

# Reconstructing partially visible models using stereo vision, structured light, and the g2o framework

Tom Botterill

Department of Computer Science  
University of Canterbury  
Christchurch, NZ

tom.botterill@canterbury.ac.nz

Richard Green

Department of Computer Science  
University of Canterbury  
Christchurch, NZ

Steven Mills

Department of Computer Science  
University of Otago  
Dunedin, NZ

## ABSTRACT

This paper describes a framework for model-based 3D reconstruction of vines and trellising for a robot equipped with stereo cameras and structured light. In each frame, high-level 2D features, and a sparse set of 3D structured light points are found. Detected features are matched to 3D model components, and the g2o optimisation framework is used to estimate both the model's structure and the camera's trajectory. The system is demonstrated reconstructing the trellising present in images of vines, together with the camera's trajectory, over a 12m track consisting of 360 sets of frames. The estimated model is structurally correct and is almost complete, and the estimated trajectory drifts by just 4%. Future work will extend the framework to reconstruct the more complex structure of the vines.

## Categories and Subject Descriptors

I.4.8 [Image Processing and Computer Vision]: Scene Analysis—*Motion, shape, stereo, time-varying imagery, tracking*

## General Terms

Algorithms, performance, measurement, theory, verification.

## Keywords

Model-based 3D reconstruction, bundle adjustment, stereo vision, structured light, sensor fusion, nonlinear optimisation, tracking, data association.

## 1. INTRODUCTION

This paper describes the development of a 3D reconstruction framework for reconstructing a topologically correct 3D model of vines viewed by a robot. The robot uses a hybrid imaging system consisting of three colour cameras and three line lasers which project a sparse structured light pattern into the scene.

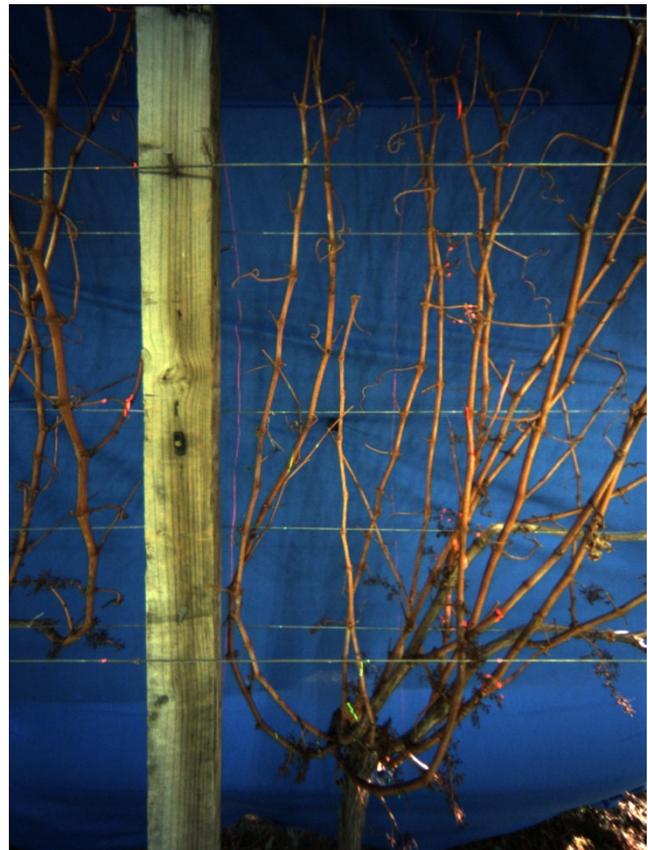
The vines which are imaged grow over a trellis consisting of wires stretched between posts (Figure 1). The computer vision system must model the trellis in addition to the vines in order to avoid confusing wires and vines, to avoid accidentally cutting the high tension wires and because the position of the wires affects the decisions made by the robot. The current system models the trellis, and is designed to incorporate measurements of vines once available.

