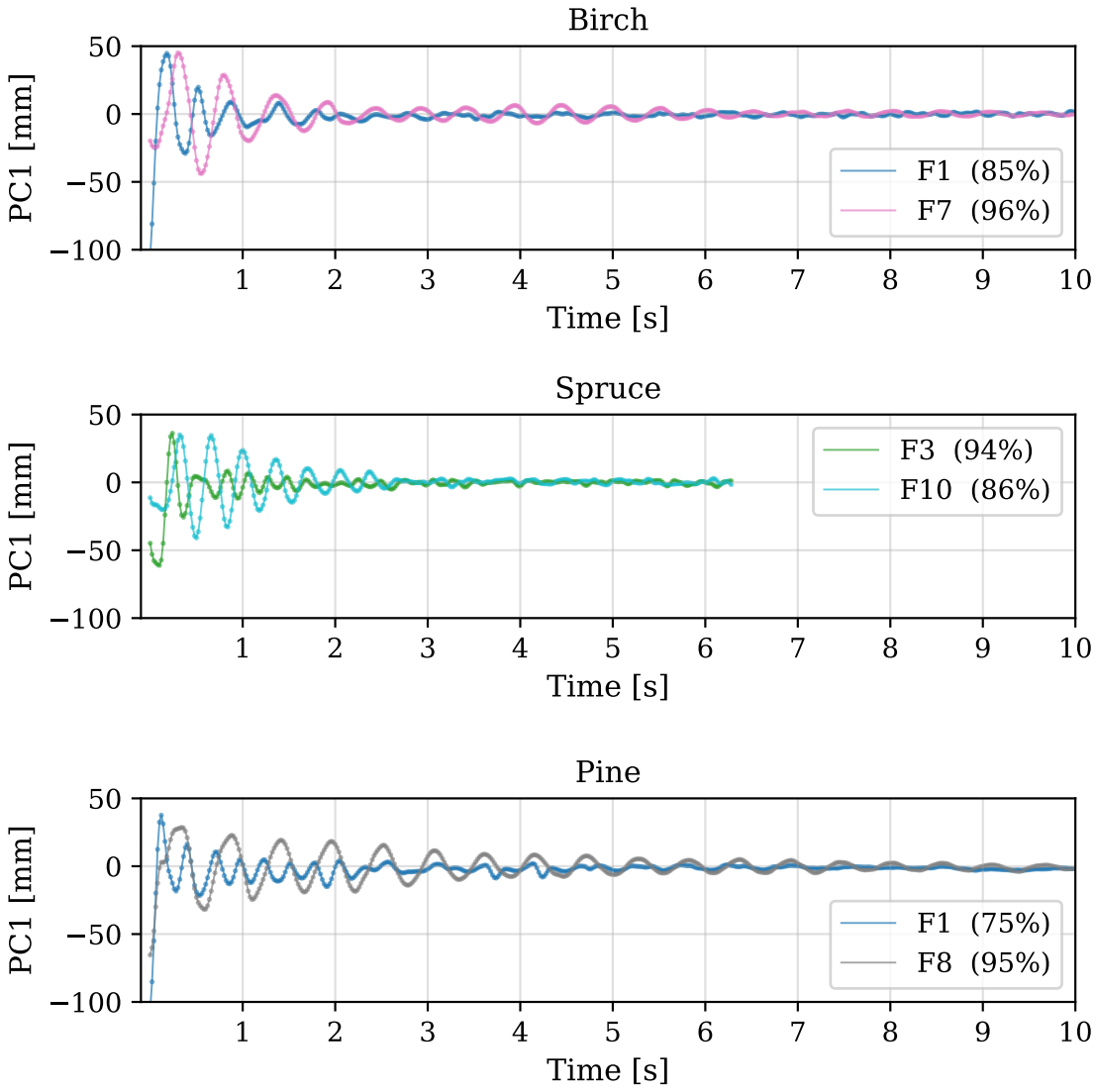

Figure 4.14: PC1 displacement of representative stem and branch features during pull-and-release tests for the birch, spruce, and pine. The birch result shows stem feature F1 and branch feature F7 during test 1, the spruce result shows stem feature F3 and branch feature F10 during test 1, and the pine result shows stem feature F1 and branch feature F8 during test 1. For each feature, the first principal component captures the dominant direction of motion independently, with the explained variance fraction given in the legend.

4.3.4.4 Trajectory Reproducibility

Figures 4.15, 4.16 and 4.17 show the PC1 displacement of a representative feature across all three pull-and-release tests for each tree. The oscillatory response is consistent across the tests. The times have not been synchronised across tests, which accounts for the time offset of test 1 relative to tests 2 and 3 in Figure 4.16. The curves also differ slightly in vertical scale since each test was transformed by its own PCA transformation.

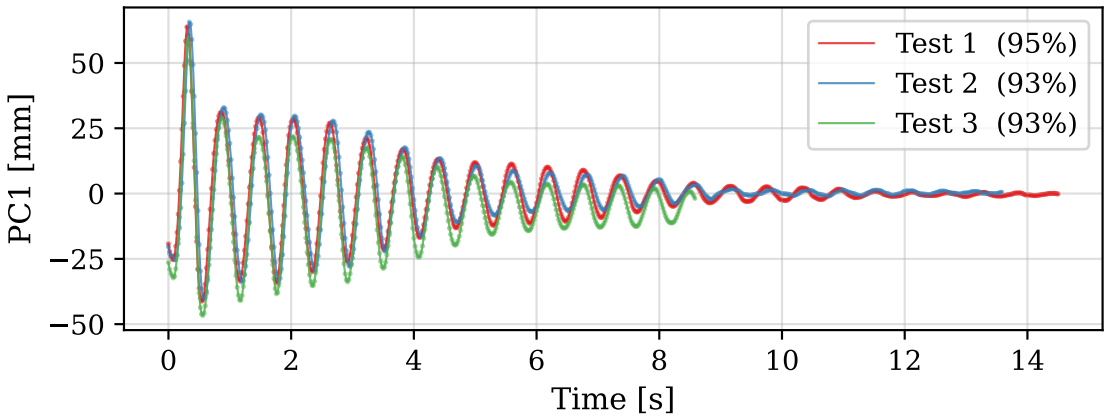

Figure 4.15: Displacement of F12 across the tree pull-and-release tests in the PC1 direction over time for birch. The small vertical differences between curves arise from each trajectory having its own PCA transformation.

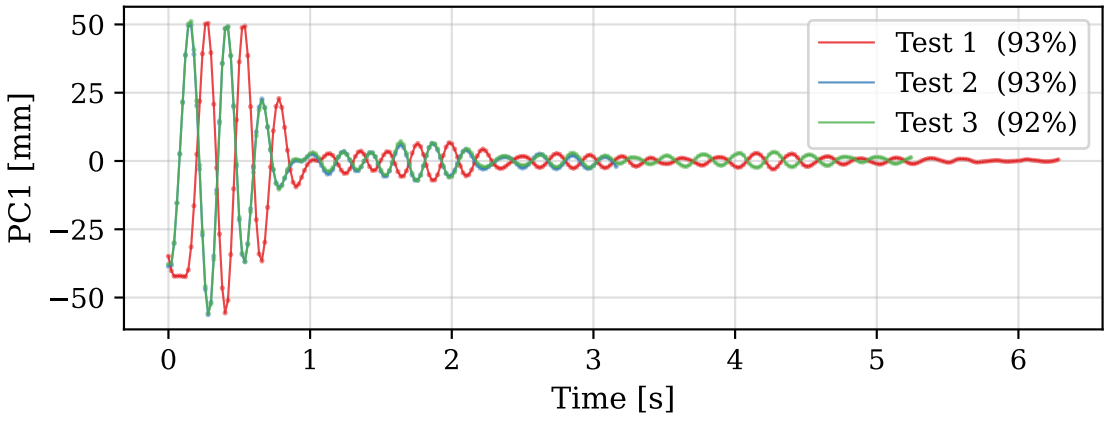

Figure 4.16: Displacement of F8 across the tree pull-and-release tests in the PC1 direction over time for spruce. Test 1 is offset in time because the recordings were not synchronised across tests. Tests 2 and 3 nearly coincide, making them hard to distinguish.

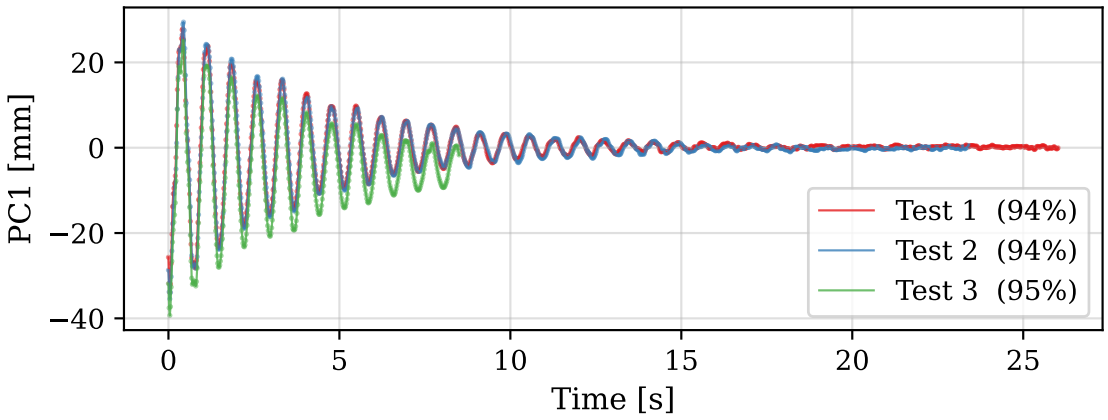

Figure 4.17: Displacement of F11 across the tree pull-and-release tests in the PC1 direction over time for pine. The small vertical differences between curves arise from each trajectory having its own PCA transformation.

4.3.5 Frequency and Damping

This section presents the frequency and damping estimates obtained from the PC1 signals of the tracked features. Two complementary methods are applied, as described in Section 3.4.5.1. The energy-weighted Welch power spectral density identifies the frequency content of each feature, resolving the dominant oscillation frequencies and any additional peaks present in the signal. The damped harmonic oscillator fit then provides, for features dominated by a single mode, an estimate of both the natural frequency and the amplitude decay rate.

