1-1-2007

Laser-scanning based tomato plant modeling for virtual greenhouse environment.

Tong Wang
University of Windsor

Laser-Scanning Based Tomato Plant Modeling for Virtual Greenhouse Environment

by

Tong Wang

A Thesis
Submitted to the Faculty of Graduate Studies
through Electrical and Computer Engineering
in Partial Fulfillment of the Requirements for
the Degree of Master of Applied Science at the
University of Windsor

Windsor, Ontario, Canada

2007

Library and
Archives Canada

Published Heritage
Branch

395 Wellington Street
Ottawa ON K1A 0N4
Canada

Bibliothèque et
Archives Canada

Direction du
Patrimoine de l'édition

395, rue Wellington
Ottawa ON K1A 0N4
Canada

Your file Votre référence
ISBN: 978-0-494-42259-5
Our file Notre référence
ISBN: 978-0-494-42259-5

Abstract

In this thesis, a new methodology of tomato plants structural model (SM) construction is presented. The methodology is based on the concepts from 3D reconstruction modeling methods, integrated with modern laser scanners and CAD tools to result in an architecturally accurate and human maniplable tomato plant model reflecting crop growth and management over time. A laser scanner is used for plant measurement and the resulting data are intelligently simplified. Critical data is extracted and exported to a CAD system as reference points to form a parametric model construction. A 2D corner detection algorithm is developed and applied for reference point extraction in 3D scans. A hierarchical structure is used to decompose the tomato plants into weekly-growth-sections (WGS) to relate age with plant growth geometry and solving occlusion and complexity problems. A mathematical plant growth model is developed to define geometry and topology of the plant components, WGS and the plant as a whole.

Key Words: Tomato Plant Modelling, CAD, Parametric model, Point cloud, Rule-based plant model, 3D reconstruction, Laser scanning, corner detection

Acknowledgements

It is my pleasure to thank the many people who made this thesis possible. I would have been lost without them.

I have my gratitude to my supervisors, Dr. Jonathan Wu and Dr. Sherman Lang, and my research project team leader, Dr. John Dickinson, with their inspiration, encouragement, and the great effort in guiding me getting through the challenges. Not only give the technical advices and suggestions, they also provided an ideal research environment in NRC London for me to explore new ideas, learn and improve.

I would like to thank Dr. N. Kar and Dr. Guoqian Zhang, my MAS committee members, for their valuable comments and suggestions on my research where I can read their enthusiasm and professionalism.

I would like to thank Shalin Khosla, Greenhouse Vegetable Specialist, Ontario Ministry of Agriculture, Food and Rural Affairs. He not only provided me the plant samples, raw statistic data from greenhouses, also his expertise and great knowledge on tomato plants management.

I appreciate Dr. Zhuming Bi, Steve Kruithof, Marcel Verner, Linghe Yang, sherry Yang, and other members of VETC and Reconfigurable Manufacturing Group, IMTI NRC London, for their support and technical assistances.

Lastly and most importantly, I wish to thank my wife Hui Qiao, daughter Adeline Wang, my son Jeffrey Wang, I could not complete this thesis without their love, support and patience.

Table of Contents

Chapters

Appendices

List of Figures

List of Tables

Tables Page

Terminology

Terminology for 3D CAD

2-D	A concept of displaying real-world objects on a flat surface showing only height and width. This system uses only the X and Y axes.
3-D	A way of displaying real-world object in a more natural way by adding depth to the height and width. This system uses the X Y and **Z** axes.
Boolean operations	Commands that allow you to add, subtract or intersect solid objects in AutoCAD.
Complex surface	Generally a curved surface. Examples: car fender, landscape contour.
Extrude	The extrude command raises the shape of a 2D outline into a 3D solid. For example, a circle would be extruded into a cylinder.
Face	The simplest true 3-D surface.
Facet	A three or four sided polygon that represents a piece (or section) of a 3-D surface.
Isometric Drawing	A simple way of achieving a '3-D' appearance using 2-D drawing methods.
Plan View	Also known as the top view, a plan view looks directly down the WCS Z-axis to the X-Y axis.
Primitive	A basic solid building block. Examples would be boxes, cones, cylinders.
Region	A 2-D area consisting of lines, arcs, etc.
Rendering	A complex way of adding photo-realistic qualities to a 3-D model you have created.
Shading	A quick way of adding color to a 3-D object you have drawn.
Solid Model	A 3-D model creating using solid 'building blocks'. This is the most accurate way of representing real-world objects in CAD.
Surface Model	A 3-D model defined by surfaces. The surface consists of polygons. (See facets.)
Thickness	A property of lines and other objects that gives them a 3-D like appearance.
vertex	a zero-dimensional part of a higher dimensional entity, e.g. part of a geometry, a polygon, or polyhedron
Viewport	A window into your drawing showing a particular view. You can have several viewports on your screen. Different from the viewports used in

	plotting.
Wire-frame Model	A 3-D shape that is defined by lines and curves. A skeletal representation. Hidden line removal is not possible with this model.

Terminology for Tomato Plant Modeling

Model	a pattern, plan, representation, or description designed to show the structure or workings of an object, system, or concept.
Rule-based modeling	Based on previous experiments or measurement, using compact rules or grammars to generate a model.
Image-based modeling	methods rely on a set of images of a scene to generate a three-dimensional model.
Range-based modeling	Model an object using 3D data.
L-system	a formal grammar (a set of rules and symbols) most famously used to model the growth processes of plant development, though able to model the morphology of a variety of organisms. L-systems can also be used to generate self-similar fractals such as iterated function systems
Parametric model	Each entities in CAD, are defined by parameters which control the various geometric properties of the entities and also the locations of these entities within the model. These parameters can be changed by the operator as necessary to create the desired part.
Feature-based model	A modeller allow operations such as create holes, fillets to be associated with specific edges and faces. when edges or faces move because of a regeneration, the feature operations move along with it, keeping the original relationships.
Laser Scanner	a device that analyzes a real-world object or environment to collect data on its shape and possibly color. The collected data can then be used to construct digital, three dimensional models that are used in a wide variety of applications.
3D reconstruction	The process of converting a point cloud into a polygonal 3D model is called reconstruction. Reconstruction involves finding and connecting adjacent points in order to create a continuous surface

Physiological model	Integrating the basic physiological functions that reflect the mechanism of plant growth and the relation between the plant and agricultural environment, such as plant growth vs. carbohydrates.
Morphogenetic model	Comprise object geometry and topology data, providing shape information.
Registration	the process of transforming the different sets of data into one coordinate system.
Solid modeling	the unambiguous representation of the solid parts of an object, that is, models of solid objects suitable for computer processing. It is also known as volume modeling.
Time-of-flight	finds the distance of a surface by timing the round-trip time of a pulse of light. A laser is used to emit a pulse of light and the amount of time before the reflected light is seen by a detector is timed
Triangulation	the triangulation laser shines a laser on the subject and exploit a camera to look for the location of the laser dot. Depending on how far away the laser strikes a surface, the laser dot appears at different places in the camera's field of view. This technique is called triangulation because the laser dot, the camera and the laser emitter form a triangle.
Topology	the connectivity of a set of points where adjacency is given by the edges regardless of the location of the points
Digitizing	the process of using a 3D digitizer to capture points and output scanned data; also referred to as scanning
3D digitizer	any device (contact or non-contact technology) that can capture points (x,y,z coordinates) that describe the surface of a physical object, as well as possibly other information (color, intensity, point organization, etc.); also referred to as a scanner
3D digitized dataset	the data (captured points) generated by a 3D digitizer; also referred to as a point cloud or a scan

1. Introduction

1.1 Objective, scopes and purposes

In this research, issues of modeling a complex object using modern technologies are addressed. The goal is to develop a methodology to model structural aspects of greenhouse tomato plants (here after called "plant"). The model needs to include detailed geometrical and topological information; sufficient to support 3D walk-through visualization, measurement and inspection. Users are allowed to manipulate the model by changing parameters to reflect plant growth over time and plant variety. The criteria considered in this work are:

- Accuracy: in terms of object geometry and topology;
- Level of automation: in terms of raw data acquisition, processing and model generation;
- Flexibility: in terms of variety of applications;
- Computational cost: in terms of data size and processing time

To accomplish the objective and satisfy the criteria required, new technologies for 3D measurement and model construction are deployed in the modeling system. In general, the scope of this research thesis can be summarised as:

1) Review and investigate the current tomato plant structural model construction methods by both theory and experiments;

2) Develop an appropriate modeling methodology for a complex object (like plants) visualization, measurement and inspection

There are two purposes to build a tomato plant structure model.

1) A plant structural model allows us visualize, measure and inspect the physical structure of a plant, by which plant functional characteristics can be derived and the interaction with the environment can be studied (see Section 1.5).

2) Modern automation and computer technologies have been dramatically changing the nature of labour tasks in most of the industries. While, today, operations (e.g. harvesting) in crop management in agriculture industry are still heavily depend on labour manual work, though automate crop management has been studied for decades. One of the many difficulties that delay the automation process in crop management is quick and accurate object detection and localization. As a digital map of a plant, a digitized structural model will help in this aspect by offering accurate and detailed positional information of components (e.g. fruits) of a plant.

There are great similarities among a tomato plant with the other crops in terms of geometry and topology. Thus, the developed modeling methodology and applications can be applied to the other crops as well.

It is necessary to mention that modeling a crop (e.g. tomato plants) needs continuously monitoring and simulating the growth process of multiple samples with different environment settings, requiring heavy human and equipment resources. This research is a pioneering work in terms of technologies used and the concept of plant decomposition. In this thesis, we are expecting to delivery a right direction and an advanced foundation for future studies in this area.

Figure 1: Typical tomato plants in a greenhouse, Leamington, Ontario, July 2006

1.2 Outlines

Chapter 1 covers general introduction of this thesis, including the research objective, scopes, background and motivations of tomato plant modeling. Chapter 2 presents current methodologies review, where rule-based and 3D reconstruction methods were investigated and detailed. The Problems statement and the proposed solutions were presented in Chapter 3, where the new modeling system overview, plant mathematical model, scan data simplification and proposed reference point extraction, model validation and system interface, as well as concept and procedures of CAD parametric model construction are illustrated. Chapter 4 describes the implementation and experiments results based on the new methodology. Plant measurement using a Minolta scanner, including the specifications of the hardware and operating procedures are detailed in section 4.1. Scan data processing using PolyWorks including scan data acquisition, registration, data simplification and reference point extraction are collected in section 4.2. Plant mathematical model construction, measurement of tomato plants are included in section 4.3. Measurement results, generated CAD components library and WGS library are shown in section 4.4. The conclusions, summary of the contributions and issues for further research are presented in Chapter 5.

1.3 Plant structural model construction overview

As stated by Van Der Heijden, P.H.B. De Visser and E. Heuvelink (2003), plant models can be classified along the axis from structural to functional models. At one side of the spectrum, models are mainly architectural or geometrical in nature (structural model, SM). At the other end, models are predominantly process-based (Bouman et al. 1996), often referred to as functional models (FM). The SM comprises geometry (i.e. the size, shape of plant components) and topology (i.e. the physical connections between plant components), providing shape information of a plant [1][16]. Conversely, FM integrate the basic physiological functions that reflect the mechanism of plant growth and the relation between plant and agricultural environment.

In structural modeling domain, crops (e.g. tomato plants), remain one of the most difficult kinds of object to model due to their complex geometry and wide variation in appearance. Compare with other objects, a tomato plant is complex and various in structures. Hundreds of plant organs (e.g. leaves and trusses) make up a mature tomato plant. Each organ is an individual object. None of the objects is identical from others in terms of geometry and topology.

Computer modelling and visualization of plant development can be traced back to 1962, when Ulam applied cellular automata to simulate the development of branching patterns, thought of as an abstract representation of plants [6]. Subsequently, Cohen presented a more realistic model operating in continuous space and Lindenmayer proposed the formalism of L-systems as a general framework for plant modelling, and Honda introduced the first computer model of tree structures. From these origins, plant modelling emerged as a vibrant area of interdisciplinary research, attracting the efforts of biologists, applied plant scientists, mathematicians, and computer scientists. Computer graphics, in particular, contributed a wide range of models and methods for synthesizing images of plants [6].

Conventionally, there are four methods developed for plant structural or geometrical model construction.

Interactive modeling tools

User constructs objects with drawing programs, e.g. mechanical CAD, like AutoCAD, Catia, Pro/Engineer, Unigraphics, SolidWorks. In this thesis we use SolidWorks 2004/2006 as our CAD tools

Scanning tools

In this category, geometry of objects is acquired with active sensors, like CAT, MRI, laser, magnetic, etc. The collected data can then be used to construct 3D models that are used in a wide variety of applications. In this thesis, we use Fero Laser scanner and Minontal Vivid 910 laser scanner to collect tomato plant 3D information.

Computer vision

Infer 3D geometry from images, using stereo, motion, constrains, etc. Obtain depth information using stereovision algorithms.

Procedural generation

The goal is to describe 3D models algorithmically using sweeps, fractals, and grammars. The advantages of using procedural generation system are: automatic generation, concise representation and parameterized classes of models. Rule-based plant modeling method is the representative of this class, in that, a structural plant model is built by applying production rules. In this research, we use L-studio to investigate rule-based modeling.

1.4 General criteria

in the purpose of evaluation of geometrical modelling, Sabry F. El-Hakim and his team made a concise summary of criteria in [25]. We use them as a guideline for our modeling methodology development.

- High geometric accuracy: in terms of positions, orientations and sizes of components and the plant as a whole.
- Capture of all details required: in terms of level of resolution that the measurement can be accomplished.
- Photorealism: in terms of the degree of faithfully reassemble of the object.
- High automation level: in terms of minimization of user interaction during the modeling process.

- Low cost: in terms of hardware (sensor) cost, and computation (time and data size) cost.
- Portability: in terms of hardware feasibility and accessibility.
- Application flexibility: in terms of the wideness of the range of the modeling method applications, i.e. the same modeling system can handle broad range of objects.
- Model size efficiency: in terms of the ability of presenting more information with fewer data size. It is desirable that the reconstructed surface description is as simple as possible while preserving its details and accuracy [26].

1.5 The Motivations

As a digital map of plants, a plant structural model is the tool that enable us to visualize, monitor and investigate plant growth, to test hypotheses about how different parts of the system interact, and to develop an understanding about the system as a whole [1].

Plant modelling started in the early 1970's and being developed significantly in 1980's. It has been motivated by the need for quantitative information to improve decision-making using the emerging computer technology. Plant models can be used at an operational level to simulate the crop processes that interact with the environment (i.e. greenhouse climate). At the tactical level, models are needed to relate the general policy of environment control and crop management to yield formation during the crop cycle [2].

The growers can utilize the model to predict a plant's growth, understand its behaviour better and improve how it behaves in certain conditions and environment, optimizing the quality of the crop production. The growers can start their desired production right at the very beginning by controlling the biochemical and physical conditions more closely.

From the grower's point of view, plant models can be used for production optimization, decision support systems and yield prediction. Use of crop simulation models incorporating local climatic conditions with management operations may increase the grower's ability to make more timely and reliable decisions.

For researchers, scientific crop growth simulation models have traditionally been used to address research problems, answer questions and most importantly, to increase knowledge on crop growth, development and yield, see example [4].

For greenhouse automation, 3D plant models enhance 3D digital map of the greenhouse that can be embedded into robotic computer system for object detect and localization, i.e. robotic navigation and harvesting.

For example, as shown In Figure 2, previous decision (dotted line) was made by growers using their experience or early experimental data; Now, by simulate and inspection a functional-structural model with the plant production with various settings, an optimal setting point can be achieved [5].

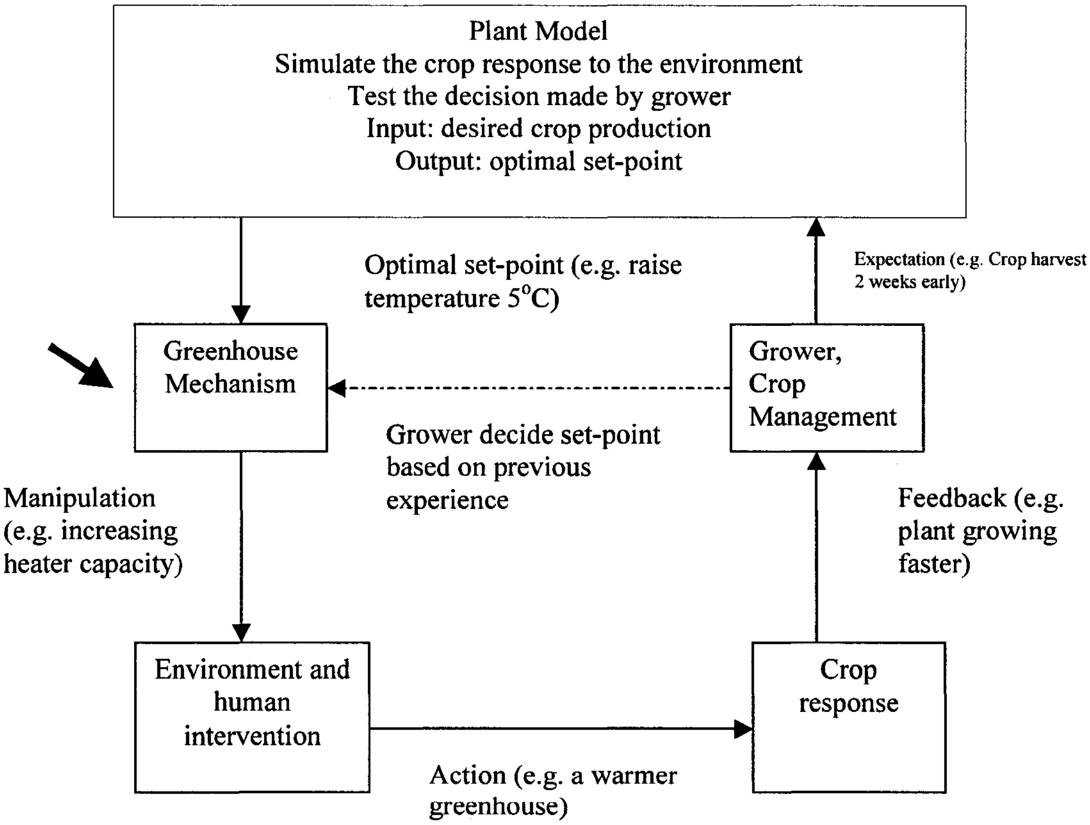

Plant Model
Simulate the crop response to the environment
Test the decision made by grower
Input: desired crop production
Output: optimal set-point
Optimal set-point (e.g. raise temperature 5°C)
Expectation (e.g. Crop harvest 2 weeks early)
Greenhouse Mechanism
Grower, Crop Management
Grower decide set-point based on previous experience
Manipulation (e.g. increasing heater capacity)
Feedback (e.g. plant growing faster)
Environment and human intervention
Crop response
Action (e.g. a warmer greenhouse)

Fig. 2: An example of a plant model used in plant crop management system

2. Plant modeling methodology review and investigation

2.1 Rule-based plant modeling

Rule-based methods use compact rules or grammar creating models of plants and trees. As a prime example, Prusinkiewicz developed a series of approaches based on the idea of the generative L-system in 1994 [6]. Weber and Penn use a series of geometric rules to produce realistic-looking trees in 1995 [12]. There are a number of techniques that take into account various kinds of plant interaction with the environment.

2.1.1 Modelling plant as geometrical object

In rule-based modelling domain, a simple plant can be modelled as a geometrical object by (a) using simple 3D-symbols, or (b) by applying texture maps to improve the visual quality of simple symbols. However, for detailed modelling to a complex plant with thousands of individual objects, algorithms for automatic object generation (i.e. L-system) have to be applied [17] (see Section 2.1.2).

For (a), there are numerous symbol sets available on the market, with a higher degree of detail and therefore larger file sizes, but still far away from real vegetation structures. For (b), a very popular method of plant visualisation in landscape architecture is the use of texture mapping in connection with simple 3D-faces. Texture mapping implies the projection of raster graphics onto a modelled surface in order to alter the surface characteristics such as the colour or the transparency [17]. In the case of plant visualisation, a photograph of a real tree can be mapped onto a rectangular vertical face; in the rendering process the areas of the image background are treated as transparent. By applying this simple method a high degree of realism can be achieved [17].

2.1.2 Automatic generation of plant structures using rule-based method

Plants consist of a large number of individual elements, however, the configuration of these elements follows relatively simple rules (e.g., the branching pattern within a genus is usually constant). Therefore plant-modelling algorithms have to find ways for a formal mathematical description of the 'genetic construction plan' of a plant [2][17].

Simple fractal structures

Benoit Mandelbrot's work on fractal geometry provides a means for the simple description of complex structures found in nature [17]. The principles of self-similarity, which form an important basis for fractal geometry, can be observed both in landscape morphology and biological organisms. It is easy to create simple tree-like structures with the help of fractal construction rules implemented in a visualisation system.

However, there are limitations to the degree of realism achievable with this approach. A typical example shows branching patterns: While it is obvious that a specific branching pattern is repeated from the major branches down to smaller twigs, this cannot be repeated infinitely. Many trees change their branching pattern with age. Also, there are phases of vegetative growth alternating with phases of generative growth, which are difficult to model within a standard fractal graphics system [17].

The AMAP system

As opposed to the deterministic approach of L-system based method, the AMAP system is a stochastic model based on the specific statistical probability of the occurrence of a certain phenomenon, such as branching, trunk length, fructification, etc [6][17]. These parameters must be derived from field data for every single plant species; therefore the data collection and parameterisation for different growth stages is a very time consuming procedure.

AMAP was originally designed for the modelling of agricultural crops, and was further developed into a commercial visualisation system for landscape architecture. A big advantage of the practical implementation in the visualisation module is the fact that the individual plants are being modelled temporarily during the rendering process. This means that only the plant parameters have to be stored, but not the millions of polygons that form a scene. This keeps file sizes for large projects in a manageable range [17].

2.1.3 Modelling plants using L-system

L-systems

L-systems are a mathematical formalism proposed by the biologist Aristid Lindenmayer in 1968 as a foundation for an axiomatic theory of biological development [6][13]. More recently, L-systems have found several applications in computer graphics. Two principal areas include generation of fractals and realistic modelling of plants. Central to L-systems, is the notion of rewriting, where the basic idea is to define complex objects by successively replacing parts of a simple object using a set of rewriting rules or productions. The rewriting can be carried out recursively [13].

With the help of this approach, plants can be modelled in the same way as they grow in reality, from an initial apical bud to the development of stems, branches, leaves etc. Different hormonal stages within a plant's life, such as juvenile growth, flowering and fructification can be simulated. With relatively few lines of program code, very complex structures can be modelled. In connection with high-end rendering systems, amazing images of many different types of organisms have been produced.

L-system formalism is able to generate detailed and realistically visualised 3D plant in computer, which is further refereed to as "virtual plant" [13][16].

L-studio and Vlab

L-studio and Virtual Laboratory (Vlab) are one of the most recent L-system based plant modelling tools developed at the University of Calgary, Canada. In that, plant growing with human intervention (i.e. pruning) can be simulated.

The simulation programs cpfg and lpfg are the heart of L-studio. Their design has been guided by two key objectives: (a) flexibility, making it possible to model and simulate a wide range of structures and developmental processes in plants, and (b) visual realism of models [15].

Both cpfg and lpfg make it possible to construct models with interactive features. The user can interact with the models by selecting a model component with a mouse to trigger a programmable action associated with this component. Some applications of this feature include: simulation of pruning and grafting, interactive placement of insects or pathogens on selected parts of the plant, selective increase or decrease of the level of detail in the model presentation, and selective display of parameters associated with the model. Based on their input (a number of files including the L-system model and optional data files), cpfg and lpfg create a three-dimensional internal representation of a model and project it on the screen. Model visualization is based on the OpenGL graphics library and supports standard modelling and rendering techniques developed in computer graphics, such as parametric surfaces, generalized cylinders, and texture mapping [15].

The output may be a static model (which can be interactively rotated and zoomed in or out by the user) or computer-generated animation that results from visualizing consecutive stages of the simulation. The visualizations may have the form of schematic or realistic representations of branching structures, as well as plots and histograms. The multi-view presentation mode, available in lpfg, makes it possible to simultaneously visualize different aspects of the simulation in separate windows on the screen. For instance, one view may realistically depict a growing plant, while another view may represent quantitative aspects of the simulation in progress as function plots [15].

A experiment using L-Studio 4.0

The purpose of the experiment is to test the functionality of L-Studio. The following figures capture some experimental results and illustrate the functionality of L-Studio. For all of the examples the software used was L-Studio 4.0 (Evaluation version), working on the Lychnis example plant, modified by Tong Wang, July 2006.

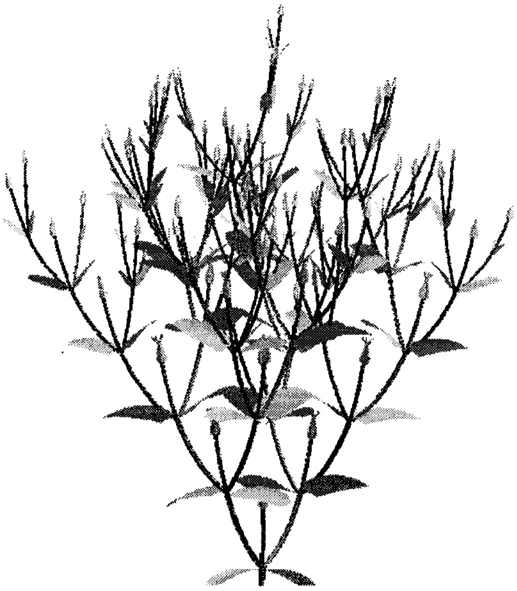


Figure 3: Lychnis example plant with flowers disabled.

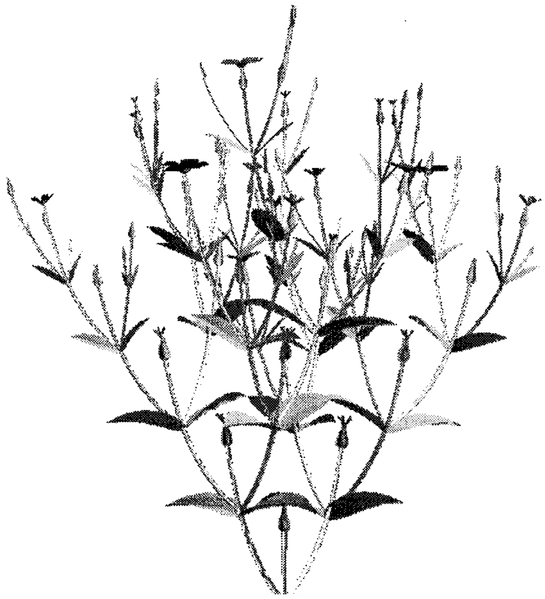


Figure 4: Lychnis example plant with flowers enabled.

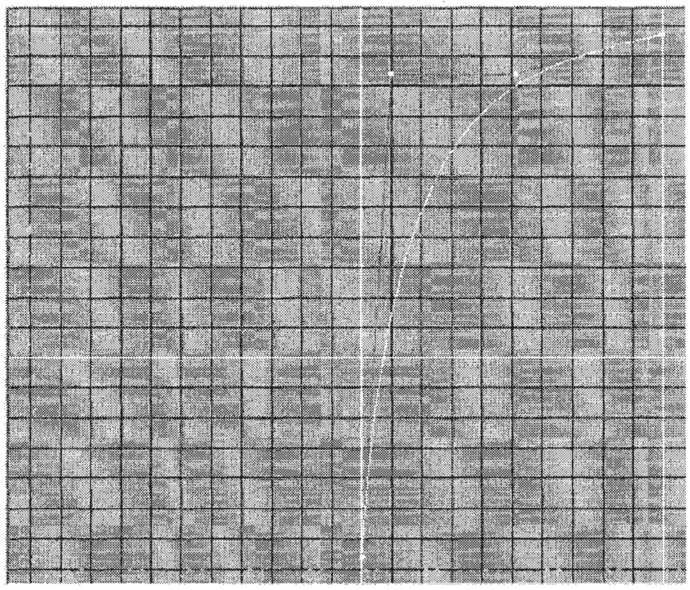


Figure 5: Original branching angle function: angle over position

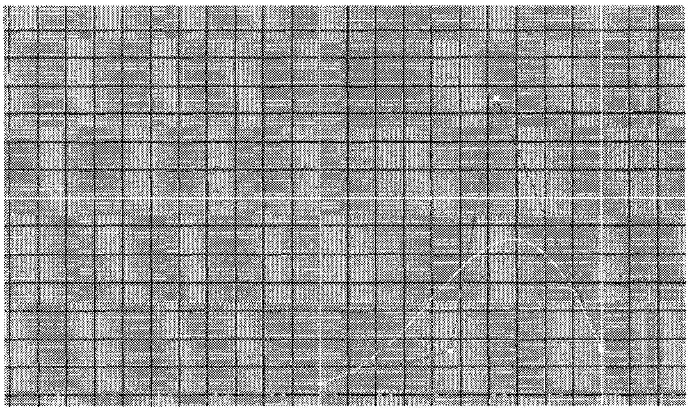


Figure 6: Modified branching angle function: angle over position

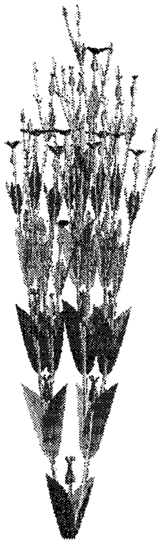


Figure 7: Lychnis example plant with modified branching angle pattern.

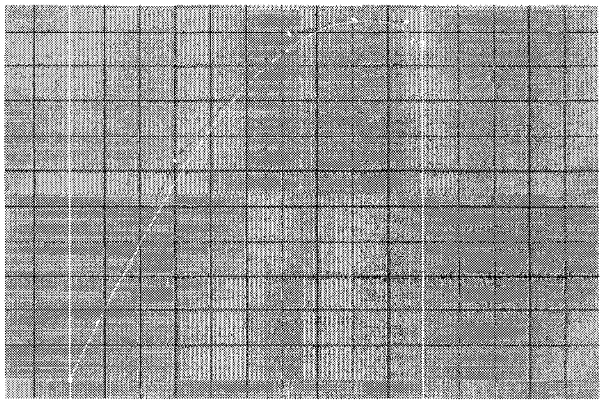


Figure 8: Original leaf growth function: size over time

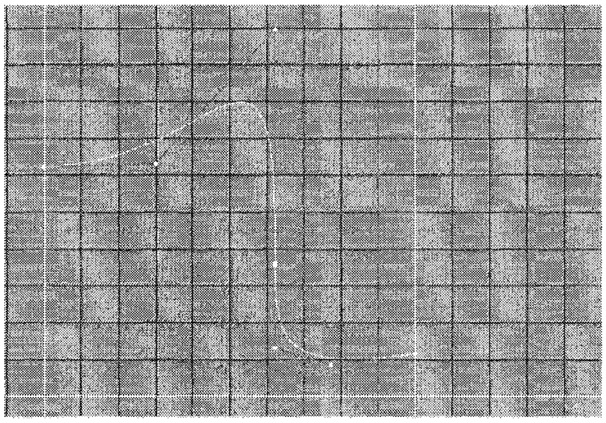


Figure 9: Modified leaf growth function: size over time

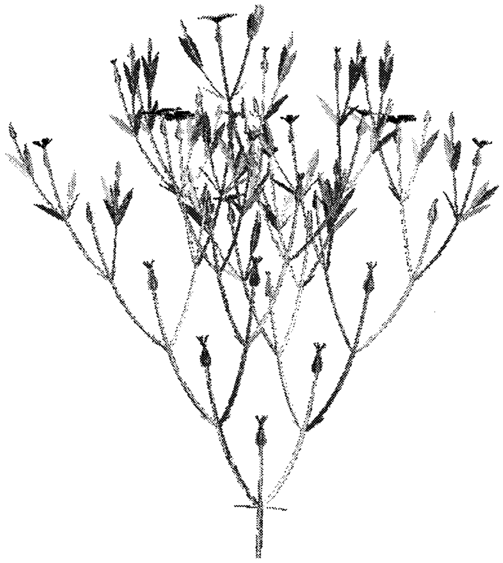


Figure 10: Lychnis example plant with modified leave pattern.

2.1.4 Summary of rule-based method

Rule-based plant modelling uses compact rules or grammars to generate realistic-looking plant synthetically [12]. The "rules" are derived from plant database, which can be obtained either from theoretical models, previous experimental data, or by 3D scan measurement [17]. "Rule-based" morphogenetic models can be parameterized and validated by 3D digitized plant. The beauty of rule-based plant models is that they provide a clear interface connecting rules and the spatial and textured structure of the plant. The rule decides the appearance of a plant [15][23].

The purpose of rule-based architecture tomato plant modelling (i.e. L-studio) is for generating a relationship between plant growth and the environment [12]. The plant models are based on previous experiments data and assumptions, which are defined by rules. However, rules can represent certain type of growing parameters, but not all. For example, a greenhouse r tomato plant is not growing naturally; on the contrary, it is in a specific environment with heavy human intervention, i.e. the plant is hanging up on the overhead wires, and with constant pruning activities. There are no rules available for this particular growing Patten. Thus, the geometrical accuracy is not satisfactory for plant architecture analysis in this case.

Moreover, rule-based plant models used in computer graphics are commonly created with procedural methods. A difficult problem is the user control of the models: a small number of parameters are insufficient to specify plant characteristics in detail, while large numbers of parameter are tedious to manipulate and difficult to comprehends [9]. The drawback of the rule-based model is that it needs expertise to use [12], and for large plant or group of plants, it is very time-consuming, as the un-structured nature of a plant having too many organs and variables involved.

From the global level, it is clear that different plant growing have different rules, even same plant may show different growing attributes with various environments. As the model is made by rule, the accuracy of the rule-based plant model is subject to the robustness of the rules as well as the input parameters.

2.2 Modeling plant using 3D reconstruction method

2.2.1 The concept

As mentioned in section 1.3, based on methodology, structural plant models can be achieved from rule-based plant modelling, or from 3D plant reconstruction. The latter can be accomplished either from image-based plant modelling, or using range-based approach. Basically, a morphogenetic plant model comprises the plant measurement and model construction based on the measurements. Rule-based modeling approach use computer graphics and rendering techniques (which are defined by rules) to construct the model based on the architectural measurement [21]. While, 3D reconstruction, use surface meshing techniques, taking 3D point cloud as reference to construct a polygonal surface model. In contrast to rule-based modelling strategy, 3D reconstruction directly models the plant using real world image and the resulting model inherits the realistic shape and complexity of the plant.