The vine imaging robot consists of a tractor-mounted module

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

IVCNZ '12, November 26 - 28 2012, Dunedin, New Zealand  
Copyright 2012 ACM 978-1-4503-1473-2/12/11...\$15.00.

which is driven along a row of vines. Three wide-angle machine vision cameras in a trinocular stereo rig image the vines, and three planes of structured light are projected into the scene by high-power line lasers. The points of structured light are extracted from the images to give a sparse set of reconstructed 3D points [2]. Around 20 3D points are found per image; all on canes and wires. Currently, the tractor moves at 0.25m/s and the cameras run at 7.5 frames per second, capturing a set of three frames every 3 to 4cm.



**Figure 1: Example of a complete image of the vines, showing posts, wires, and structured light pattern.**

To fit in the confined space under the tractor, while still imaging enough of the vines to reconstruct a model, a high field of view (110 degrees) and depth of field (20cm to 80cm) are necessary. This creates challenges for the machine vision system, including variation in lighting [3] and feature sizes at different depths. In addition, the structure being imaged consists of a network of overlapping, occluding, and similar looking canes and wires, making the problem of correctly matching features to other features and to the model's component parts challenging.

In order to make cut decisions, and to plan a path for a robot arm, a topologically correct 3D model of the complete vine plants is

required. The approach described in this paper is to incrementally build a model, consisting of connected wire and post segments (and eventually vine segments), each parameterised by 3D control points. Knowledge of the structure of the features (posts, wires and vines) is used to identify and localise model parts in the images, despite occlusions. The structured light pattern visible in the images is used to assign 3D points to some of these features. These 3D points, together with the usual epipolar constraints are used to correspond features between images and to initialise 3D model components. Further observations are assigned to 3D components as they are observed, and an accurate 3D structure, together with the robot’s trajectory, are estimated using the modern optimisation framework *g2o* (a “general framework for graph optimisation”; [14]).

This paper is organised as follows: Section 2 discusses bundle adjustment, the *g2o* framework, and other options considered for building 3D vine models from images; Section 3 describes the proposed system; Section 4 evaluates the system on real images, and on simulated rendered images; Section 5 discusses the strengths and limitations of the system, and Section 6 describes our conclusions and plans for further work.

## 2. BACKGROUND

Many methods for 3D reconstruction from images have been proposed. The most successful general purpose approaches include various dense stereo algorithms, and approaches based on feature matching followed by bundle adjustment [19]. Of these, dense stereo algorithms are generally unsuitable for the vine reconstruction problem, because most cannot model the variation in ordering in the appearance of features between images, and many perform poorly along occlusion boundaries [16]. Bundle adjustment (BA)-based approaches are much more flexible however, and can incorporate prior information on the scene [21].

BA is a generic framework for photometric reconstruction which computes the maximum likelihood estimate of both the positions of the cameras and the positions of objects by minimising an appropriate function of the reprojection error (the difference between the position of objects back-projected into the image, and their measured locations). The classic formulation of BA corresponds point features detected in the image with 3D points in the world, however any image measurement can be used, for example line segments or detected model parts. As BA is fundamentally a nonlinear optimisation, additional measurements (e.g. 3D structured light points), and constraints from knowledge of the structure of the scene, can easily be incorporated, and will improve the accuracy of the solution [19].

BA is traditionally an algorithm for photogrammetry, however the problem is equivalent to the Simultaneous Localisation And Mapping (SLAM) problem in robotics. When a mobile robot explores an unknown environment, the position of the robot must be reconstructed, while simultaneously estimating a map of the environment, so that the robot can navigate without getting lost. Leading contemporary SLAM algorithms estimate both the robot’s trajectory (i.e. its sensors’ positions) and the positions of observed landmarks in a nonlinear optimisation framework (e.g. [13, 20]). For both SLAM and for photogrammetry, thousands of camera/sensor poses, and millions of 3D points/landmarks may be reconstructed, so optimisation algorithms must exploit the sparse structure of the problem in order to be efficient [21]. Recently, there has been convergence between SLAM and photogrammetry algorithms, and the general purpose frameworks *g2o* [14] and *iSAM2* (incremental Smoothing And Mapping; [12]) are designed to be applied to either problem. Both are Levenberg-Marquardt-

based optimisers which operate directly on sparse graphs, where the nodes of the graphs are parameters (i.e. camera poses and 3D points), and the edges represent the constraints between nodes introduced when measurements are made. These graphs can be built incrementally as the camera/robot moves, with the 3D structure being refined as more measurements are made.