4.3.5.1 Energy-Weighted Welsh PSD

Frequencies were estimated by applying the energy-weighted Welch method to the first principal component (PC1) of motion for each tracked feature, as described in Section 3.4.5. PCA was applied to decouple the dominant direction of oscillation from the fixed world coordinate system, as motivated by trajectories visible in figures 4.7, 4.8 and 4.9. The Welch estimate has a frequency bin spacing of 0.2 Hz, set by the segment length $L = 250$ samples. Reported peak frequencies are therefore quantised to the nearest bin, giving a readout precision of ± 0.1 Hz. The resulting normalised power spectral densities of PC1, with dominant peaks annotated, are shown for birch, spruce and pine in figures 4.18, 4.19 and 4.20 respectively with one panel per pull-and-release test.

For the birch, shown in Figure 4.18, the branch features F6-F12 generally exhibit a single dominant peak, while the stem features F1-F5 exhibit a broader spectral density with multiple peaks. The lower of the stem peaks is close to or coincides with the dominant branch peaks, while the stem features additionally show higher-frequency content not present in the branch spectra. The branch peaks of features F7-F12 are consistent across the three tests to within the ± 0.1 Hz readout precision, whereas among the stem features only F2 has a dominant peak consistent across all

three tests, with the remaining stem features varying by more than one bin between tests. Feature F6 is the exception among the branch features, with peaks at 1.6 and 2.0 Hz across tests, a difference of two bins that exceeds the readout precision.

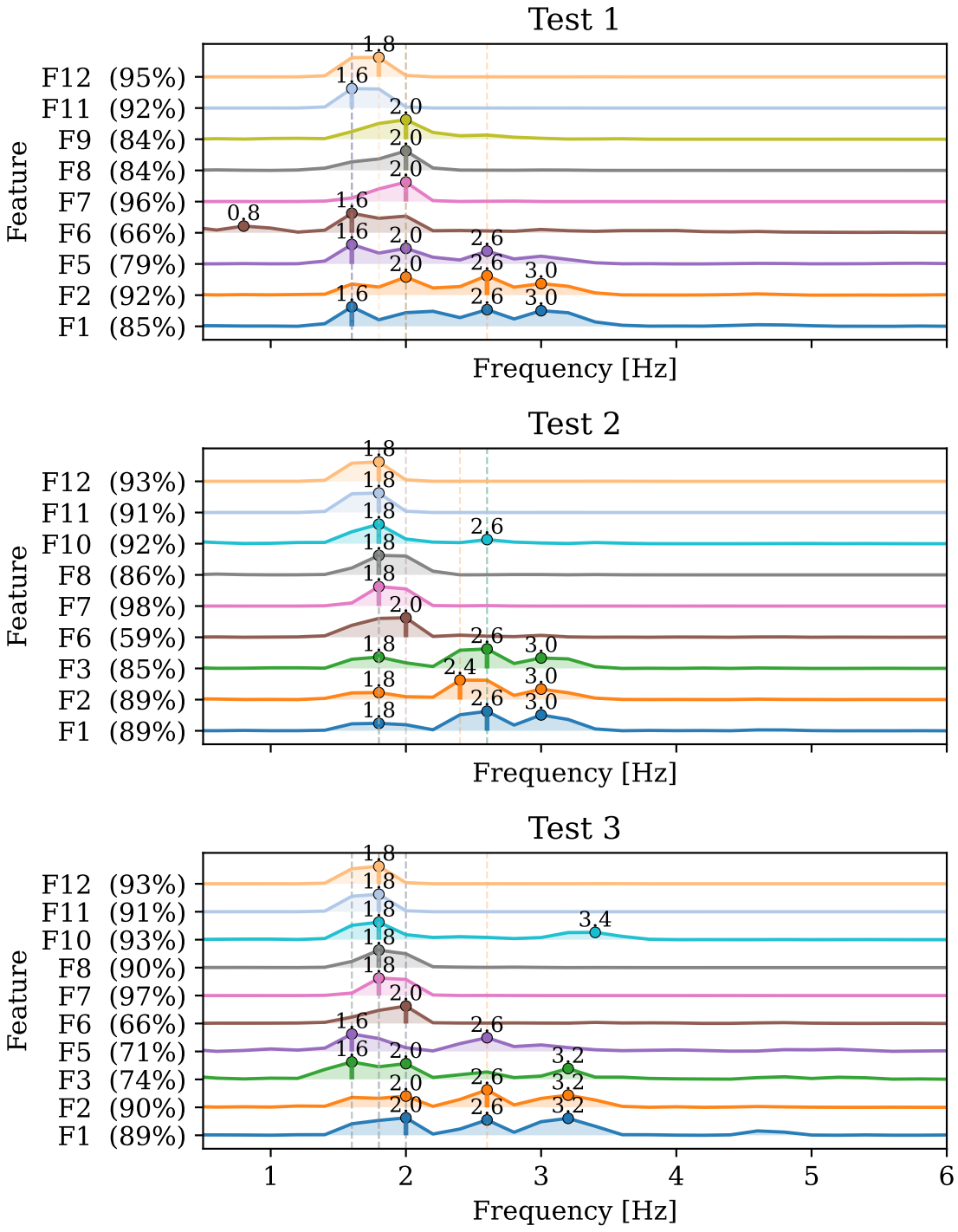

Figure 4.18: Energy-weighted Welch power spectral density of the first principal component (PC1) across the three pull-and-release tests for the birch. Features F1-F5 are stem features while features F6-F12 are branch features. Each feature's PC1 signal is truncated to its usable length T^* prior to spectral estimation, suppressing noise-dominated frames. The explained variance of PC1 is shown for each feature. Ridgelines within each subplot correspond to individual tracked features where dominant frequency peaks are marked. Only successfully tracked features are shown.

For the spruce, shown in Figure 4.19, the branch features F9-F11 have dominant peaks at frequencies that do not coincide with the stem peaks, in contrast to the birch. The branch features F7 and F8 are the exception, with peaks that align with the stem content. Across tests, the branch features F7 and F9-F11 are consistent to within the ± 0.1 Hz readout precision, while F8 is not. The stem features do not produce consistent dominant peaks across all tests, with the exception of F6. Test 2 produces broader, smoother spectral peaks than tests 1 and 3, because the shorter video sequence yields fewer Welch segments and a coarser effective frequency resolution, broadening the estimated peaks.

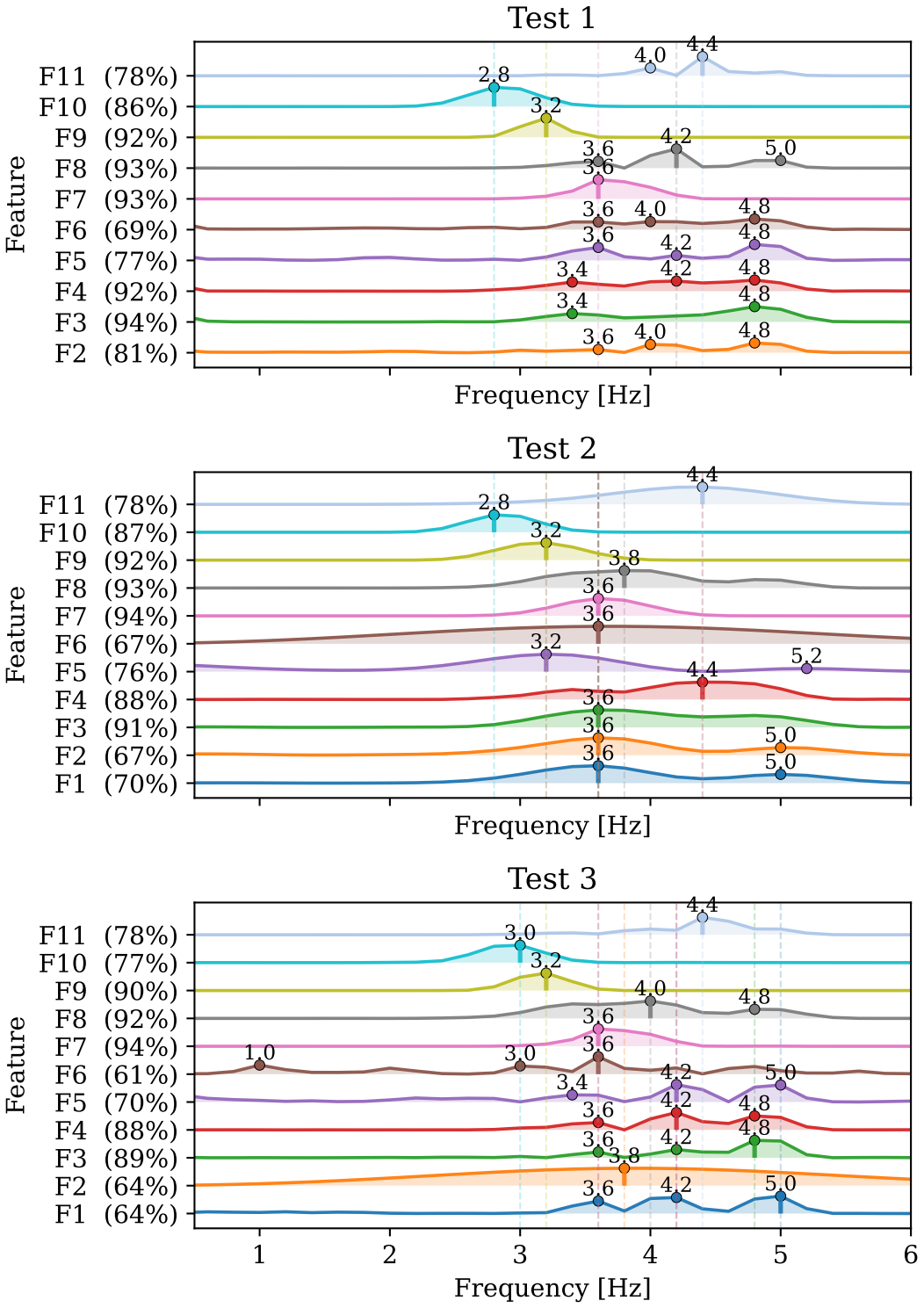

Figure 4.19: Energy-weighted Welch power spectral density of the first principal component (PC1) across the three pull-and-release tests for the spruce. Features F1-F6 are stem features while features F7-F11 are branch features. Each feature's PC1 signal is truncated to its usable length T^* prior to spectral estimation, suppressing noise-dominated frames. The explained variance of PC1 is shown for each feature. Ridgelines within each subplot correspond to individual tracked features where dominant frequency peaks are marked. Only successfully tracked features are shown.