3D reconstruction builds models based on object measurements. Fabio described the goal of surface reconstruction as follows [34]: "given a set of sample points P, assumed to lie on or near an unknown surface S, create a surface model S' approximating S". A surface reconstruction procedure cannot guarantee the recovering of S exactly, since we have information about S only through a finite set of sample points. In general, as the sampling density increases, the output results S' is more likely topologically correct and converges to the original surface S.

From the above statement, we can conclude that 3D reconstruction comprises 2 steps:

1. Measurement: capture 3D information in point cloud format (called unorganized points).
2. Intelligently compress point's density, forming a polygonal surface model

2.2.2 Image-based approach

The principle

Image-based plant modelling uses stereovision reconstruction techniques found in the field of computer vision. It can be described as "A process of recovering 3D structure from 2D images at their overlapping area. That is the ability to interpret information on the 3D structure and distance of an object from two or more images taken from different viewpoints" [10]. To capture a full 3D representation of an object, close-spaced images surrounding the object are required [25][31].

Basically, stereo vision must solve two problems [8][10].

- Correspondence problem: determine which pixel on the left corresponds to which pixel on the right. It is a "search" problem: find features to match or similarity measure to compare features.
- Reconstruction problem: given a number of correspondence pairs and camera geometry information, find location and 3D structure of the observed objects.

Deriving 3D measurements from images naturally requires that interest points be visible in the image. Often, this is not possible for a complex object or combination of many objects either because a region is hidden or occluded behind an object or a surface or there is no mark, edge or visible feature to extract, (like tomato plant). In objects such as monuments in their normal settings, restrictions also stem from there being limited locations from which to take the images and from the existence of other objects, shadows, and uncontrolled illumination [25].

Reconstructing a 3D plant model from 2D images

At plant structural modeling domain, stereovision has been used by Ivanov et al in 1995, where a canopy of maize plants has been reconstructed. In this set-up, two cameras at a fixed distance apart record the same scene. Extensive calibration of cameras leads to an estimate of the perspective-transform matrix for each of the cameras. From the shift of corresponding points in both views, the 3D position in real-world coordinates can be computed. Sequential manual removal of leaves and image recording was necessary to obtain a full reconstruction of the stand. A standard deviation of about 1 cm in Xand Y-direction and 5 cm in Z-direction could be obtained, whereas bias was rather low. Stereovision as applied by Ivanov et al. required extensive manual intervention [31].

Nowadays, many different photogrammetry software tools exist, where pictures of a scene are taken under a large number of angles with a calibrated camera. By interactively indicating object features like points, lines or edges, the software will try to reconstruct the points across the photos, resulting in a 3D model. Phattaralerphong and Sinoquet (2005) created artificial 3D scenes of trees, using digitized data at leaf scale. The hence created scene was used to synthesize plant images using POV-Ray®. From these images, canopy volume parameters were estimated, using photogrammetic principles.

In summary, there are 2 major problems in correspondence pair setting:

- Occlusion: leaves overlap, branches are covered by leaves.
- Illumination levels vary for different viewpoints: This fact causes pixel intensity values to vary which can lead to correspondence search failure.

To overcome these problems, Quan presented his approach in [12]. There were three parts in the system:

- **Image acquisition and structure construction from motion**: to recover the camera parameters and extract point correspondence, obtain 3D point cloud.
- **Leaf segmentation and recovery**: to get a "sample leaf", a deformable generic model which will be used to fit the other segmented data to model all the other visible leaves which are under occlusion.
- **Interactive branch recovery**: the branching structure is difficult to reconstruct automatically from images due to occlusion. The solution is to design a data-driven editor that allows the user to easily recover the branch structure. The user is presented with an interface with two windows: an area showing the current synthesized tree, and the other showing the synthetic tree superimposed on an input image. The image can be switched to any other image at anytime. The user can have four basic operations: draw curve, move curve, edit radius, specify leaf.

In this system, for a given plant, the user first segments out a leaf; this is used as a deformable generic model, which is subsequently used to fit the other segmented data to model all the other visible leaves. The system is also designed to use the images as guides for interactive reconstruction of the branches. The resulting model of the plant very closely resembles the appearance and complexity of the real plant [12].

Summary of image-based approach

The image-based modelling approach is a well-proven and reliable technique for 3D object reconstruction. It is easy to use, and the necessary hardware is usually very portable, low to medium cost and integrated with a wide availability of commercial modelling software packages [25] [31].

However, intensity values variation (i.e. illumination) from the images will cause errors when searching for corresponding pairs. In addition, issues remaining in image-based modelling include the capture of details on unmarked and sculpted surfaces and the full automatic creation of the 3D model [25]. In this framework, capturing the data for a realistically model would require a large number of closed spaced images, and deriving the depth require a large amount of user input [11][25][26]. Difficulties in using this method to model a complex plant like a tomato plant can be listed as below:

- Occlusion problem: information of components being covered or shadowed can not be recovered [12];
- Un-uniformed illumination from different viewpoints, causing difficulties on corresponding point pair searching [31];
- For detailed plant model, images are close-spaced. That means for a full plant modelling, "a lot of images" are needed [31].
- Camera calibration is required [10][31];
- Human interaction is required to initialize and supervise the computation process [12][31].

2.2.3 Range-based approach

2.2.3.1 The concept

The rapid development of laser scanning and computer graphic technology has been advancing 3D plant reconstruction using active range data recently. In contrast with the image-based method, an active sensor has the advantage of acquiring 3D information directly [25]. In this arrangement, a laser scanner is used to capture the 3D structure and a digital (or video) camera is incorporated to provide colours and textures which are then mapped on to the range image.The measured data alone, usually represented as a point cloud, lacks topological information and is therefore often processed and modeled into a more usable format such as a triangular faced mesh, a set of NURBS surfaces or a CAD model. Applications like PolyWorks is used to process the point clouds themselves into formats usable in other applications such as 3D CAD, CAM, CAE or visualization.

In this research, a 3D scanner is used for object measurement in the form of point cloud. These points can then be used to extrapolate the shape of the subject. If color information is collected at each point, then the colors on the surface of the subject can also be determined. Like cameras, 3D scanners have a cone-like field of view. They can only collect information about surfaces that are not obscured. While a camera collects color information about surfaces within its field of view, 3D scanners collect distance information about surfaces within its field of view. The "picture" produced by a 3D scanner describes the distance to a surface at each point in the picture. For most situations, a single scan will not produce a complete model of the subject. Multiple scans, even hundreds, from many different directions are usually required to obtain information about all sides of the subject. These scans have to be brought in a common reference system, a process that is usually called alignment or registration, and then merged to create a complete model.

At the following example (Fig. 11), S. Fleck, D. van der Zande, M.Schmidtc, P. Coppin used laser scans to model a tree structure, and extract the tree structural information (e.g. the height of the tree) to interpret the functional aspect of the sampled tree [21].

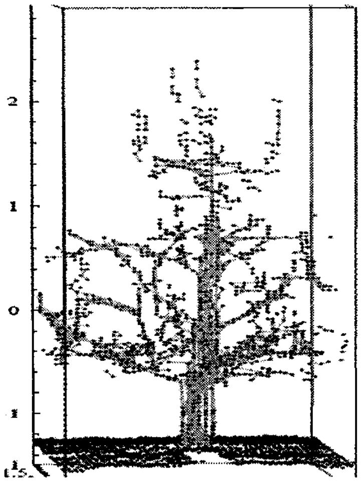


Fig.11: Reconstruction of tree structure from laser-scans: 3D point cloud of an apple tree, 1352 points after thinning. See [21]

In general, the entire modelling process can be briefly summarized as follows [30]:

- 3D measurement, multiple range data and colour images acquisition and data processing: capture needed points, remove unwanted
- Multi-scan registration: align scans as a complete view
- Fusion intensity image with range image: add colour and texture
- Transform range data into 3D CAD model: from point to surface

2.2.3.2 Methods of 3D measurement

Plant architectural measurement is the bases of model construction. Traditionally, Two main approaches exist: contact measurements, where the 3D coordinate of each point is individually captured, requiring extensive human interaction, and non-contact point-cloud measurements.

Contact measurements

A commonly used contact measurement device within the plant community (e.g. Sinoquet and Rivet 1997; Rakocevic et al. 2000; Evers et al. 2005) is the FASTRAK® magnetic 3D digitizer (Polhemus, Colchester, VT, USA). It includes a magnetic signal receiver and pointer, allowing the user to record the 3D spatial coordinates of the pointer within a hemisphere of 3 m diameter from the receiver. Individual plants are digitally reconstructed by recording a series of point coordinates and the relevant connectivity between the points. Due to its principle of creating a magnetic field, it can be used outdoors at relative ease, but in greenhouse environments the surrounding iron frames can disturb measurements.

Non-contact point-cloud measurements

Active scanners emit kinds of radition and detect its reflection in the order to probe an object or environment. Possible types of radiation used include light, ultrasound or x-ray **Error! Reference source not found.**.

Time-of-flight

The time-of-flight 3D laser scanner is an active scanner that uses laser light to probe the subject. At the heart of this type of scanner is a time-of-flight laser range finder. The laser range finder finds the distance of a surface by timing the round-trip time of a pulse of light. A laser is used to emit a pulse of light and the amount of time before the reflected light is seen by a detector is timed. Since the speed of light *c* is a known, the round-trip time determines the travel distance of the light, which is twice the distance between the scanner and the surface. If *t* is the round-trip time, then distance is equal to $(c \cdot t)/2$. Clearly the accuracy of a time-of-flight 3D laser scanner depends on how precisely we can measure the *t* time: 3.3 picoseconds (approx.) is the time taken for light to travel 1 millimetre [47].

The laser range finder only detects the distance of one point in its direction of view. Thus, the scanner scans its entire field of view one point at a time by changing the range finder's direction of view to scan different points. The view direction of the laser range finder can be changed by either rotating the range finder itself, or by using a system of rotating mirrors. The latter method is commonly used because mirrors are much lighter and can thus be rotated much faster and with greater accuracy. Typical time-of-flight 3D laser scanners can measure the distance at of 10,000~100,000 points every second [47].

Laser triangulation

The triangulation 3D laser scanner is also an active scanner that uses laser light to probe the environment. The triangulation laser shines a laser on the subject and exploit a camera to look for the location of the laser dot (Fig. 12). Depending on how far away the laser strikes a surface, the laser dot appears at different places in the camera's field of view. This technique is called triangulation because the laser dot, the camera and the laser emitter form a triangle. The length of one side of the triangle, the distance between the camera and the laser emitter is known. The angle of the laser emitter corner is also known. The angle of the camera corner can be determined by looking at the location of the laser dot in the camera's field of view. These three pieces of information fully determine the shape and size of the triangle and gives the location of the laser dot corner of the triangle. In most cases a laser stripe, instead of a single laser dot, is swept across the object to speed up the acquisition process. The National Research Council of Canada was among the first institutes to develop the triangulation based laser scanning technology in 1978 [47].

By laser triangulation, a laser point or laser line (sheet of light) is projected on the scene. This method is rather robust, although shininess, absorption by the surface or abundance of ambient light may cause insufficient contrast. Translucency of plant material makes the position estimation less accurate. Occlusions can occur if the object has concavities. Texture mapping can add colour/texture to the object.

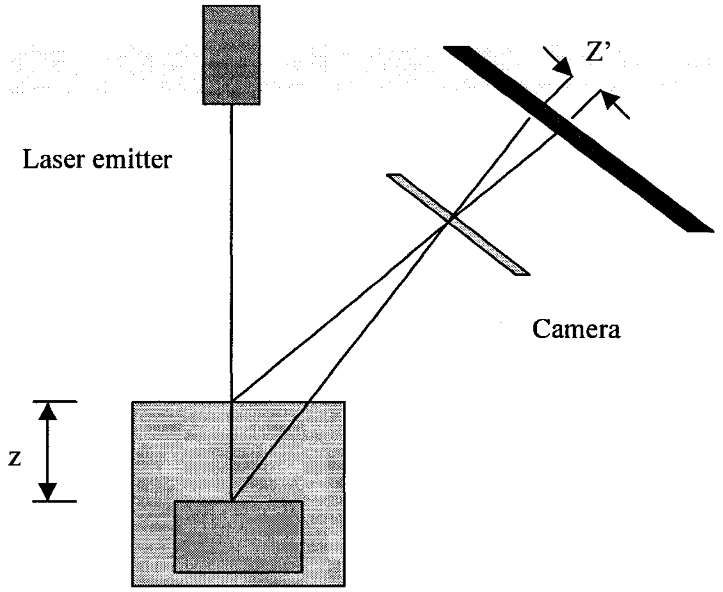


Fig.12: Principle of a laser triangulation sensor. Two object positions are shown.

Major advantages of laser triangulation are its high accuracy, the dense cloud of points that can be measured, and its robustness. It can measure distance from the camera to plant parts, still generating a reasonable 3D outline (profile) of the object in complex situations.

Disadvantages are the fixed resolution (depending on the thickness of the laser beam), which may be crucial for scanning small flowers, and scattering as a result of the translucent character of most plant parts. Kaminuma et al. (2004) use a laser range finder (type Voxelan Hew- 50HS, Hamano Eng, Japan) to reconstruct *Arabidopsis* plants. Loch (2004) used the Polhemus FastSCAN hand-held laser scanner (triangulation principle) extensively for the generation of leaf-surface models. The hand-held device keeps track of its position by means of a magnetic field, hence being vulnerable for nearby metallic objects or electromagnetic fields [47].

2.2.3.3 Range data acquisition and processing

Capturing range data

3D digitizing consists of simultaneously recording plant geometry, (i.e. the spatial location of plant components) and plant topology (i.e. the physical connection between plant components) [22]. Capturing range data for plant digitising raises several issues.

The first is that the surface may not produce sufficient laser contrast for the scanner to detect a profile. This may be caused by shininess or by the colour of a surface. Since a green surface will absorb light of any colour but green and laser light is relatively pure in colour, not enough of a red laser beam may be reflected from the object and received by the scanner cameras to calculate positions of the data points (see Fig. 13). Rather than an expensive change from a red to a green laser, the surface may be treated to change its reflective properties [20].

Secondly, plant digitising by laser scanner cannot be done under broad daylight. The reason for this is that scanning errors can occur when the scanners view direct sources of bright, broad-spectrum light [20].

Lastly, at least one out of the set of scanner viewpoints needs to have a direct line of view of the profile; otherwise the information for this part will be missing. This means that hidden plant parts cannot be scanned unless enough scans from different viewpoints are available [20].

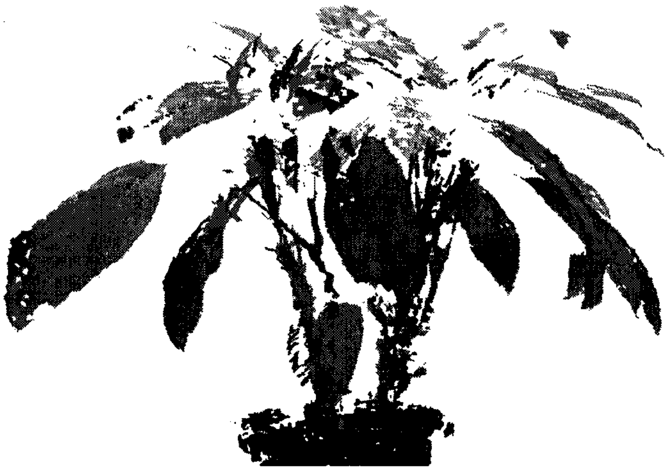


Figure 13: A test plant model: Software: PolyWorks V9; Scanner: Minolta VIVID 910, July 2006, NRC London. Measuring range setting: 1450mm; Accuracy: 0.068mm/2.5m; Total Scan taken: 12 (6 from top view, 6 from bottom view, 60 degree per step); Scanning Time: 25 Minutes; Registration Time: Manual + Auto, 90 Minutes

Potential solutions include:

- Change from a red to a green laser
- Treat surface to change its reflective properties
- Increase the laser output power

Balancing image resolution and computation cost

There is a trade-off between image resolution and computation cost. 3D geometric models extracted from laser range data provide a high amount of detail and accuracy of the scene. However, the acquisition and processing of the range data can be difficult and time consuming, and in a complex environment the size of a detailed 3D model can quickly become too large. A possible solution is to use accurate geometric models from laser range data only for the parts of the scene that are most interesting and to use alternative visualisation techniques to obtain a rough model of the general scene and provide a realistic spatial context [24] [26].

We can use high-resolution scan for object of interest, low resolution for background.

In this case, we define a special window, which is intended to be surveyed. Important objects can be measured down to the smallest details with this feature, as we can easily set a very high resolution for this window [27] (Fig.14 and Fig.15).

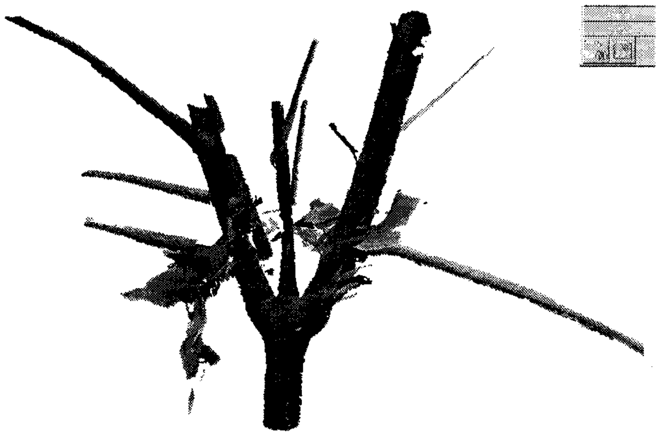


Figure 14: High resolution scan of a section of a Maple tree branch. Minolta Vivid 910, Resolution: 625 dot in 1 inch.

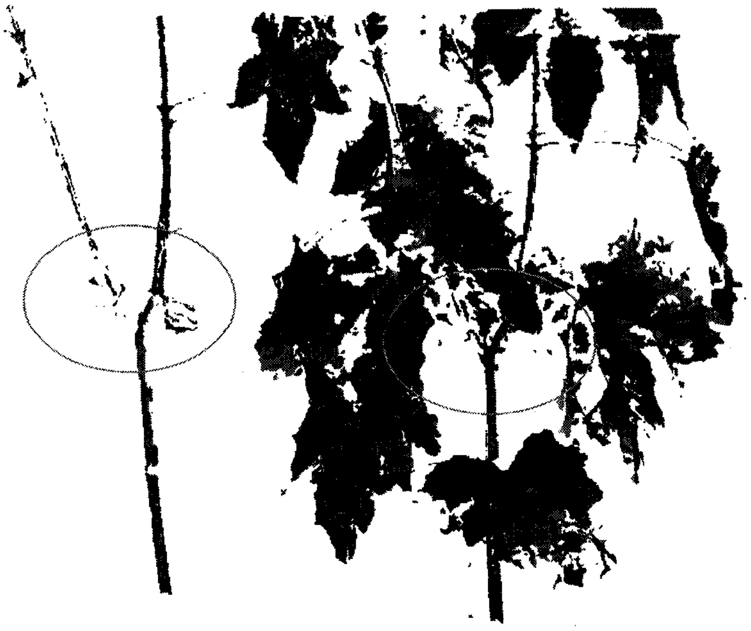


Figure 15: Low resolution scan of the same branch as in Figure 15, Minolta Vivid 910, Resolution: 286 dot in 1inch.

Point cloud filtering

The advent of laser profile scanners means that 3D plant data can be captured holistically. The data acquired from laser surface scanners is in unordered point cloud form, while points to be collected are those only on the surface of the object under study. This means the structure of the plant must be extracted from the cloud of 3D co-ordinates [20].

In addition, since background information is often captured, editing of the data with a point manipulation program such as Point Picker developed at the ACMC to remove unwanted points is desirable [20].

In practice, associate software from the laser scanner manufacturers is available for outlier removal [30][32]. However, user interaction is required for criteria setting and filtering supervision. The purpose of the supervision is to make sure that valuable data is kept and unwanted data is removed.

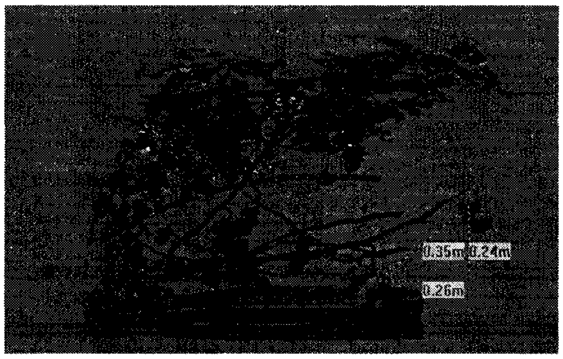


Figure 16: Laser scan at a greenhouse, Leamington, Ontario. Point cloud, low resolution, filtered with distance setting: 0 ~ 2.50m, Scanner: FARO LS 880.

2.2.3.4 Multi-scan registration and ICP Algorithm

It is not possible to have a complete 3D representation from only one viewpoint. To capture the complete geometry of a plant, it is necessary to acquire data from multiple viewpoints (Fig. 17), which are also advancing on higher resolution images in specific regions of interest. Scans to the same object from various viewpoints are registered. The purpose of data registration is to bring different scans from different coordinates into a common coordinate, that is, the transformation.

Sequeira and his team presented a system in [26] where embedded software performs several automatic functions, including triangulation of the range data, range to video registration, registration and integration of data acquired from different capture points, planning the next optimal capture position in an initially unknown large-scale scene (indoor or outdoor) (see [26]). However, fully automatic registration of multiple range images is still an area of active research in the computer vision community. State-of-the-art systems often still rely on user-interaction to determine the initial transformation, making the pre-registration a tedious and time-consuming task [25] [28] [30].

Iterative closest point (ICP) is a dominating algorithm for 3D data registration. The principle was developed by Besl & Mckay in 1992. Consider two datasets, called p and q. If we manage to get the correct transformation (rotation matrix r and translation matrix t) between p and q (Fig.18), then we can minimize the error function [45] below:

Error Function:
$E=| rp_i + t - q_i |^2$

The transformation can be derived if we know correct corresponding point pairs (at least 3 pairs for 6 unknown variables r and t). So the algorithm start from "guess correspondence point pairs" (begin with the closest point, i.e. p and q). If they are corresponding points, we are lucky to get the correct transformation. Other wise, try other point again.

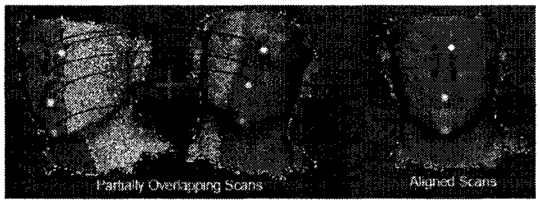


Fig. 17: Data registration using ICP: overlapping areas are required [45].

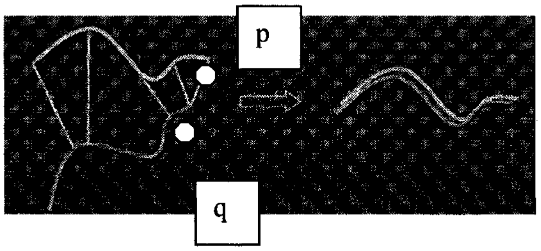


Fig. 18: Data registration using ICP [45]

Successful convergence of applying ICP algorithm requires the two data set "close" enough, as shown above. In practical application, if the scans are not posed "close" enough, user initialization is required. The purpose of the initialization is to establish data transformation. Effective transformation establishment can be derived from the following arrangements.

Using relative viewpoint positions of different scans

Additional information available from the scanning process can be exploited to derive the initial transformation. For instance, the relative viewpoint position might be known, e.g. from tracking the scanner position or by using a turntable on which the object to be digitized was located [28]. Although direct and convenient, this is not always feasible due to the nature of the object, its dimensions or location.

Using pre-defined feature points

A more general approach is to derive an initial transformation by aligning a small set of corresponding feature points in the range images. These feature points are either found as local geometric features on the surface of the object or by placing additional markers on or in the surrounding of the object [28]. For example, FARO scene uses paper targets and spheres placed in the scene for this purpose [32]. In the former case, robustness of the feature detection is of vital importance, whereas in the latter, special care has to be taken in the placement of the markers, as markers should be visible from as many viewpoints as possible whilst casting preferably no shadows on the object.

Using feature point from intensity images

On the other hand, scanning devices commonly capture not only geometry but also color information or light intensities for the scene. These intensity images are far less subject to noise and, as opposed to range images, do not exhibit missing values [28]. As a consequence, feature points extracted from these images are more robust than those extracted from range images, making them more suitable for correspondence computation [28].

2.2.3.5 Fusion intensity and range image

To achieve virtually realistic models, 2D digital images are used to "paint" the 3D representation of the scanned areas [26] that is Texture mapping. Commercial application examples can be found from FARO LT system (see [32]) and Minolta VIVID 910 system (see [30]).

To add colour and texture into point cloud, the first step corresponds to the normalisation and registration of the visual and spatial data. To achieve these, the photographic and laser reflectance images are used [26].

2.2.3.6 Summary of range-based approach

Laser scanner automatically, accurately and quickly captures 3D plant information from a real plant in the form of points cloud. Polygons are usually the idea way to accurately represent the results of measurements, providing an optimal surface description. Therefore, with the accurate 3D measurement by laser scanner and advanced software tools to convert point clouds to polygons (and further convert to shaded CAD model), it is feasible to derive accurate 3D models with high level of automation [34]. Thus, this approach owns the following two advantages:

- **Accurate and reliable results:**
  Laser scan is a faithful "resemble" of the real object. With more dense point, more details can be obtained. With matured point cloud processing algorithm and CAD modelling software, an accurate 3D representation with manageable data size is achieved.

- **Automatic data capturing**

  In terms of measurement, in contrast with rule-based approach, which still mainly relies on manual measurement, range-based modelling approach captures plant 3D data automatically.

Fig. 19 shows an experiment result of tomato plant model using 3D reconstruction approach, in that, total 64 scans are aligned. The registration was not successful due to the plant complexity. Due to occlusion, only surface data are collected. The level of details is not satisfactory for plant geometrical analysis.

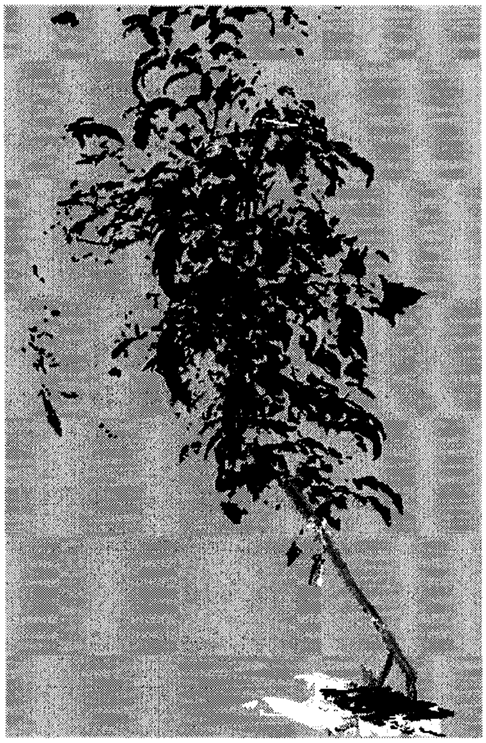


Fig.19: A typical greenhouse tomato plant polygonal model from scans. Sample taken from a greenhouse at Leamington, Ontario, Canada, 2006

3. Developing the new methodology

3.1 The challenges and the solutions

A plant is a complex structure with different combinations of many different organs. This various structured nature makes plant-modelling works more complicated than other modelling applications (i.e. single object modelling).

As mentioned in section 2.1.4, current rule-based modeling systems presents the interaction between plant growth and the environment [1], but are not geometrically accurate [3]. As each plant grows differently, there is no single rule valid to every plant.

3D reconstruction techniques promise an accurate geometry and topology model. However, based on our experiment (Fig. 13), low-resolution scans with wider views make 3D registration easy to convergence, but they do not provide details for individual component. While high-resolution scans offer great details, but their 3D registration hardly ends with successful convergence due to plant self-occlusion and high complexity. Occlusion, complexity and poor registration, make 3D reconstruction method incapable of capturing the useful geometric and topological information of a tomato plant. Further more, as an as-is model, 3D reconstruction offers no connection from plant growth to the environment. User interaction with the model is not allowed.

From the view point of 3D measurement, all current non-contact methods (see Section 4.2) only record the outline of the plant. For complex plants, it needs to revert to expensive internal imaging methods, like X-Ray CT-scans or MRI to capture a full 3D structure of a plant. Furthermore, although a 3D volume of the plant might be created automatically with point-cloud techniques, plant features still have to be extracted interactively since current software is not yet capable of fully automatic extraction of plant features in complex images.

Obviously, common rule-based method or solely 3D reconstruction method cannot fulfill our objective. In this research, we use a hierarchical structure to break tomato plants into weekly-growth-sections (WGS) to simplify the scan dataset and also relate age with plant growth geometry. We scan each WGS using 3D reconstruction concept, and exporting scan point cloud into CAD, constructing a parametric CAD plant model.

3D scans offer a digital map of a plant. A laser scanner is used for 3D measurement, which rapidly captures the geometry of the exposed surface in the form of dense, accurate point clouds [15]. With matured point cloud processing software (e.g. PolyWorks), point clouds are intelligently thinned. With advanced data extracting algorithms, critical measurements are obtained and exported to CAD system as reference to form a parametric 3D model for visualization and inspection.

A Components Library (CL) is used to capture the characteristic of elements of the plant variety. Based on measurements from scans, a mathematical plant growth model is constructed to define the geometry and topology of components and WGS. In addition, a model validation system is used to increase model accuracy and flexibility. The entire modeling system starts from plant decomposition and scanning. The CL is constructed from scan data. Each WGS is an assembly of components; and the plant is an assembly of WGSs.

3.2 System pipeline

The modeling pipeline is illustrated in Fig. 14. The sample plants data were measured and recorded during 2005 and 2006 at several Leamington, Ontario, Canada area greenhouses. The hardware used was a Minolta Vivid 910 Laser Scanner. The software used were InnovMetric's PolyWorks V9, SolidWorks 2004/2007, Visual Basic 6.0 and Matlab 7.0.

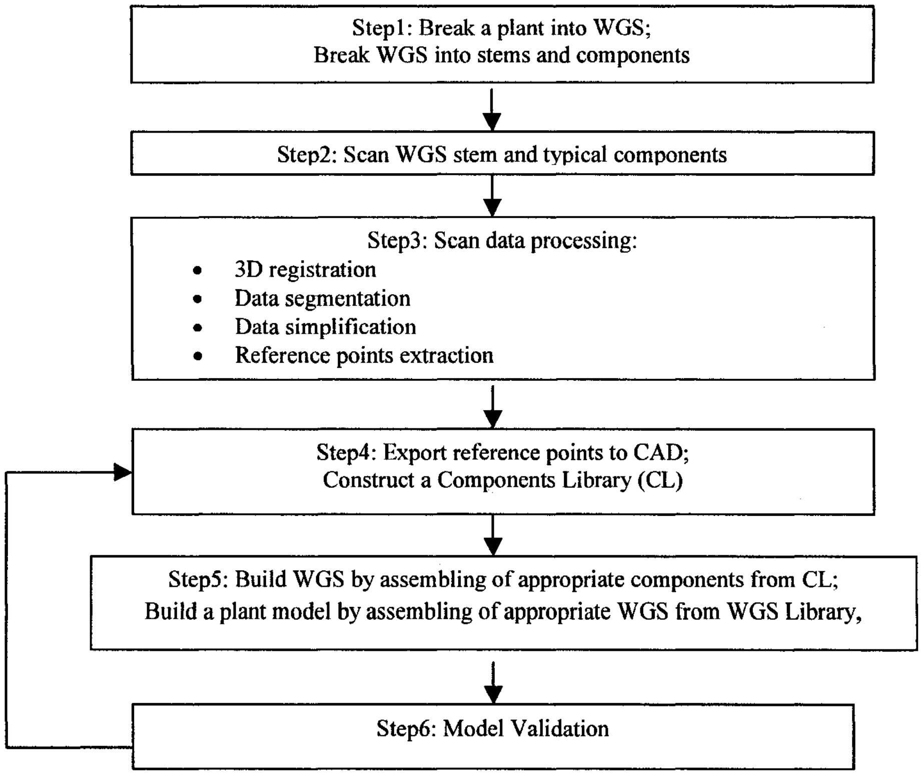

Step1: Break a plant into WGS;
Break WGS into stems and components
Step2: Scan WGS stem and typical components
Step3: Scan data processing:
3D registration
Data segmentation
Data simplification
Reference points extraction
Step4: Export reference points to CAD;
Construct a Components Library (CL)
Step5: Build WGS by assembling of appropriate components from CL;
Build a plant model by assembling of appropriate WGS from WGS Library,
Step6: Model Validation

Fig. 20: The pipeline of the proposed methodology

Step1 decomposes the plant into weekly growth sections using a plant mathematical model. For each WGS, the stems and associate components are measured and recorded. See section 3.3 for details.

Step2 takes measurements of WGS stems and selected components using a laser scanner. The advantages of using laser scanning technology for this application are demonstrated in section 3.4.1.

The purpose of Step3 is to optimize scan data efficiency while maintain the model accuracy. We use ICP algorithms (Besl & Mckay 1992) to align multiple scans into a common coordinate (3D registration), where, additional feature markers and user initialization are required, see section 2.2.3. To segment the object to be modeled (e.g. a fruit truss) from the others, combination of filters (e.g. distance, texture and color) and user interaction are used. Data simplification and reference point extraction are described in section 3.4.2 & 3.4.3 respectively..

Step4 constructs a components library (CL) using reference points that are exported into CAD from point cloud environment (e.g. PolyWorks). The procedures and a experiment are illustrated in section 4.4.1.

In Step5, with topology defined by the mathematical model and selected components, the WGS are accurately built. See section 4.4.2 for details.

In step6, Plant models are validated by comparing the model output against real plant measurements of attributes considered relevant by the agriculture industry. See section 3.3 for details explanation and section 4.5 for experiment results.

In addition, in the system, a model interface is developed to connect interesting geometry and topology information about plant components, WGS, and their assembly. It allows the user to navigate from tomato plant visualization model to detailed measurements and to access components or editing functions for quick model modification and update, see section 3.7.

3.3 Tomato plant growth mathematical model

To ensure the geometric and topologic accuracy, a mathematical model is developed to constrain the model as "parts/assemblies" in CAD environment. That is, the size, shape, position and connectivity of each plant components are defined.