Before any BA algorithm can be applied, a matching between features observed in images and the 3D model must be established. This problem is challenging when many features are similar in appearance (e.g. wires), or are only partially visible, as correspondences are ambiguous and may be matched incorrectly. Incorrect measurements can corrupt the 3D model and camera trajectory<sup>1</sup>, so must be avoided.

When many feature measurements (e.g. points) in an image are matched to each other, or to 3D structure, RANSAC [9], or one of its variants, can be used to find a large, mutually compatible subset of measurements which are likely to be inliers. RANSAC only works when the number of inlier measurements is considerably greater than that the number of degrees of freedom of the model however, so can’t be applied to small sets of measurements, such as is the case when only a few incomplete parts of a larger model are imaged. Instead, measurements can be considered individually, with measurements which are unlikely or ambiguous given the estimated camera pose rejected. This idea is used in 1-point RANSAC [5], in Innovation Gating [1], and when fitting Active Shape Models to images [6], however when outliers are incorrectly accepted by these approaches the solution can drift, leading to subsequent incorrect data associations and eventually positioning failure.

In our system, knowledge of the structure of the trellis (and vines) is used to guide the entire reconstruction process, however another option is to attempt to build a 3D mesh or point cloud from images, and only then to fit a model. A 3D point cloud can be computed by detecting and matching point-features, removing outliers using RANSAC, and then applying BA to reconstruct a 3D point cloud. This approach is used by Dey et al. [7] to reconstruct 3D points on images of vines with foliage. Local structure is classified as branch, leaf or fruit, so that yields can be estimated, however the method is clearly not suited to reconstructing topologically correct structures, as results show many obvious gaps and errors in the structure of branches where no suitable features were matched. This is because the points where canes overlap (occlusion boundaries) cause image features which will be detected by corner or blob detectors, but which do not actually correspond to 3D points [4].

An alternatively 3D reconstruction method, voxel carving, has considerable potential for reconstructions where the foreground is sparse compared with the background. Paproki et al. [18] use voxel carving on images of cotton plant roots, with a root network model fitted to the voxels after reconstruction. This method was considered for our application, however preliminary experiments gave poor results in regions of high cane density.

An alternative to using vision for the 3D reconstruction would be to use a 3D camera or scanner, e.g. a LiDAR laser scanner or a

---

<sup>1</sup> BA can be made partially robust to outliers by assuming heavy-tailed distributions for measurement errors (e.g. by minimising a Huber cost function of the reprojection error), however outliers can still bias the solution, leading to subsequent errors, and gross outliers which are unlikely even under these assumptions will still corrupt the model estimate.

For each new frame:

1. Detect 2D features (wires, posts and vines)
2. Detect and reconstruct structured light points
3. Assign SL points to 2D features

For each new set of stereo frames:

1. Initialise new robot pose
2. Repeat until no change in assignment:
  - Assign 2D features to 3D objects
  - Optimise with g2o
3. Correspond unassigned 2D features to initialise new structure
4. Optimise with g2o

**Figure 2: Overview of proposed framework**

Time-of-Flight camera. For our application, the lighting conditions, FOV requirements, and robot’s motion limit options for using these sensors, however LiDAR scanners have been used successfully for reconstructing the structure of tree branches by Gorte and Pfeifer [11], and by Livney et al. [15].

### 3. PROPOSED SYSTEM

This section describes how the g2o optimization framework is applied to reconstruct 3D models from high-level features extracted from 2D images. Figure 2 gives an overview of the processing pipeline, and the following sections describe each stage in turn.