For the pine, shown in Figure 4.20, only a single stem feature F1 was successfully tracked. Unlike the birch and spruce stem features, F1 exhibits one dominant frequency peak along PC1 with a smaller secondary peak. The two successfully tracked branch features, F8 and F11, show dominant peaks at 1.4 Hz and 1.8 Hz respectively, which are lower than the stem peak. The peaks are consistent across all tests.

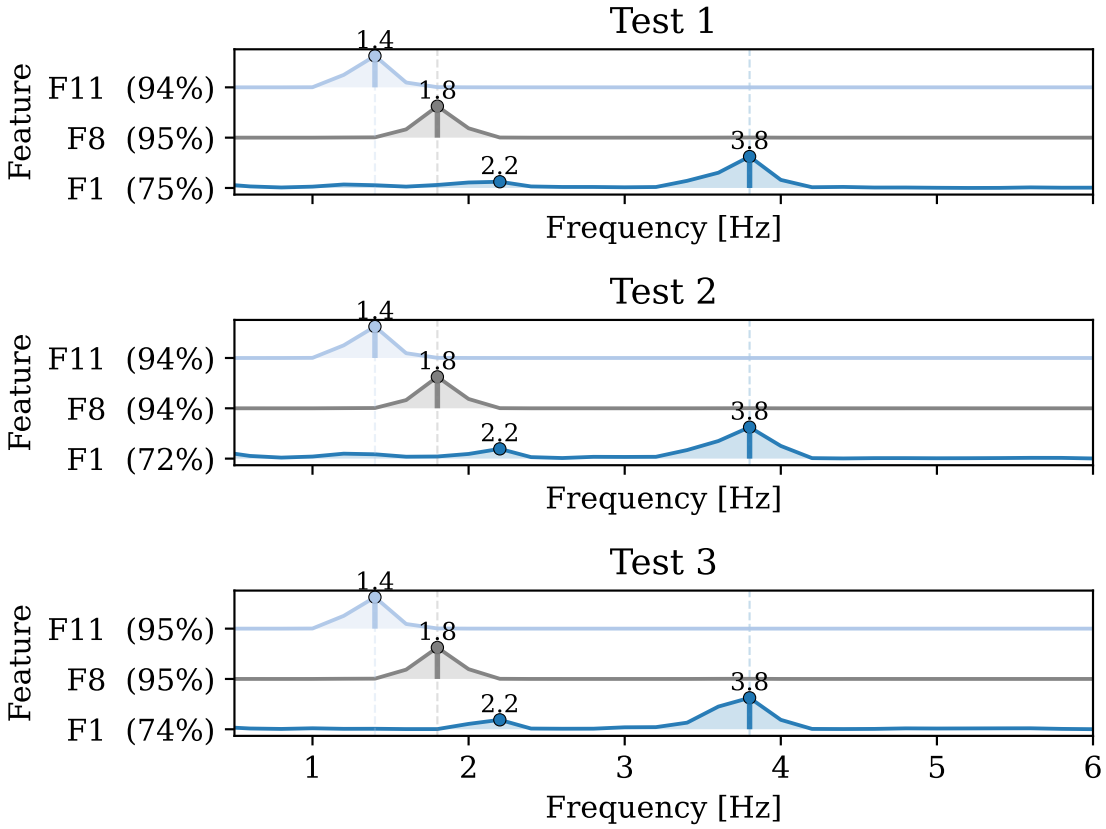

Figure 4.20: Energy-weighted Welch power spectral density of the first principal component (PC1) across the three pull-and-release tests for the pine. Feature F1 is a stem feature while feature F8 and F11 are branch features. Each feature's PC1 signal is truncated to its usable length T^* prior to spectral estimation, suppressing noise-dominated frames. The explained variance of PC1 is shown for each feature. Ridgelines within each subplot correspond to individual tracked features where dominant frequency peaks are marked. Only successfully tracked features are shown.

4.3.5.2 Damped Harmonic Oscillator

Figures 4.21 and 4.22 show the fitted DHO curve for two representative pine features, the branch feature F11 and the stem feature F1. The quality of the fit depends strongly on the signal. For the branch feature F11, whose response is close to a single decaying oscillation, the fitted curve follows the measured signal closely (Figure 4.21). For the stem feature F1, the large initial displacement and the presence of multiple frequency components cause the single-mode model to fit poorly (Figure 4.22), although the fit still returns approximate frequency and damping values. The variation in fit quality, most pronounced for the stem features, should be kept in

mind when interpreting the fitted parameters summarised across all trees and tests below.

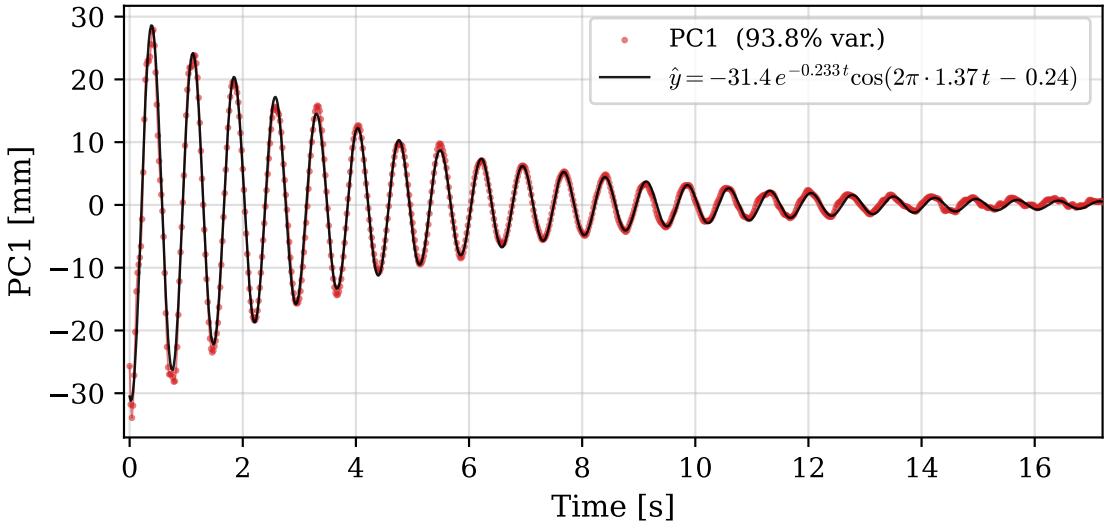

Figure 4.21: Displacement of branch feature F11 on the pine in the PC1 direction over time from test 1 with the fitted DHO curve shown in black.

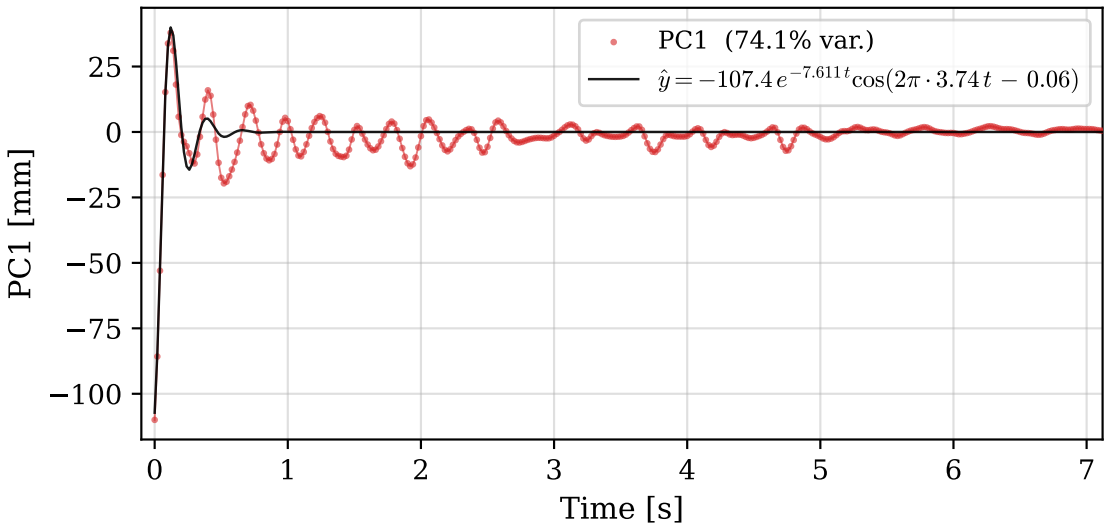

Figure 4.22: Displacement of stem feature F1 on the pine in the PC1 direction over time from test 3 with the fitted DHO curve shown in black.

Figures 4.23, 4.24 and 4.25 shows the estimated frequencies and damping from the damped harmonic oscillator fit, as described in Section 3.4.5.1.3, across all tests for the birch, spruce and pine respectively.

For the birch, two points appear as outliers where the fit failed, the singly tracked feature F9 and feature F6 in test 1. Excluding these, the fitted parameters show

a clear separation between the stem features F1-F5 and the branch features F6-F12. The stem features have consistent fitted frequencies and damping coefficients across the three tests and with one another, as do the branch features, which sit at lower frequencies and have lower damping than the stem. The agreement between the fitted frequencies and the Welch peaks in Figure 4.18 differs between the two groups. The fitted branch frequencies of F7, F9, F11 and F12 coincide with their spectral peaks, and the fitted frequency of the stem feature F5 also matches its peak. The remaining features do not coincide with any spectral peak, which is expected given that their spectra contain multiple peaks rather than a single dominant one which the DHO fit assumes. The damping estimates separate the two groups in the same way, with the branch features being less damped than the stem features. Feature F6 is the exception, with damping closer to the stem values, consistent with its position nearer the stem than the other branch features. The branch features F7, F8, F11 and F12 have an estimated damping an order smaller than of the stem features. Features F11 and F12 agree particularly closely in both fitted frequency and damping, which is consistent with their being spatially close to one another.