Correct decomposition and attributes identification are the key issues of mathematic model building. In general, a plant mathematic model includes:

- Defining WGS: constrain a stem section with specified weekly growth rate, nodes information, and connections with respective components.
- Defining a Components Library (CL): constrain geometry and topology of each element forming a right component (i.e. leaflets forms a compound leaf).
- Defining a plant with respective WGS assemblies: constrain topology of WGS.

An effective method to monitor tomato plant growth is measuring weekly growth rate in Leamington greenhouse industry. Using this concept in our modeling system, tomato plants are decomposed into WGS. Each WGS have their characteristic, which can be captured and modeled. Based on the plant measurement, a mathematical model is derived to define attributes on the respective WGS as well as their variations. The next step is rebuilding a WGS model using the mathematic model with user-selected age as input. Components are considered as constant and taken from CL.

Advantages of using WGS can be summarised as:

- Relating plant geometry with age;
- Overcomes occlusion problem, maintaining high geometrical accuracy
- Simplify model-building process, offering manageable working scales.
- Simplify model validation, offering a comparable working conditions;
- Relating components to the main stem structure. Topology accomplishment;

With strategy of hierarchical structure and WGS, key attributes for a plant model can be defined as:

- Numbers of WGS for a given age
- For each WGS:
    - Height, diameters
    - Node info, Position, orientation, type
    - Components information (from CL), i.e. compound leaf, truss, suckers

The algorithm can be illustrated as:

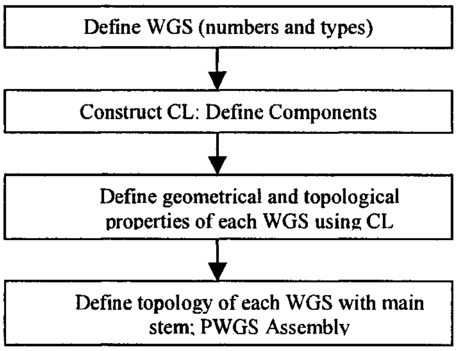

Define WGS (numbers and types)
Construct CL: Define Components
Define geometrical and topological properties of each WGS using CL
Define topology of each WGS with main stem: PWGS Assembly

Fig. 21: Constructing a plant mathematical model: the pipeline

3.4 Combining laser scan and CAD

3.4.1 Introduction

Modeling a free-form object (e.g. a leaflet) merely using interactive 3D CAD is time-consuming. A tomato plant, comprising hundreds of such objects, makes the situation even worse. Our solution is laser scanning. We use laser scans as a source for intelligent measurements to create entities in the 3D CAD.

To obtain complete 3D information, scans from multi-viewpoint are registered [25][26]. To build a high quality polygonal model using 3D reconstruction method, dataset registration requires a great number of local scans and extensive user interaction [40]. In contrast, in our system, only accurate control points from scan are vital; a watertight polygonal model is not required. This idea greatly reduce the number of field laser scans, simplifying the scanning work and allow scanning to focus on critical areas resulting in improved accuracy.

In general, surface models are generated in CAD. To achieve accurate parameters, two techniques are used:

- Creating a geometrical surface from selected control points in point environment, i.e.PolyWorks, and export them into CAD (i.e. SolidWorks) as a feature for CAD model construction.
- Directly export selected primitives (i.e. points or curves) to CAD and then, use surface construction functions to form a parametric surface model.

3.4.2 Point Cloud simplification

The purpose of point cloud simplification is to thin the original dense points, while maintain the model accuracy. That is to balance the efficiency of the data processing system and the quality of the output. Efficient simplification of the point cloud is a fundamental for reference point extraction.

Currently, several point cloud simplification algorithms are developed. Schroeder et.al proposed Decimation of triangle meshes method in 1992, in that, a vertex and the associate triangles are deleted if the specified decimation criteria are met. Model simplification using vertex clustering method was developed by Low et.al in 1997, where, the closeness of the vertices and determined and the vertex are grouped together based on the proximity. Eventually, they are replaced by a new representative vertex. The other algorithms include: Geometric optimization (Hinker et.al.1993), Voxel based object simplification (He et.al. 1995), Simplification Envelopes (Cohen et.al. 1996), Simplification using Quatric Error Metrics (Garland et.al. 1997).

Reducing or compressing the mesh is an operation where some triangles are removed from the mesh without changing the accuracy of the overall model. In PolyWorks, based on decimation of triangle meshes and vertex clustering methods, the user can adjust the density of polygons by setting meshing distance and surface sample steps [10]. During this operation, triangles are removed over flat areas and small triangles are kept in the highly curved or highly detailed areas.

In addition, as a part of data compression process, while importing a polygonal model, IMEdit automatically verifies its quality. If the following topological anomalies are found, the associated points will be removed.

- Degenerate triangle: 2 or 3 vertices of a single triangle have the same X,Y,Z coordinate;
- Duplicate triangle: 2 triangles use the same 3 vertices;
- Degenerate edges: more than 2 triangles share a common edge;
- Inconsistent edge: vertices are ordered in the opposite direction or 2 adjacent triangles have opposite normal directions (the front and back faces are flipped).

Table 3 demonstrates the data size of meshing from points to polygonal surface models yielded by different configurations in PolyWorks. As shown in Fig.22, the numbers of the points is significantly reduced when we use Decimation of triangle meshes algorithm.

Table 1: Dataset configuration of meshing a surface model of a young fruit truss

	(Original) High-Resolution	Decimation points alg'm 1	Decimation points alg'm 2
Max.Distance	4.0	10	20
Surface Sampling Step	0.68	2.0	5.0
Standard deviation	0.59	1.0	1.0
Numbers of Points	231,452	20,830	1,122
Numbers of triangles	24,050	6,301	657
Size in IGES format	43MB	2.95MB	364KB

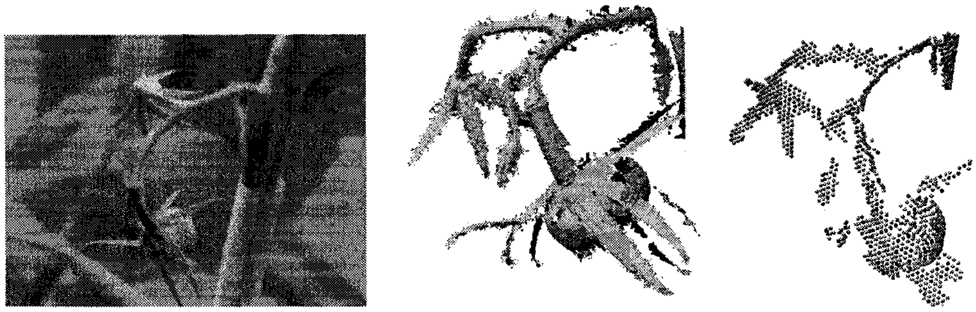


Fig. 22, Left: Photo of a fully opened flower truss with forming fruit of the sample tomato plant Middle: a High-Resolution scan of the plant, with 231,452 points. Right: a Low-Resolution scan., thinned point cloud to 1,122 points using Decimation point algorithm 2.

3.4.3 Reference point extraction using corner detection method

3.4.3.1 The concept

As mentioned early, the output of a laser scanner is an unorganized point cloud. Point simplification can reduce point density, but current algorithms have their limitations. With the cube experiment (see section 3.4.4), data compression was saturated at 2079 triangles. If further compression is attempted, the model will become distorted. However, for a simple cube, only 8 points is sufficient to define its structural property in space. To reduce data computation cost and increase system efficiency, beside data simplification, reference point extraction is also required.

Unfortunately, automatic extraction of 3D architectural objects directly from unorganized point cloud data is still under research. Currently, this process largely depends on user's interaction [16][39]. In PolyWorks, there are two methods available. One method is numerical generation [30], in that, point cloud are converted into .txt format. After numerical filtering (e.g set distance threshold), the extracted points are kept and exported back to PolyWorks for visualization and verification. The other method is manual selection in an interactive graphic window. In this research, we developed an image-based method for critical point extraction. That is using corner detection algorithms to extract anchor points in range image, and transform them into 3D point cloud environment.

Each pixel of a range image expresses the distance between a known reference frame and a visible point in the scene. Therefore, a range image reproduces the 3D structure of a scene, and is best thought of as a sampled surface. Range image can be represented in two basic forms. One is a list of 3D coordinates in a given reference frame, called xyz or point cloud, for which no specific order is required (i.e. a scan). The other is a matrix of depth values of points along the directions of an x, y image axis, called r_{ij} data, where, the pixel values is representing the depth, with organized x, y coordinates, i.e. I (x, y) = 20cm. Furthermore, scans can be interpreted as a combination of views of 2D images, that is: slices of 2D image with various range values, e.g. z = 1; z = 2, etc.

In a 3D scan, points are corresponding to the pixels in a 2D image, which is an oblique projection of the scan. For us, the coordinates of the pixels in image frame are of interest as they represent the location of the points.

Thus, given a point pair, in 3D scan, we have P (X, Y, Z); in 2D image frame, we have corresponding pixel coordinates I(c, r). If we define a pose of a 3D scan (i.e. view +x, or +y, or +z), and derive its oblique projection (a 2D image) then there is a relationship (a transformation) between P (X, Y, Z) and I (c, r). If we have the transformation, we can transform coordinates of the detected corners in 2D image frame to a 3D scan frame: I(c, r) → P (X, Y, Z). P is a collection of critical points we need for CAD.

3.4.3.2 Corresponding points transformation between 2D images and 3D scans

P (X,Y,Z) is a point in 3D scan coordinate. While, I (c,r) is a point in 2D image frame which is an oblique projection of the scan, with pixel coordinate (c, r).

If we have a transformation between I and P, then, P (X,Y,Z)=f (I(c, r)). To calculate the transformation, based on the nature of our application, we propose preconditions are:

1. There is no rotation between the two frames. And,

2. Images taken from scans must be oblique projection along +x view, +y view, or +z view. And,
3. At least 3 correspondence point pairs can be determined (for calculation of distance between adjacent pixels along horizontal axis and vertical axis, d_h and d_v, and set reference frame).

Let us picking a correspondence point pair which in image frame is at (c_0, r_0); while in the scan frame is at (X_0, Y_0, Z_0). If we have an image with oblique projection along +y view, an input point (c, r) in image reference frame becomes: c_r=c - c_0; r_r=r - r_0.
Thus, X= X_0 + $c_r * d_h$; Z=Y_0+$r_r * d_v$, where, Y is considered as a constant, and Z is the coordinate along horizontal axis.

3.4.3.3 The Harris Corner Detection Algorithm

Harris and Stephens improved upon Moravec's corner detector by considering the differential of the corner score with respect to direction directly, instead of using shifted patches [8][46]. It should be noted that this corner score is often referred to as autocorrelation, since the term is used in the paper in which this detector is described. However, the mathematics in the paper clearly indicate that the SSD is used [47].

Without loss of generality, we will assume a grayscale 2-dimensional image is used. Let this image be given by *I*. Consider taking an image patch over the area (*u*,*v*) and shifting it by (*x*,*y*). The SSD between these two patches, *S* is given by:

$$S = \sum_u \sum_v (I(u,v) - I(u-x, v-y))^2 \quad .$$

The Harris matrix (denotes *A*) is found by taking the second derivative of *S* around (*x*,*y*) = (0,0). *A* is given by:

$$A = \begin{bmatrix} \langle I_x^2 \rangle & \langle I_x I_y \rangle \\ \langle I_x I_y \rangle & \langle I_y^2 \rangle \end{bmatrix},$$

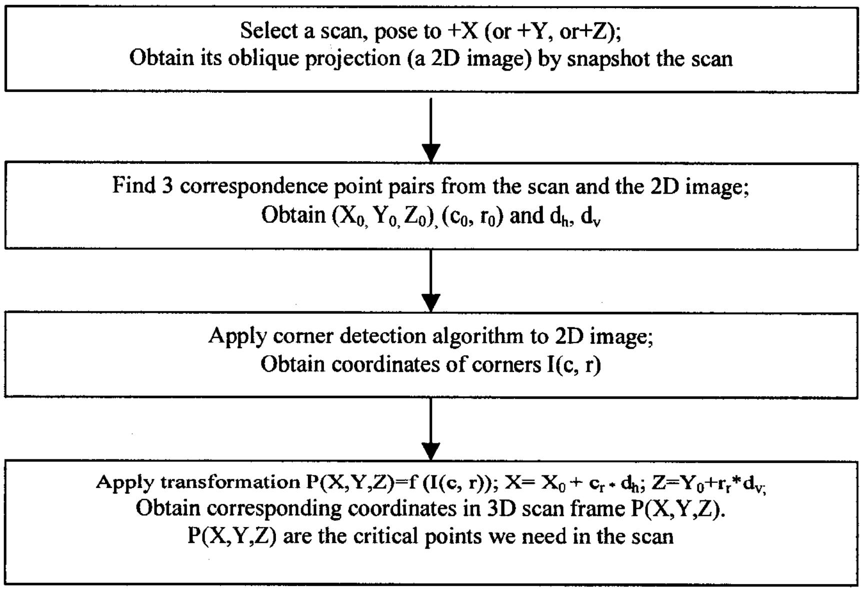
where angle brackets denote averaging (summation over (u,v)), and the typical notation for partial derivatives is used. If a circular window (or circularly weighted window, such as a Gaussian) is used, then the response will be isotropic [8][34][47].

The strength of the corner is determined by `how much' second derivative there is. This is done by considering the eigenvalues (λ_1 and λ_2) of A. Based on the magnitudes of the eigenvalues, the following inferences can be made based on this argument:

1. If $\lambda_1 \approx 0$and $\lambda_2 \approx 0$then there are no features of interest at this pixel (x,y).
2. If $\lambda_1 \approx 0$and λ_2 is some large positive values, then an edge is found.
3. If λ_1 and λ_2 are both large, distinct positive values, then a corner is found.

3.4.3.4 The data extraction algorithm and a experiment result

As illustrated in Fig.29, the input is a scan of an object in the unorganized, dense point cloud format; the output is a set of point P(X,Y,Z), which is able to describe the geometrical nature of the object. The main function blocks are: 2D corner detection and 2D-3D transformation.



Select a scan, pose to +X (or +Y, or+Z);
Obtain its oblique projection (a 2D image) by snapshot the scan

Find 3 correspondence point pairs from the scan and the 2D image;
Obtain $(X_0, Y_0, Z_0), (c_0, r_0)$ and d_h, d_v

Apply corner detection algorithm to 2D image;
Obtain coordinates of corners I(c, r)

Apply transformation P(X,Y,Z)=f (I(c, r)); $X= X_0 + c_r \cdot d_h$; $Z=Y_0+r_r*d_v$;
Obtain corresponding coordinates in 3D scan frame P(X,Y,Z).
P(X,Y,Z) are the critical points we need in the scan

Fig. 23: Extracting reference point using corner detection algorithm

The implementation of the algorithm can be illustrated by the following example (Fig.6). A WGS is scanned, and a 2D image is obtained from +X view of the scan. A transformation is established. After corner detection process in the 2D image, 44 corners were detected. Using the established transformation, 44 point in (X,Y,Z) format are derived. We convert these data into .txt format and export them into the original scan in PolyWorks. We find that all the points match the expected critical positions. Thus the accuracy of the algorithm, as well as the transformation has been proven. The code and output data are included in section 10.6.1.4.

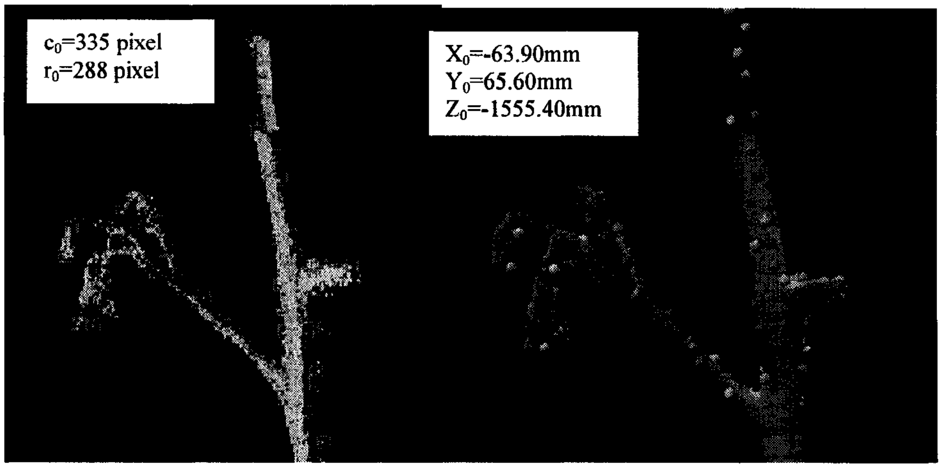


Fig. 24. Left: Modeling WGS#2. Applying Harris corner detection algorithm, Standard derivative, sigma=1; Threshold level, t=0.05; Neighbourhood size, Size=5. Total 44 corners detected. Right: Modeling WGS#2. Transform 44 corners into scan coordinate, and export the X,Y,Z into PolyWorks for visualization. d_h=0.7170mm, d_v=0.6810mm.

3.4.4 A experiment: Modeling a Box

To analyze data efficiency with different configurations at simplification and extraction stages, an experiment is conducted.

As shown in Fig. 31 and 32, a simple shaped box requires only 8 reference points to define in space, while a set of aligned scans may return us 607,452 triangles. Using data decimation algorithm may reduce the data size significantly, However, from 2D image processing knowledge, that corner detection method may offer us a perfect data efficiency level. The comparison results are summarised in Table 4 (note: for comparison purpose, the 8 points are picked manually at this stage).

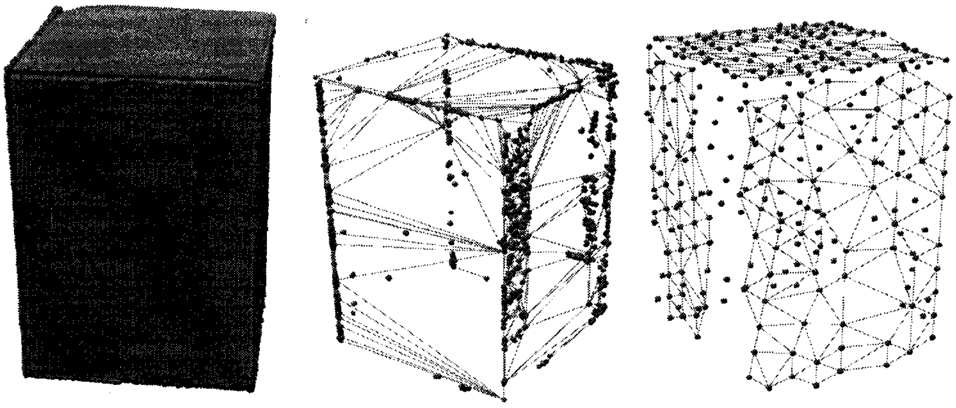


Fig. 25 (1). Left: a scan of a simple box, 6 scans aligned, total 607,452 triangles generated Middle: a simplified point view of the same box. Reduced mesh to 2079 triangles, using decimation algorithm, researching saturation status.

Right: a distorted model. If reduce triangles by increasing maximum distance and surface sampling step, we can get model with 407 triangles, but the shape is distorted.

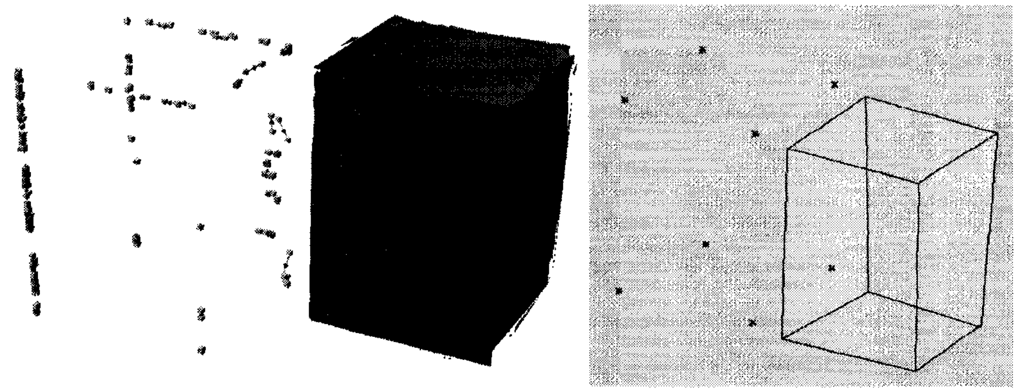


Fig. 25 (2). Left: View of reference pointed extracted using edge detection algorithm, 236 curves are derived. Right: 8 points to be extracted from scans and exported to CAD for a CAD box model construction.

Table 2: Experiment summary

Data Simplification					Data Extraction	
Max distance	2.0	Decimation of mesh: targeted triangle: 1000; Tolerance 5	Targeted triangle: 500; Tolerance 50	20	Edge detection	Corner detection
Surface Sampling Step	0.73	N/A	N/A	20	N/A	N/A
Standard Deviation	0.20	N/A	N/A	1	N/A	N/A
# Of Triangles/points	607,452 Triangles	2372 Triangles	2079 Triangles	407 Triangles	236 points	8 points
Data size in stl	29.6MB	116KB	102KB	20KB	-	-
In wrl	50.6MB	223KB	189KB	41KB	-	-
In dxf	131MB	511KB	448KB	89KB	-	-
In igs	150MB	1.1MB	1MB	197KB	205KB	4.72KB
Data efficiency (Minimum points / generated points	Very low 8/607452	Low 8/2372	Low 8/2079	Model distorted	Good 8/236	Perfect 8/8

3.5 Building a Parametric CAD model

3.5.1 Introduction

The capabilities of modern CAD systems relevant to plant modeling include [39]:

- wireframe geometry creation
- 3D parametric feature based modelling, solid modeling
- freeform surface modeling
- Ease of modification of design of model and the production of multiple versions
- Automatic generation of standard components of the design
- Output of design data directly to manufacturing facilities
- Output directly to a rapid prototyping or Rapid Manufacture Machine for industrial prototypes
- maintain libraries of parts and assemblies
- Programmable design studies and optimization

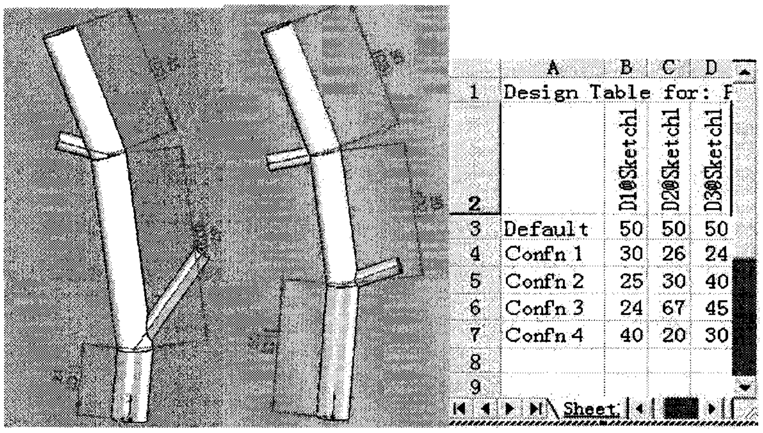
Many CAD programs use what is called ``wireframe" modeling, in either a 2D or 3D representation. In these programs, the operator uses lines, circles, arcs, and other similar entities to create an outline of the part. It is called wireframe modeling because it is analogous to building a physical model of the part using wires to represent the edges of the part. These models can be used for blueprints, engineering drawings, and other applications that require only pictorial information about a part.

Programs that are capable of solid modeling can be much more powerful than simple wireframe modelers. These programs are used to build parts that are actually solid objects instead of simply a wireframe outline of the part. Since these parts are represented as solids, they have volume, and if given a density can have a weight and mass as well. The computer can calculate many physical properties of these parts, such as center of gravity and moments of inertia. These calculations can even be performed for irregularly shaped parts, for which manual calculations would be extremely difficult. Finite Element Analysis techniques can also be used to perform stress analyses of these parts]

There are two basic methods used to create solid models. They are Constructive Solid Geometry (CSG) methods, and Boundary Representation (Brep) methods. CSG uses solid primitives (rectangular prisms, spheres, cylinders, cones, etc.) and boolean operations (unions, subtractions, intersections) to create the solid model. Brep methods start with one or more wireframe profiles, and create a solid model by extruding, sweeping, revolving or skinning these profiles. The boolean operations can also be used on the profiles themselves and the solids generated from these profiles. Solids can also be created by combining surfaces, which often have complex shapes, through a sewing operation. This can be used, for example, to create the body of an aerodynamic vehicle such as an airplane, with its carefully designed wing profiles. Further details on these two different methods can be found in Zeid [Zeid]. These two methods can often be combined in order to create the desired parts. Each of these methods has its limitations, and parts which are very difficult to create using just one or the other method can be created much more easily using a combination of both methods. Thus, most commercial solid modeling systems are hybrids using both CSG and Brep methods.

3.5.2 A Parametric CAD model

An important feature of modern 3D CAD tools is its ability to generate a parametric model, which allows quick model modification. In a parametric model, parameters control the various geometrical properties of the entity, as well as the locations of these entities within the model. Parameters can be changed by operators to regenerate desired entities or parts. Parametric model that use a history-based method keep a record of the model building procedures [39][44]. When operators change parameters in the model, the program repeats the operating procedures from the history with new parameters, thus parts are regenerated.



	A	B	C	D
1	Design Table for: F			
2		D1@Sketch1	D2@Sketch1	D3@Sketch1
3	Default	50	50	50
4	Confn 1	30	26	24
5	Confn 2	25	30	40
6	Confn 3	24	67	45
7	Confn 4	40	20	30
8				
9				

Fig. 26 Reconfigurating WGS using Excel. WGS are automatically regenerated according to different configurations. Left: Conf'n 3; Right: Conf'n 1.

The great advantage in using parametric modeling system does not rest solely in the development of mathematically correct models, but also in the ability to quickly edit or reconfigurate models and compare it with laser scan (from real world) to access the accuracy of the CAD model. In this paper, the advantage of using parametric CAD model can be summarised as:

- Relate model geometry with plant mathematical model,
- Offering high level of automation for model building, reconfiguration and regeneration (i.e. Excel can be inserted for easy data management, Fig.3),
- Offering a user-friendly interface. Users are able to direct the model output by selecting appropriate parameters.

3.6 Model validation

E.Heuvelink claimed in [1], that current poor validation methodology of plant models limited the application of plant models. Before a model can be used it must be validated, i.e. the model output has to be compared with the real plant, even though, the algorithm for input and output quantification is still under development so far. Finding appropriate measures for validation of 3D virtual plants with real plants in term of geometry and topology data is another area for future research.

In general, real plant measurements are divided into two independent parts: one for model development, the other for model calibration – control data [1]. Plant models can be validated by comparing the model output against the control data [3]. That is, the modelled and measured geometry and topology data are compared. The comparison results were verified mainly by human's experience and measurement.

In this research, general procedures of validation are:

- Define attributes to validate: components (e.g. position of fruit truss), WGS (e.g. position of nodes) and plant overall structure (e.g. stem length).
- Define the reference (control data), i.e. from grower's field measurement or scans.

Plant models are validated through a hybrid method. For structured entities, like geometry of main stems and nodes, we use traditional field measurement as reference comparing the model output [1]. For unstructured object, like plant components, we export the CAD model back to point cloud environment (PolyWorks) to verify with scan reference [27][30] or vice versa.

For example (see Section 4.5), to validate a forming fruit truss model in CL (T2), we take one truss with same age from greenhouse field, scan it as "scan reference" (Fig.58 Left) and verify them. Errors are detected and corrected using CAD parametric functions; then CL is updated. In this system, triggered by the feedback, updated components rebuild the assembly. Eventually, the whole plant model is updated. This feedback modeling system can be demonstrated as followings:

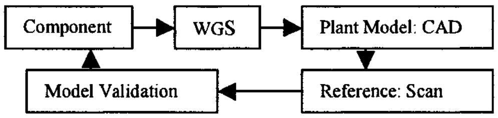
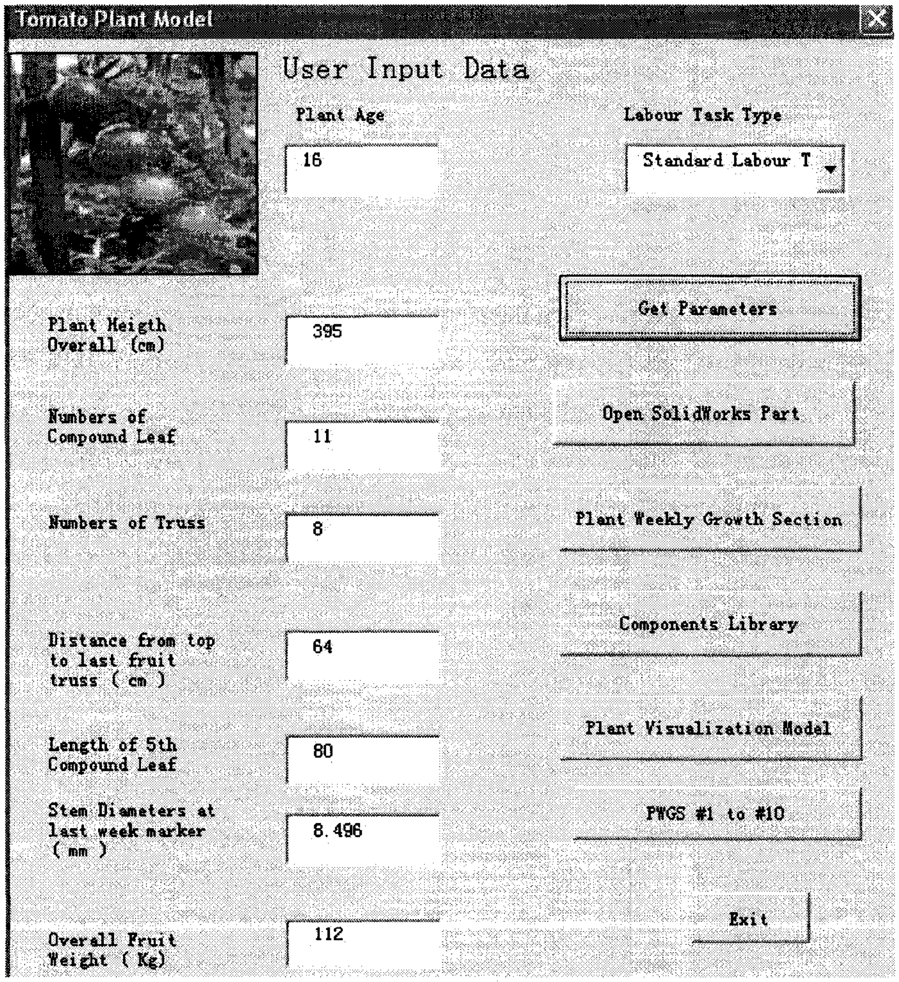


Fig. 27: A feedback model validation system

In addition, using this strategy, the system itself can be fine-tuned and "calibrated" to ensure the accuracy and flexibility to multiple applications.

3.7 System interface

The purpose of building a modeling interface is to provide an easy connection between interest geometry and topology information about plant models of components, WGS, and their assembly. With selected plant age, allows user navigating from tomato plant visualization model to detailed measurements and accessing components editing functions for quick model modification and update. As a meaningful future work, the interface could be carried out with SolidWorks API, using VB 6.0. The concept is illustrated as below (see Fig. 28 & 29).



User Input Data

Plant Age: 16

Labour Task Type: Standard Labour T

Plant Heigth Overall (cm): 395

Numbers of Compound Leaf: 11

Numbers of Truss: 8

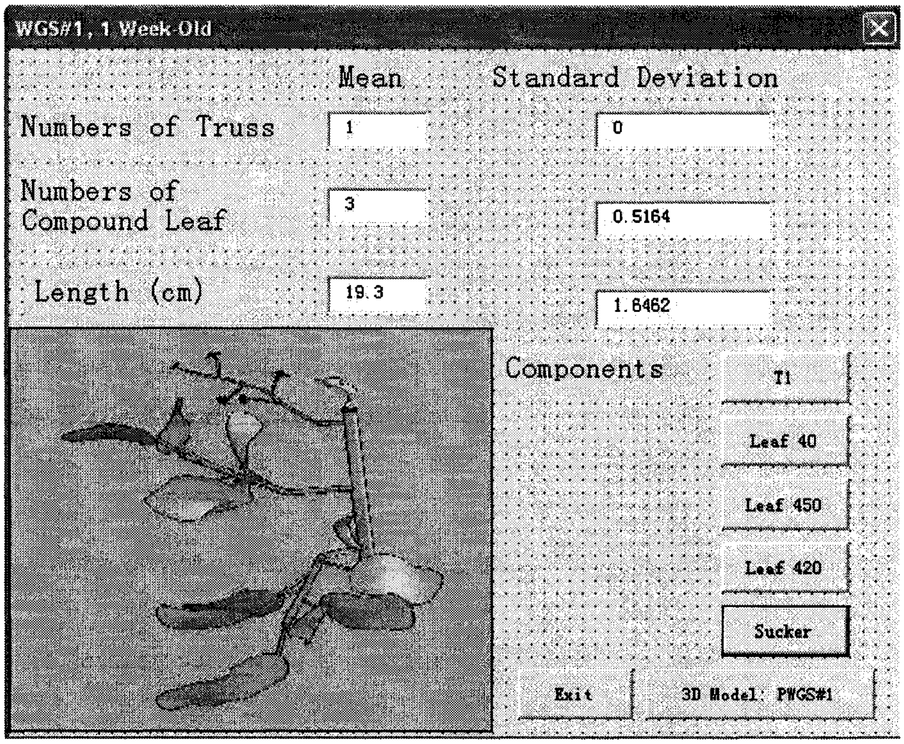
Distance from top to last fruit truss (cm): 64

Length of 5th Compound Leaf: 80

Stem Diameters at last week marker (mm): 8.496

Overall Fruit Weight (Kg): 112

Fig. 28: Main page of a plant modeling interface



WGS#1, 1 Week-Old

	Mean	Standard Deviation
Numbers of Truss	1	0
Numbers of Compound Leaf	3	0.5164
Length (cm)	19.3	1.6462

Components

Fig. 29: WGS#1 page of a plant modeling interface

4. Implementation and experiment results

4.1 Plant measurement using a Minolta Vivid 910 Laser scanner

In our experiment, 6 samples of 16 weeks old plants were taken from a greenhouse at Leamintong, Ontario, Canada, on 01 Nov. 2006. They were spring-fall (two-crop per year) tomato plants, transplanted at 6 July 2006 . The first measurement marker, Marker #1 was taken after 3 weeks of plant delivery. Hardware used is Minolta Vivid 910 Laser Scanner; while the softwares are InnovMetric's PolyWorks V9, SolidWorks 2004/2006, Visual Basic 6.0, and Matlab 7.0.