#### 3.1 DETECTING POSTS AND WIRES

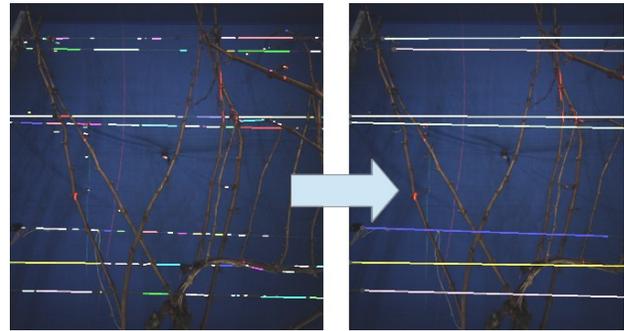
The first stage in processing each frame is to detect and localise posts and wires. A series of standard computer vision techniques is used; these methods perform well but are somewhat heuristic. Other methods may perform better.

The wires in the trellis are supported by wooden or steel posts every few metres (Figure 1). The posts are approximately vertical, and the wires approximately horizontal, and both generally extend off the top and bottom of images. Both vary significantly in size in the images, because of variation in both depth and size.

The first stage in reconstruction is to separate the foreground and background. Bayes formula is used to compute the probability of each pixel being foreground or background, given a background image. Sensor noise is assumed to be normally distributed (with a standard deviation of 5 greylevels for each channel), and shadows on the background are modelled (each pixel may be darker than the background image, with the level of shading uniformly distributed between 0% and 100%). This gives a ‘foreground image’, where each pixel’s value is the probability, under this model, that the pixel shows foreground.

To detect posts, the foreground images are filtered with a box filter, then thresholded, to remove small features (wires and many canes). The image derivative in the horizontal direction is computed, and a Hough transform [8] is applied to find the approximately-straight edges of the post. A hypothesis test is used to reject detections which are too narrow or too short to be posts.

Wires appear in the foreground image at thicknesses of between 1 and 6 pixels. To detect wire pixels, each pixel’s neighbours above and below are considered—those with background both above and below are likely to be wires. Wire pixels are merged into



**Figure 3: Wire pixels are detected, and merged into short wire segments (left). Short segments are merged to form longer segments. Different colours show different candidate segments.**

candidate wire segments by scanning left to right across the image, and merging pixels collinear to the pixels already in each candidate segment. Collinear segments are merged (Figure 3), and remaining short segments are discarded.

Long, thin canes are often detected as wires; most of these are rejected by finding the average colour of each detected wire and estimating the likelihood of that colour—on average wire pixels are more grey and vines more brown.

Preliminary work has been carried out on extracting the structure of the vines by skeletonisation [10], however currently there are too many structural errors in the 2D skeletons for successful 3D reconstruction.

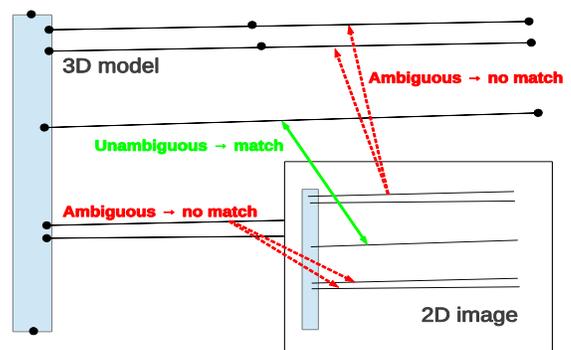
#### 3.2 STRUCTURED LIGHT SYSTEM

The design and calibration of the structured light system was described in [2]. Three planes of laser light are projected into the scene, and the locations of the red and green points in each image are identified by colour. The 3D location of each of these points can then be computed by finding the point where the ray through that pixel intersects the plane of structured light.