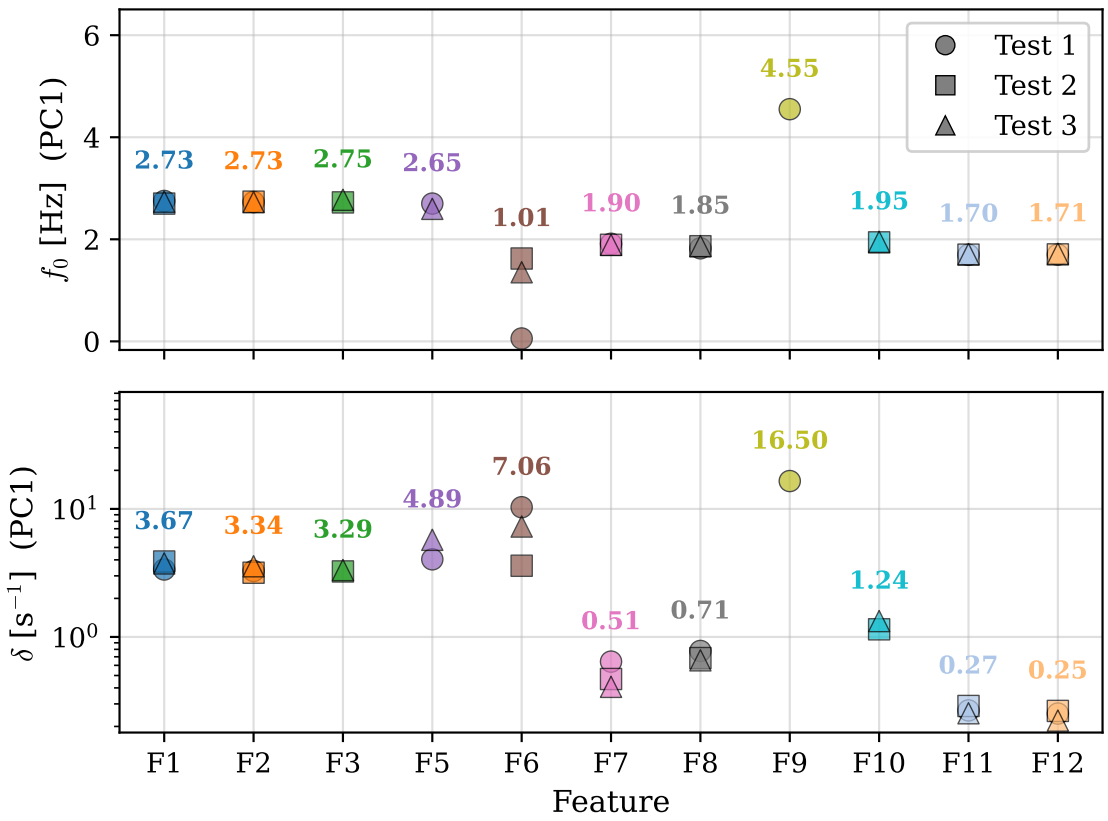

Figure 4.23: Fitted damped harmonic oscillator parameters for each successfully tracked feature along the first principal component (PC1) of birch. The upper panel shows the frequency f_0 [Hz] and the lower panel shows the damping coefficient δ on a logarithmic scale. Each marker represents a single pull-and-release test, with marker shape distinguishing tests and colour indicating the feature. The annotated value above each cluster is the mean across tests. The fit was performed on the signal truncated to the usable window T^*.

For the spruce, we notice that the stem features are not consistent across tests per feature as test 1 shows a lower frequency compared to tests 2 and 3. There are also two outliers stem feature F6 test 1 and branch feature F9 test 2 where the fit failed. The branch features F7, F9, F10, F11 are consistent with the Welch peaks in Figure 4.18. Similarly, as seen in the birch, we notice that the stem features F1-F6 exhibit higher damping than the branch features F7-F11.

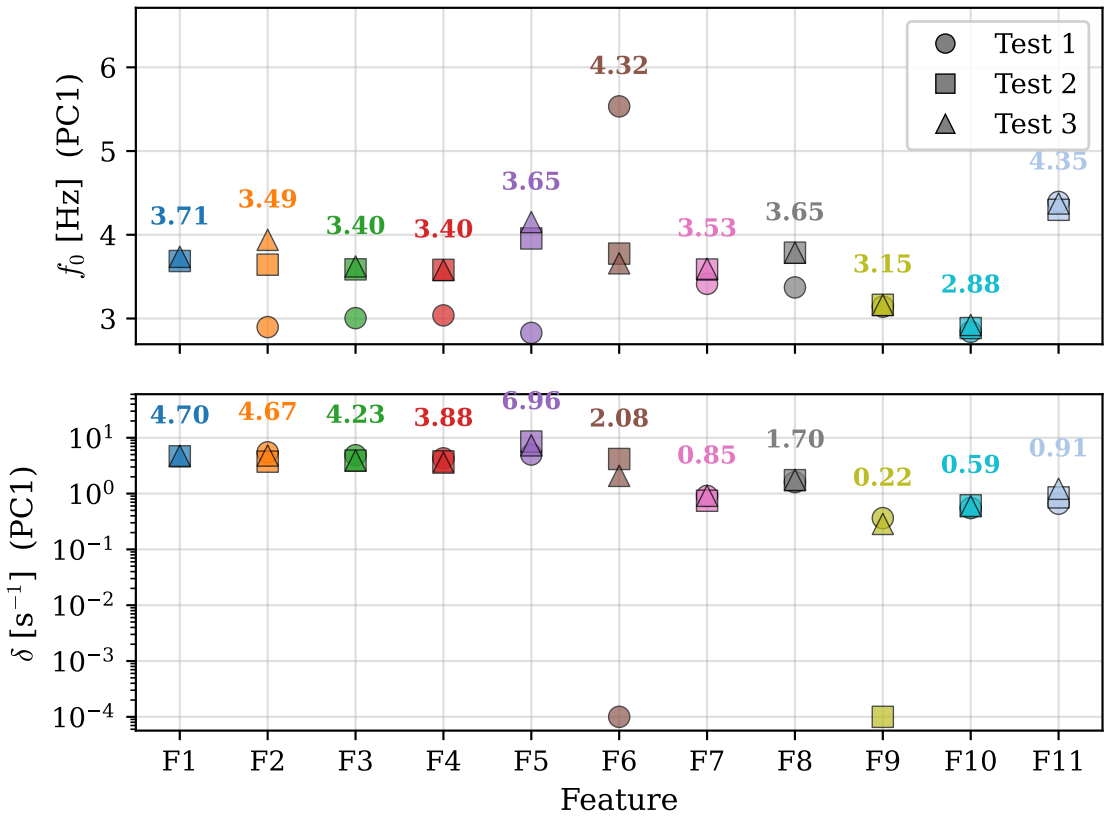

Figure 4.24: Fitted damped harmonic oscillator parameters for each successfully tracked feature along the first principal component (PC1) of spruce. The upper panel shows the frequency f_0 [Hz] and the lower panel shows the damping coefficient δ on a logarithmic scale. Each marker represents a single pull-and-release test, with marker shape distinguishing tests and colour indicating the feature. The annotated value above each cluster is the mean across tests. The fit was performed on the signal truncated to the usable window T^*.

For the pine, we notice that all features exhibits consistent frequency estimates and damping estimates across all tests. In particular, the frequencies are consistent with the frequency peaks shown in Figure 4.20 as they all are within the frequency resolution. Similarly, as we saw in the birch and spruce, the two branch features exhibit lower damping than the stem feature which is consistent with Figure 4.14

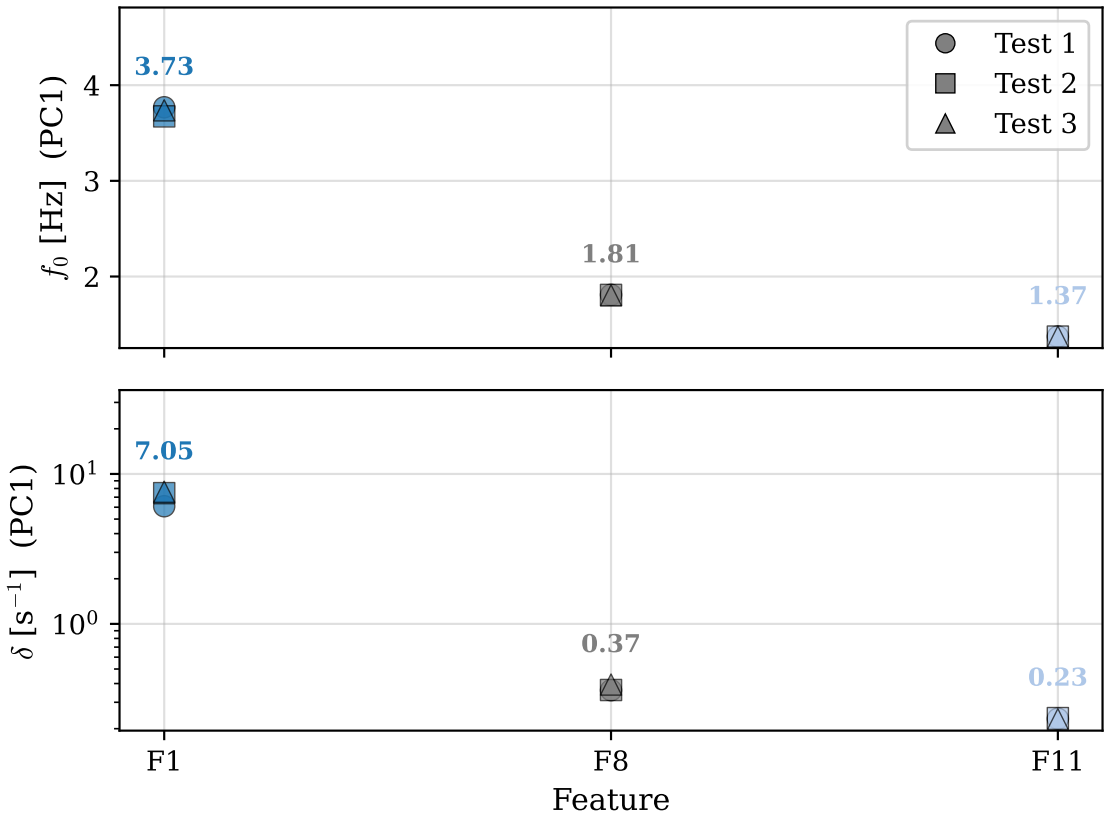

Figure 4.25: Fitted damped harmonic oscillator parameters for each successfully tracked feature along the first principal component (PC1) of pine. The upper panel shows the frequency f_0 [Hz] and the lower panel shows the damping coefficient δ on a logarithmic scale. Each marker represents a single pull-and-release test, with marker shape distinguishing tests and colour indicating the feature. The annotated value above each cluster is the mean across tests. The fit was performed on the signal truncated to the usable window T^*.