General procedures of plant scanning and data processing can be described below:

1. Performing laser scan for interested sections of a sample tomato plant, obtain 3D, color and texture data; relate the scanned sections with WGS.
2. Using PolyWorks IMAlign and IMMerge build components and WGS polygonal models of the plant;
3. Data segmentation, registration, simplification, and reference points extraction;
4. Export reference points to CAD;

4.1.1 The Hardware

The Minolta VIVID910 implements laser-beam light sectioning technology to scan workpieces using a slit beam. Light reflected from the workpiece is acquired by a CCD camera, and 3D data is then created by triangulation to determine distance information (see section 2.2.3.2). The laser beam is scanned using a high-precision galvanometric mirror, and each scan is capable of measuring 640 x 480 individual points. In addition to distance data, this 3D digitizer can also be used to acquire color image data. Employing a rotating filter to separate the acquired light, the VIVID910 can create color image data for 640 x 480 points with the same CCD as used for distance data [37]. The VIVID910 is a non-contact 3D digitizers that set new standards for low cost, camera sensitivity, and convenient operation. When operating in Fast mode, this advanced product requires only 0.3 seconds for data input (approximately twice as fast as the VIVID900); furthermore, when operating in Fine mode, precision of ± 0.008mm and accuracy of ± 0.10 mm can be achieved on the Z-axis.

Table 3: Principle specifications of Minolta Vivid 910 (see [37])

Model	VIVID 910 Non-Contact 3D Digitizer
Measurement method	Triangulation light block method
Incident lenses	Tele: focal distance f=25mm MIDDLE: Focal distance f=14mm WIDE: Focal distance f=8mm
Positioning range for image input	0.6~2.5m
Laser safety class	2 (IEC60825-1)
Laser scanning method	Galvanometric mirror
	111~463mm (TELE), 198~823mm (MIDDLE), 359~1196mm (WIDE)
Y-direction input range	83~347mm (TELE), 148~618mm (MIDDLE), 269~897mm (WIDE
Z-direction input range	40~500mm (TELE), 70~800mm (MIDDLE), 110~750mm (WIDE/FINE mode)
Accuracy	Z: ± 0.10 mm with respect to standard Z plane (Fine mode, Konica Minolta's standard)
Input time	0.3s (Fast mode), 2.5s (Fine mode), or 0.5s (Color mode)
Transfer time to host computer	Approximately 1s (Fast mode) or 1.5s (Fine mode)

Ambient lighting condition	500 lx or less
Number of Output pixels	3D data: 640 x 480 (Fine mode) or 320 x 240 (Fast mode) Color data: 640 x 480
Output format	3D data: Konica Minolta format (converted into 3D data by utility software provided as a standard accessory) Color data: RGB, 24-bit raster scan data
Data file size	Total 3D data and color data: 1.6 MB (Fast mode) or 3.6 MB (Fine mode) per data
Output interface	SCSI II (DMA-compatible simultaneous transmission)
Power supply	Commercial 100V or 240V AC supply (50 or 60Hz), rated current 0.6A (at 100V)

4.1.2 Scanning procedures

Plant surface treatment prior to scanning

Surface may not produce sufficient laser contrast for the scanner to detect a profile because of shininess or by the colour of a surface: a green surface will absorb light of any colour but green and laser light is relatively pure in colour, not enough of a red laser beam may be reflected from the object and received by the scanner.

Solution:

- □ Change from a red to a green laser
- □ Treat surface to change its reflective properties: i.e. apply chalk powers
- □ Increase the laser output power

Preparing feature markers for registration

The similarity nature, make plant organs looks same. However, they are all different in geometry and topology. These coursed difficulty in finding reference point for registration. Since so far, fully automatic registration is not available, "Registration still rely on user-interaction to determine the initial transformation, making the pre-registration a tedious and time-consuming task" [26], see Section 6.3. Thus, additional feature markers are important for successful scans registration.

The followings preparations are used in our scanning setup:

- Having at least 4 distinguishable markers as high as plant; making sure at least 3 poles can be "seen" from every single scan. (That is similar to FARO system, 3 spheres are required).
- Taking global scan from at least 4 directions with reference markers covered.
- Taking local scan with high resolution, with at least 2 reference markers covered.

Detailed scanning plan of a section of a plant

- scan bottom stem with high resolution, 6 directions;
- scan bottom mature leaves with stick, at least 3 sets;
- scan top portion, 6 directions;
- scan top young leaves with stick, at least 3 sets;
- scan up mid-mature leaves with stick, at least 3 sets;
- scan bud and flowers with close look;
- scan young fruits with vine;
- scan mature fruits with vine;
- remove all leaves, scan stem and branches, 6 directions;

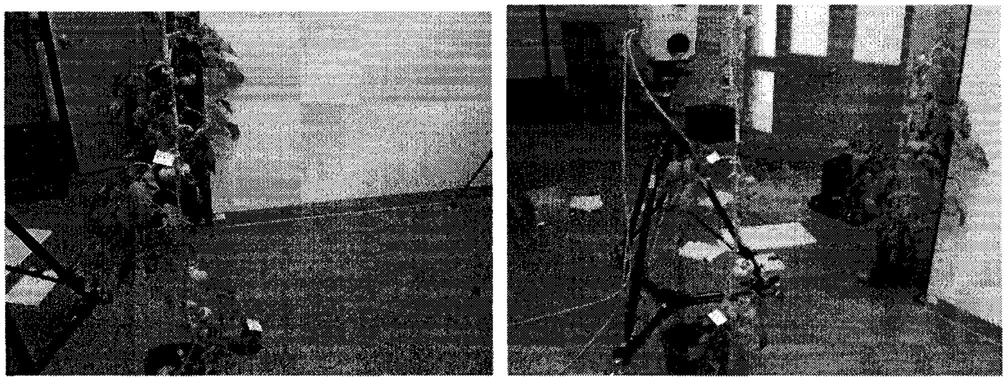


Fig. 30: Left Scanning the plant sample#1 in IMTI NRC, Nov.2006
Right: Decomposing plant sample#1into WGS for sectional scanning

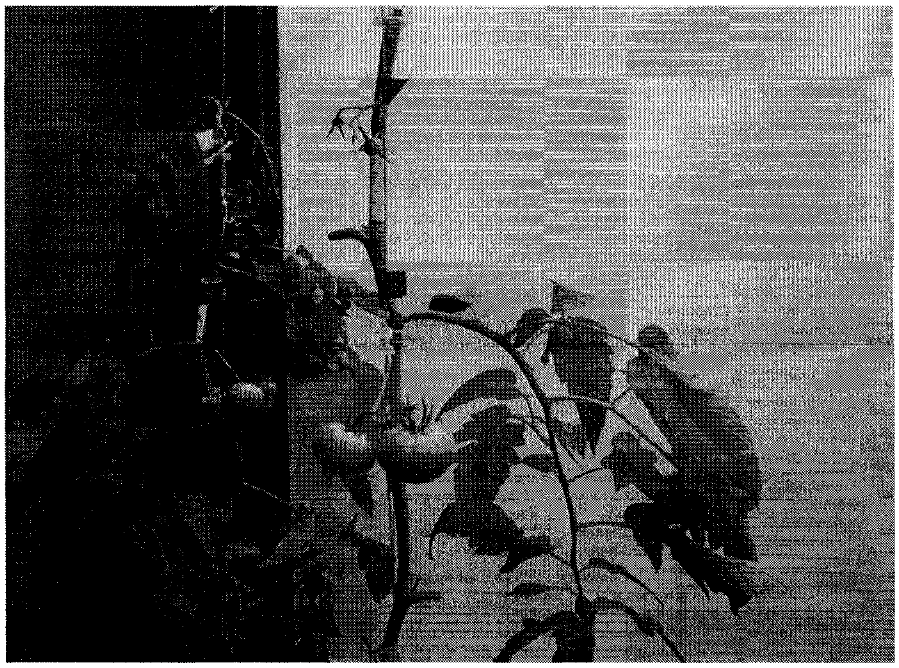


Fig. 31: Scanning a WGS of plant sample #1

Fig. 32: Scanning the 1-week-old leaf (Leaf 40 in CL) on top position of plant sample#1

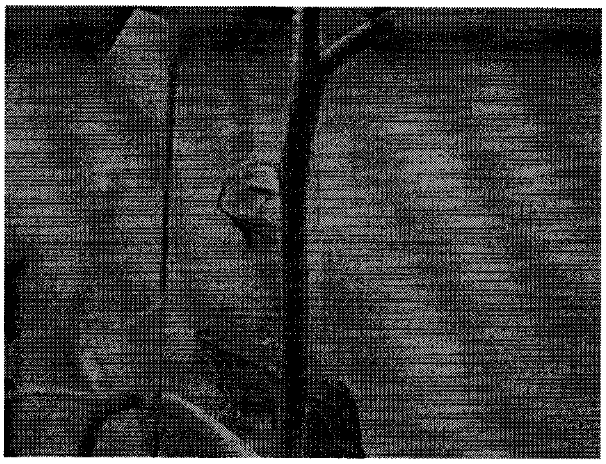


Fig. 33: Scanning a flower truss of plant sample#1

Fig. 34: Top: scanning a sucker of plant sample#1

4.2 Scan data processing using PolyWorks

4.2.1 Introduction of PolyWorks

PolyWorks is a comprehensive software solution for point cloud data processing and creating accurate and smooth polygonal models and NURBS surfaces. Under PolyWorks Workplace Manager (WM) window, among the other features, there are 4 sections assembling the full process of model building.

IMAlign – Data acquisition

Acquiring data is the first item in the PolyWorks workflow. It consists of two steps. Scanning and scans alignment.

IMAlign allow user-applying plug-ins interfacing with scanner. Once connected, the scanner is remotely controlled by TMAlign. Firstly, the scan range is adjusted either by auto or by manual. A distance-colored preview with scan range map is presented for user decides the optimal range setting. Scanner must be calibrated whenever, the object position or the scanner's position change. User can select different angles for step-scan; the less the angle the more scans will be produced. Once scans are captured, user defined feature points in the scans can be used for scan registration.

PolyWorks answers the scan alignment needs with its IMAlign module. Situate all of the scans in the same coordinate system, based on the shapes of the scanned object. For the operation to succeed, the scans must meet two criteria: each scan must share some redundant information with adjacent scans, and each scan should contain at least one change of shape (e.g., corner, hole, angle) of the object, since IMAlign does a shape-based alignment (changes in shape help lock the scans).

IMMerge - From point cloud to surface model

An important part of 3D reconstruction is a process that involves converting the aligned scans into a high-quality, highly accurate surface represented by polygons. The model building process consists of two steps.

Table 4: Steps of polygonal model building using reverse engineering concept

Steps	
1	Generate a Polygonal model Using IMMerge to create a polygonal model from the points in the aligned datasets. IMMerge offers smoothing reduction that reduces the number of triangles in planar regions while respecting object curvature.
2	Improve the model It is normal that a model made from digitized dtasets has some imperfections. Use IMEdit to improve the model by filling holes, extracting sharp edges, optimizing triangles, performing local reductions.

PolyWorks' IMMerge module is employed to create an accurate polygonal model from an IMAlign project, which contains scans aligned using the IMAlign module. The polygonal mesh is adapted to the object's curvature, and it is possible to control the triangle size, smooth the input data by removing digitizer noise, and significantly reduce the model size while respecting the object's shape. The IMMerge module is composed of a main parameter window, a 3D scene, and a text output area. The 3D scene shows the evolution of the meshing process and the text output area gives precise information as to the processing currently being performed. The 3D scene can be transformed using standard, mouse-based operations.

In the following experiment (Fig. 35), 4 scans merged surface. Meshing parameters: smooth level: low, maximum distance: 4.0 mm, surface sampling step: 0.26, standard deviation: 0.11, number of model triangle: 85,426, number of displayed triangles: 85,426.

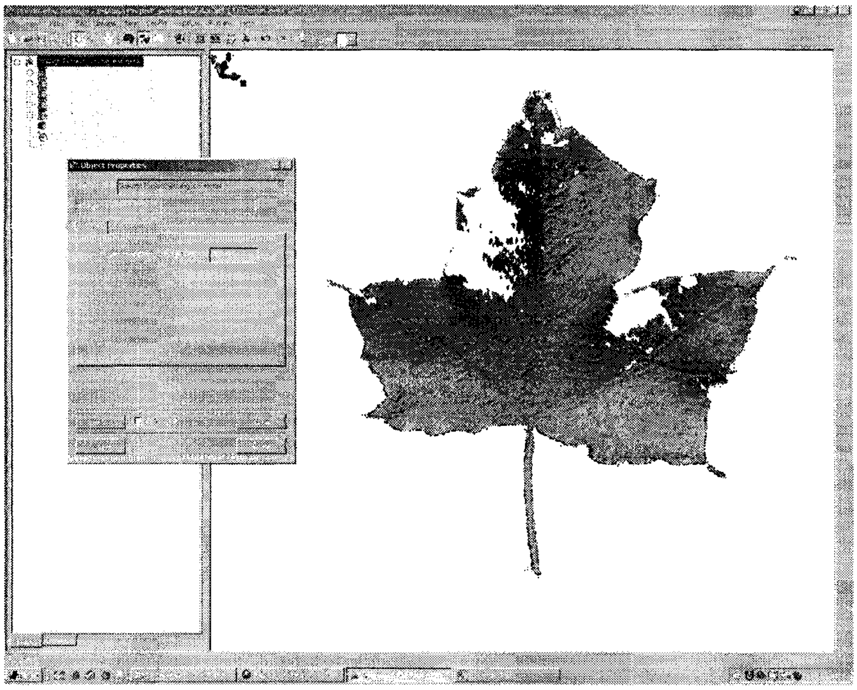


Figure 35: Using IMMerge, Leaf polygonal model

IMEdit - Improve the polygonal model

The IMEdit module is InnovMetric's toolbox for preparing your polygonal model for the applications. You use composite Bézier curves and surfaces to enhance the model, Point and Plane primitives for a number of functions, and cross-sections to export the object's shape. Curve networks can be created, and NURBS surfaces fitted to them and then exported to IGES. IMEdit tools allow you:

- analyze topology watertightness
- reduce the mesh (selection)
- extrude boundaries
- subdivide the mesh
- fill holes
- smooth the mesh
- optimize the mesh
- trim the model
- reconstruct edges, corners, and fillets
- slice the model (dowels), and more

In IMEdit, you will import a polygonal model and edit it. The import operation consists of making a copy of the model and storing it in an internal PolyWorks format. Once the editing is completed, the improved polygonal model may be exported to a number of standard formats.

Once the project is finished, the polygonal model (or other objects) must be exported to a specified standard format for use in another software. The export operations are found in the File menu. The following formats are supported in PolyWorks for data transfer.

- Autodesk files (*.dxf)
- Iges files (*.igs)
- Invertor files (*.iv)
- Nastran files (*.nas)
- Wavefront files (*.obj)
- PLY files (*.ply)
- InnovMetric files (*.pol)

- STL files (*.stl)
- ASCII STL files (*stla)
- Bianry STL files (*stlb)
- VRML files (*wrl)

It allows user doing hole-filling, surface smoothing, feature creating and measurement.

IMInspect – Model inspection

Load the IMAlign project as the data object (digitized object). Load reference object (theoretical model). Align the data to the reference (Fig. xx). Then perform global comparison using color maps, generate cross-sections, and take measurements.

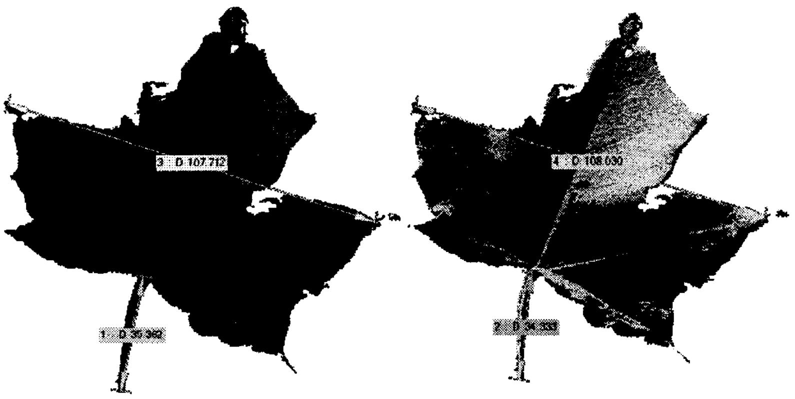


Figure 36: Polygonal surface models of maple leaves. Scan taken by Tong Wang, 24 Aug. 2006, using IMInspect tools to compare a reference leaf (left) and modeled leaf (right).

4.2.2 Data simplification

On importing a polygonal model, IMEdit automatically verifies its quality. If topological anomalies are found, they must be corrected before the software allows further work on the model. A correction wizard pops up. It informs you that IMEdit will delete some triangles to remove anomalies and asks for your approval. IMEdit detects the following four kinds of topological anomalies:

- Degenerate triangle: 2 or 3 vertices of a single triangle have the same X,Y,Z coordinate;
- Duplicate triangle: 2 triangles use the same 3 vertices;
- Degenerate edges: more than 2 triangles share a common edge;
- Inconsistent edge: vertices are ordered in the opposite direction or 2 adjacent triangles have opposite normal directions (the front and back faces are flipped).

Reducing or compressing the mesh is an operation where some triangles are removed from the mesh while maintaining the required accuracy of the overall model. During this operation, triangles are removed over flat areas and small triangles are kept in the highly curved or highly detailed areas (Fig.37). In IMEdit, you can reduce part or all of the polygonal mesh.

Reduction is normally done once the editing work is finished. The reduction window offers you four options for specifying a reduction level:

- Specify a target number of triangles in the Number of triangles text box.
- Specify a reduction percentage on the number of triangles in the Reduction percentage spin box. A 0% reduction percentage preserves the current triangulation, while a 100% percentage guarantees maximum reduction of the selected area.
- Specify a relative 3D tolerance in the Relative tolerance text box, as a percentage of the largest side of the model's bounding box.
- Specify a 3D tolerance in model units in the Tolerance text box. This is a maximum absolute distance in model units that a surface area can be lowered by removing a vertex and retriangulating.

In addition, the window provides a Max edge length combo box that allows you to set a maximum triangle edge length, which is set to Not constrained by default. To specify a maximum edge length, enter a value in the combo box.

Finally, the window provides a Max dihedral angle text box for detecting sharp edges and corners prior to compression. Sharp edges and corners will then be preserved by the polygon reduction algorithm. Edges are reduced along their length using the same tolerance applied to the surface triangulation reduction. Reduction is performed by pressing the Apply button. It should be noted that IMEdit preserves the boundary of the selected area throughout the reduction process.

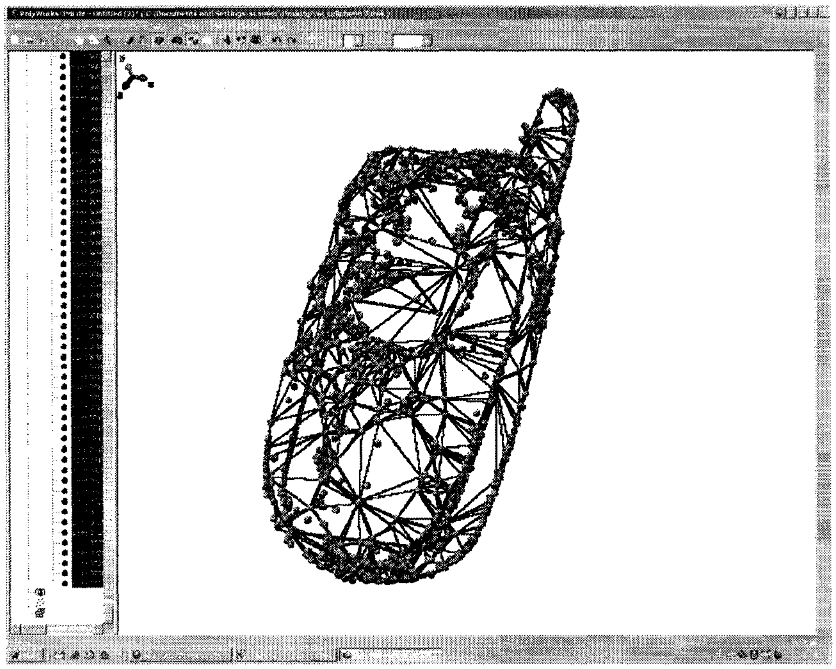


Fig. 37: An example of dataset simplification: triangles are removed over flat areas and small triangles are kept in the highly curved or highly detailed areas

4.2.3 Reference point extraction

In this research, with help of PolyWorks, there are 3 basic methods used for reference point extraction.

Cross-section

In this case, the level of detail is important. As examples, a plant fruit truss (Fig. 38 & 39), and leaflets (Fig. 44) are free-form objects whose geometry in space can not be generated unless references are given. In Fig. 38, We use cross-section functions in IMMedit, to derive the diameters of the fruit truss as well as the shape and positional data from scans.

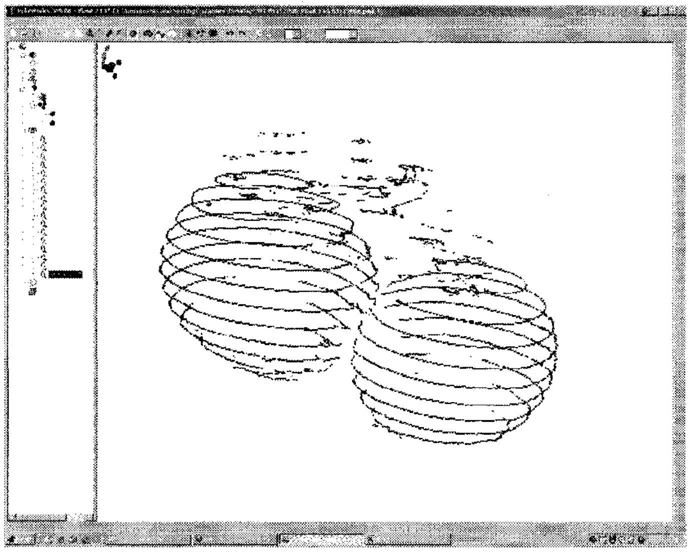


Fig. 38: Extracting reference data (e.g. diameters and shape) from cross section

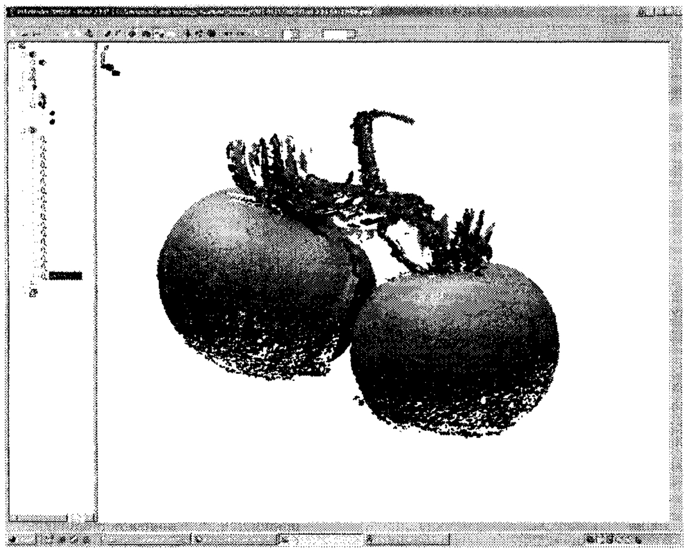


Fig. 39: The original scan

Manually picking

Manually select critical points can be done using interactive graphic windows in IMEdit [30]. As shown below (Fig. 40), the main structural data, used to define the shape of WGS, are derived from manual picking. In this case, level of detail is not essential, only the main stem geometry and nodes topology data are important. So that, 30 points per meter long for the main stem is sufficient to describe the plant shape in space. We can manually select critical points and export them using IGES format to CAD for parametric model generation.

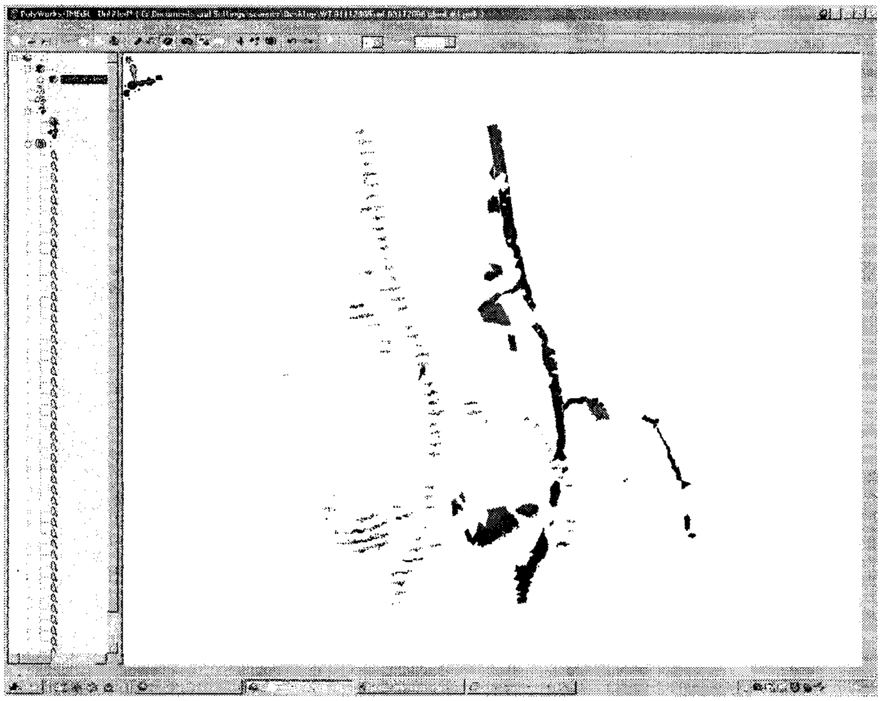


Fig. 40: Reference point extraction by manual picking

Corner detection method

To increase the accuracy and the level of automation of reference point extraction, we proposed using corner detection method to detect the critical points. The concept, the algorithm and the experiment are introduced in Section 3.4.3. The Matlab code and experiment results are listed in Appendix 1.

4.3 Building a plant mathematical model

4.3.1 Plant decomposition

As stated in section 3.3, a plant mathematical model is developed to define the geometry and topology of plant components, WGS, and the plant structure as a whole. In our experiment, the 16 weeks old plant sample#1 was decomposed into 16 WGSs. The structure of the decomposed plant are shown below (Fig. 41).

Overall Plant height (from top to bottom): 392 cm

Distance from marker #1 to rockwool is: 75 cm.

Distance from top to first fruit is: 140 cm.

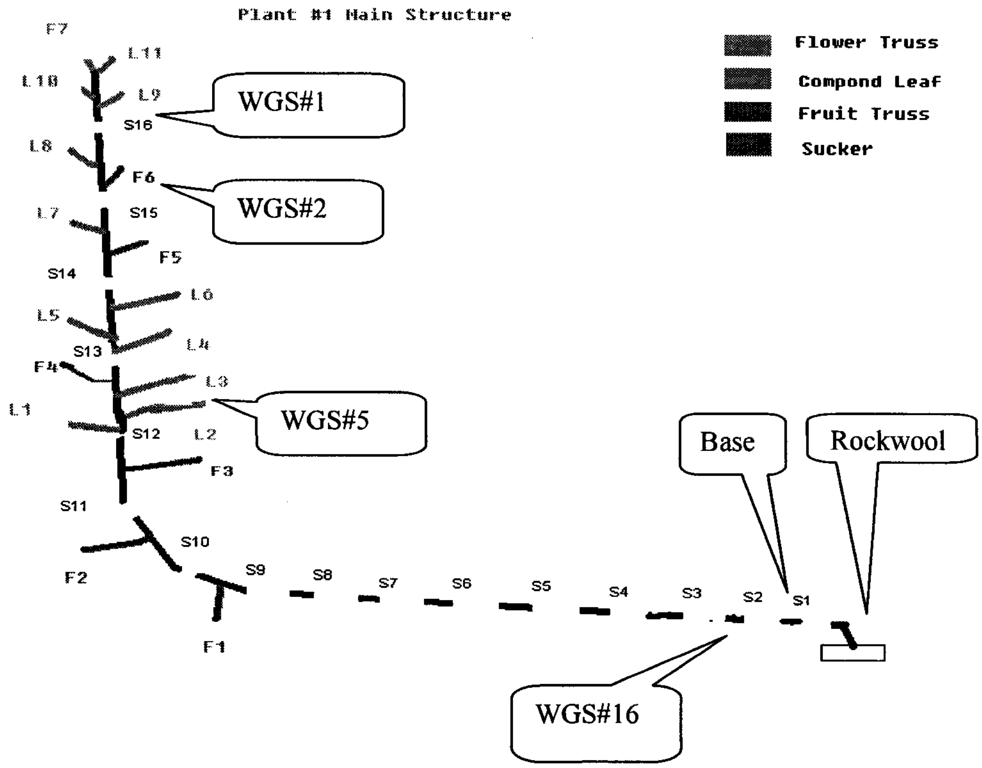


Fig.41: A 16-WGS representing a 16 week-old plant.
F: Fruit or flower truss; L: Compound Leaf

4.3.2 Tomato plant measurement

Tomato plant manual measurement includes geometry and topology measurement for each structured entities of components, WGS, and the plant structural as a whole. The following measurements are obtained from the 6 sample plants mentioned at the beginning of Chapter 4.

Stem Weekly Growth Rate (cm)

Table 5: Plant weekly growth rate (cm)

WGS	1	2	3	4	5	6	7	8
Sample #1	18	18.5	19.00	17.5	20.0	21.0	21.5	22.0
Sample #2	18.5	21.5	18.00	20.5	19.0	21.0	22.5	19.0
Sample #3	20.5	18.5	19.00	19.5	22.0	22.0	19.5	21.0
Sample #4	18	18.5	19.00	18.5	20.0	21.0	21.5	22.0
Sample #5	21	20.5	20.00	19.5	18.0	18.0	19.5	20.0
Sample #6	19	21.5	19.00	20.5	21.0	19.0	20.5	21.0
Mean	19.3125	19.6250	19.2500	19.6250	20.000	20.3333	20.8333	20.9871
Standard Derivative	1.6462	1.3562	0.8864	1.2464	1.4142	1.5055	1.2111	1.1690
Probability [16 ~ 22]	0.9266	0.9563	0.9989	0.9698	0.9190	0.8639	0.8323	0.8408

Week	9	10	11	12	13	14	15	16
Sample #1	18	18.5	19.00	17.5	22.0	21.0	21.5	22.0
Sample #2	18.5	21.5	18.00	20.5	19.0	21.0	22.5	19.0
Sample #3	20.5	18.5	19.00	19.5	22.0	22.0	19.5	21.0
Sample #4	18	18.5	19.00	18.5	20.0	21.0	21.5	22.0
Sample #5	21	20.5	20.00	19.5	18.0	18.0	19.5	20.0
Sample #6	19	21.5	19.00	20.5	21.0	19.0	20.5	21.0
Mean	19.1667	19.8333	19.000	19.3333	20.3333	20.3333	20.8333	20.8333
Standard Derivative	1.2910	1.5055	0.6325	1.1690	1.6330	1.5055	1.2111	1.1690
Probability [16 ~ 22]	0.9788	0.9195	1	0.9866	0.8423	0.8639	0.8323	0.8408

Numbers of Leafs Per WGS

Table 6: Numbers of compound leaves per WGS

WGS	1	2	3	4	5	6	7	8
Sample #1	3	1	1	3	3	0	0	0
Sample #2	3	1	1	3	2	1	0	0
Sample #3	2	1	1	2	3	0	0	0

Sample #4	2	1	2	1	3	0	0	0
Sample #5	3	2	1	2	3	0	0	0
Sample #6	3	1	1	3	2	1	0	0
Mean	3	1	1	3	3	0.3333	0	0
Standard Derivative	0.5164	0.5477	0.5477	0.8165	0.5477	0.5164	0	0
Probability [1 - 3]	0.8775	0.8163	0.8163	0.7417	0.8163	-	-	-

Numbers of Truss Per WGS

Table 7: Numbers of truss per WGS

WGS	1	2	3	4	5	6	7	8
Sample #1	1	1	1	0	1	1	1	1
Sample #2	1	1	0	1	1	0	1	1
Sample #3	1	1	1	0	1	1	1	1
Sample #4	1	1	1	0	1	0	1	1
Sample #5	1	1	0	1	1	0	0	1
Sample #6	1	1	1	0	1	1	1	1
Mean	1	1	1	0	1	1	1	1

Standard Derivative	0	0	0.5164	0.5477	0	0.5142	0.8623	0
Probability [0-2]	-	-	0.8967	0.8163	-	0.8543	0.8975	-

Due to deleafing operations, there is no fruit truss on WGS#9 and backward to the rockwool.

Stem Diameters (mm) Vs age (week)

Measurement taken at the location of 5th leaf from top of the plant

Table 8: Stem diameters Vs. age

WGS	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
Stem Diameter (mm)	8	8.5	8.8	9.3	9.5	10.5	11.3	12.18	12.6	12.83	11.80	11.68	11.70	11.57	10.96	11.89

Plant components measurement

As shown in Fig. 42 & 43, plant components are measured to build the CL. The measurement results are listed in Table 15.