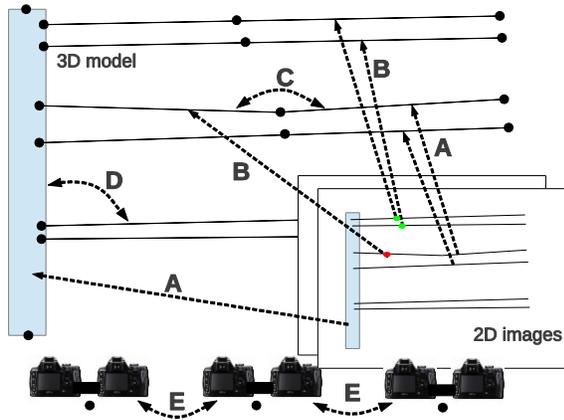
Each of the 3D points from the structured light system is assigned to a 2D feature extracted from the images. Those with ambiguous assignments are rejected.

#### 3.3 CORRESPONDANCE

The 2D image features observed are now matched with parts of the 3D model. Firstly, image features are assigned to existing parts of the model, and secondly, unassigned image features are



**Figure 4: Features are only assigned to model parts if the match is unambiguous.**



**Figure 5: Control points for the 3D model components, and camera poses form the vertices in the  $g2o$  graph (marked with dots). Every time a set of frames is captured, a new camera pose and set of constraints ( $g2o$  edges or multiedges; dashed arrows) are added to the  $g2o$  graph (see text for explanation).**

corresponded together to initialise new model components.

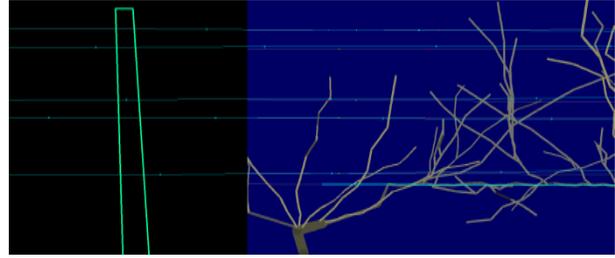
For matching 2D features to 3D model parts, a simple and safe approach is taken, which minimises the chance of incorrectly matching features. The estimated camera pose, and its associated uncertainty, is used to project each model part (post or wire segment) into the image, then the distance of the projected model from each 2D feature is measured (the reprojection error if the assignment is correct). All matches which could be ambiguous are rejected (Figure 4).

2D image features which are not assigned to any model parts, and which are not marked as ambiguous, are now used to initialise new 3D parts of the model. Every pair of 2D image features from two frames which could possibly be matched is reconstructed in turn, and the likelihood of each candidate reconstructed model is computed, given knowledge of the scene structure (e.g. wires are approximately parallel to the direction of motion, and go close to, but do not intersect posts). The matches which are most likely to be correct and where each 2D features is matched unambiguously are accepted, and new 3D model parts are initialised.

For both matching and assignment, 3D structured light points associated with the 2D features can be used to exclude many incompatible matches.

### 3.4 MODEL FOR $G2O$ OPTIMISATION

The  $g2o$  framework estimates the parameters associated with vertices in a graph, subject to constraints between these vertices. For our application, the vertices are the set of camera poses, and the 3D control points in the model. When a 2D feature is assigned to a model part, constraints (edges) are added to the graph to represent this measurement. When a new camera pose is added, edges are added to impose a motion model. When new 3D structure is initialised, constraints are added between 3D components, to incorporate knowledge of the scene’s structure. Some of these constraints connect more than two vertices; these are known as multiedges (technically,  $g2o$  operates on a ‘hypergraph’), and each residual has an associated standard deviation  $\sigma$ . Examples of these constraints are shown in Figure 5, and include the following:



**Figure 6: 3D model backprojected onto a simulated rendered image (Frame 350), after passing the second post.**

- (A) Each 2D feature (wire or post segment) introduces constraints (multiedges) between the camera pose and the two model control points. The residuals used are the shortest vectors between the measured 2D feature’s endpoints and the reprojected model ( $\sigma=1$  pixel for wires,  $\sigma=10$  pixels for posts).
- (B) Each reconstructed laser line point introduces a constraint between the camera pose and two control points. The residual is the shortest vector between the laser line point and the 3D model ( $\sigma=0.02m$ ).
- (C) The high-tension wires are piecewise-straight, with small kinks. A multiedge connecting each set of three consecutive control points imposes this constraint ( $\sigma=0.02$  rads).
- (D) Wires pass close-to, but not through posts. A multiedge connecting wire and post control points imposes this constraint ( $\sigma=0.1m$ ).
- (E) A constant velocity motion model ( $\sigma=1m/s^2$ ) is imposed with a constraint between each triple of consecutive camera poses.