5

Discussion and Conclusion

This chapter interprets the results of the two experiments. The controlled motion experiment is discussed first, focusing on the accuracy of the tracking method and the sources of error identified against the reference sensors, followed by the tree experiments, where the method was applied to real trees. The chapter then summarises the main findings and outlines directions for future work.

5.1 Controlled Motion Experiment

The base case provided a controlled validation of the motion estimation method under conditions where independent ground truth was available from two reference sensors. The video-based method recovered three-dimensional marker trajectories agreeing with the robot ground truth to within a mean Euclidean distance of 6.16 mm and with the LiDAR to within 4.40 mm across the four measured endpoint positions. As shown in the following sections, these absolute figures are dominated by a systematic offset from the world frame registration and the per-axis precision of roughly 0.8 mm is the more representative measure of the method's tracking accuracy.

5.1.1 Camera Calibration Quality

The sub-pixel intrinsic reprojection errors (Table 4.1) confirm that the calibration procedure yielded well-estimated camera models, and the tracking method proved robust to the choice of intrinsic calibration, with the mean Euclidean residual varying by only 0.31 mm across the five datasets (Table 4.7). This insensitivity held even for the anomalous camera 1 c_y estimate in dataset 1, indicating that intrinsic errors of this magnitude do not propagate meaningfully into tracking accuracy.

The extrinsic calibration was the more influential of the two, though its effect on tracking remained modest. The pre-experiment set carried an elevated reprojection error for the $(2, 3)$ camera pair (Table 4.4), yet this did not translate into a larger tracking residual, which was in fact slightly smaller than the post-experiment one. The difference between the recordings is more plausibly explained by the cameras being displaced between sessions, as they were powered on and off and mounted on tripods susceptible to minor movement. The choice of recording shifted the mean tracking residual by 0.66 mm (Table 4.8), roughly twice the 0.31 mm intrinsic spread and largest in the Z component, consistent with a small downward nudge displacing

the cameras along the checkerboard-normal direction that defines the world Z-axis.

Overall, the reconstruction proved robust to the choice of calibration parameters. Both the intrinsic dataset and the extrinsic recording had only a minor effect on the mean tracking residual, which stayed below 6 mm throughout with small deviations across the sets, indicating that the tracking method does not critically depend on any single calibration.

5.1.2 LiDAR Scan Quality

The individual scans were precise, all falling within the scanner's ±2 mm ranging error, but the dominant source of LiDAR uncertainty was the co-registration between scan positions. Because registration used scene reference objects in FARO SCENE rather than the tracked spheres, the resulting 4.58 mm effective endpoint uncertainty reflects the alignment of independent scans rather than the instrument's intrinsic ranging precision. It therefore sets a practical floor on the LiDAR endpoint accuracy and should be kept in mind when interpreting the sensor comparisons, where the LiDAR is treated as a reference.

5.1.3 Coordinate Frame Registration

Each sensor was registered to $\mathcal{W}$ by a different procedure, so the three registrations contribute to the sensor comparison in different ways and to differing degrees.

For the robot, the 1.29 mm RMS residual is large relative to the robot's ±0.03 mm positional repeatability, so it is attributable to the manual positioning procedure rather than the robot itself. The ball was placed manually at the four outermost inner corners of the board as close as possible without making physical contact. Each correspondence carries a manual placement error in addition to the fitted residual. With only four point correspondences, the fit has little redundancy to average out such placement errors, leaving the residual sensitive to the accuracy of each individual point. The residual therefore characterises the accuracy of the corner correspondences, not the robot's precision.

That the video registration errors are comparable to the stereo calibration errors, and larger than the intrinsic errors, is to be expected for two reasons. First, the intrinsic parameters and camera poses were held fixed during the board pose estimation, so any residual calibration error is absorbed into the registration residual rather than being re-optimised. Second, the board pose was estimated over the fixed camera parameters, so the registration cannot reproject the corners more accurately than the underlying calibration allows. Its reprojection error is therefore bounded below by the stereo calibration error and would not be expected to fall beneath it regardless of how well the single board pose is estimated.

For the LiDAR, the plane normal defines the Z-axis of $\mathcal{W}$, so the 0.47 mm plane-fit residual represents the out-of-plane uncertainty of the extracted points and prop-

agates directly into the Z-axis accuracy of the registration. The dominant source of LiDAR positional uncertainty in $\mathcal{W}$ nonetheless remains the co-registration error from the previous section, not the world registration itself.

5.1.4 Z-axis Bias

The systematic Z-axis bias seen in the sensor comparisons appears in all three sensor pairs, including Robot-LiDAR, where the video tracking method plays no role. It's therefore not that likely that the bias originates from the video tracking method or the camera calibration.

The Z-axis is the direction normal to the checkerboard plane and is the least constrained by a planar target. Each sensor's registration estimates this plane normal independently, so any small error in a normal propagates directly into that sensor's Z-axis orientation in $\mathcal{W}$, and any inconsistency between the three estimated normals produces a systematic Z offset between sensors. This is consistent with the bias being confined almost entirely to Z, while the in-plane X and Y biases remain smaller.

A calibration-induced bias would most likely change with the calibration sets, yet the Z bias stays at a similar large magnitude across all five intrinsic sets and both extrinsic sets. This points to the world frame registration, rather than the calibration as the main source. The small variation that does occur, the 0.64 mm shift in $\bar{e}_Z$ between the two extrinsic recordings, is consistent with the minor camera displacement discussed earlier.

5.1.5 Motion Tracking Accuracy

The mean Video-Robot endpoint error of 6.16 mm represents the combined effect of several sources, the camera calibration, the world frame registration, the offset between the robot's tool centre point (TCP) and the ball centre, and the motion tracking itself. The following paragraphs separate these contributions, and show that the tracking adds only a small random component while the systematic sources account for the bulk of the error.

The error is dominated by a large, near-constant offset in the Z direction. Over the full trajectory $\bar{e}_Z = -5.64$ mm, far exceeding the lateral biases $\bar{e}_X = 1.60$ mm and $\bar{e}_Y = 0.11$ mm, and this offset is essentially the same at every position. On top of this offset is a smaller structure in the residual, which peaks near the trajectory turning points rather than remaining constant with position. A purely constant offset would produce a position-independent residual, so this remaining structure indicates that the video and robot frames are not perfectly rotationally aligned. The rotational contribution is small relative to the constant Z offset, which remains the dominant systematic term and is consistent with the world frame registration identified in the previous sections.

Separate from these systematic effects, the random scatter of the residual is small.

The per-axis standard deviations of $\sigma_X = 0.58$ mm, $\sigma_Y = 0.78$ mm and $\sigma_Z = 0.79$ mm are an order of magnitude below the 5.64 mm constant Z offset, and the largest of them, 0.8 mm, matches the image-to-scene scale of 0.60 mm/px estimated from the projected size of the sphere marker. This correspondence indicates that the random component is near the precision limit set by the camera resolution, so reducing it would require higher-resolution cameras or a shorter working distance rather than changes to the tracking algorithm.

The inter-endpoint distance comparison isolates the tracking accuracy from the registration and TCP offsets. Because both sensors measure the same displacement from $\boldsymbol{P}_0$, the constant offsets cancel, leaving only the position-dependent and random terms. The Video-Robot distance error then falls to a mean of 1.67 mm, far below the 6.16 mm absolute endpoint error, which confirms that the constant offset dominates the absolute error and that the method tracks relative displacements to within a few millimetres.

These results can also be read against each reference's own uncertainty. The Video-LiDAR mean distance of 4.40 mm is comparable to the LiDAR's 4.58 mm effective endpoint uncertainty, so the two agree at the level of the reference's own error and the discrepancy cannot be ascribed to the video tracking. The robot, by contrast, is precise to ± 0.03 mm, so the larger 6.16 mm Video-Robot distance reflects genuine disagreement rather than reference uncertainty. That disagreement is the constant Z offset discussed above, and once it is removed in the inter-endpoint comparison the agreement improves to 1.67 mm. The method is therefore consistent with both references at the level each one permits.

Taken together, the random component of the tracking error sits near the resolution limit of the camera hardware, while the absolute error is dominated by the constant offset from the world frame registration and the TCP offset rather than by the tracking itself. This distinction matters for the tree experiments. A constant offset shifts every position equally, so it leaves relative quantities, the displacements, amplitudes and frequencies that characterise the tree motion, unaffected. What limits those measurements is instead the random component, whose sub-millimetre magnitude is well below the displacement amplitudes expected in pull-and-release tests confirming that the method is suitable for the intended application.

5.2 Tree Experiments

This section discusses the pull-and-release experiments, in which the motion estimation method was applied to three small trees, a birch, a spruce and a pine. Since each species was represented by a single tree and only three tests were performed, the results are interpreted as a demonstration of the motion tracking method rather than as statistically representative species-level conclusions.