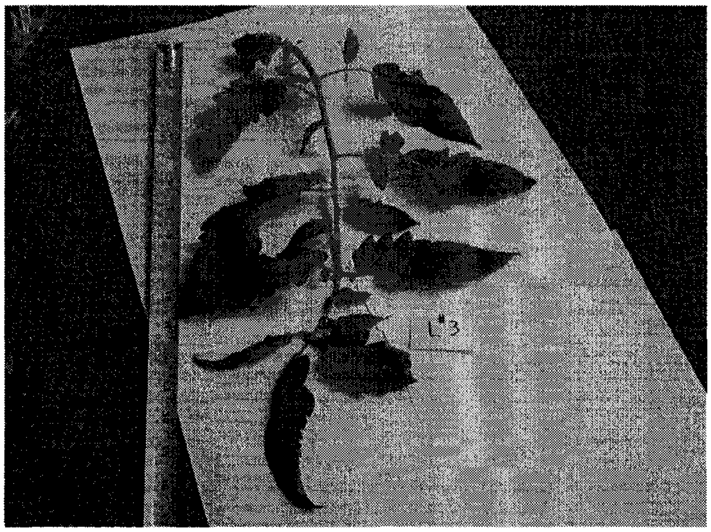


Fig. 42: measuring a compound leaf

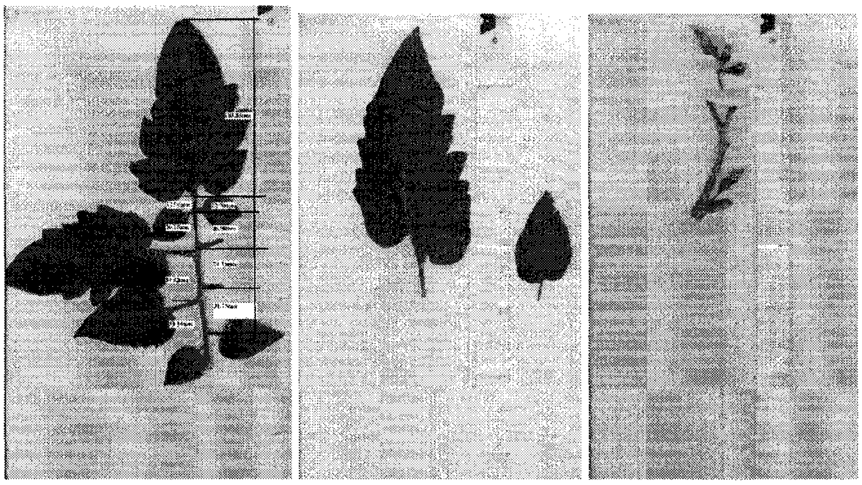


Fig. 43: measuring leaflets

Components measurement on Sample#1:

Table 9: Components measurement on Sample#1

Component	Age	Section	Position (cm) down to next component	Orientation Degree to Front	Numbers of Leaflets	Length (cm)	Diameter (cm)	Weight (grams)
		3						
F1-1	8	9	96 to marker #1	270			8.2	245.3
F1-2	8	9	96 to marker #1	270			9.1	286.3
F2-1	7	10	26 to F1	300			7.1	152.95
F2-2	7	10					7.2	170.91
F3-1	6	11	25 to F2	310			7.1	168.29
F3-2	6	11	25 to F2	310			4.4	36.2
F4-1	5	12	6 to L3	270			6.1	90.58
F4-2	5	12	6 to L3	270			6.2	86.43
F4-3	5	12	6 to L3	270			3.5	23.66
F4-4	5	12	6 to L3	270			3	23.66
F5	3	13	11 to L6	190			young fruit	
F6	2	15	7 to L7	200			fully open flower truss	
F7	1	16	connect to L11				forming flower truss	
L1	5	12	6 to F3	210	17	42		
L2	5	12	7 to L2	100	15	45		
L3	5	12	6 to L2	270	12	40		
L4	4	13	4 to F4	90	11	46		
L5	4	13	8 to L4	0	15	49		
L6	4	13	7 to L5	100	15	46		12
L7	3	14	6 to F5	290	12	44		
L8	2	15	4.5 to F6	340	9	28		
L9	1	16	12 to L8	340	12	39		
L10	1	16	6.5 to L9	120	11	34		
L11	1	16						

4.3.3 Observations

The flowing observations were made based on measurements shown section 11.2, and discussions with greenhouse vegetable specialist, as well as greenhouse field studies.

- With standard labour task implied, numbers of compound leaf are normal distributed, central mean: 11 for spring-fall crop plant. 2 or 3 for each WGS. So, we have 5 or 6 WGSs with compound leaves.

- With standard labour task implied, weekly growth rate are normally distributed. Central means for different WGSs are shown at table 11 for spring-fall crop plant.

- One truss per week developed on one WGS from WGS#1 to #8.

- Flower truss need 1 to 2 weeks to mature. So newly forming flower truss is on WGS#1. A fully opened flower truss is on the 2nd week section, WGS#2.

- Because fruit setting start from 3rd week flower truss, and getting mature 6 weeks later. So a fully mature fruit is on an 8 weeks old truss, WGS#8 (if not picked); and a forming fruit truss is just on a 3 weeks old truss, WGS#3.

- For spring-fall crop, the growth cycle is 8 week. That is to say, the geometry of WGS#1 to WGS#8 will be remained as constant over the whole season. Because of deleafing and fruit picking operations, only top 8 sections, WGS#1 to #8, have components with them. There is no component on WGS #9 and backwards. For example, a 20 weeks old plant will have WGS#1 to WGS#8 with components on, and no components on WGS#9 to WGS#20.

4.3.4 Mathematically define plant WGSs

As stated in section 5.2, the Plant WGS data can be defined by the measurement data as follows.

Table 10: WGS data for plant sample #1

WGS	Age (Weeks)	Components		Section
		Leaf	Truss	Length (cm)
1	1	L9, L10, L11	F7, Sucker	18
2	2	L8	F6	18.5
3	3	L7	F5	19
4	4	L4,L5, L6		17.5
5	5	L3, L2, L1	F4	20
6	6		F3	21
7	7		F2	21.5
8	8		F1	22
9	9			18
10	10			18.5
11	11			19
12	12			17.5
13	13			22
14	14			21
15	15			21.5
16	16			22
Base	As Delivered			75

4.4 A Laser-scanning based CAD parametric plant model

4.4.1 Constructing a plant components library

In the CL, typical trusses and compound leaves are constructed in CAD. Trusses are identified by ages (in weeks), and compound leaves are identified by length (in cm). Components in the CL can be utilized with respective WGS and with respective connections. For example, a 2 weeks old WGS requires 1 set of 2 week-old fruit truss, 1 set of 200mm Leaf with specified connectivity (see example WGS data below).

Within the CL, a component is made up of various elements. The geometry of each element (e.g. leaflet) is measured from scans and their roles in the topological structures are recorded (see Fig. 44). The geometry and topology of suckers, fully opened flowers, fruit clusters, nodes and compound leafs are obtained from laser scans in the form of reference points which are exported to CAD as anchors for model construction.

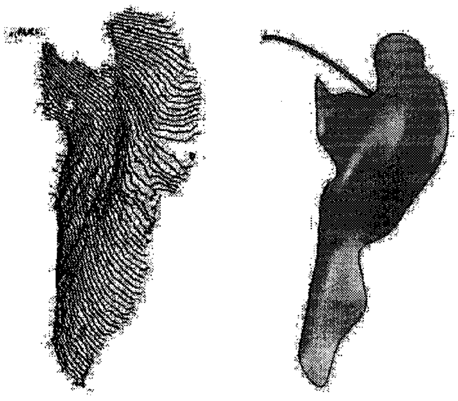


Fig. 44. Left: Surface measurements from 3D scan: Curves captured from PloyWorks. Rigth: Reconstructed leaflet in SolidWorks.

Compound Leaf

Similar to Long Quan's plant modeling method [4], we consider leaves in the same plant are typically very similar. We extract a generic leaf model with certain length from a sample leaf and use it to fit all the other leaves with the same length. This strategy turns out to be more robust as it reduces uncertainty due to noise and occlusion by constraining the shapes of leaves [4]. In our experiment, the measurement of sample#1 plant are listed in Table 17. The corresponding CAD models are shown in Appendix 3.

Table 11: Compound leaf data for plant sample #1

C L ID		Leaf Length cm	Leaflets		Max. Leaflet length		Min. Leaflet length	
			Numbers	OD cm	Leaflet length cm	Stick length/OD cm	Leaflet length cm	Stick length/OD cm
Leaf	40	4	1		4			
Leaf	80	8	9	0.5	8	4/0.1	3.5	0.5
Leaf	180	18	10	0.1	5	1.5/0.1	1	0.5/0.1
Leaf	280	28	9	0.2	6	2/0.1	2	2/0.1
Leaf	310	31	10	0.2	8	2/0.1	4	1/0.1
Leaf	340	34	13	0.2	8.5	3	2	0.8
Leaf	390	39	12	0.4	11	2.5	2	0.8
Leaf	400	40	16	0.3	11	3/0.1	1.5	0.5/0.05
Leaf	450	45	15		18	12/0.18	2	
Leaf	460	46	15	0.6	17	11/0.15	2	2.5/0.1
Leaf	490	49	17	0.8	14	11/0.2	2	2.5/0.1

The numbering of Compound Leaf is based on length of leaf, i.e. “Leaf 440”, means a compound leaf with 44 cm in length, see Fig. 45.

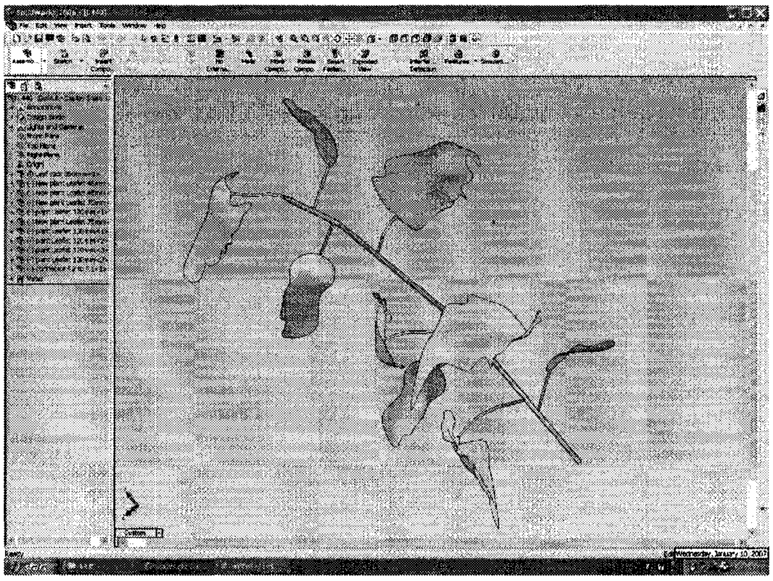


Fig. 45: Leaf 440

Truss

Similar to compound leaf, the truss are measured as shown in Table 18. the CAD models are shown in Appendix 3. The numbering of trusses in the CL are based on the age, i.e. “T2”, means 2nd week-old truss. T1, T2 and T3 are built based on the scans, see Fig. 46, 47 & 48.

Table: 12: Truss data of plant sample#1

CL ID	Age week	Description	F 1 (mm)		F 2 (mm)		F3 (mm)	
			Height	Radio	Height	Radio	Height	Radio
T1	1	forming flower, from scan						
T2	2	fully open flower, forming fruit, from scan						
T3	3	developing fruit	25	15	28	17.5		
T4	4		25	15	32	20	38	22.5
T5	5		32	20	50	30	55	35
T6	6		55	37.5	58	38.5		
T7	7		55	37.5	58	38.5		
T8	8		65	45	58	38.5		

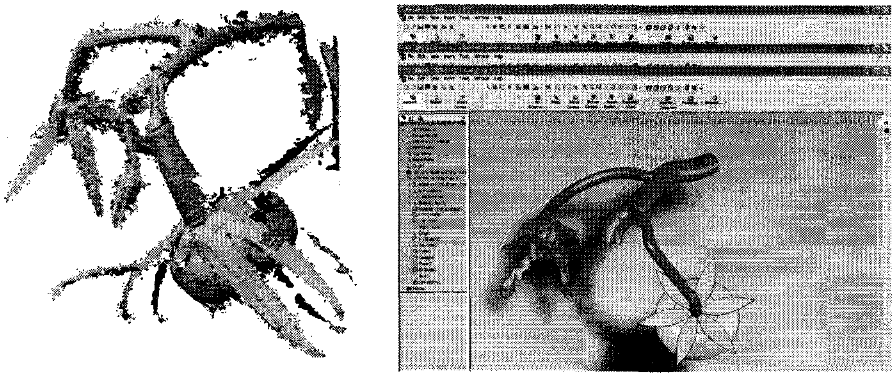


Fig. 46: Constructing a CAD model of a Forming fruit truss T2. Left: the scan; right: CAD model based on scan data

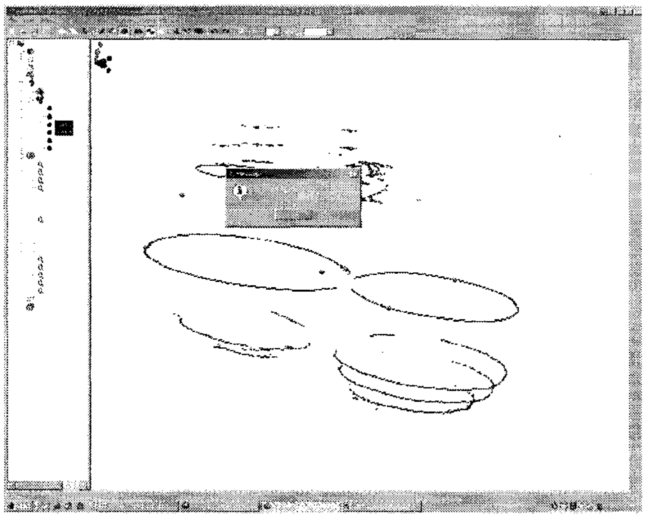


Fig. 47: Cross section data of a fruit truss

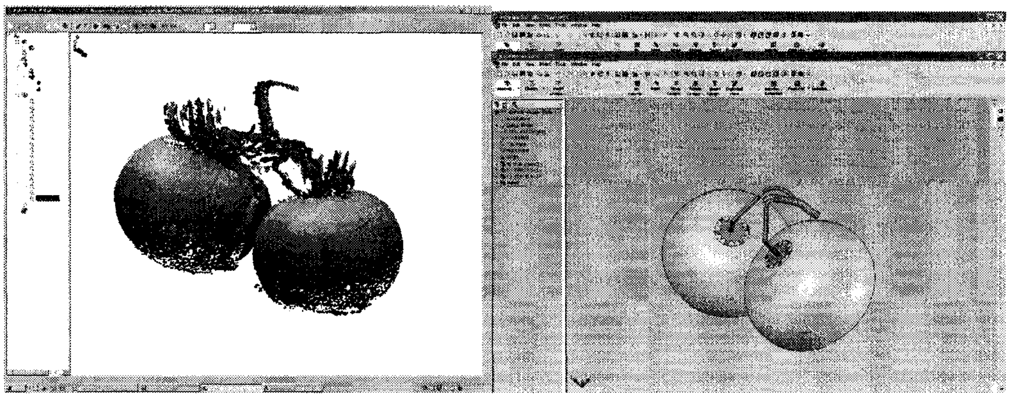


Fig. 48. Left: The scan of T3. Right: the CAD model of T3

Sucker

The positional and size information of a sucker is an entity of interest to plant researchers. There is one sucker at Sample#1 plant. The CAD model is constructed based on the scan reference points.

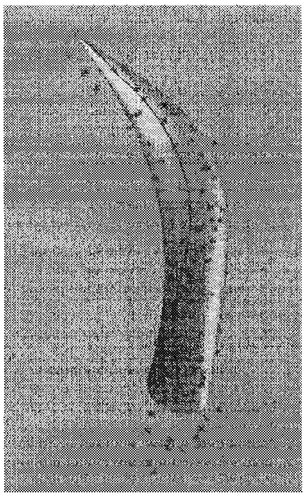


Fig. 49: Sucker

4.4.2 Modeling plant WGS

Based on the mathematical model described earlier, a WGS database is developed to define parameters to be used in CAD. To define the nodes geometry and topology in a WGS, raw data can be obtained from 1) scans or 2) traditional manual measurement. As mentioned in Section 3.4, laser scans promise quick, accurate and automatic data acquisition, while it is expensive on hardware in comparison with manual measurement. In this experiment, for comparison purpose, we use 1) to construct WGS#2, and use method 2) to construct the rest of WGSs. The results of method 2) are included in Appendix 2.

To define the nodes geometry and topology in a WGS, reference points from scan are used.

A simple point in space gives connectivity information between node and respective component. With the exported 44 reference points in CAD, WGS#2 model is constructed (Fig. 51). Table 13 and Table 14 show an example description of WGS #2.

Table 13: Example WGS #2 data

WGS	Normal distribution	Length (cm)	# of Compound Leaf	# of Truss	# of Suckers
# 2	Current measurement	18.5	1	1	NA
	Mean	19.625	1	1	
	Standard Derivative	1.3562	0.5477	0	

Table 14: Example Components data for WGS #2

Components	CL ID	Connectivity to WGS
Compound leaf #1	Leaf 200	Node #1 of scan WGS#2
Truss #1	T2	Node #2 of scan WGS#2

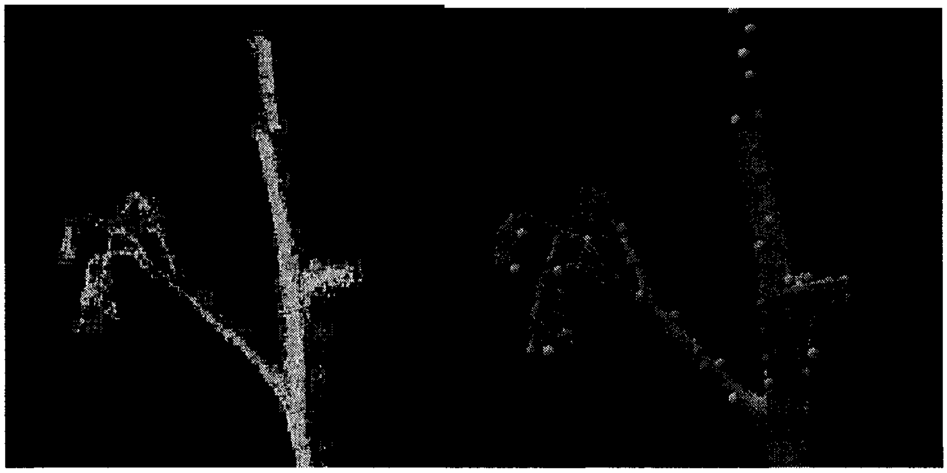
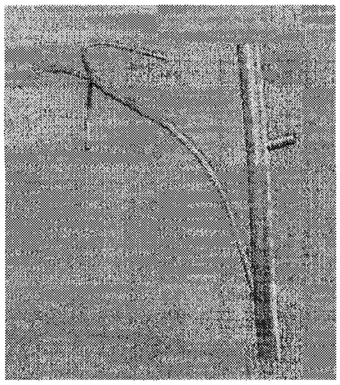


Fig. 50. Left: Critical points are detected by corner detection algorithm, 44 corners detected; Right: Detected corner points are transformed to scan coordinate, ready for exporting to CAD



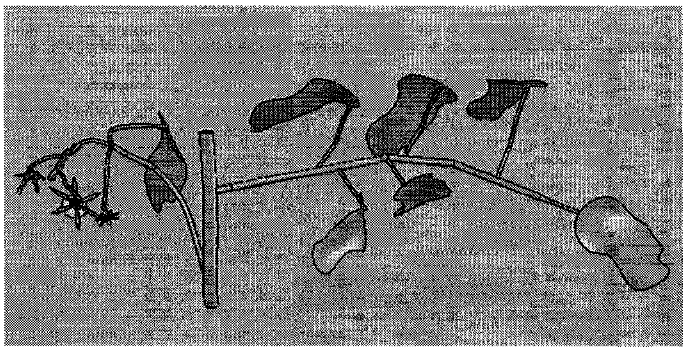


Fig. 51: WGS#2 structure CAD model are constructed from imported 44 reference points

Fig. 52: Components are added to WGS#2 structure CAD model

The WGS#2 CAD model can also be constructed using cross section method (Fig. 53).

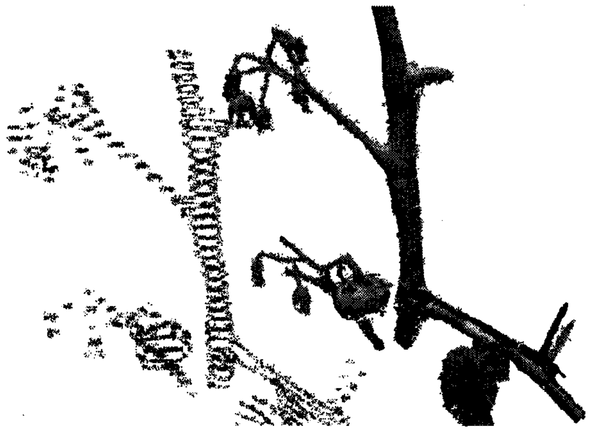


Fig.53: Capture WGS cross section data from scan

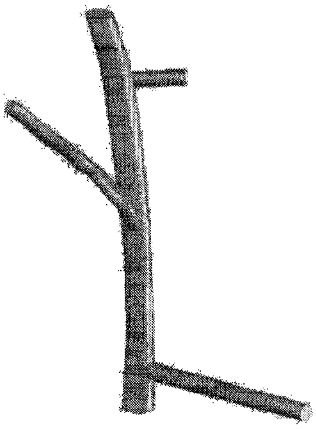


Fig.54: CAD model of the WGS based on the scan

4.4.3 Assembly a tomato plant model from WGS

With selected plant age and the mathematical model, we can define numbers of WGS. A plant model is achieved by correctly assembling the appropriate WGSs.

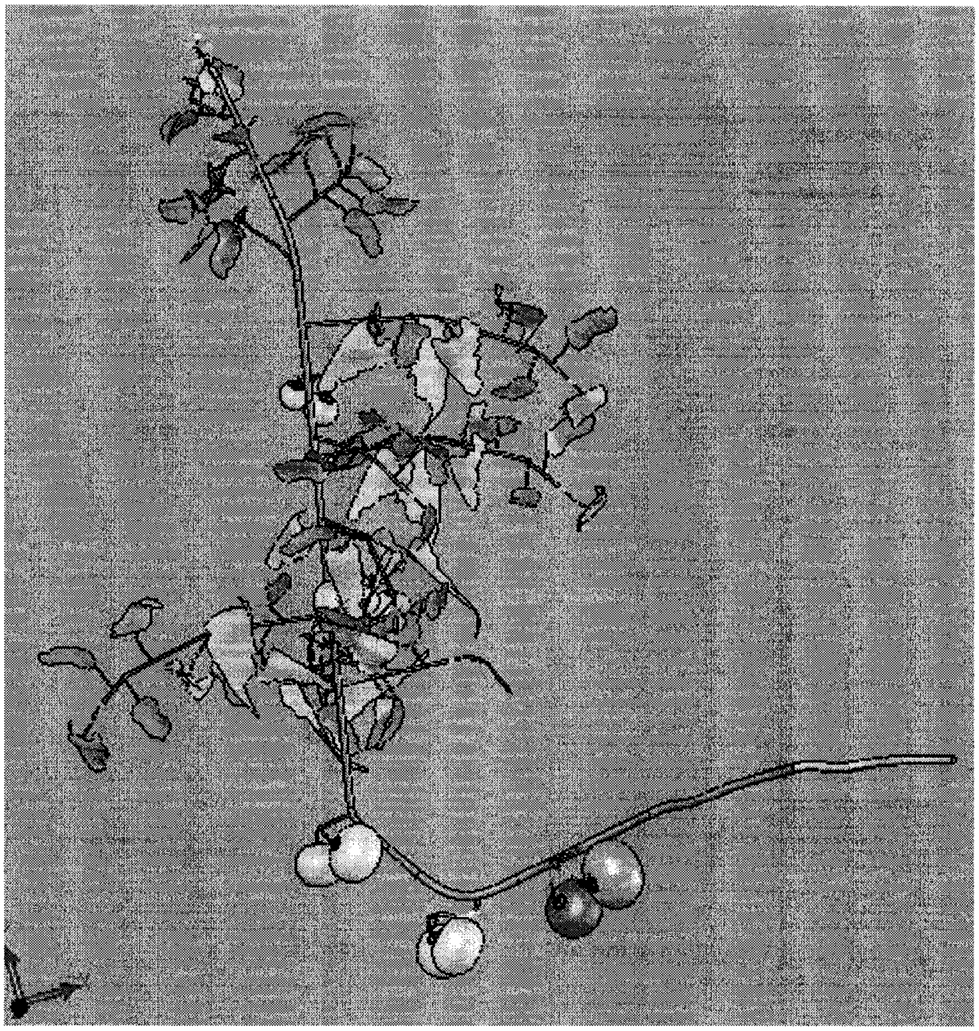


Fig.55: a 16-week-old Plant Model, Assembly of WGS #1 to #16

4.5 Model Validation

As mentioned in Section 3.6, a plant model validation is carried out at different levels. For structured entities, like the geometry and topology of main stems and nodes, we use manual measurement (at field or from scans, see Fig. 56, stem node information are recorded from a scan) for comparison at WGS level.

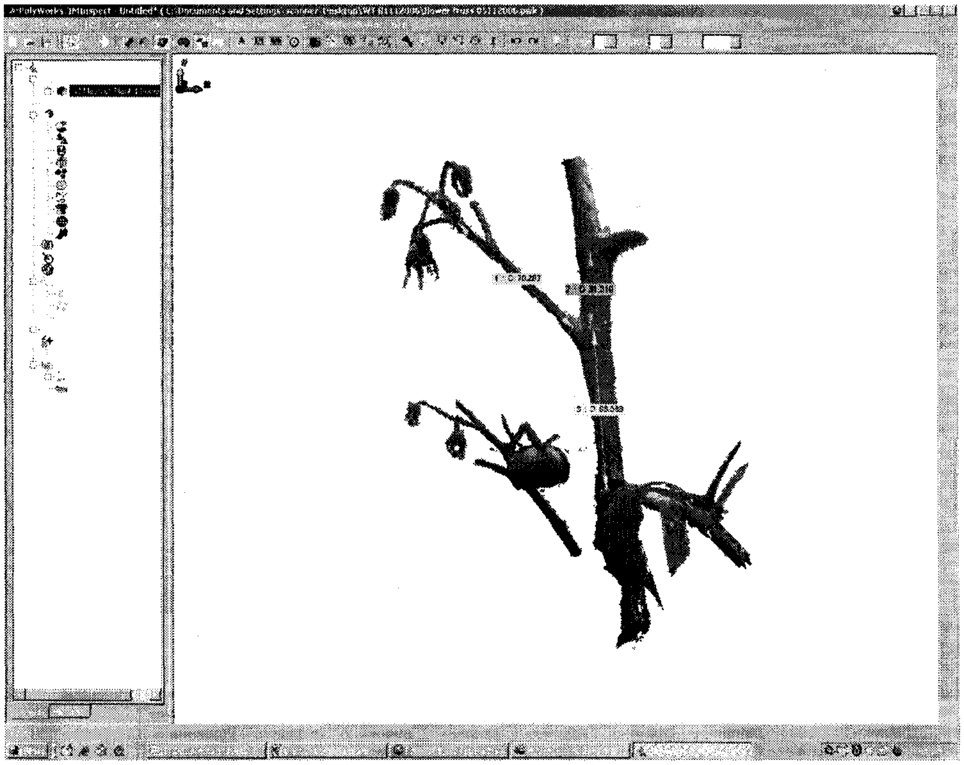


Fig. 56: Model validation using measurements obtained from scans

The following example is to validate the accuracy of node positions of a WGS CAD model with scans (Fig. 57).

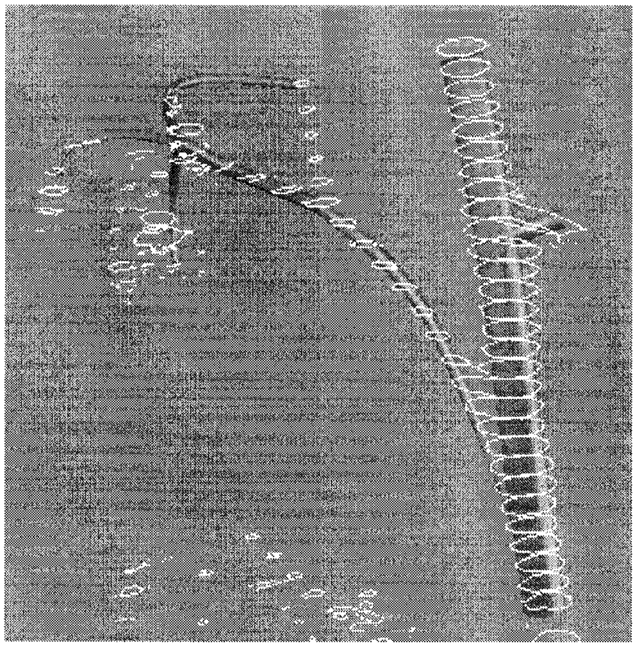


Fig. 57: WGS#2 is validated with scan data

At component level, interested entities are verified with scan data. We export the CAD model of the component back to point cloud environment scan data to make comparison with reference data or vice versa. The following examples show that truss (Fig. 58, Left), leaflet (Fig. 58, Right), sucker (Fig. 59, Left) and a young leaf (Fig. 59, Right) are validated with corresponding scan data.

In this research, we use point-to-point method manually identify the interested attributes and make comparison. At the experiment shown in Fig. 58 Left, interested attributes, such as fruit position data from model and reference are compared. The error are recorded and transferred to the CL for correction and model regeneration. Because current validation is conducted at component level point-to-point, thus, it is very time consuming. A better automate solution is considered a future improvement and listed in Chapter 5, future works.

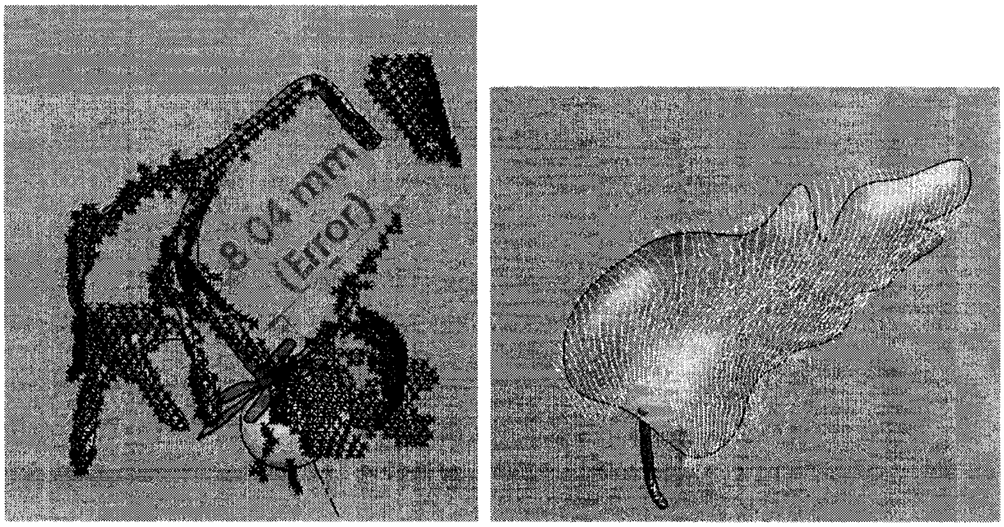


Fig. 58. Left: a fruit truss is validated with scans; Right: a leaflet is validated with scans

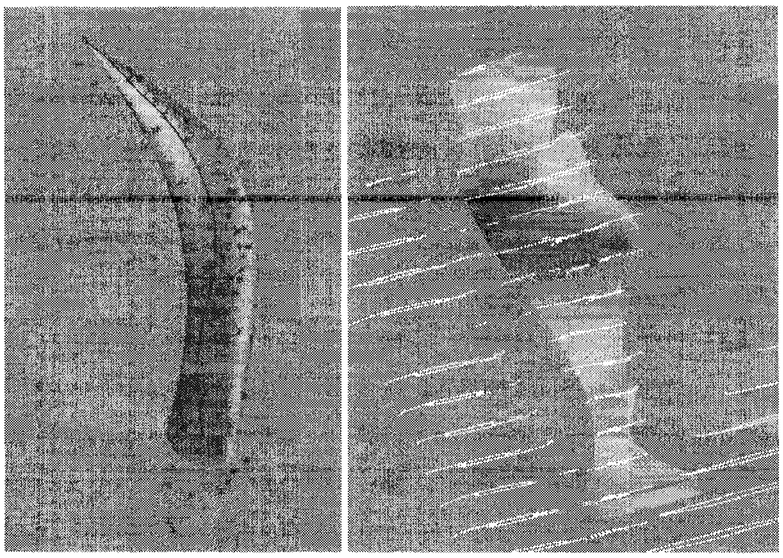


Fig. 59. Left: A sucker is validated with scan data; Right: a young leaf is validated with scans

5. Conclusion and Future Works

We have developed a new methodology for tomato plant modeling which is robust enough for many other similar varieties. In the proposed modeling system, we combine several sources of data to achieve a parametric CAD model based on 3D scans. We gathered weekly plant data to describe WGS and thus connect plant geometry with age and used laser scans and CAD modelling tools to create a library of components. Through modeling interface, users can quickly edit the current models to improve accuracy and accomplish plant variety.

The characteristics of the methodology include:

- High accuracy: the method is based on plant scan data and plant growth records; which are derived from real world measurements. In addition, this system enables model validation starting from component level, resulting in a calibrated plant model.
- Semi-Automatic: with help from laser scans, 3D measurements are automatically captured; with help from CAD's parametric functions, models are automatically regenerated according to re-entered parameters. However, user interaction is required to process the point cloud and generate a CAD 3D model.
- High manipulability: through advanced CAD, parameters are accessed and edited to fine tune and customize different type of plant model (i.e. plants with various ages).
- Wide application: this method can be applied to other complex objects whose structure can be decomposed.
- Computational cost: The user can customize the data size for control point extraction, polygonal model meshing and numbers of reference points to be transferred. A comfortable and manageable data size can thus be achieved by balancing the efficiency of the system and the accuracy of the output model.

The contributions of this research can be summarized into 3 aspects.

1. Traditionally, the input of 3D reconstruction is a set of scans, while the expected output is a watertight polygonal model. Because of conditions of applying 3D registration algorithms and dense dataset handling, 3D reconstruction modeling method are constrained to "simple" structured object, or the quality of the output model are compromised.

   In this research, we utilize the advantage of 3D reconstruction method, while, expecting accurate reference points, rather than a high-quality polygonal model. This arrangement greatly eases the difficulty of 3D registration and dense data processing problems. With reference point exported to CAD software, a parametric model can be made. By this approach, the advantages of 3D reconstruction method and CAD are integrated, which can be used to model more complex objects, like tomato plant.

2. Unorganized point cloud data simplification and extraction are not fully developed today. Practically, user interaction is required. In this research, issues about improving the automation level of this operation are addressed. We propose extracting reference data using corner detection algorithm from a 2D image and transform the detected corner points back to 3D scan coordinate, and further exported to CAD. As a new direction on this topic, the advantage of this approach is that mature 2D image processing technology can be utilized in 3D data processing domain.