Currently Gaussian error distributions are assumed, so the optimisation minimises the squared residual errors, with residuals normalised by the appropriate  $\sigma$ .

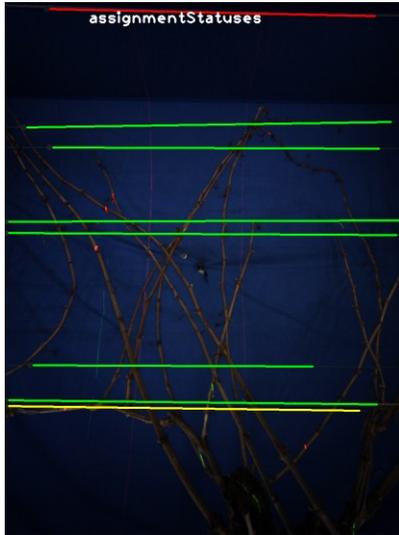
These constraints were chosen to model observed properties of the scene, and illustrate how easily additional knowledge and measurements can be incorporated into the framework.

The  $g2o$  optimisation is run after 2D features are assigned to model parts, and after new model parts are initialised. Currently the full structure and trajectory are optimised, however for long-term operation, the optimisation should be restricted to nearby structure so that computational costs do not grow with time.

## 4. RESULTS

The system is first evaluated on rendered images of simulated vines. These images are designed to recreate all of the features of the vines, posts and wires, and include a simulated structured light pattern.

The system is started when a post comes into view, and the post is reconstructed first. Not all wires are in view in every frame, but five wires are reconstructed after the first four frames, and the sixth wire is added after 18 frames. After 24 frames the post goes out of view, however positioning can continue because of the combination of the wire observations and the motion model. Five wires are reconstructed and tracked all of the way to the next post, which is ten metres, 260 frames, and 35 seconds away. In many frames, some wire features are partly occluded, and are not detected, but overall, sufficient measurements are made that five wires are reconstructed all the way to the next post. One additional wire is reconstructed incorrectly from a thin, straight cane, however this does not affect the rest of the reconstruction,



**Figure 8: Most detected wire parts are assigned to the 3D model (green) although some are ambiguous (yellow) and one false detection is unassigned (red).**

and several other false detections are ignored. When the second post comes into view, it is reconstructed and added to the 3D model (Figure 6).

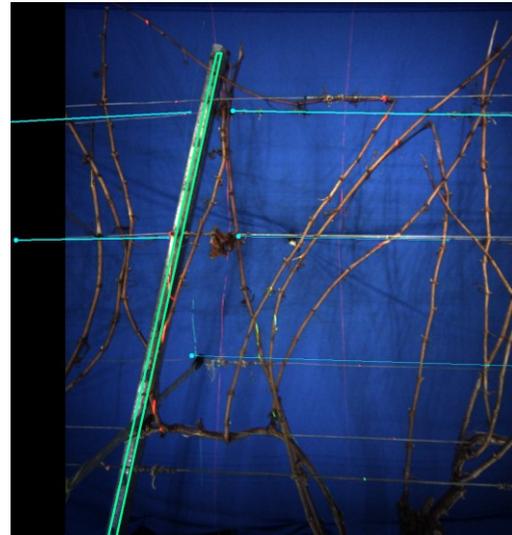
Positioning continues beyond the second post, however the optimisation becomes slow as the size of the structure grows, so the system is stopped after 360 frames. At this point, the drift in estimated camera pose is 0.48m, or 4% of the 12m travelled, however this is partly due to the motion closely matching that predicted by the motion model. The drift does not affect the relative accuracy of the local 3D reconstruction. The final optimisation includes approximately 40 000 edges between 150 3D point vertices and 360 camera pose vertices, with the optimisation itself still taking less than one second per iteration.