5.2.1 Tracking Performance

Tracking reliability varied considerably across the three species, and this variation maps onto differences in foliage density, marker placement and feature visibility rather than anything specific to the species themselves. The spruce, with well-separated and visible markers, was tracked most reliably, the birch comparably so for most features, while the pine, with markers placed close together and partly obscured by denser foliage, yielded few usable tracks. The method is therefore best suited to sparse, clearly visible and well-separated markers and is less robust when foliage causes frequent occlusion.

The visibility statistics support this reading. Successful tracks were generally carried by two- or three-camera observations, so the method was effective whenever the marker remained at least intermittently visible, but the rarity of complete-occlusion frames among successful tracks suggests that prediction alone could not sustain a track through longer occlusions. This is consistent with the observed failure modes, where unsuccessful tracks either diverged or were associated with a neighbouring marker.

A contributing factor is the process model. The Singer acceleration model treats acceleration as a temporally correlated random process and does not encode the oscillatory return of a pull-and-release response. During longer occlusions the prediction has no mechanism to anticipate the turning point of the oscillation. A more dynamics-specific model, one that embeds the oscillatory motion, could improve the prediction through occlusion and is returned to in the future work.

5.2.2 Reconstructed Motion and Reproducibility

The reconstructed motion is not confined to a single coordinate direction, which supports the use of a three-dimensional reconstruction rather than a one-dimensional measurement of tree motion. Because PC1 captures the dominant direction but a smaller oscillation persists in PC2, the PC1-based frequency estimates describe the principal oscillation direction rather than the full trajectory, and should be read as such.

The oscillatory response is generally consistent across the three repeated tests, which indicates that the reconstructed motion reflects a genuine, repeatable response of the tree rather than potential tracking errors. The fact that three independent recordings, processed separately, produced similar trajectories further supports the reliability of the method.

5.2.3 Stem and Branch Dynamics

The reconstructed motion shows that stem and branch features differ in both their decay and their frequency content, so the trees do not respond as single rigid oscillators. This is the basic argument for resolving the motion at several points, since

a single-point measurement would capture only the local response.

Branches are widely described in the tree-dynamics literature as tuned mass dampers, or coupled damped oscillators, attached to the stem [4][11]. By moving relatively independently of the stem they can draw mechanical energy out of it and distribute it through the crown, where it is dissipated more effectively by viscous and aerodynamic damping across the branches than in a stiff stem alone. A stated prerequisite for this energy exchange is that the resonance spectra of the stem and branches overlap, so that the coupled oscillators can exchange energy [4][11].

The birch is the clearest case in which this prerequisite is met. Its stem spectra are multi-peaked, and the lowest stem peak coincides with the dominant branch peak while the stem additionally carries higher-frequency content of its own. The branch frequency is therefore present in the stem motion, which is the spectral overlap the mechanism requires. The much lower damping of the birch branches, mostly an order of magnitude below the stem, is consistent with energy persisting in lightly damped branch modes after the stem has settled. The spruce and pine fit this picture less cleanly. For the pine the branch peaks lie below the stem peak, so the spectra are separated rather than overlapping, and for the spruce only some branch features share frequencies with the stem while others do not, with the stem peaks themselves varying between tests. The trees therefore differ in how closely they match the multiple-resonance picture.

These readings should be treated as consistent with, rather than proof of, the damping mechanism. The experiment does not measure the stem-to-branch energy exchange directly, only the resulting motion of each feature. The excitation was also applied at the stem, since the rope was attached near the top of the stem, so the stem was driven directly while the branches moved only through their attachment to it. The stem and branches therefore did not start from comparable initial conditions, and the difference in observed decay reflects both their differing dynamic properties and their unequal excitation. With a single specimen per species, these observations indicate dynamic differences between tracked features rather than species-level behaviour.

5.2.4 Frequency and Damping Estimates

The oscillatory response was characterised with two complementary methods, the energy-weighted Welch PSD, which identifies frequency content without assuming a single mode, and the DHO fit, which provides a compact estimate of one frequency and one damping coefficient. Their relative strengths shape how the results should be interpreted.

The Welch estimates carry two limitations. The pull-and-release signals are transient and decaying, and for several features, especially the stem features, the usable oscillatory part is short relative to the Welch segment length, so the spectra indicate dominant spectral content rather than a precise modal decomposition. The finite

resolution is the second, with a 0.2 Hz bin spacing and an associated ± 0.1 Hz readout precision, so smaller differences carry no meaning. Within these limits, the branch features generally produced more reproducible peaks across repeated tests than the stem features, consistent with branch motion being closer to a single decaying mode.

The DHO fit rests on the assumption of a single damped oscillation, which is restrictive for the multi-peaked stem features. The fitted stem frequencies generally do not agree with the Welch peaks, because the stem signals contain several frequency components that a single-mode model cannot represent, and because the large initial displacement of the stem features further distorts the fit. The fit is correspondingly more trustworthy where the response is dominated by a single oscillation, as generally for the branch features, where the fitted frequencies generally matched the Welch peaks within resolution. It should be emphasised that the fits were of varying quality. Several features produced unstable estimates or failed outright, appearing as outliers in fitted frequency and damping, which reinforces that the single-mode model does not suit every signal.

Despite these limitations, the faster decay of the stem features is supported by two independent observations. The DHO returns consistently higher damping coefficients for the stem than for the branches, in agreement with the qualitative time-domain comparison, where the stem displacement decays more rapidly after release. This should be read as an observed difference in decay between tracked stem and branch features rather than as a measurement of a tree-level damping mechanism.

A further qualification applies to both methods, that the frequency and damping estimates were computed from the PC1 signal alone. As shown in the trajectory results, oscillatory motion is also present in the other principal directions, so the reported values describe the dominant direction of motion rather than the full three-dimensional response. A feature whose motion is not well captured by a single direction is therefore only partially described by its PC1 estimate.

Overall, the two methods are best used together. The Welch PSD reveals when a response contains more than one frequency component, while the DHO supplies a compact frequency and damping estimate whose reliability depends on how well the single-mode assumption holds. Agreement between them, as for the pine and several branch features, strengthens confidence in the estimated frequency, whereas disagreement, as for several stem features, is itself informative, indicating that those signals are not well described by a single damped oscillator.

5.2.5 Interpretation of the Tree Experiment Results

The tree experiments demonstrate that the method can reconstruct reproducible three-dimensional marker trajectories and extract dynamic information from pull-and-release tests, provided the markers remain sufficiently visible. The spruce and birch yielded the most successfully tracked features, while the birch additionally gave the clearest dynamical structure, with a well-separated stem and branch re-

sponse. The pine experiment exposes the main practical limitation of the method, since its dense foliage, closely spaced markers and poor marker visibility reduced the number of usable trajectories. The few pine features that were tracked nonetheless produced consistent frequency and damping estimates across the repeated tests, indicating that the method can still recover useful dynamic information wherever stable trajectories are obtained.

The results show that the dynamic response depends on marker location. Stem and branch features do not necessarily share the same frequency content or decay behaviour, and the oscillatory motion is not confined to a single coordinate direction. These observations support the value of the three-dimensional multi-marker approach, since it provides spatially resolved motion data rather than a single response for the whole tree.

The interpretation is nonetheless limited by the experimental scope. Each species was represented by a single small tree and only three tests were performed per tree. The results should therefore be read as a demonstration of the measurement method and as an indication of dynamic differences between tracked features, not as statistically representative species-level conclusions.

5.3 Conclusion

This thesis presented a video-based method for reconstructing the time-resolved three-dimensional motion of trees from a calibrated three-camera system with physically attached markers and an Extended Kalman Filter state estimator. The method was validated against robot and LiDAR ground truth data in a controlled setting before being applied to pull-and-release experiments on small spruce, pine and birch trees posing additional challenges such as oscillating motions, occlusion and data association.

The validation demonstrated that the method achieves sub-centimetre absolute accuracy, with a mean Euclidean distance of 6.16 mm between the video and robot estimates and 4.40 mm between the video and LiDAR estimates. The per-axis precision, characterised by standard deviations of approximately 0.8 mm is consistent with the resolution limit imposed by the camera image scale suggesting that the tracker performs at the level expected from the hardware. A systematic depth bias of approximately -5.6 mm was identified and attributed to the world frame registration procedure and it does not impair the method's ability to resolve relative displacements. The calibration procedure was found to be robust, the tracking residual varied by only 0.31 mm across five intrinsic calibration datasets and by 0.66 mm between two extrinsic recording sessions.

In the tree experiments, the method tracked markers along the stems and branches of all three species, producing reproducible oscillatory trajectories across repeated tests. The frequency estimates were more consistent for branch features than for stem features. The pine was the most difficult species, with denser foliage and mark-

ers that were placed close together and not positioned to ensure visibility, which led to frequent occlusion and data association failures. Although the few features that were tracked still produced consistent estimates. The branch features sustained oscillations longer than the stem features after release, which is consistent with the role branches are thought to play in damping tree motion. The birch in particular showed the frequency overlap between stem and branches that this mechanism requires, though the energy transfer itself was not measured and the comparison is affected by the trees being excited from the stem.

Taken together, the results show that a small, calibrated multi-camera system with physical markers can provide time-resolved, three-dimensional displacement measurements of tree motion at sub-millimetre precision in a controlled indoor setting. The method is non-destructive, requires no training data and produces physically interpretable, spatially resolved outputs that are not accessible from single-camera or point-sensor approaches. The primary constraints identified are the requirement for marker installation, the sensitivity to occlusion and the indoor, small-tree scope of the experiments. Extending the method to mature trees under natural wind loading remains a challenge and is the subject of the directions outlined in the following section.

5.4 Future Works

The method was developed and validated indoors, and the tree experiments served as a proof of concept under repeatable conditions. The next step is to apply it to a mature tree under natural wind loading, which brings several challenges that define the main directions for future work.