3. We introduced the concept of WGS to decompose a tomato plant into meaningful sections. The idea of WGS not only simplifying the complexity of a plant, solving occlusion problems in the plant modeling domain, but also in the crop research domain. WGS relates plant growing characteristics and the interaction to its environment with its geometrical structure. For example, plant age and its generative aspects (e.g. labour task efficiency) can be analyzed from WGS.

To improve the proposed modeling method in terms of level of automation and data efficiency, the following issues can be pursuit in the future works:

- Point cloud simplification: reduce data size, while keep high accuracy
- 3D feature segmentation: extract specified features to model
- Improving the capacity and robustness of data acquisition during laser scanning: surface treatment, 3D registration
- Developing and applying CAD API: automatic CAD model construction with given specifications
- Incorporate more attributes into modeling system, building a components property library: e.g. mass, color, texture.
- Developing a robust plant modeling validation system to increase the automation level and the accuracy.

Appendix 1

Extracting reference points using Harris Corner Detection Algorithm

1. The Harris Corner Detection code

```
% Tong Wang  07 FEB 2007
% Harris Corner detector
% this is a user friendly program, allowing user to key in parameters and
% using dragging down window to select interested object to be corner-detected.

%%%%%%%%%% Initializing data%%%%%%%%%%%%%%%
disp('Enter Standard Derivation');
sigma=input('sigma=');

disp('Enter Threshold Level');
Threshold=input('T=');            %Threshold=0.01;


disp('Enter Neighborhood Size');
s=input('Size=');


%%%%%%%%%%%%% Object Selection%%%%%%%%%%%%%%%
% this section can automatically adjust threshold level to compromise the required
%number of corners expected to be detected


disp('Enter Minimum Number of Corners Expected ');
min_N=input('N_min=');

disp('Enter Maximum Number of Corners Expected ');
max_N=input('N_max=');

 %set how many numbers of corner you are expecting to get

A=imread('p1+z.jpg');                %read-in image
```

```
figure, imview(A);
imshow(A);%show image

%A=imnoise(AA,'salt & pepper', 0.02);
%A= imnoise(AA,'gaussian', 0,0.04);          %implement noise-gaussian

%imshow(A)

% optional - First filtering-reduce noise - gaussian

I=double(A);                        %conversion- unsigned 8-bit integer to double


% prepare windows for corner detection which can be made by mores dragging-down

k = waitforbuttonpress;
point1 = get(gca,'CurrentPoint');  %button down detected
rectregion = rbbox;  %%%return figure units
point2 = get(gca,'CurrentPoint');%%%%button up detected
point1 = point1(1,1:2); %%% extract col/row min and maxs
point2 = point2(1,1:2);
lowerleft = min(point1, point2);
upperright = max(point1, point2);
ymin = round(lowerleft(1)); %%% define 4 - coordinate of the window
ymax = round(upperright(1));
xmin = round(lowerleft(2));
xmax = round(upperright(2));

A=8;
cmin=xmin-A; cmax=xmax+A; rmin=ymin-A; rmax=ymax+A; % set allowance

%%%%%% Calculate gradient  %%%%%%%%%

  dx = [1 0 -1; 1 0 -1; 1 0 -1];          % mask for x-axis, from textbook, A2
  dy = dx';                               % mask for y-axis

  Ix = conv2(I(cmin:cmax,rmin:rmax), dx, 'same');     %get gradient Ix
  Iy = conv2(I(cmin:cmax,rmin:rmax), dy, 'same');     %get gradient Iy
  g = fspecial('gaussian',max(1,fix(6*sigma)), sigma); %%%%%% Gaussien Filter

  %%%%% Forming C, getting Eigenvalue %%%%%%%%%%%%%%%%

  Ix2 = conv2(Ix.^2, g, 'same');
  Iy2 = conv2(Iy.^2, g, 'same');
  Ixy = conv2(Ix.*Iy, g,'same');
  LM=((Ix2+Iy2)-(4*Ixy.*Ixy+(Ix2-Iy2).^2).^(1/2))/2;
```

```
%eigenvalue from "mathworld.wolfram.com"

  R=LM;                              % buffering values to R
  ma=max(max(R));                    %get globle maximum
  sz = 2*s+1;                        % Q size of neighborhood

MX = ordfilt2(R,sz^2,ones(sz));
% perform filtering, replace elements in R by maximum value of Q,
% eliminating non-maxima point

 LM = (R==MX)&(R>Threshold);
% if R (representing a point) is the local maxima and greater than Threshold, increment
%LM by 1

  count=sum(sum(LM(5:size(LM,1)-5,5:size(LM,2)-5)));

%count how many corners detected.

  %%%%%%%%%%%% Dealing with Your expected number of
corners %%%%%%%

  loop=0;
  while (((count<min_N)|(count>max_N))&(loop<30))  % if over-estimate, relax T;

    if count>max_N
      Threshold=Threshold*1.5;
    elseif count < min_N                      % if under-estimate, reduce T;
      Threshold=Threshold*0.5;
    end

    LM = (R==MX)&(R>Threshold);  % LM is maxima list
    count=sum(sum(LM(5:size(LM,1)-5,5:size(LM,2)-5)));
    loop=loop+1;
  end

      R=R*0;           %clear R, and re-load R with LM – maxima list
  R(5:size(LM,1)-5,5:size(LM,2)-5)=LM(5:size(LM,1)-5,5:size(LM,2)-5);
      [r1,c1] = find(R);
            %[I,J] = FIND(X) returns the row and column indices (location) of
            %the nonzero entries in the matrix X.  This is often used
            %with sparse matrices.


  PIP=[r1+cmin,c1+rmin]%% IP

 Size_PI=size(PIP,1);
 for r=1: Size_PI
```

```
  I(PIP(r,1)-3:PIP(r,1)+3,PIP(r,2)-3)=255;
  I(PIP(r,1)-3:PIP(r,1)+3,PIP(r,2)+3)=255;
  I(PIP(r,1)-3,PIP(r,2)-3:PIP(r,2)+3)=255;
  I(PIP(r,1)+3,PIP(r,2)-3:PIP(r,2)+3)=255;
  % will add in numbering label here
  end

%%%%%%% mark the corners %%%%%%%%%%%%%%%%

  cor=PIP;

  inter_x = cor(:,2);
  inter_y = cor(:,1);

%%%%%%%%%%%%%%%%    options to display %%%%%%%%
% draw an "X" at the point of intersection

  figure,
  imshow(uint8(I));          %Convert to unsigned 8-bit integer.

  %%%%%%%%%prepare labelling on each corner %%%%%%%%%%%%

        hold on;
        text(10,10,'Applying Harris Corner Detection','Color','y','FontWeight', 'bold');
        aa=size(PIP);
        disp('Numbers of the corner detected ');
        aa(1)
```

2. The detected corners

The followings is the Matlab output with the input image of Fig. 30, Left. The 44 corners detected are also shown on the same Figure.

```
Enter Standard Derivation
sigma=1
Enter Threshold Level
T=0.05
Enter Neighborhood Size
Size=5
Enter Minimum Number of Corners Expected
N_min=5
Enter Maximum Number of Corners Expected
N_max=25

PIP=[r, c];
```

```
PIP =

   287   270
   290   282
   300   287
   290   292
   302   300
   293   306
   280   309
   305   309
   252   310
   305   315
   290   319
   248   323
   285   325
   290   331
   233   334
   304   335
   289   339
   254   342
   318   343
   244   344
   271   347
   293   347
   263   348
   251   350
   307   350
   324   351
   270   357
   298   359
   253   362
   308   363
   263   364
   310   373
   313   387
   306   388
   315   395
   306   396
   318   403
   309   408
   311   416
   322   417
   314   425
   325   425
   329   437
   319   438
```

```
Numbers of the corner detected

ans =

    44
```

3. The transformation between 2D image and 3D scan

The principle of the transformation is shown at Section 7.2.3. The following is the detailed implementation with the input data PIP shown at section 10.5.4.2.

```
>> r=PIP(1:44);
>> r'

ans =

   287
   290
   300
   290
   302
   293
   280
   305
   252
   305
   290
   248
   285
   290
   233
   304
   289
   254
   318
   244
   271
   293
   263
   251
   307
   324
   270
   298
   253
   308
   263
   310
   313
   306
   315
   306
   318
   309
   311
   322
   314
   325
   329
   319

>> rnew=r'-288

rnew =

 -1
  2
 12
  2
 14
  5
 -8
 17
-36
 17
  2
-40
 -3
  2
-55
 16
  1
-34
 30
-44
-17
  5
-25
-37
 19
 36
```

```
-18
 10
-35
 20
-25
 22
 25
 18
 27
 18
 30
 21
 23
 34
 26
 37
 41
 31

>> c=PIP(45:88);
>> c'

ans =

   270
   282
   287
   292
   300
   306
   309
   309
   310
   315
   319
   323
   325
   331
   334
   335
   339
   342
   343
   344
   347
   347
   348
```

```
  350
  350
  351
  357
  359
  362
  363
  364
  373
  387
  388
  395
  396
  403
  408
  416
  417
  425
  425
  437
  438

>> cnew=c'-335

cnew =

  -65
  -53
  -48
  -43
  -35
  -29
  -26
  -26
  -25
  -20
  -16
  -12
  -10
   -4
   -1
    0
    4
    7
    8
    9
   12
```

```
    12
    13
    15
    15
    16
    22
    24
    27
    28
    29
    38
    52
    53
    60
    61
    68
    73
    81
    82
    90
    90
   102
   103

>> X=-63.9;
>> Y=65.6+cnew*0.717

Y =

   18.9950
   27.5990
   31.1840
   34.7690
   40.5050
   44.8070
   46.9580
   46.9580
   47.6750
   51.2600
   54.1280
   56.9960
   58.4300
   62.7320
   64.8830
   65.6000
   68.4680
   70.6190
   71.3360
   72.0530
   74.2040
   74.2040
   74.9210
   76.3550
   76.3550
   77.0720
   81.3740
   82.8080
   84.9590
   85.6760
   86.3930
   92.8460
  102.8840
  103.6010
  108.6200
  109.3370
  114.3560
  117.9410
  123.6770
  124.3940
  130.1300
  130.1300
  138.7340
  139.4510

>> Z=rnew*0.681-1555.4

Z =

  1.0e+003 *

   -1.5561
   -1.5540
   -1.5472
   -1.5540
   -1.5459
   -1.5520
   -1.5608
   -1.5438
   -1.5799
   -1.5438
   -1.5540
   -1.5826
   -1.5574
   -1.5540
```

-1.5929
-1.5445
-1.5547
-1.5786
-1.5350
-1.5854
-1.5670
-1.5520
-1.5724
-1.5806
-1.5425
-1.5309
-1.5677
-1.5486
-1.5792
-1.5418
-1.5724
-1.5404
-1.5384
-1.5431
-1.5370
-1.5431
-1.5350
-1.5411
-1.5397
-1.5322
-1.5377
-1.5302
-1.5275
-1.5343

With derived Y and Z, corresponding X can be found from the point cloud (X, Y, Z) dataset.

Appendix 2

Obtain WGS information from manual measurement

List of Tables

List of Figures

Tables Page

WGS #1

Table 2-1: WGS#1 data

WGS age	1		
Items	Current	Mean	Stan'd Deviation
Numbers of Truss	1	1	0
Numbers of compound leaf	3	3	0.5164
Length cm	18.00	19.31	1.6462
OD mm	8.00		

Component	Position (cm) down to next comp't	Orientation to Front (Degree)	CL ID
L11	On top	300	Forming leaf
F7	1 to L11	290	Forming flower truss, T1
L10	6.5 to F7	120	Leaf 110
L9	5 to L10	340	Leaf 120
Sucker	On L9	340	Sucker

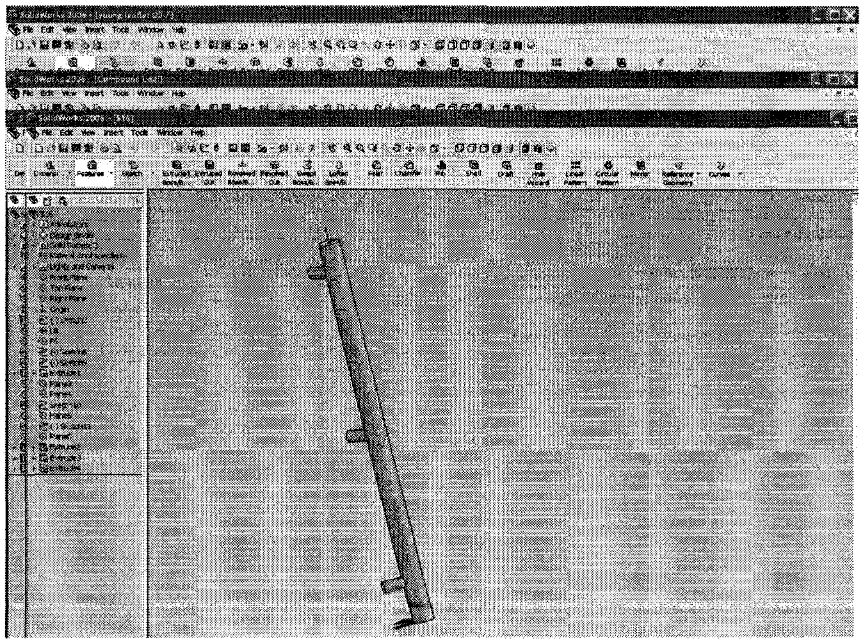


Fig. 2-1: WGS#1

WGS#1

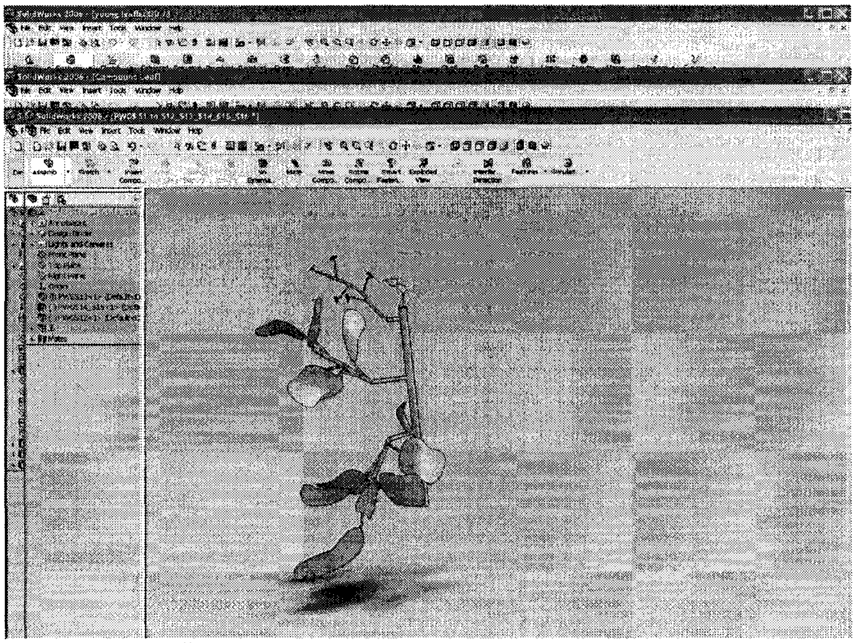


Fig. 2-2: WGS#2+CL

WGS #2

Table 2-2: WGS#2 data

WGS age	2		
Items	Current	Mean	Stan'd Deviation
Numbers of Truss	1	1	0
Numbers of compound leaf	1	1	0.5477
Length cm	18.50	19.625	1.3562
OD mm	8.50		

Component	Position (cm) down to next comp't	Orientation to Front (Degree)	CL ID
L8	6 to top	340	Leaf 280
F6	9 to L8	200	Fully opened flower truss with forming fruit, T2

WGS#2 is constructed using scan data. Reference points derived from corner detection algorithm. Refer to Fig. 50, 51 and 52 at section 4.4.2 for details.

WGS #3

Table 2-3: WGS#3 data

WGS age	3		
Items	Current	Mean	Stan'd Deviation
Numbers of Truss	1	1	0.5164
Numbers of compound leaf	1	1	0.5477
Length cm	19.00	19.25	0.8844
OD mm	8.80		

Component	Position (cm) down to next comp't	Orientation to Front (Degree)	CL ID
L7	6 to top	290	Leaf 440
F5	8 to L7	190	Young fruit, T3

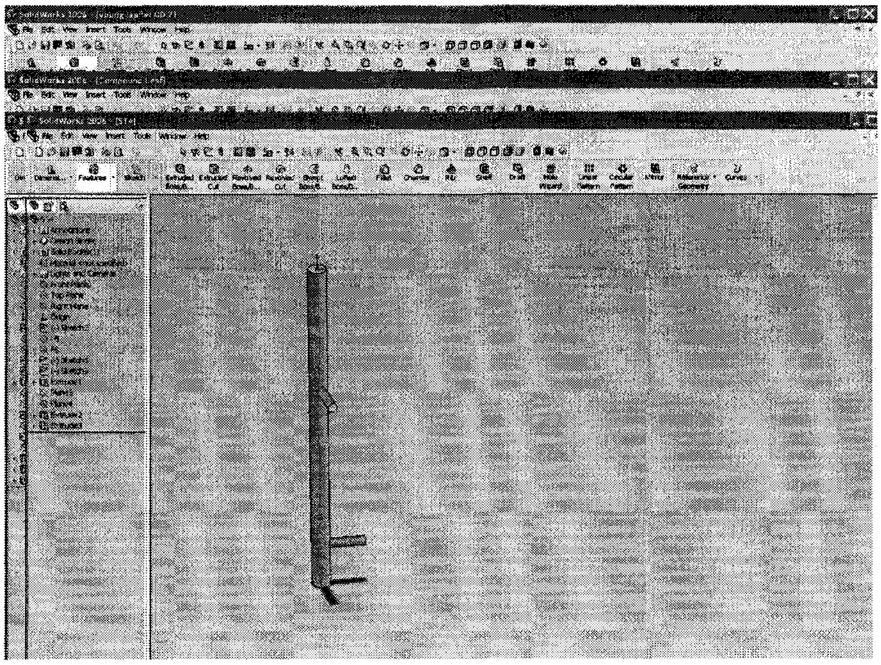


Fig. 2-3: WGS#3

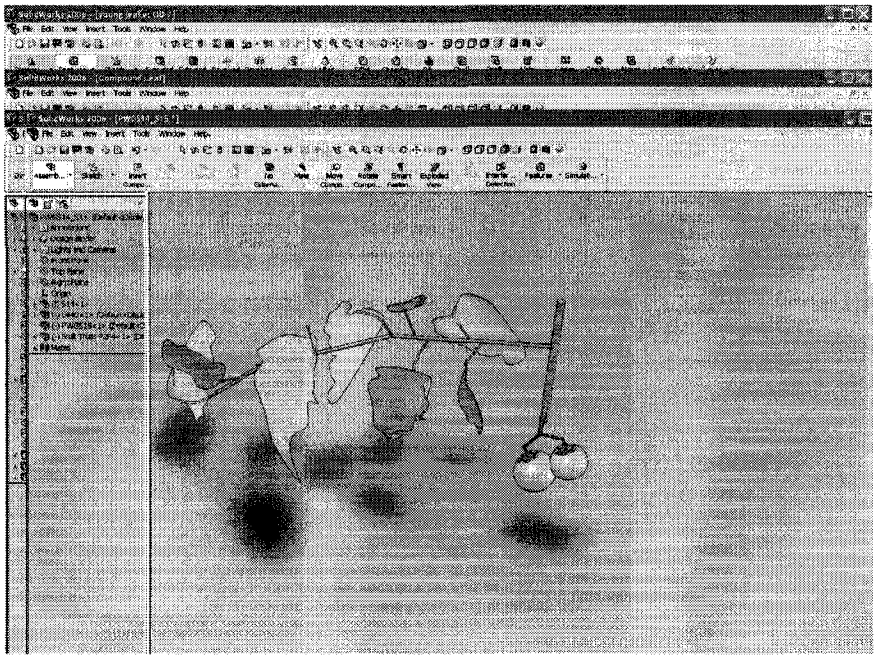


Fig. 2-4: WGS#3+CL

WGS #4

Table 2-4: WGS#4 data

WGS age	4		
Items	Current	Mean	Stan'd Deviation
Numbers of Truss	0	0	0.5477
Numbers of compound leaf	3	3	0.7417
Length cm	17.5	19.625	1.2462
OD mm	9.30		

Component	Position (cm) down to next comp't	Orientation to Front (Degree)	CL ID
L6	5 to top	100	Leaf 460
L5	7 to L6	0	Leaf 490
L4	8 to L5	90	Leaf 460

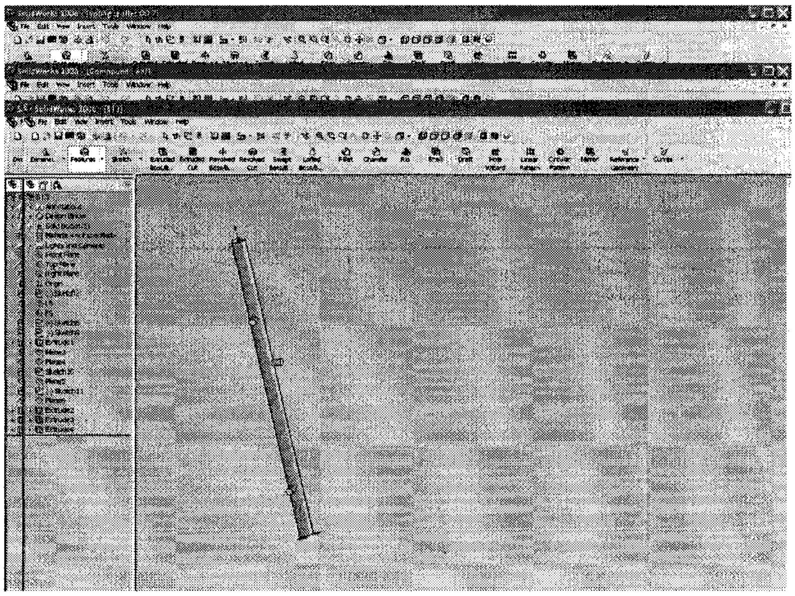


Fig. 2-5: WGS#4

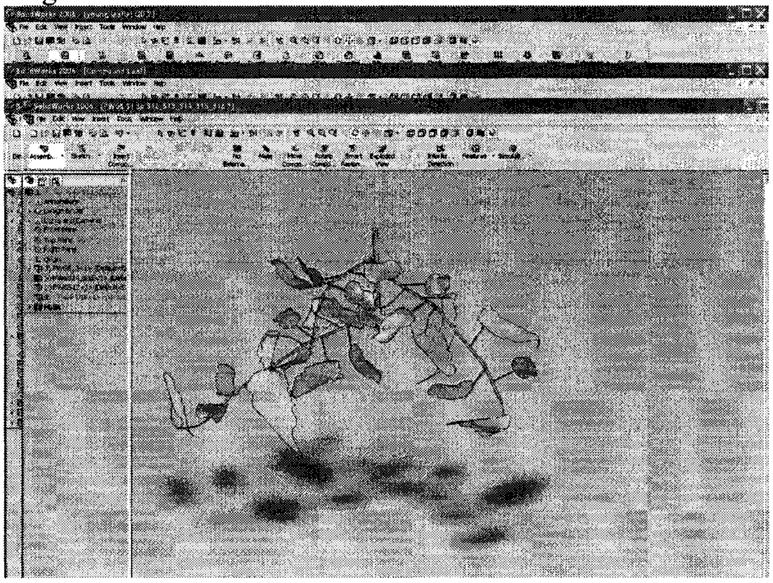


Fig. 2-6: WGS#4+CL

WGS #5

Table 2-5: WGS#5 data

WGS age	5		
Items	Current	Mean	Stan'd Deviation

Numbers of Truss	1	1	0
Numbers of compound leaf	3	3	0.5477
Length cm	20.00	20.00	1.4142
OD mm	9.50		

Component	Position (cm) down to next comp't	Orientation to Front (Degree)	CL ID
F4	4 to top	270	T5
L3	6 to F4	170	Leaf 400
L2	6 to L3	100	Leaf 450
L1	3 to L2	210	Leaf 420

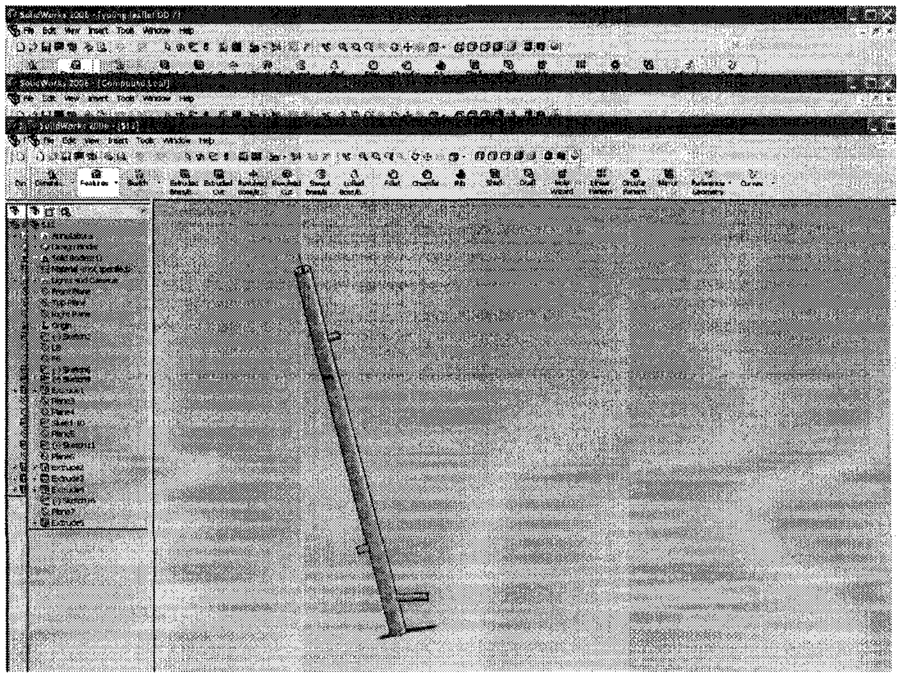


Fig. 2-7: WGS#5

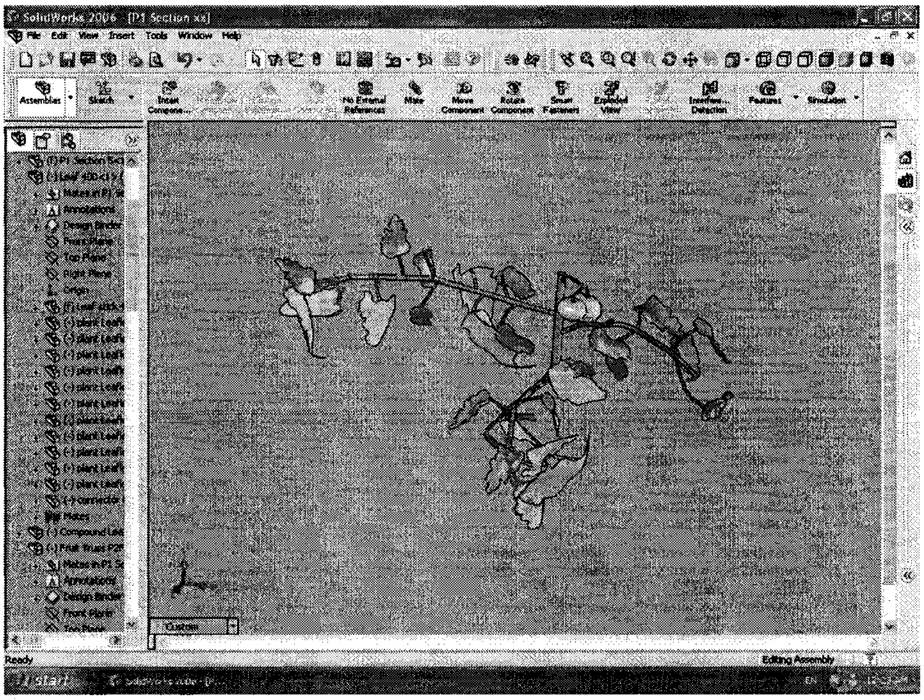


Fig. 2-8: WGS#5+CL

WGS #6

Table 2-6: WGS#6 data

WGS age	6		
Items	Current	Mean	Stan'd Deviation
Numbers of Truss	1	1	0
Numbers of compound leaf	0	0.3333	0.5477
Length cm	21.00	20.3333	1.5055
OD mm	10.50		

Component	Position (cm) down to next comp't	Orientation to Front (Degree)	CL ID
F3	6 to top	310	T6

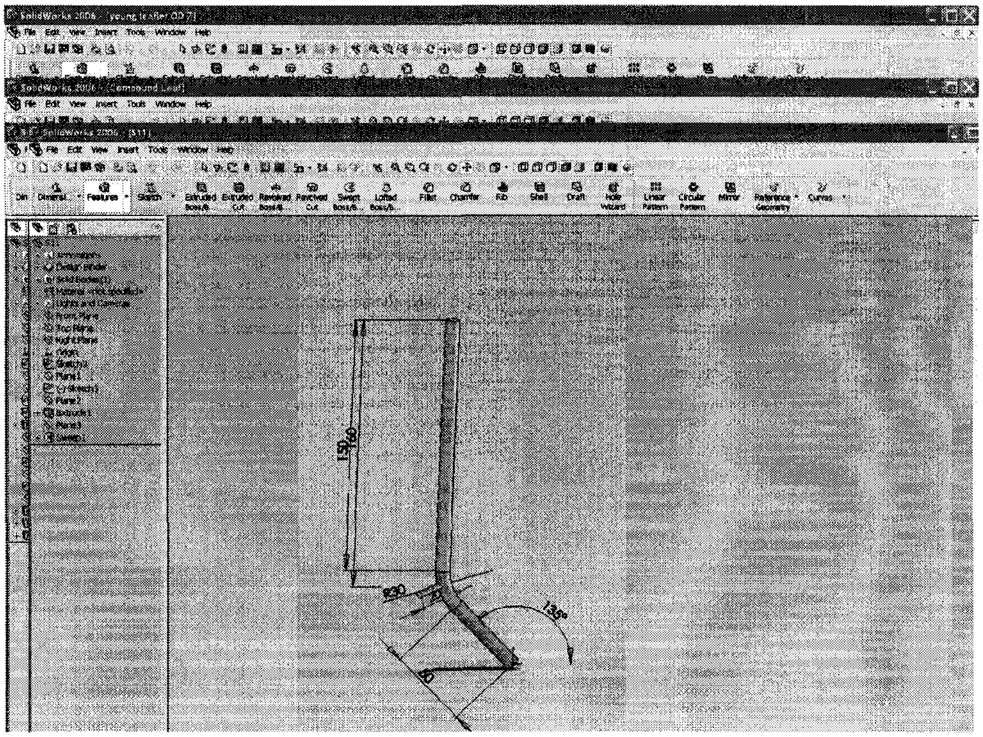


Fig. 2-9: WGS#6

WGS #7

Table: 2-7: WGS#7 data

WGS age	7		
Items	Current	Mean	Stan'd Deviation
Numbers of Truss	1	1	0.8623
Numbers of compound leaf	0	0	0
Length cm	21.50	20.8333	1.2111
OD mm	11.3		

Component	Position (cm) down to next comp't	Orientation to Front (Degree)	CL ID
F2	16 to top	300	T7

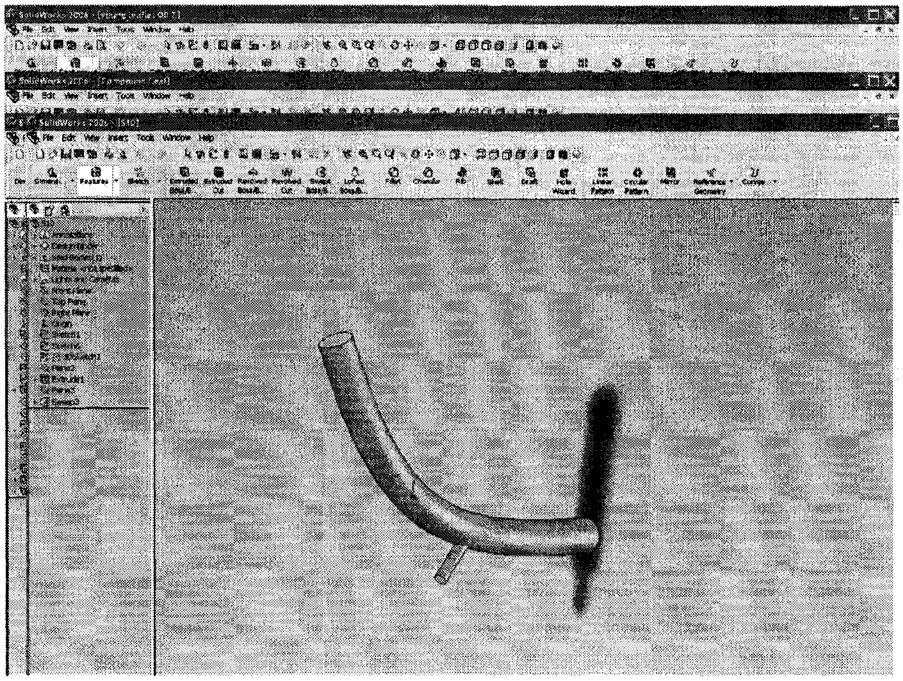


Fig. 2-10: WGS#7

WGS #8

Table 2-8: WGS#8

WGS age	8		
Items	Current	Mean	Stan'd Deviation
Numbers of Truss	1	1	0
Numbers of compound leaf	0	0	0
Length cm	22.00	20.9871	1.1690
OD mm	12.18		

Component	Position (cm) down to next comp't	Orientation to Front (Degree)	CL ID
F1	10 to top	300	T1