The system is now run with the structured light disabled. The structure and trajectory are reconstructed correctly over the first few frames, however without structured light, the data association is more likely to be ambiguous. By frame 24, no wires are assigned to any part of the existing model, and the estimated trajectory drifts away from the 3D model.

The system is now evaluated on real data collected in a vineyard. Again, wires and posts are initialised successfully, and are matched to measurements in subsequent frames. Figure 8 shows assignment statuses of some of the wires detected in a frame, and Figure 7 shows the 3D model backprojected into the frame 10. Before long, an incorrect correspondence causes a wire to be reconstructed incorrectly. The correspondence was not rejected as ambiguous because other nearby wires were occluded by canes, and were not detected. Subsequently, too few image features are matched unambiguously to the model, and the position drifts away after 16 frames. In another real dataset, positioning also fails after just a few frames because of a different kind of data association error: this time two 3D structured light points (on a small vine in the foreground) are incorrectly assigned to two wire parts. As a result, two new wires are initialised incorrectly, and subsequent data association fails.

## 5. DISCUSSION

The framework proposed in this paper is demonstrated to be capable of reconstructing the camera trajectory and a trellis



**Figure 7: 3D model backprojected to a real vine image. Not all wires are reconstructed, due to matching ambiguities.**

structure from images of real and simulated vines. Positioning continues even when only wires are visible, and a model is reconstructed despite ambiguous correspondences, poor lighting, and the presence of complex vine structures. Modelling the properties of the scene being imaged is key for robust feature detection and for reconstructing the model.

The main weakness of the system is the challenges involved in data association. Many features cannot be matched unambiguously because there are multiple similar-looking match candidates. In addition, some features are matched incorrectly, because other features are occluded and are not detected. These incorrect assignments can cause subsequent incorrect assignments, which eventually cause reconstruction to fail. Work on robustifying the system, and on better modelling uncertainty in pose is required in order to address these challenges. One promising solution could be to consider all possible assignments jointly, with a mutually compatible subset selected. The Joint Compatibility Branch and Bound algorithm [17] searches over possible assignments of image features to 3D points, and a similar approach could be applied to matching parts to models. An alternative approach being investigated is to use the optimisation framework to detect gross outliers: measurements which lead to structure which is unlikely given the model will be detected and removed.

The processing time per-frame is around one second (after 50 frames) half of which is feature extraction. The system has not yet been optimised, and currently all 3D structure is refined on each iteration. By limiting the optimisation to recent structure, and by parallelising and optimising the feature extraction, real-time (7.5Hz) operation should easily be achievable.

## 6. CONCLUSIONS AND FUTURE WORK

This paper has described how the g2o framework can be applied to a model-based 3D reconstruction problem. A partially-visible trellis which supports vines is reconstructed, and the system will now be extended to reconstruct the vines as well. By fitting a model to high-level features extracted from the image, including wire sections and posts, an accurate model is obtained. On rendered images of simulated vines, the reconstructed camera trajectory drifts by just 4% over 12 metres. On real data from a

vineyard, the 3D structure is initially reconstructed successfully, despite poor lighting and occlusion, although eventually incorrect data association causes reconstruction to fail.

Sparse 3D information from a structured light system is used to help match image features to model parts, and this enables matches to be resolved that would otherwise be ambiguous. Using the structured light, positioning continues for over 360 frames on the simulated images.

The model-based framework proposed will now be extended to reconstruct a 3D model of the vines. The vines have a considerably more complex structure than the trellis however, and many canes are partially visible or occluded. Before the vines can be reconstructed in 3D, two challenging problems must be solved: firstly, the problem of extracting the topological structure of the vines from the images, and secondly, the problem of correctly corresponding 2D vine segments. We propose to address both of these problems by modelling the connected tree structure of the vines. As demonstrated with posts and wires, incorporating knowledge of the scene into the reconstruction framework enables complete models to be built even when components are only partly visible.