The first concerns calibration. The checkerboard-based extrinsic routine relies on the cameras remaining fixed after calibration, which cannot be assumed outdoors where wind may disturb the cameras during recording. A targetless calibration that recovers the camera geometry from the scene itself rather than from a physical board [53], would be better suited to these conditions and ideally would do so continuously so that slow drift in the camera poses can be tracked over a long recording. Long outdoor recordings raise a second, related issue. The volume of data grows with the recording length, which may require an online tracking that processes each frame as it arrives rather than the entire video sequence. The fixed-interval smoother used here is incompatible with such operation, since it requires the entire recording before producing smoothed estimates. It could be replaced by a fixed-lag smoother, which returns smoothed estimates after only a short, bounded delay [35]. Over such durations the tracker should also be able to recover features that have been lost, rather than relying on the manual initialisation used in this work.

The second direction concerns how features are detected and sensed. The method depends on physical markers, which can be difficult to install on a mature tree. Even where markers remain practical, the red tape used here is not ideal, since its flat, irregular shape projects inconsistently across viewing angles and its colour is

still sensitive to changing illumination. A better-designed marker would address both issues. A spherical marker, or a band wrapped fully around a branch, would present a consistent appearance from any viewpoint, and a more distinct colour or a retro-reflective surface would segment more reliably under varying illumination. A markerless approach that tracks features already present in the scene would remove the installation problem altogether, at the cost of the robustness that well-defined markers provide. A further possibility is to combine other sensing methods. Because the estimator accepts any measurement with a known measurement model, a sensor able to see through foliage, such as radar, could be incorporated to mitigate the occlusion that limited the present experiments.

The third direction is the tracking method itself. The data association currently commits to a single assignment per feature in each frame, which is the source of the failures observed in cluttered scenes. A multi-hypothesis tracking approach, which maintains several candidate associations rather than one, would handle this ambiguity more robustly [54][55]. The dynamic model could also be reconsidered. The Singer model treats acceleration as a temporally correlated random process and does not anticipate the oscillatory return of a pull-and-release response, so a model that embeds the oscillation, such as a damped harmonic oscillator, could maintain better predictions through occlusion. Finally, the present method uses only the segmentation centroids, whereas the recordings contain considerably more information that a richer detection or appearance model could exploit.

A more substantial change would be to constrain the features relative to one another. Articulated pose estimation represents a body as a connected skeleton whose joints constraint parts relative to each other, which allows occluded parts to be inferred from their visible neighbours, in both humans and animals [56][57][58]. A tree could be modelled in the same way, as a connected structure in which branches are linked by joints, so that an occluded feature is constrained by the visible features it is attached to rather than tracked in isolation.

A final direction concerns the analysis of the reconstructed trajectories. Here the motion of each feature was reduced to a single direction by principal component analysis before estimating its frequency and damping, but the remaining directions also carry oscillatory information that the deployed method discards. Subspace identification methods from operational modal analysis, such as stochastic subspace identification, could estimate the frequencies and damping of several modes at once, and from many tracked points simultaneously, making fuller use of the three-dimensional, multi-point data the method produces [59][60].

Bibliography

[1] T. Jackson, A. Shenkin, J. Moore, A. Bunce, T. van Emmerik, B. Kane, D. Burcham, K. James, J. Selker, K. Calders, N. Origo, M. Disney, A. Burt, P. Wilkes, P. Raumonen, J. Gonzalez de Tanago Menaca, A. Lau, M. Herold, R. C. Goodman, T. Fourcaud, and Y. Malhi, "An architectural understanding of natural sway frequencies in trees," *Journal of The Royal Society Interface*, vol. 16, p. 20190116, 06 2019.

[2] M. Rudnicki, S. J. Mitchell, and M. D. Novak, "Wind tunnel measurements of crown streamlining and drag relationships for three conifer species," *Canadian Journal of Forest Research*, vol. 34, no. 3, pp. 666–676, 2004.

[3] S. Vollsinger, S. J. Mitchell, K. E. Byrne, M. D. Novak, and M. Rudnicki, "Wind tunnel measurements of crown streamlining and drag relationships for several hardwood species," *Canadian Journal of Forest Research*, vol. 35, no. 5, pp. 1238–1249, 2005.

[4] K. R. James, N. Haritos, and P. K. Ades, "Mechanical stability of trees under dynamic loads," *American Journal of Botany*, vol. 93, no. 10, pp. 1522–1530, 2006.

[5] T. D. Jackson, S. Sethi, E. Dellwik, N. Angelou, A. Bunce, T. van Emmerik, M. Duperat, J.-C. Ruel, A. Wellpott, S. Van Bloem, A. Achim, B. Kane, D. M. Ciruzzi, S. P. Loheide II, K. James, D. Burcham, J. Moore, D. Schindler, S. Kolbe, K. Wiegmann, M. Rudnicki, V. J. Lieffers, J. Selker, A. V. Gougherty, T. Newson, A. Koeser, J. Miesbauer, R. Samelson, J. Wagner, A. R. Ambrose, A. Detter, S. Rust, D. Coomes, and B. Gardiner, "The motion of trees in the wind: a data synthesis," *Biogeosciences*, vol. 18, no. 13, pp. 4059–4072, 2021.

[6] H. Holbo, T. Corbett, and P. Horton, "Aeromechanical behavior of selected douglas-fir," *Agricultural Meteorology*, vol. 21, no. 2, pp. 81–91, 1980.

[7] H. Mayer, "Wind-induced tree sways," *Trees*, vol. 1, no. 4, pp. 195–206, 1987.

[8] B. Gardiner, B. Marshall, A. Achim, R. Belcher, and C. Wood, "The stability of different silvicultural systems: a wind-tunnel investigation," *Forestry: An International Journal of Forest Research*, vol. 78, pp. 471–484, 10 2005.

[9] M. Rudnicki, U. Silins, V. Lieffers, and G. Josi, "Measure of simultaneous tree sways and estimation of crown interactions among a group of trees," *Trees*, vol. 15, pp. 83–90, 02 2001.

[10] A. Geitmann and J. Gril, *Plant Biomechanics: From Structure to Function at Multiple Scales*. 06 2018.

[11] H.-C. Spatz, F. Brüchert, and J. Pfisterer, "Multiple resonance damping or how do trees escape dangerously large oscillations?," *American Journal of Botany*, vol. 94, no. 10, pp. 1603–1611, 2007.

[12] K. R. James, "A study of branch dynamics on an open-grown tree," *Arboriculture & Urban Forestry*, vol. 40, pp. 125–134, May 2014.

[13] F. Zanotto, L. Marchi, and S. Grigolato, "Wind-tree interaction: Technologies, measurement systems for tree motion studies and future trends," *Biosystems Engineering*, vol. 237, pp. 128–141, 2024.

[14] W. Y. Chau, C. N. Loong, Y.-H. Wang, S.-W. Chiu, T. J. Tan, J. Wu, M. L. Leung, P. S. Tan, and G. L. Ooi, "Understanding the dynamic properties of trees using the motions constructed from multi-beam flash light detection and ranging measurements," *Journal of The Royal Society Interface*, vol. 19, p. 20220319, 08 2022.

[15] J. H. Ammatelli, E. D. Gutmann, S. A. Bush, H. R. Barnard, D. M. Ciruzzi, S. P. Loheide, M. S. Raleigh, and J. D. Lundquist, "Measuring tree sway frequency with videos for ecohydrologic applications: Assessing the efficacy of eulerian processing algorithms," *Agricultural and Forest Meteorology*, vol. 373, p. 110751, 2025.

[16] B. Lucas and T. Kanade, "An iterative image registration technique with an application to stereo vision (ijcai)," vol. 81, 04 1981.

[17] J.-Y. Bouguet, "Pyramidal implementation of the lucas kanade feature tracker," 1999.

[18] J. Diener, L. Reveret, and E. Fiume, "Hierarchical retargetting of 2d motion fields to the animation of 3d plant models," 09 2006.

[19] C. Der Loughian, L. Tadrist, J.-M. Allain, J. Diener, B. Moulia, and E. de Langre, "Measuring local and global vibration modes in model plants," *Comptes Rendus Mécanique*, vol. 342, 12 2013.

[20] J. Shi and Tomasi, "Good features to track," in *1994 Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, pp. 593–600, 1994.

[21] A. Barbacci, J. Diener, P. Hémon, B. Adam, N. Donès, L. Reveret, and B. Moulia, "A robust videogrametric method for the velocimetry of wind-induced motion in trees," *Agricultural and Forest Meteorology*, vol. 184, pp. 220–229, 2014.

[22] D. Bolme, B. Draper, and Y. Lui, "Visual object tracking using adaptive correlation filters," pp. 2544–2550, 06 2010.

[23] A. Wang, X. Yang, and D. Xin, "The tracking and frequency measurement of the sway of leafless deciduous trees by adaptive tracking window based on mosse," *Forests*, vol. 13, no. 1, 2022.

[24] J. Jia, J. Kang, L. Chen, X. Gao, B. Zhang, and G. Yang, "A comprehensive evaluation of monocular depth estimation methods in low-altitude forest environment," *Remote Sensing*, vol. 17, no. 4, 2025.

[25] J. Phattaralerphong and H. Sinoquet, "A method for 3d reconstruction of tree crown volume from photographs: assessment with 3d-digitized plants," *Tree Physiology*, vol. 25, pp. 1229–1242, 10 2005.

[26] S. Park, H. Park, J. Kim, and H. Adeli, "3d displacement measurement model for health monitoring of structures using a motion capture system," *Measurement*, vol. 59, pp. 352–362, 2015.

[27] A. Cappozzo, A. Cappello, U. Croce, and F. Pensalfini, "Surface-marker cluster design criteria for 3-d bone movement reconstruction," *IEEE Transactions on Biomedical Engineering*, vol. 44, no. 12, pp. 1165–1174, 1997.

[28] S. Hu, Z. Zhang, H. Xie, and T. Igarashi, "Data-driven modeling and animation of outdoor trees through interactive approach," *The Visual Computer*, vol. 33, no. 6, pp. 1017–1027, 2017.

[29] S. Hu, P. He, and D. He, "Motion capture and estimation of dynamic properties for realistic tree animation," in *Next Generation Computer Animation Techniques* (J. Chang, J. J. Zhang, N. Magnenat Thalmann, S.-M. Hu, R. Tong, and W. Wang, eds.), (Cham), pp. 18–34, Springer International Publishing, 2017.