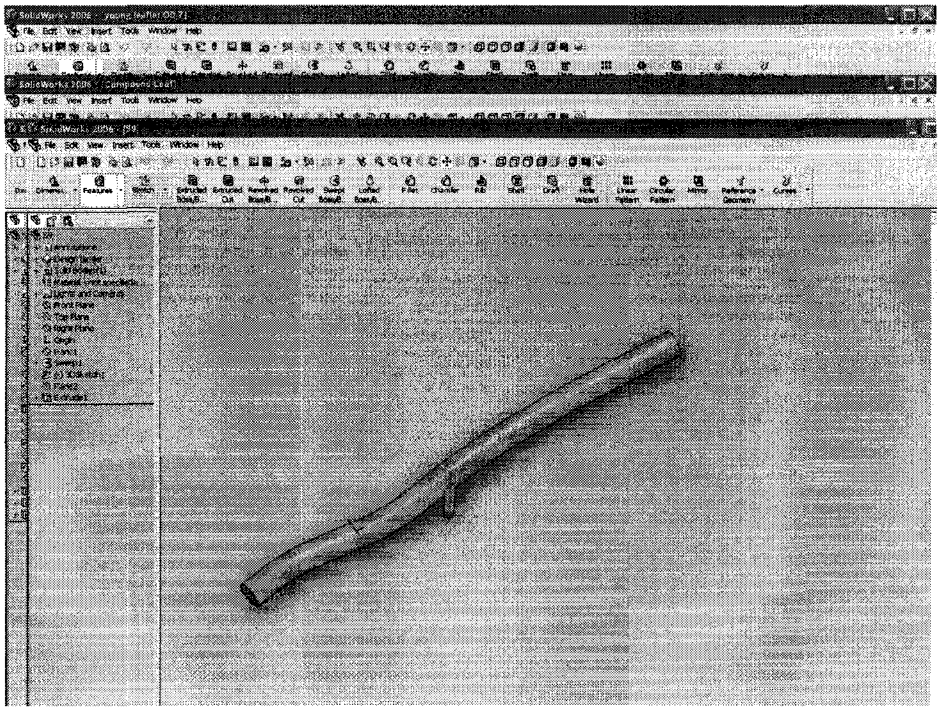


Fig. 2-11: WGS#96

WGS #9

Table 2-9: WGS#9

WGS age	9		
Items	Current	Mean	Stan'd Deviation
Numbers of Truss	0	0	0
Numbers of compound leaf	0	0	0
Length cm	19.000	19.1667	1.2910
OD mm	12.6		

Component	Position (cm) down to next comp't	Orientation to Front (Degree)	CL ID

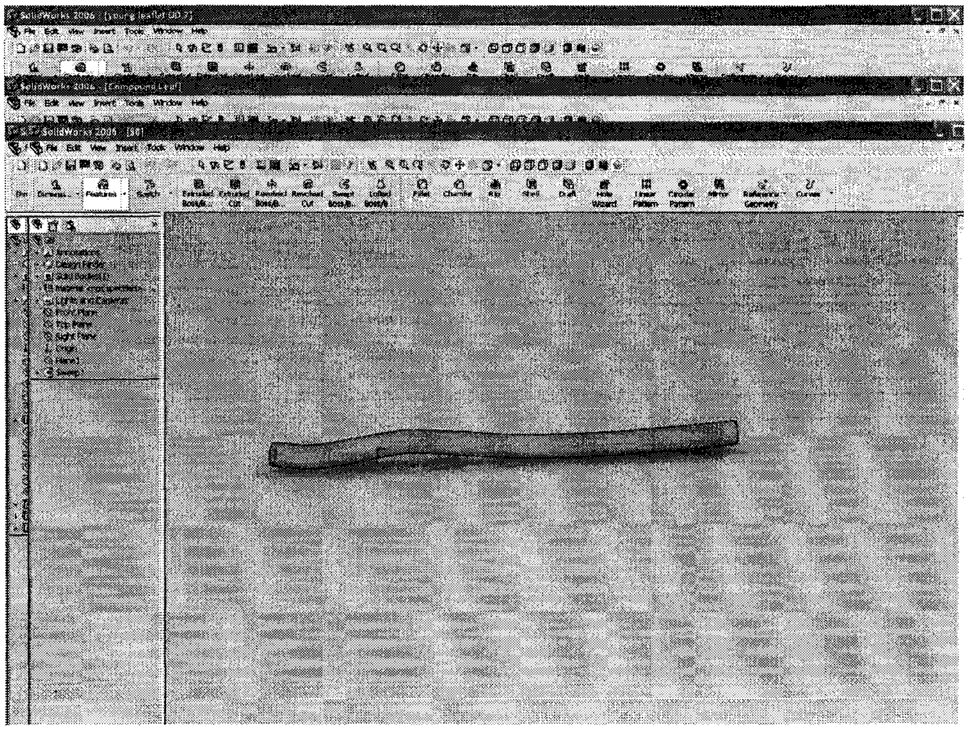


Fig. 2-12: WGS#9

WGS #10

Table 2-10: WGS#10 data

WGS age	10		
Items	Current	Mean	Stan'd Deviation
Numbers of Truss	0	0	0
Numbers of compound leaf	0	0	0
Length cm	18.50	19.8333	1.5055
OD mm	12.83		

Component	Position (cm) down to next comp't	Orientation to Front (Degree)	CL ID

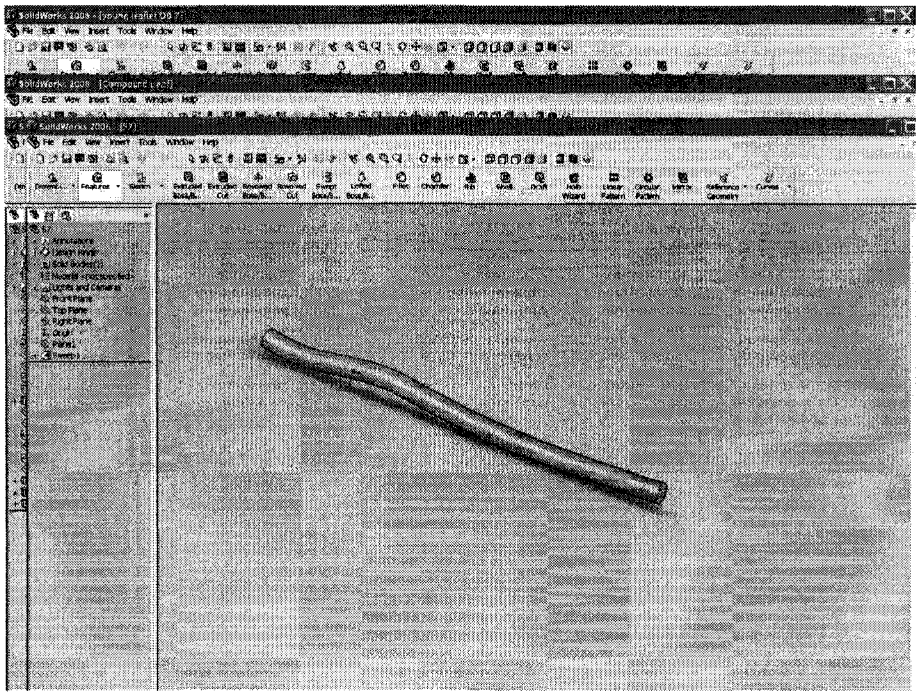


Fig. 2-13: WGS#10

WGS #11

Table 2-11: WGS#11

WGS age	11		
Items	Current	Mean	Stan'd Deviation
Numbers of Truss	0	0	0
Numbers of compound leaf	0	0	0
Length cm	19.00	19.00	0.6325
OD mm	11.80		

Component	Position (cm) down to next comp't	Orientation to Front (Degree)	CL ID

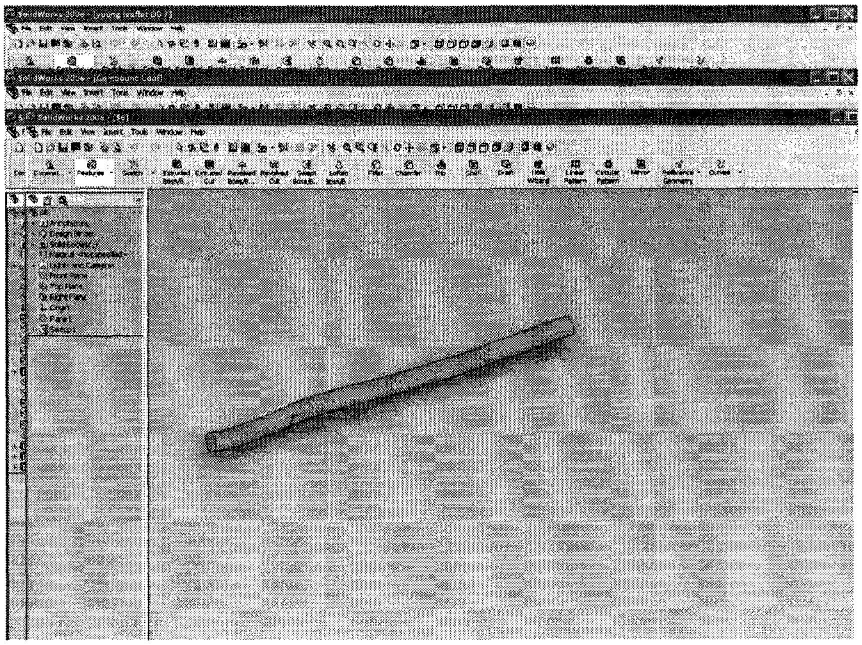


Fig. 2-14: WGS#11

WGS #12

Table 2-12: WGS#12

WGS age	12		
Items	Current	Mean	Stan'd Deviation
Numbers of Truss	0	0	0
Numbers of compound leaf	0	0	0
Length cm	17.50	19.33	1.1690
OD mm	11.68		

Component	Position (cm) down to next comp't	Orientation to Front (Degree)	CL ID

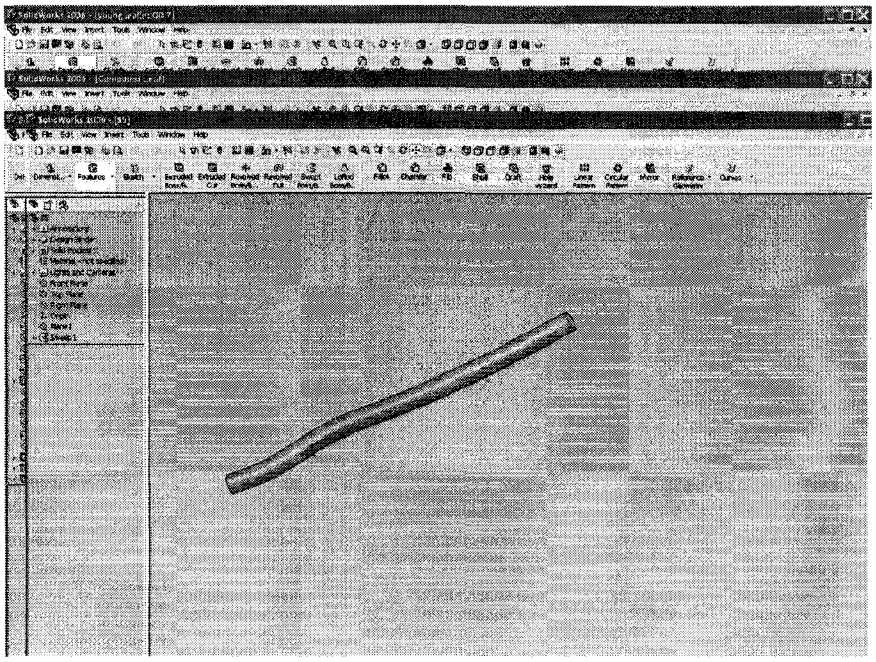


Fig. 2-15: WGS#12

WGS #13

Table 2-13: WGS#13

WGS age	13		
Items	Current	Mean	Stan'd Deviation
Numbers of Truss	0	0	0
Numbers of compound leaf	0	0	0
Length cm	22.00	20.3333	1.6330
OD mm	11.70		

Component	Position (cm) down to next comp't	Orientation to Front (Degree)	CL ID

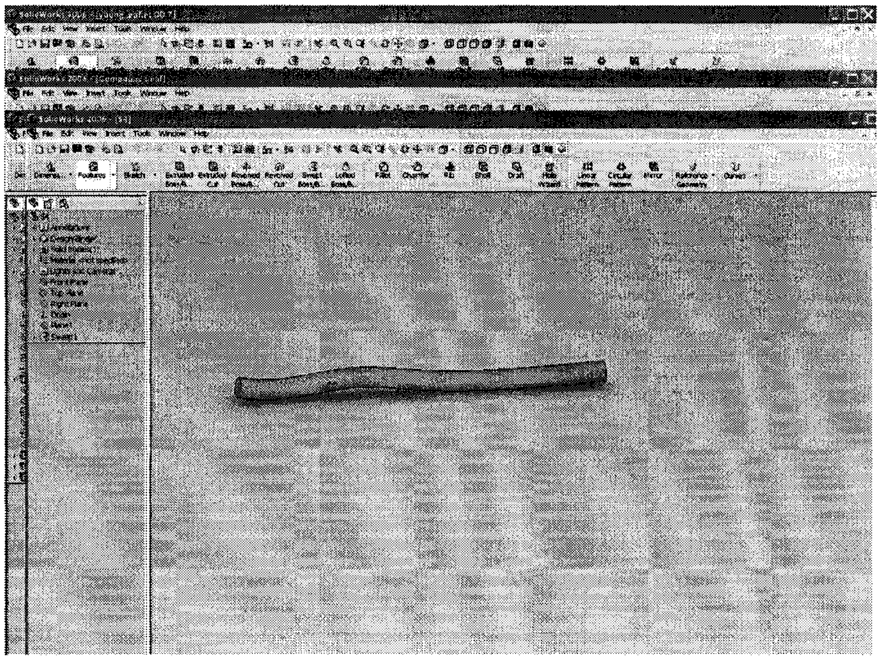


Fig. 2-16: WGS#13

WGS #14

Table 2-14: WGS#14 data

WGS age	14		
Items	Current	Mean	Stan'd Deviation
Numbers of Truss	0	0	0
Numbers of compound leaf	0	0	0
Length cm	21.00	20.3333	1.5055
OD mm	11.57		

Component	Position (cm) down to next comp't	Orientation to Front (Degree)	CL ID

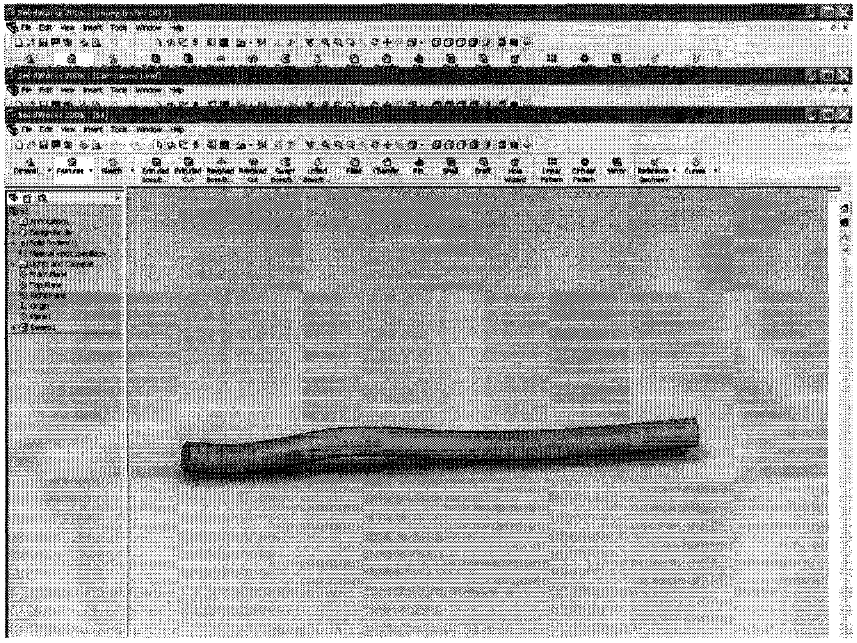


Fig. 2-17: WGS#14

WGS #15

Table 2-15: WGS#15 data

WGS age	15		
Items	Current	Mean	Stan'd Deviation
Numbers of Truss	0	0	0
Numbers of compound leaf	0	0	0
Length cm	21.50	20.8333	1.2111
OD mm	10.96		

Component	Position (cm) down to next comp't	Orientation to Front (Degree)	CL ID

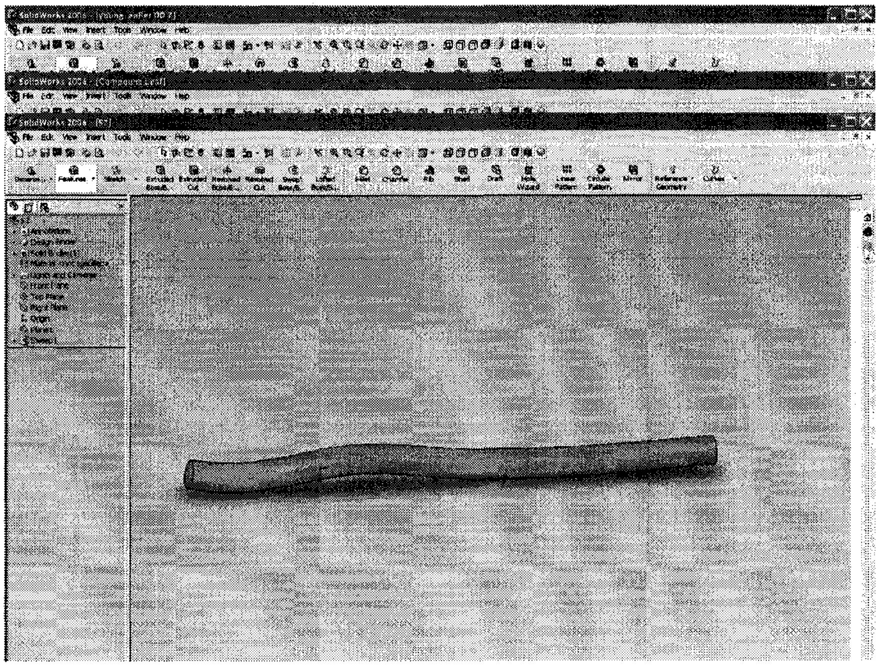


Fig. 2-18: WGS#15

WGS #16

Table 2-16: WGS#16 data

WGS age	16		
Items	Current	Mean	Stan'd Deviation
Numbers of Truss	0	0	0
Numbers of compound leaf	0	0	0
Length cm	22.00	20.8333	1.1690
OD mm	11.89		

Component	Position (cm) down to next comp't	Orientation to Front (Degree)	CL ID

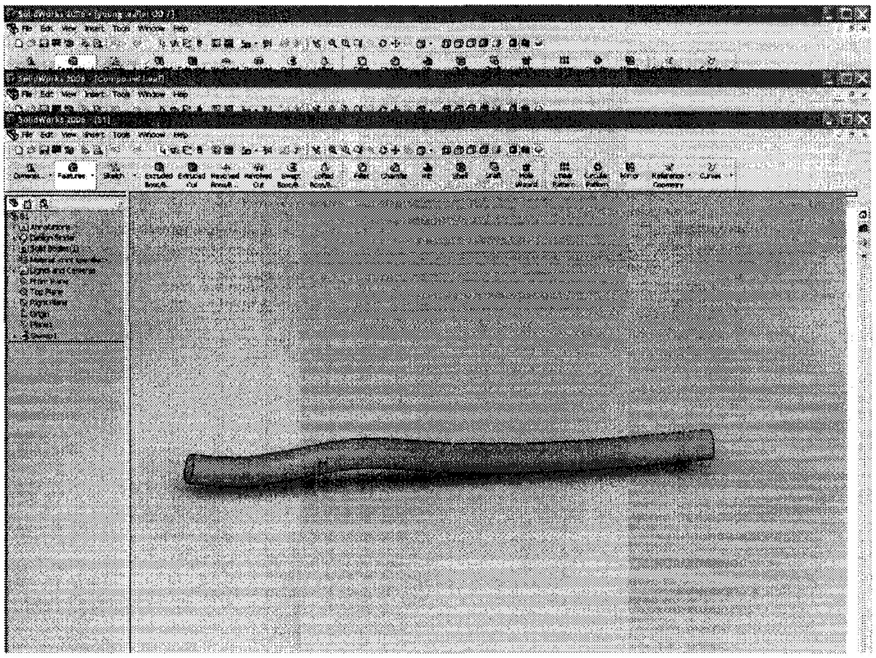


Fig. 2-19: WGS#16

WGS # base

Table 2-17: WGS Base data

WGS age	base	
Items	Mean	Stan'd Deviation
Numbers of Truss	0	
Numbers of compound leaf	0	
Length cm	75	
OD mm	11.5	

Component	Position (cm) down to next comp't	Orientation to Front (Degree)	CL ID

Appendix 3:

Plant Components Library

List of Figures

Leaflet

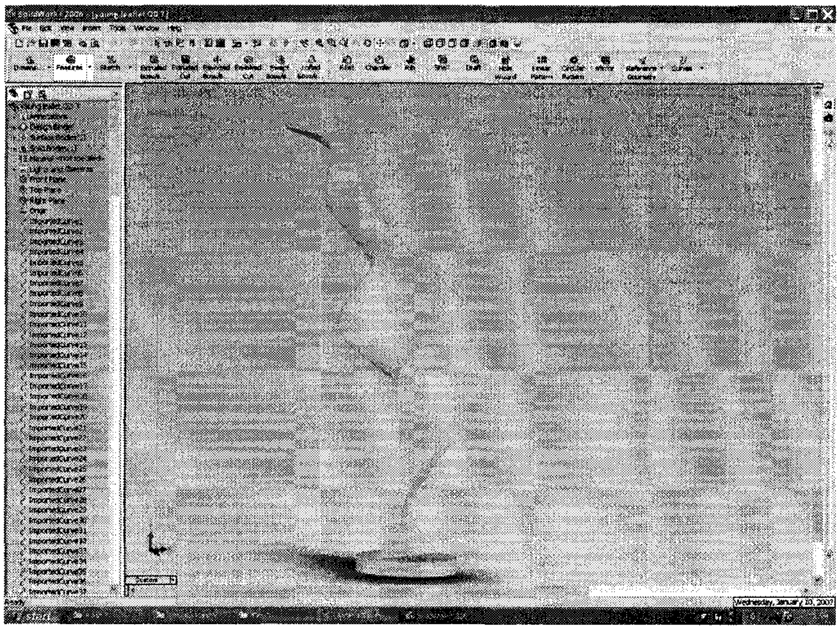

SolidWorks 2006
File Edit View Insert Tools Window Help
Wednesday, January 10, 2007