## REFERENCES

- [1] Y. Bar-Shalom and T. E. Fortmann. *Tracking and Data Association*. Academic Press, 1988.
- [2] T. Botterill, S. Mills, and R. Green. Design and calibration of a hybrid computer vision and structured light 3D imaging system. In *Proc. International Conference on Automation, Robotics, and Applications (ICARA)*, 2011.
- [3] T. Botterill, S. Mills, R. Green, and T. Lotz. Optimising light source positions to minimise illumination variation for 3D vision. In *To appear in Proc. 3DIMPVT*, pages 1–8, 2012.
- [4] O. Chum. *Two-View Geometry Estimation by Random Sample and Consensus*. PhD thesis, Czech Technical University in Prague, 2005.
- [5] J. Civera, A. J. Davison, J. A. Magallón, and J. M. M. Montiel. Drift-free real-time sequential mosaicing. *International Journal of Computer Vision*, 81:128–137, 2009.
- [6] T. Cootes, C. Taylor, D. Cooper, J. Graham, et al. Active shape models—their training and application. *Computer vision and image understanding*, 61(1):38–59, 1995.
- [7] D. Dey, L. Mummert, and R. Sukthankar. Classification of plant structures from uncalibrated image sequences. In *Workshop on the Applications of Computer Vision (WACV)*, 2012.
- [8] R. O. Duda and P. E. Hart. Use of the Hough transformation to detect lines and curves in pictures. *Communications of the ACM*, 15(1):11–15, 1972.
- [9] M. A. Fischler and R. C. Bolles. Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. *Communications of the ACM*, 24(6):381–395, 1981.
- [10] W. Gittoes, T. Botterill, and R. Green. Quantitative analysis of skeletonisation algorithms for modelling of branches. In *In Proceedings of Image and Vision Computing New Zealand*, 2011.
- [11] B. Gorte and N. Pfeifer. 3D image processing to reconstruct trees from laser scans. In *Proc. Conf. of the Advanced School for Computing and Imaging*, 2004.
- [12] M. Kaess, H. Johannsson, R. Roberts, V. Ila, J. Leonard, and F. Dellaert. iSAM2: Incremental smoothing and mapping using the Bayes tree. *The International Journal of Robotics Research*, 31(2):216–235, 2012.
- [13] G. Klein and D. Murray. Parallel tracking and mapping for small AR workspaces. In *Proc. International Symposium on Mixed and Augmented Reality*, pages 225–234, 2007.
- [14] R. Kummerle, G. Grisetti, H. Strasdat, K. Konolige, and W. Burgard. g2o: A general framework for graph optimization. In *International Conference on Robotics and Automation*, pages 3607–3613. IEEE, 2011.
- [15] Y. Livny, F. Yan, M. Olson, B. Chen, H. Zhang, and J. El-Sana. Automatic reconstruction of tree skeletal structures from point clouds. In *ACM Transactions on Graphics (TOG)*, volume 29, page 151, 2010.
- [16] W. Miled, J. Pesquet, and M. Parent. Dense disparity estimation from stereo images. 2006.
- [17] J. Neira and J. D. Tardos. Data association in stochastic mapping using the joint compatibility test. *IEEE Transactions on Robotics and Automation*, 17(6):890–897, 2001.
- [18] A. Paproki, J. Fripp, O. Salvado, X. Sirault, S. Berry, and R. Furbank. Automated 3D segmentation and analysis of cotton plants. In *Int. Conf. Digital Image Computing Techniques and Applications (DICTA)*, pages 555–560, 2011.
- [19] F. Remondino and S. El-Hakim. Image-based 3D modelling: A review. *The Photogrammetric Record*, 21(115):269–291, 2006.
- [20] H. Strasdat, J. M. M. Montiel, and A. J. Davison. Scale drift-aware large scale monocular SLAM. In *Proc. Robotics: Science and Systems Conference*, 2010.
- [21] B. Triggs, P. F. McLauchlan, R. I. Hartley, and A. W. Fitzgibbon. Bundle adjustment – a modern synthesis. In *Proc. International Workshop on Vision Algorithms: Theory and Practice*, volume 1883/2000, pages 1–71, Corfu, Greece, 1999.