[30] A. Bewley, Z. Ge, L. Ott, F. Ramos, and B. Upcroft, "Simple online and realtime tracking," in *2016 IEEE International Conference on Image Processing (ICIP)*, pp. 3464–3468, 2016.

[31] N. Wojke, A. Bewley, and D. Paulus, "Simple online and realtime tracking with a deep association metric," in *2017 IEEE International Conference on Image Processing (ICIP)*, pp. 3645–3649, 2017.

[32] X. Weng, J. Wang, D. Held, and K. Kitani, "3D Multi-Object Tracking: A Baseline and New Evaluation Metrics," *IROS*, 2020.

[33] X. Weng, J. Wang, D. Held, and K. Kitani, "AB3DMOT: A Baseline for 3D Multi-Object Tracking and New Evaluation Metrics," *ECCVW*, 2020.

[34] F. Rajič, H. Xu, M. Mihajlovic, S. Li, I. Demir, E. Gündoğdu, L. Ke, S. Prokudin, M. Pollefeys, and S. Tang, "Multi-view 3d point tracking," in *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 2025.

[35] F. Gustafsson, *Statistical Sensor Fusion*. Lund, Sweden: Studentlitteratur, 3 ed., 2018.

[36] R. A. Singer, "Estimating optimal tracking filter performance for manned maneuvering targets," *IEEE Transactions on Aerospace and Electronic Systems*, vol. AES-6, no. 4, pp. 473–483, 1970.

[37] R. Szeliski, *Computer Vision: Algorithms and Applications*. Cham: Springer, 2 ed., 2022.

[38] J. G. Fryer and D. C. Brown, "Lens distortion for close-range photogrammetry," *Photogrammetric Engineering and Remote Sensing*, vol. 52, pp. 51–58, Jan. 1986.

[39] G. Bradski, "The OpenCV Library," *Dr. Dobb's Journal of Software Tools*, 2000.

[40] Z. Zhang, "Flexible camera calibration by viewing a plane from unknown orientations," in *Proceedings of the Seventh IEEE International Conference on Computer Vision*, vol. 1, pp. 666–673 vol.1, 1999.

[41] Z. Zhang, "A flexible new technique for camera calibration," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 22, no. 11, pp. 1330–1334, 2000.

[42] R. Hartley and A. Zisserman, *Multiple View Geometry in Computer Vision*. Cambridge, UK: Cambridge University Press, 2 ed., 2004.

[43] F. Bolelli, S. Allegretti, L. Baraldi, and C. Grana, "Spaghetti labeling: Directed acyclic graphs for block-based connected components labeling," *IEEE Transactions on Image Processing*, vol. 29, pp. 1999–2012, 2020.

[44] R. E. Kalman, "A new approach to linear filtering and prediction problems," *Journal of Basic Engineering*, vol. 82, pp. 35–45, 03 1960.

[45] D. F. Crouse, "On implementing 2d rectangular assignment algorithms," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 52, no. 4, pp. 1679–1696, 2016.

[46] K. S. Arun, T. S. Huang, and S. D. Blostein, "Least-squares fitting of two 3-d point sets," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 9, pp. 698–700, May 1987.

[47] P. D. Welch, "The use of fast fourier transform for the estimation of power spectra: A method based on time averaging over short, modified periodograms," *IEEE Transactions on Audio and Electroacoustics*, vol. AU-15, pp. 70–73, June 1967.

[48] C. R. Harris, K. J. Millman, S. J. van der Walt, R. Gommers, P. Virtanen, D. Cournapeau, E. Wieser, J. Taylor, S. Berg, N. J. Smith, R. Kern, M. Picus, S. Hoyer, M. H. van Kerkwijk, M. Brett, A. Haldane, J. F. del Río, M. Wiebe, P. Peterson, P. Gérard-Marchant, K. Sheppard, T. Reddy, W. Weckesser, H. Abbasi, C. Gohlke, and T. E. Oliphant, "Array programming with NumPy," *Nature*, vol. 585, pp. 357–362, Sept. 2020.

[49] P. Virtanen, R. Gommers, T. E. Oliphant, M. Haberland, T. Reddy, D. Cournapeau, E. Burovski, P. Peterson, W. Weckesser, J. Bright, S. J. van der Walt, M. Brett, J. Wilson, K. J. Millman, N. Mayorov, A. R. J. Nelson, E. Jones, R. Kern, E. Larson, C. J. Carey, İ. Polat, Y. Feng, E. W. Moore, J. VanderPlas, D. Laxalde, J. Perktold, R. Cimrman, I. Henriksen, E. A. Quintero, C. R. Harris, A. M. Archibald, A. H. Ribeiro, F. Pedregosa, P. van Mulbregt, and SciPy 1.0 Contributors, "SciPy 1.0: Fundamental Algorithms for Scientific Computing in Python," *Nature Methods*, vol. 17, pp. 261–272, 2020.

[50] L. Song, Y. Wang, X. Li, Z. Xia, Y. Tong, and H. Wang, "Joint calibration method for non-scanning lidar and camera," *Engineering Research Express*, vol. 7, p. 0452d1, dec 2025.

[51] F. Gao and L. Han, "Implementing the nelder-mead simplex algorithm with adaptive parameters," *Comput. Optim. Appl.*, vol. 51, p. 259–277, Jan. 2012.

[52] "CloudCompare." GPL software, 2026. Version 2.14.

[53] J. L. Schönberger and J.-M. Frahm, "Structure-from-motion revisited," in *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.

[54] D. Reid, "An algorithm for tracking multiple targets," *IEEE Transactions on Automatic Control*, vol. 24, no. 6, pp. 843–854, 1979.

[55] C. Kim, F. Li, A. Ciptadi, and J. M. Rehg, "Multiple hypothesis tracking revisited," in *2015 IEEE International Conference on Computer Vision (ICCV)*, pp. 4696–4704, 2015.

[56] C. Zheng, W. Wu, C. Chen, T. Yang, S. Zhu, J. Shen, N. Kehtarnavaz, and M. Shah, "Deep learning-based human pose estimation: A survey," *ACM Comput. Surv.*, vol. 56, Aug. 2023.

[57] A. Monsees, K.-M. Voit, D. J. Wallace, J. Sawinski, E. Charyasz, K. Scheffler, J. H. Macke, and J. N. D. Kerr, "Estimation of skeletal kinematics in freely moving rodents," *Nature Methods*, vol. 19, pp. 1500–1509, Nov. 2022.

[58] P. F. Felzenszwalb and D. P. Huttenlocher, "Pictorial structures for object recognition," *International Journal of Computer Vision*, vol. 61, pp. 55–79, Jan. 2005.

[59] Z. Li, J. Fu, Q. Liang, H. Mao, and Y. He, "Modal identification of civil structures via covariance-driven stochastic subspace method," *Mathematical Biosciences and Engineering*, vol. 16, pp. 5709–5728, 01 2019.

[60] B. Peeters and G. De Roeck, "Reference based stochastic subspace identification in civil engineering," *Inverse Problems in Engineering*, vol. 8, pp. 47–74, 02 2000.

A

Appendix 1

A.1 Intrinsic Calibration Result

Table A.1: Detailed intrinsic calibration results for each camera and dataset. N denotes the number of images used after outlier rejection. RPE_{RMS} is the mean reprojection error.

Camera	Dataset	N	RPE_{RMS} [px]	f_x [px]	f_y [px]	c_x [px]	c_y [px]	k_1	k_2	k_3	p_1	p_2
1	1	66	0.2855	1289.5	1289.2	538.7	1005.4	0.0638	−0.1989	0.1752	0.0051	−0.0026
	2	67	0.3192	1297.9	1294.1	540.4	971.2	0.0255	−0.0416	−0.0026	0.0000	−0.0024
	3	67	0.2780	1304.0	1300.8	543.7	978.7	0.0265	−0.0747	0.0454	−0.0003	−0.0031
	4	66	0.3328	1301.2	1297.0	539.4	970.8	0.0338	−0.0837	0.0619	−0.0010	−0.0033
	5	64	0.2966	1299.1	1295.2	529.4	978.1	0.0455	−0.1264	0.0997	0.0008	−0.0049
	Mean		0.302	1298.3	1295.3	538.3	980.9					
	Std		0.023	5.5	4.3	5.4	14.2					
2	1	70	0.3082	1298.9	1294.5	521.3	950.1	0.0294	−0.0626	0.0463	−0.0048	−0.0034
	2	67	0.2901	1292.2	1291.3	526.5	953.9	0.0167	0.0129	−0.0651	−0.0042	−0.0027
	3	66	0.2801	1294.7	1293.0	521.1	954.1	0.0053	0.0566	−0.1122	−0.0043	−0.0035
	4	72	0.2731	1300.2	1297.0	525.6	955.6	0.0116	0.0164	−0.0525	−0.0021	−0.0027
	5	68	0.2690	1297.4	1292.8	516.5	945.9	0.0174	−0.0288	0.0135	−0.0051	−0.0034
	Mean		0.284	1296.7	1293.7	522.2	951.9					
	Std		0.016	3.2	2.2	4.0	3.9					
3	1	75	0.3301	1293.9	1293.1	583.7	979.2	0.0334	−0.0662	0.0218	0.0022	0.0067
	2	73	0.3576	1290.4	1289.7	570.1	987.0	0.0333	−0.0844	0.0577	0.0043	0.0043
	3	76	0.3543	1288.5	1285.5	570.2	974.2	0.0258	−0.0615	0.0288	0.0005	0.0040
	4	76	0.3364	1293.6	1290.9	571.0	990.6	0.0258	−0.0343	−0.0210	0.0050	0.0038
	5	75	0.3684	1298.5	1295.7	558.6	967.8	0.0462	−0.1255	0.1079	−0.0006	0.0028
	Mean		0.349	1293.0	1291.0	570.7	979.7					
	Std		0.016	3.8	3.8	8.9	9.3					

DEPARTMENT OF ELECTRICAL ENGINEERING
CHALMERS UNIVERSITY OF TECHNOLOGY
Gothenburg, Sweden
www.chalmers.se

CHALMERS
UNIVERSITY OF TECHNOLOGY