Fig. 3-1: 1 week-old leaflet, Leaf 40

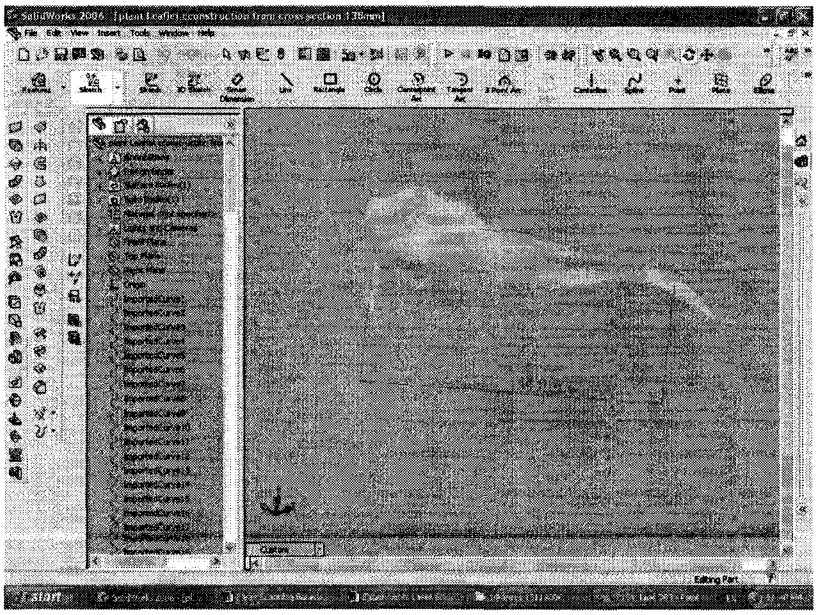


Fig. 3-2: leaflet 45

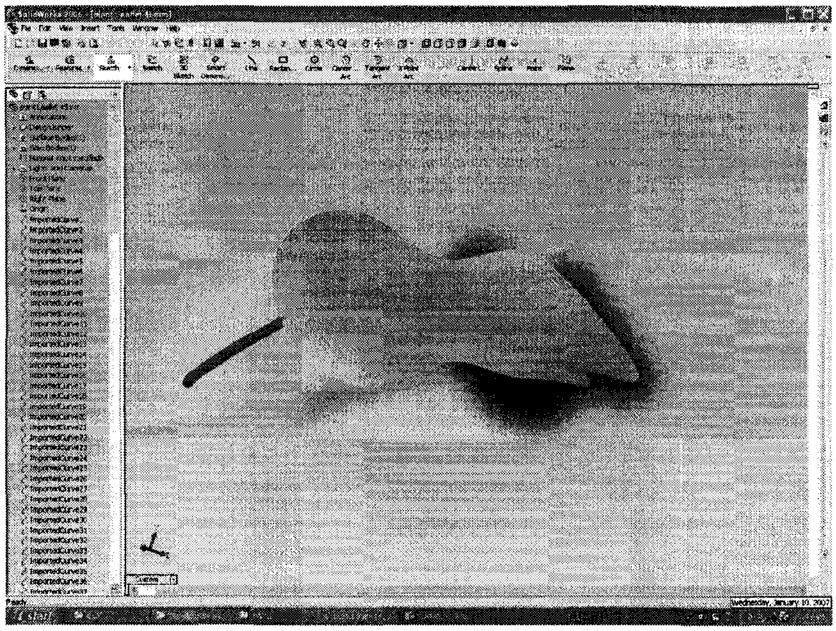


Fig. 3-3 : leaflet 75

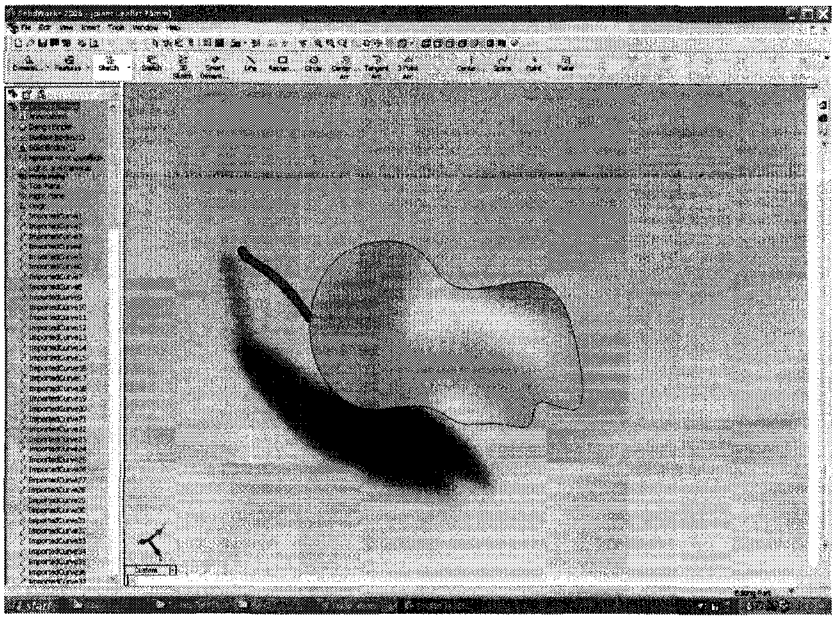


Fig. 3-4: leaflet 90

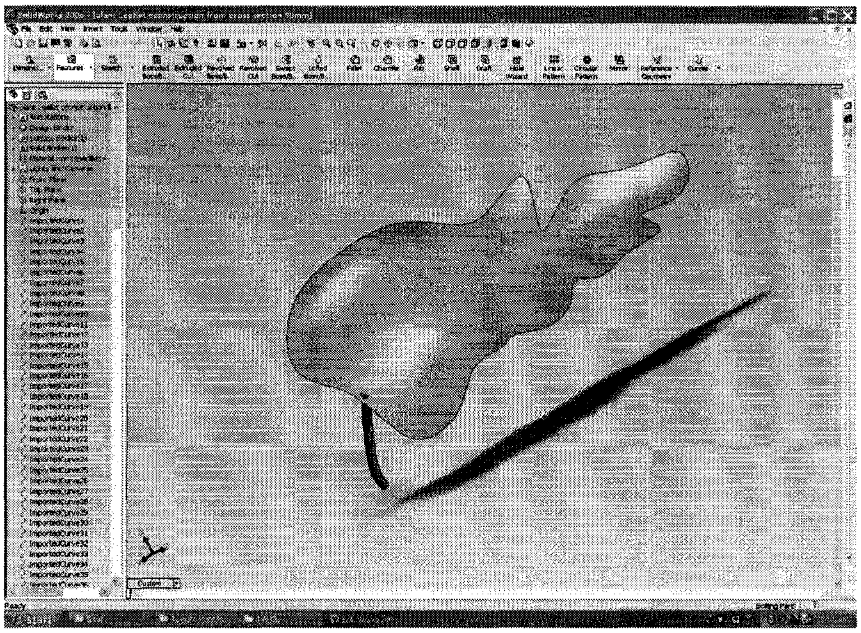


Fig. 3-5: leaflet 120

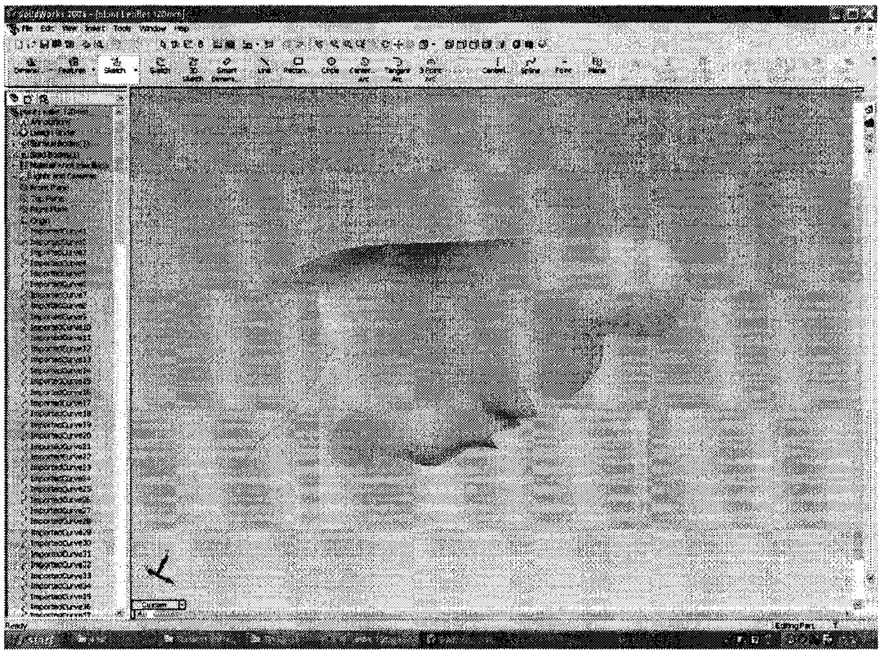


Fig. 3-6: leaflet 138

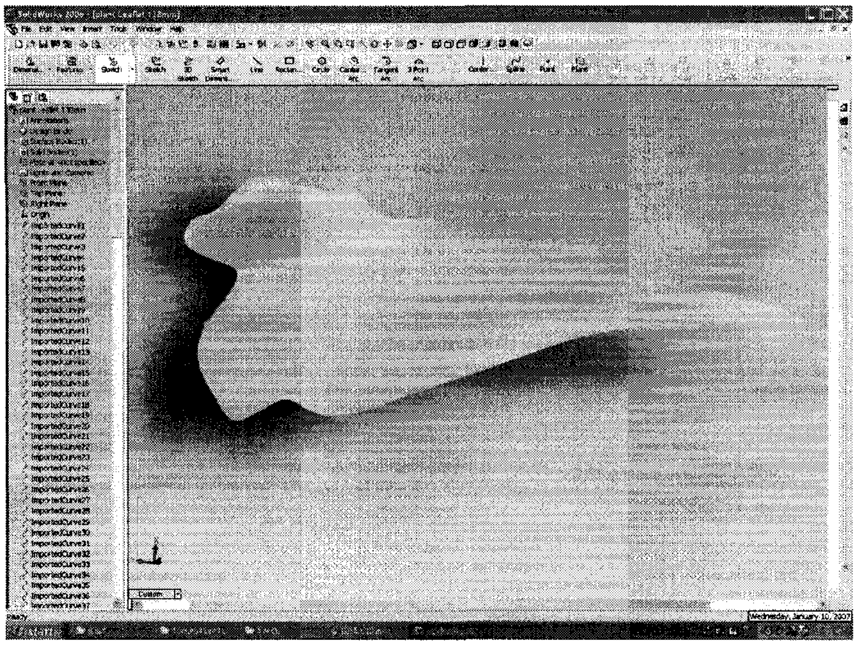


Fig. 3-7: leaflet 150

Components Leaf

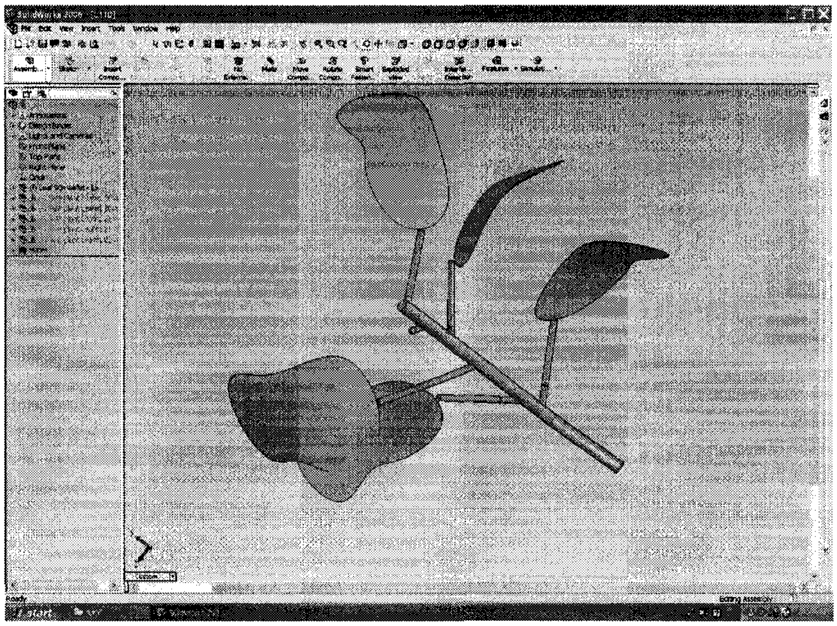


Fig. 3-8: Leaf 110

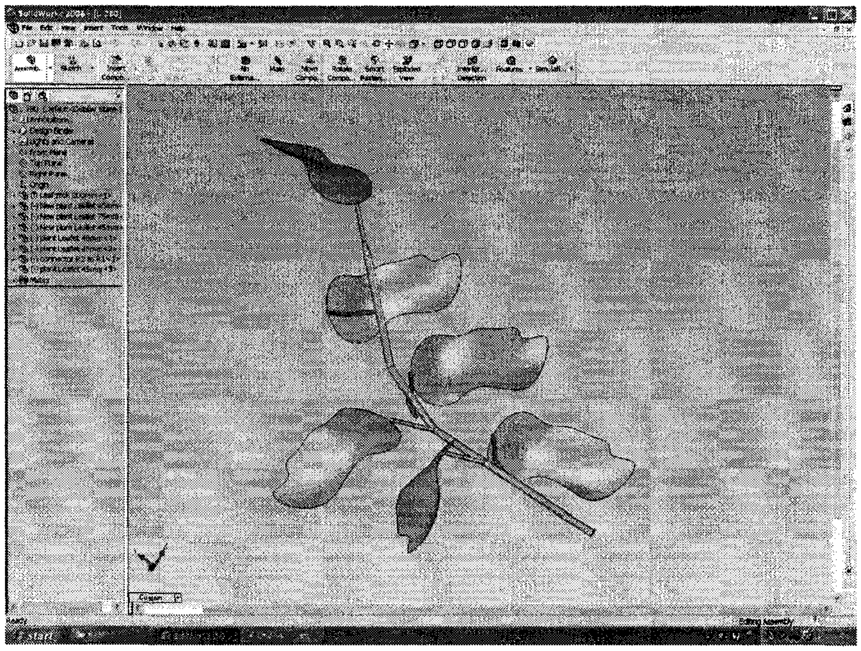


Fig. 3-9: Leaf 128

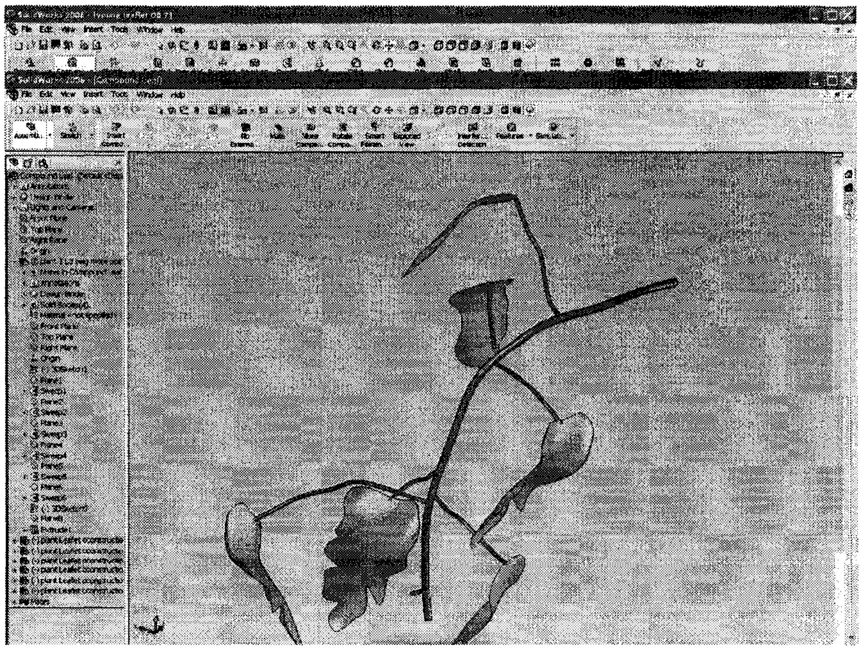


Fig. 3-10: Leaf 300

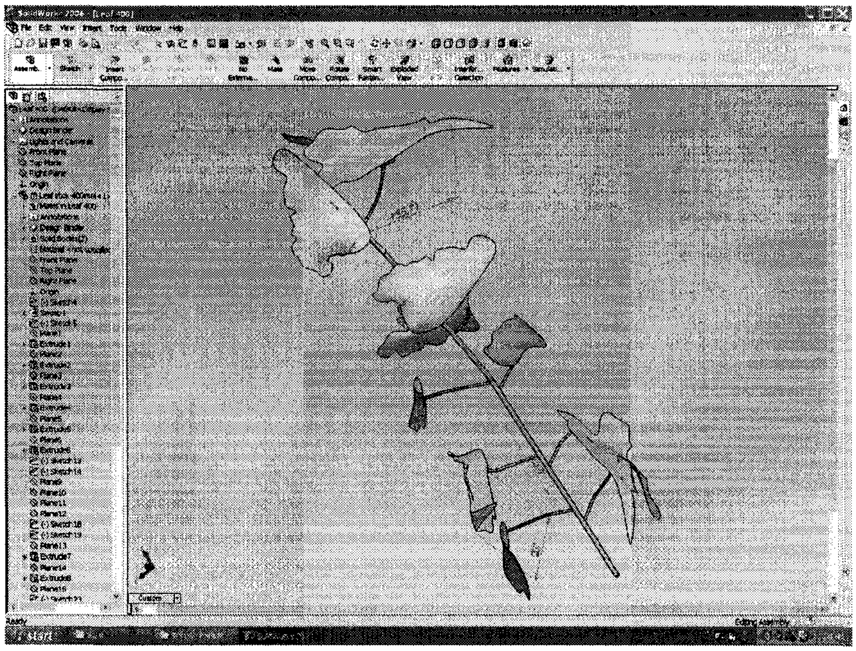

SolidWorks 2006 - [Leaf 400]
File Edit View Insert Tools Window Help

Fig. 3-11: Leaf 400

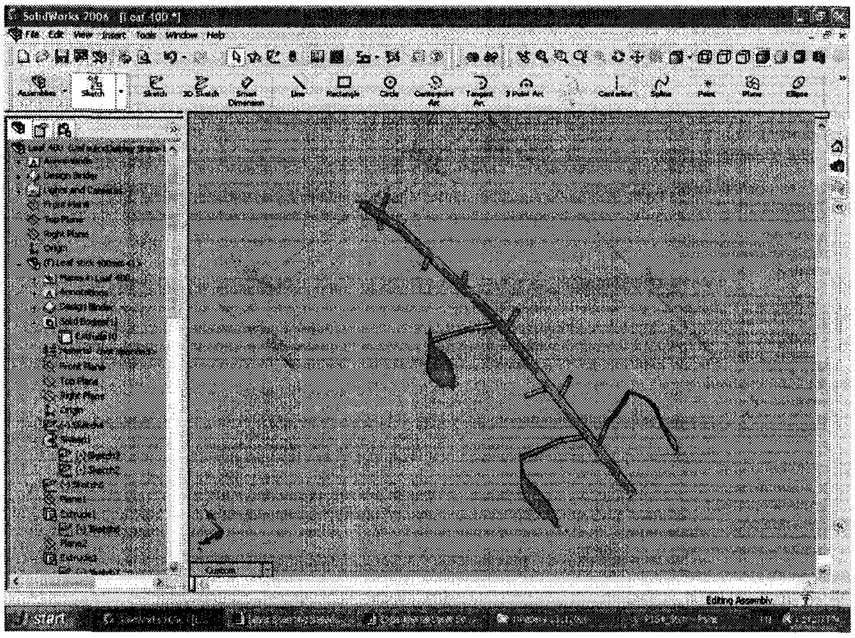


Fig. 3-12: Structure of Leaf 400

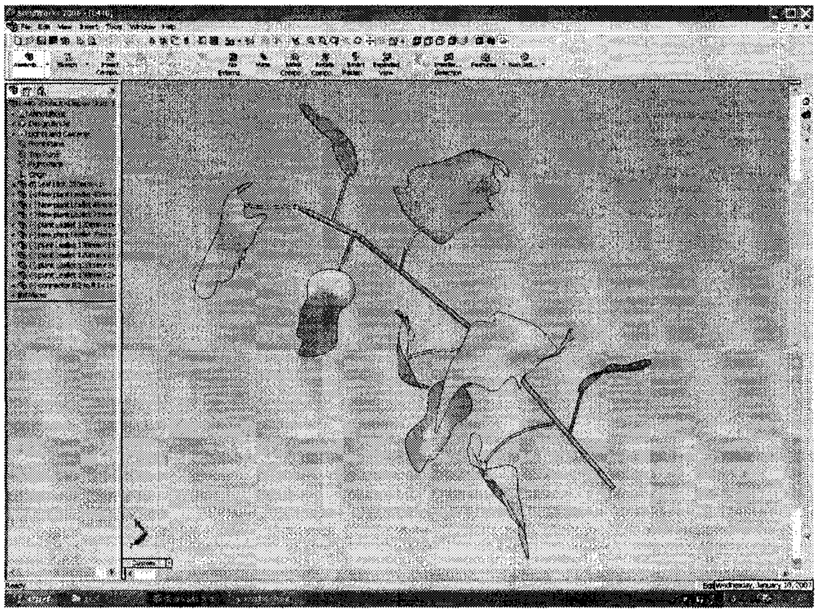


Fig. 3-13: Leaf 440

Truss

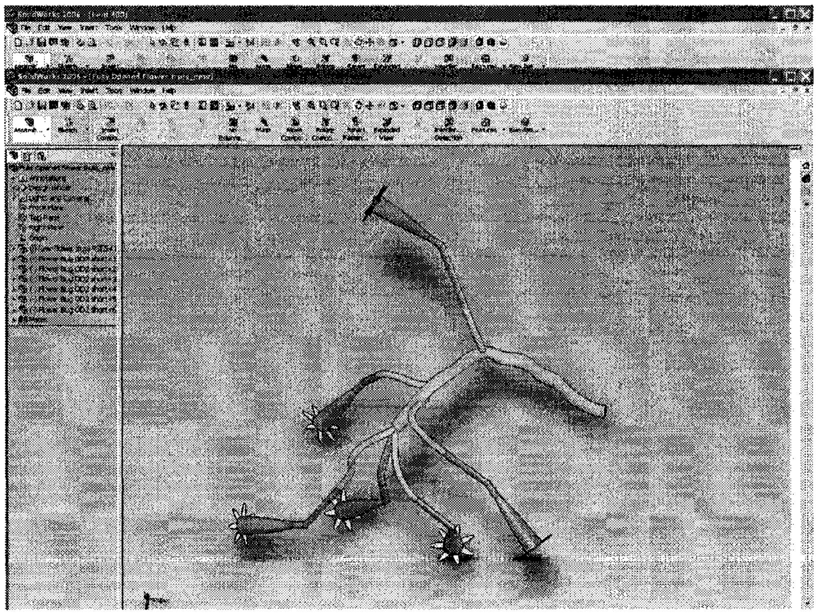


Fig. 3-14: Fully opened flower truss T1

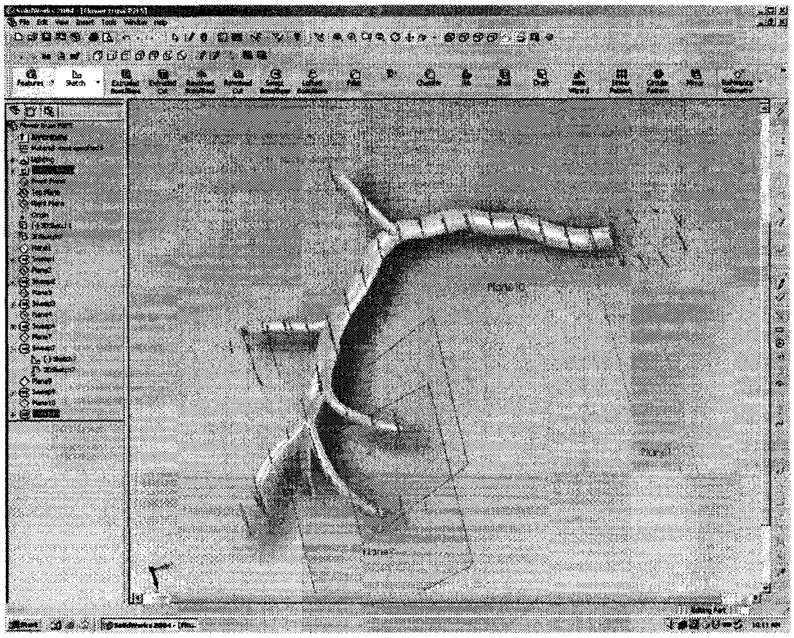


Fig. 3-15: Building truss T1 from scans

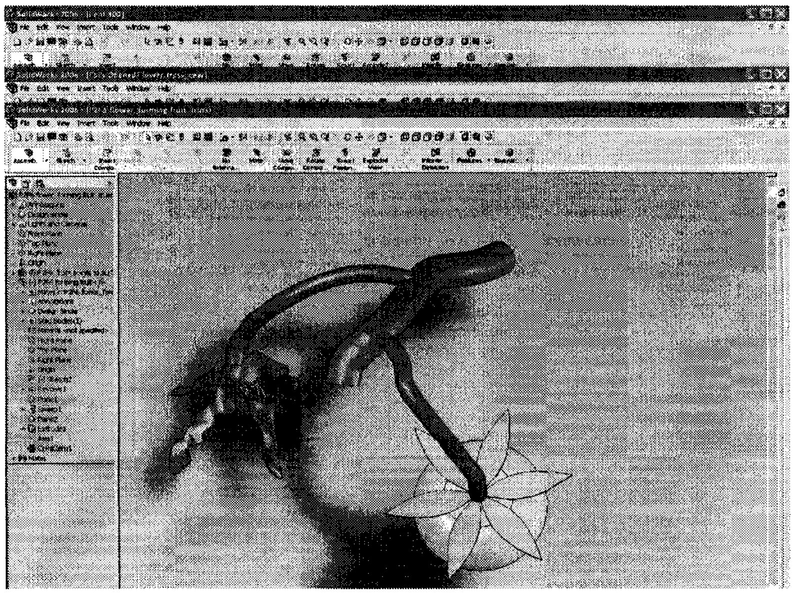


Fig. 3-16: Constructing a CAD model of a Forming fruit truss T2

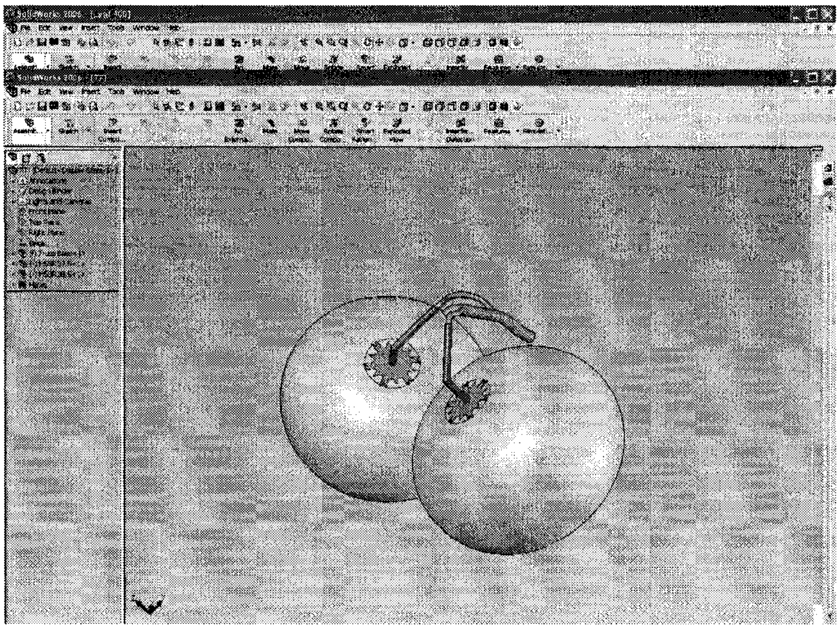


Fig. 3-17: T3

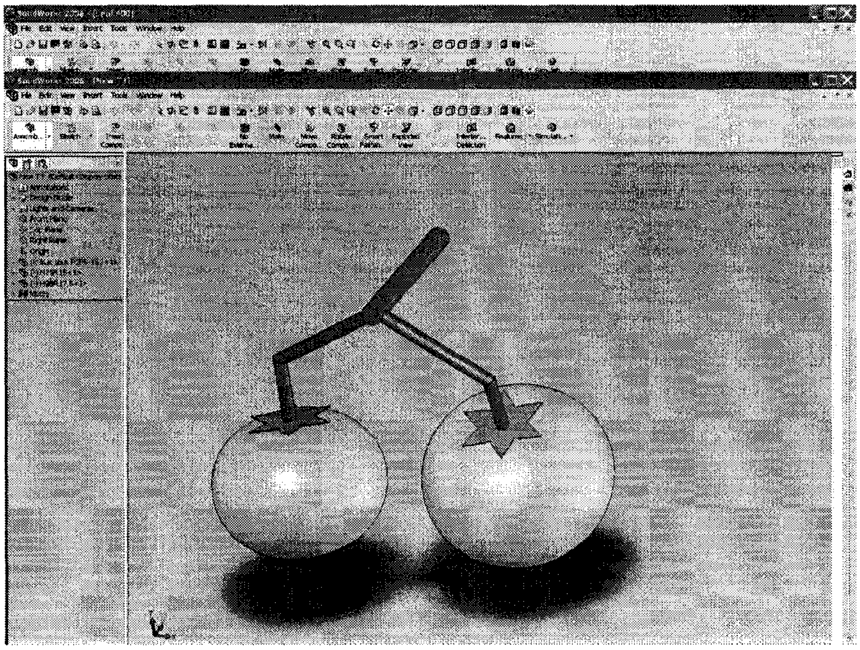


Fig. 3-18: T4

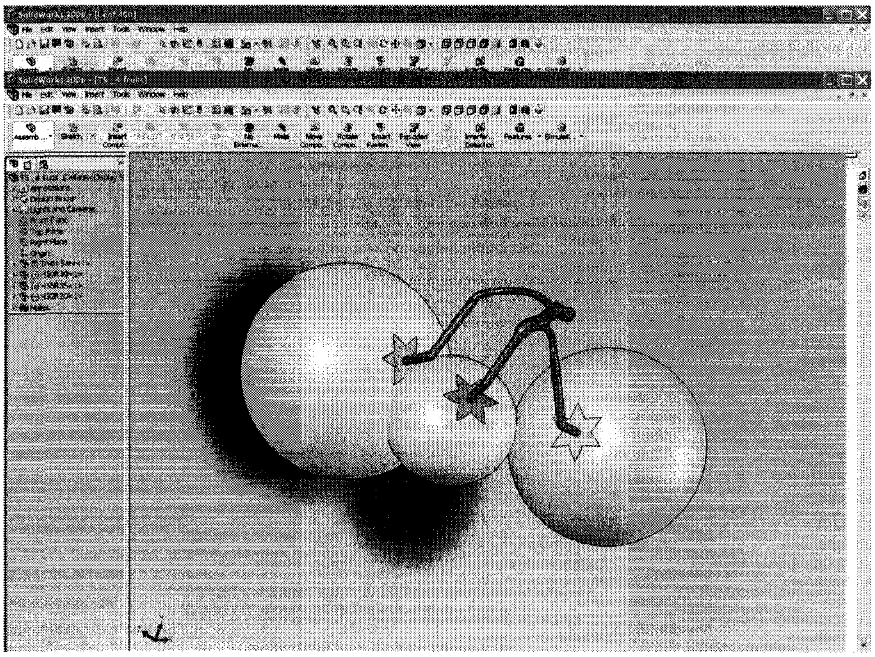


Fig. 3-19: T5

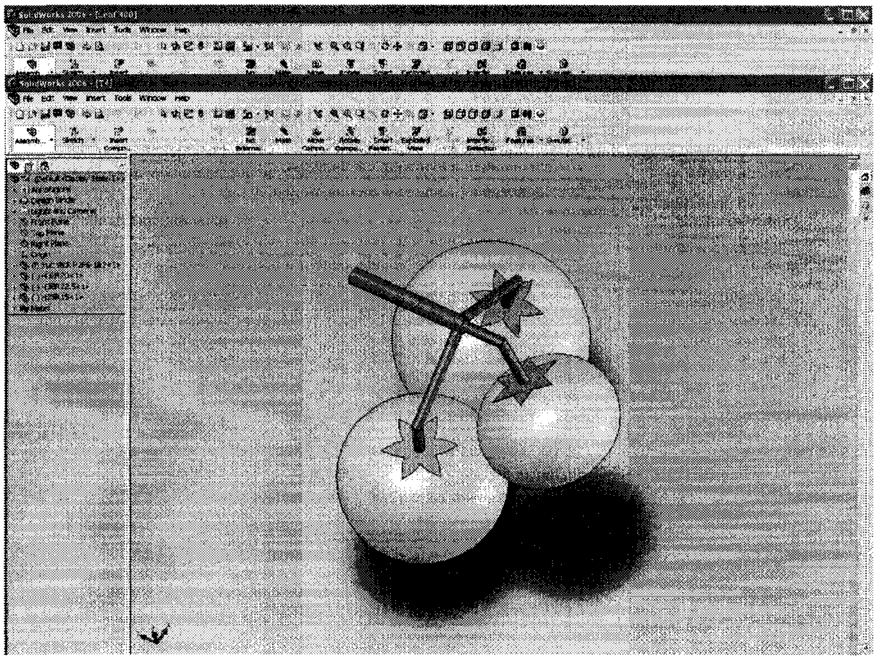


Fig. 3-20: T6

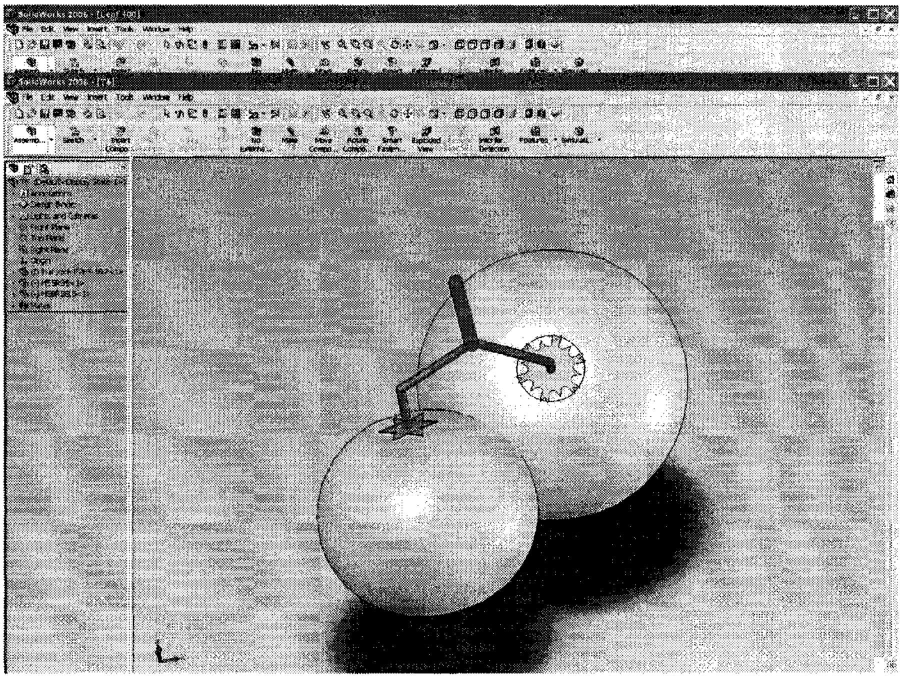


Fig. 3-21:   T7

Sucker

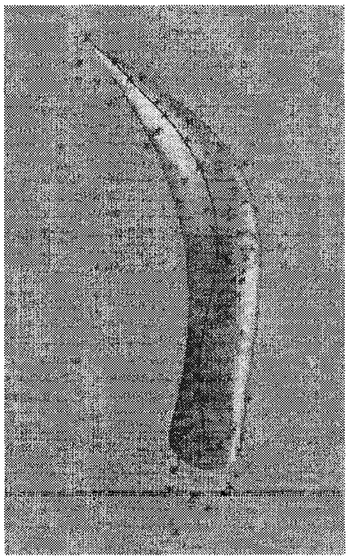


Fig. 3-22: Sucker

References

[1] E. Heuvelink, Evaluation of a Dynamic Simulation Model for Tomato Crop Growth and Development, Annals of Botany 83: 413-422, 1999

[2] Christian Gary, Modelling greenhouse crops: state of the art and perspectives, ISHS Acta Horticulturae 495: World Conference on Horticultural Research (WCHR), June 1998, Rome, Italy

[3] Tomonari Watanabe, Jim S. Hanan, Peter M. Room, Toshihiro Hasegawa, Hiroshi Nakagawa, and Wataru Takahashi, Rice Morphogenesis and Plant Architecture: Measurement, Specification and the Reconstruction of Structural Development by 3D Architectural Modelling, Annals of Botany 95, pp. 1131–1143, 2005

[4] University of Nebraska, Soil and Crop Modelling, http://csp.unl.edu/public/G_model.htm, last visited: 06 Nov 2006

[5] Heiner Lieth, Crop management model for decision support and automated optimization, Acta Horticulturae 507, pp.271-277

[6] Radomir Mech and Przemyslaw Prusinkiewicz, Visual Models of Plants Interacting with Their Environment, Computer Graphics Proceedings, Annual Conference Series, New Orleans, Louisiana, August, 1996, ACM SIGGRAPH, p. 397–410.

[7] Ding Wei-long, Xiong Fan-lun, Cheng Zhi-jun, Study and Implementation of the Expert System for Greenhouse Tomato Planting, International Technology and Applications, ICITA 2005, Volume 1, P.325-329

[8] Jonathan Wu, Automobile sensor system, January 2006, class notes, University of Windsor, Canada

[9] M. Cuerif, M. Launary, C.Duke, Remote sensing as a tool enabling the spatial use of crop models for diagnosis and prediction, p.1477~1479, 2000, IEEE, http://ieeexplore.ieee.org/iel5/6913/18598/00857245.pdf, last visited: 06 Nov 2006

[10] Emannele Trucco, Alessandro Verri, Introductory Techniques for 3-D computer vision, Prentice Hall, 1998

[11] Paul E. Debebec, Camillo J. Taylor, Modelling and Rendering Architecture from Photographs: a hybrid geometry- and image-based approach, Siggraph conference proceedings, 1996, p. 11—20

[12] Long Quan, Ping Tan, Gang Zeng, Lu Yuan Jingdong Wang, Sing Bing Kang, Image-based Plant Modelling, ACM Transactions on Graphics, Volume 25, Issue 3, Proc. of ACM SIGGRAPH, 2006, P. 599 - 604

[13] Gabriela Ochoa, An Introduction to Lindenmayer Systems, School of Cognitive and Computing Sciences, July 19, 2006, The University of Sussex

[14] M. T. Allen, P. Prusinkiewicz and T. M. DeJong, Using L-systems for modelling source–sink interactions, architecture and physiology of growing trees: the L-PEACH model, New Physiologist 166, pp. 869-880.

[15] Radoslaw Karwowski, Prizemyslaw Prusinkiewicz, The L-system-based plant-modelling environment L-studio 4.0, 4th International Workshop on Functional-Structural Plant Models, 7-11 June 2004, Montpellier, France, p. 403-406

[16] P.H.B. de Visser, L.F.M. Marcelis, E. Heuvelink, Functional-structural modelling of Chrysanthemum, 4th International Workshop on Functional-Structural Plant Models, 7-11, June 2004, Montpellier, France, p. 307-310

[17] Andreas Muhar, Three-dimensional Modelling and Visualisation of Vegetation For Landscape Simulation, Manuscript submitted for Landscape and Urban Planning, Special issue "Our Visual Landscape"; Ascona, Switzerland 1999

[18] Juha Hyyppa, Feasibility for estimation of single tree characteristics using laser scanner, p. 981~983, IEEE, 0-7803-6359, 2000, Hyyppa, J, Geoscience and Remote Sensing Symposium, 2000. Proceedings. IGARSS 2000. IEEE 2000 International Volume 3, P. 981 – 983

[19] Norbert Pfeifer, Ben Gorte and Daniel Winterhalder, Automatic Reconstruction of Single Trees from Terrestrial Laser Scanner Data, IAPRS Vol XXXV, Istanbul, Turkey, 2004

[20] Jim Hanan, Birgit Loch, Tim McAleer, Processing laser scanner plant data to extract structural information, 4th International Workshop on Functional-Structural Plant Models, 7-11 June 2004 –Montpellier, Franc, Edited by C. Godin et al., pp. 9-12

[21] S. Fleck, D. van der Zande, M.Schmidtc, P. Coppin, Reconstructions of tree structure from laser-scans and their use to predict physiological properties and processes in canopies, International Archives of Photo Grammetry, Remote Sensing and Spatial Information Sciences, Vol XXXVI – 8/W2.

[22] Frederic Boudon, Estimating the fractal dimension of plants using the two-surface method. An analysis based on 3D digitized tree foliage, 2006, Journal: Fractals, Volume 14

[23] Bo Zeng, Pruning Chinese trees: an experimental and modelling approach, Publication: 2001 Tekst. Proefschrift Universiteit Utrecht, NBC: 42.90: ecologie: algemeen, http://igitur-archive.library.uu.nl/dissertations/1977882/inhoud.htm, last visited: 06 Nov. 2006

[24] G. Salemi, F.Liberi, S.Mischi, Multi-resolution modelling from multiple range views: the laser scanner survey of porta portello, Padua, Italy, Computer. Graphics, Proceedings of CVPR'92, available at www.isprs.org/istanbul2004/comm5/papers/557.pdf, last visited: 06 Nov. 2006

[25] Sabry F. El-Hakim, J.-Angelo Beraldin, Michel Picard, and Guy Godin, Detailed 3D Reconstruction of Large-scale Heritage Sites with Intergrated Techniques, IEEE Computer Graphic and Applications, 2004, 0272-1716/04

[26] Vítor Sequeira, João G.M. Gonçalves, 3D Reality Modelling: Photo-Realistic 3D Models of Real World Scenes, Proceedings of the First International Symposium on 3D Data Processing Visualization and Transmission (3DPVT.02), 2002 IEEE

[27] Stephan, A.; Heinz, I.; Mettenleiter, M.; Härtl, F.; Fröhlich, C., Interactive Modelling of 3D-Environments, IEEE Int. Workshop on Robot and Human Interactive Communication, Sept. 2002, Berlin Germany

[28] Y. Chrysanthou, K. Cain, N. Silberman, F. Niccolucci (Editors), Gerhard Heinrich Bendels, Patrick Degener, Roland Wahl, Marcel Körtgen, and Reinhard Klein, Image-Based Registration of 3D-Range Data Using Feature Surface Elements, The 5th International Symposium on Virtual Reality, Archaeology and Cultural Heritage VAST (2004), Institute for Computer Science II – Computer Graphics, University of Bonn, Germany

[29] Puulo Diad, Vitor Sequeird, JoCo G.M Gonqulves, Frunsisco), Fusion of intensity and range data for improved 3d models, Image Processing, 2001. Proceedings. 2001 International Conference on, 2001, Volume: 3, pp. 1107-1110

[30] InnovMetric Software Inc., PolyWorks V9, May 2005

[31] F. Remondino, A. Guarnieri, A. Vettore, "3d modelling of close-range objects: photogrammetry or laser scanning?" Proceedings of SPIE-IS&T Electronic Imaging: Videometrics VIII, San Jose, California. Vol. 5665, 374, pp. 216–225. http://www.photogrammetry.ethz.ch/general/persons/fabio/fabio_laser_vs_photo.pdf, last visited: 06 Nov 2006

[32] FARO Technologies Inc., FARO scene 3.0 User's Manual, 2005

[33] Abmay T., Hartl F, Mettenieiter M., Realistic 3D Reconstruction – Combining Laser Scan Data with RGB Color Information, Zoller+Frohlich (Z+F) GmbH, Wangen, Germany, www.isprs.org/istanbul2004/comm5/papers/549.pdf, last visited: 06 Nov. 2006

[34] Remondino Fabio, From Point Cloud to Surface: The Modelling and Visualization Problem, International Archives of Photogrammetry, Remote Sensing and Spatial Information Science, Vol XXXIV-5/W 10, 2003

[35] GABRIELE Fangi, Faderica Fiori, Gianluca Gagliardini, Fast and Accurate Close Range 3D Modelling by Laser Scanning System, http://cipa.icomos.org/fileadmin/papers/potsdam/2001-21-gf01.pdf?PHPSESSID=b46638220d3b64083f011b1e8dbfc514, last visited: 06 Nov. 2006

[36] Soon-Yong Park and Murali Subbarao, Automatic 3D Model Reconstruction Using Voxel Coding and Pose Integration, p. 533-536, IEEE ICIP 2002.

[37] Minolta Vivid 900/910 3D laser scanner specifications, 2005

[38] Dragan Tubic, Patrick Hebert, Jean-Deschenes and Dennis Laurendeau, A unified Representation for Interactive 3D Modelling, Proceeding of the 2nd international symposium on 3D data processing, visualization, and transmission (3Dpvt '04), 2004

[39] Gregory M. Marr, Development of a Methodology for Creating Families of Parts, A Thesis Submitted to the Faculty of the Worcester Polyechnic Institute, 1996

[40] Remondino Fabio, From Point Cloud to Surface: The Modelling and Visualization Problem, International Archives of Photogrammetry, Remote Sensing and Spatial Information Science, Vol XXXIV-5/W 10, 2003

[41] Ontario, Ministry of agriculture and food, Growing Greenhouse vegetables, publication 371, 2005

[42] Matthieu Deveau, Nicolas Paparoditis, Marc Pierrot-Deseilligny, Xin Chen, Guillaum, Thibault, Strategy for the extraction of 3D architectural objects from laser and image data acquired from the same viewpoint, Proc. of the ISPRS Working Group V/4 Workshop, Mestre-Venice, Italy, August, 2005

[43] Mark Pauly, Markus Gross, and Leif P. Kobbelt. Efficient simplification of point-sampled surfaces. Proc. of the conference on Visualization '02, Boston, Massachusetts, Session P5, P.163 – 170

[44] Fangi, G. Fiori, F. Gagliardini, G. Malinverni, E. S. Fast and Accurate Close Range 3D Modelling by Laser Scanning System, International Archives of Photogrammetry Remote Sensing and Spatial Information Sciences, 2002, vol 34; part 5/c7, p. 196-203

[45] Michael Kazhdan, Shape machinng for model alignment, 3D scan and registration, ICCV 2995 short course, Johns Hopkins University, 2005

[46] S.V. Mikhlyaev, Relaxation Technique for Stereo Images Matching, Proceeding (372) Automation, Control, and Information Technology, 2002, Novosibirsk, Russia

[47] Wikipedia, http://en.wikipedia.org/wiki/3d_scanner

Vita Auctoris

Tong Wang was born in 1965 in Liaoning, China. He graduated from No. 1 High School of Fushun in 1984. From there she went on to the Wuhan University of Technology at Electrical Engineering Department from 1984 to 1988. He is currently a candidate for the Master's degree in Electrical engineering at the University of Windsor and hopes to graduate in Summer 2007.