MARVIN: high speed 3D imaging for seedling classification

N.J.J.P. Koenderink, M. Wigham, F. Golbach, G. Otten, R. Gerlich and H.J. van de Zedde
Wageningen UR – GreenVision, Bornsesteeg 59, 6708 PD Wageningen, the Netherlands; nicole.koenderink@wur.nl

Abstract

The next generation of automated sorting machines for seedlings demands 3D models of the plants to be made at high speed and with high accuracy. In our system the 3D plant model is created based on the information of 24 RGB cameras. Our contribution is an image acquisition technique based on volumetric intersection which is capable of the required order of speed and accuracy and which can easily be calibrated by non-expert operators. The use of 24 cameras leads to a non-trivial calibration procedure. A calibration procedure has been developed which is based on commercial single-calibration routines, offering robustness. This will be incorporated into the MARVIN-machine in such a way that a non-expert operator can easily calibrate the system with minimal intervention. In this paper we show a proof of principle for the fast and accurate image acquisition method.

Keywords: image acquisition, seedling inspection, multi-camera, calibration

Introduction

In horticulture, seedlings are sorted on their expected proficiency to effectively produce quality fruits or vegetables. This pre-selection of young plants on potential productivity significantly increases the total yield of the whole crop. At present, the quality assessment process is performed by highly trained experts. The experts use the morphological properties of the plants to assign them to a quality class. One of the reasons for researching the possibility of automated seedling inspection is the observation that experts are subjective in assessing quality. This is caused by the fact that experts perform their seedling inspection task based on explicit knowledge, tacit knowledge gained by experience and training, and heuristics. An example of the subjectivity of experts is as follows. If the quality inspection is performed by an expert, the result of the inspection may depend on previously inspected batches of seedlings. By replacing the human inspection with an automated inspection, the unwanted subjectivity is removed. However, the automation of the quality inspection task is complex due to the fact that seedlings are biological products with inherent variation between plants of the same quality class. Seedlings are assessed by a broad range of sorting criteria. They have to be inspected on simple criteria such as leaf area, stem length, and leaf curvature, but also on more complex issues concerning e.g. the likelihood that a plant is budless, and the regularity of the leaf shape. In the MARVIN-project we automate the quality assessment process taking all of these quality factors into account.
This paper describes the image acquisition process used in the MARVIN-project. The MARVIN method (Koenderink *et al.,* 2006) requires a 3D input model for each seedling. One of the major issues that had to be solved was to obtain a 3D input model that is *accurate* and *fast* enough for a commercial application of the computer vision system. The image acquisition method has to be able to cope with at least 10,000 seedlings per hour. It has to be able to recognize details in a resolution of 1 mm. We have developed a multi-camera solution based on the volumetric intersection technique that is described in the theory section. In the experimental section, we describe an experiment to estimate the speed and accuracy of our method. We then evaluate the results before ending the paper with conclusions.

Related work

At present, three types of automated seedling inspection systems are commercially available. One type takes a picture of a tray of very young seedlings to give an estimation of the average cotyledon area. This feature is used as a rough quality indication. A second system looks at individual seedlings. Two cameras – one above, one from the side – are used to determine the cotyledon area. The third existing machine can not only determine cotyledon area but also true leaf area. All these machines are hampered by two limitations. Firstly, although leaf area is an important feature in seedling inspection, it is far from being the only relevant feature. The inspection knowledge that is used by experts is more detailed than this. Secondly, these machines use a two-dimensional (2D) image of the seedlings. This implies imprecise measurements due to distortions introduced by the projection of the three-dimensional seedling onto a 2D image. A three-dimensional (3D) recording of the seedlings is essential to perform inspection with sufficient precision.

An enormous body of work exists on the subject of 3D reconstruction, which mostly falls in one of two domains. The first involves high quality scanning of objects, such as archaeological finds (Rocchini *et al.,* 2001). Here the time required is not a critical issue. The second focuses on real-time applications such as tele-immersion or augmented reality (Grau, 2004; Sakamoto *et al.,* 2005). Here quality is less important than interactive rates. Our application demands both high resolution and high speed. Further we must be able to use the model to accurately measure features of the plants. This means that inadequacies of the model cannot be masked using texture mapping or filtering – a model that merely *appears* realistic is not good enough. Finally, the cost and reliability of the equipment cannot be ignored.

Techniques for 3D modelling can be active or passive. Active techniques include the projection of structured light patterns onto the object. This is most often used for high-quality scanning, with a consequently high price tag. Low cost versions do exist (Rocchini *et al.,* 2001), but the required time for scanning is excessively long for our purposes. Zhang and Huang (2006) achieve good quality at interactive frame rates, but for only a partial model.

Passive techniques offer the advantage of requiring only ordinary cameras. Two major types are (1) those that use the differences between different images to build a depth map of the object, and (2) those that produce a volumetric reconstruction. Volumetric reconstruction is better suited to fine structures with occlusions (Eisert, 2005). The speed/quality trade-off remains a problem. Grau (2005), Kehl *et al.* (2005), Matusik *et al.* (2002) and Sakamoto *et al.* (2005) come reasonably close to the speed required by the sorting application. In each of these cases, the level of detail in the captured model is much less than that required for our application. Methods reported in literature indicate that higher quality models require significantly longer, ranging from tens of seconds to hours (Kutulakos and Seitz, 2000; Zeng *et al.,* 2006).

Regardless of the specific technique used, any 3D acquisition system using multiple cameras must be calibrated. Calibration of a single camera is a simple, widely implemented procedure. Calibration of multiple cameras is more difficult, in particular in our application as the necessary positioning of the cameras to capture all plant details precludes the possibility of them all being able to view the same calibration object. Papers do exist which offer solutions to this more difficult problem (Kurillo *et al.,* 2008; Svoboda *et al.,* 2005; Sebe and Chen, 2002). The technique described by Svoboda *et al.* (2005) is even available as a free MATLAB toolbox. However our application is intended for use in a commercial environment where time and manpower are limited. It is therefore undesirable to use third party software which does not offer full technical support or where redistribution is limited, and the calibration should ideally proceed fully automatically. These requirements cannot be fulfilled by either current commercial products or the algorithms described in the aforementioned papers.

3D-reconstruction of seedlings

For our purpose, we build upon the volumetric intersection method. With this technique, cameras can be placed at arbitrary positions around the object. In order to obtain a good view of the seedling under inspection, we use 20 to 30 cameras distributed in a half dome around the plant. In the related work discussed above, volumetric intersection is often used to create 3D models of large objects: people, houses, mature plants, statues or indoor scenes. Since the algorithm starts with a cube filled with voxels that are gradually carved away to reveal the recorded object, the amount of detail in the 3D object and the memory needed to contain the voxel cube are strongly related (Zeng *et al.,* 2006). The available memory limits the details that are recognizable in the model and the speed with which the model can be processed. A high level of accuracy limits the speed at which the algorithm can be performed.

Our research involves the creation of a 3D model of a seedling. High speed and high accuracy are both required. Since seedlings are at most 10×10×10 cm^3, the voxel cube containing the object is relatively small and hence we can ensure that the required accuracy is achieved. Therefore, the issue that needs to be resolved is to implement the 3D image acquisition method with sufficiently high speed. Moreover, the final application is to be used in an industrial setting, hence it must be user friendly and robust.

Volumetric intersection can be decomposed into three steps (Eisert, 2005): (1) calibration of cameras, (2) segmentation of the object, and (3) intersection of the viewing cones with silhouettes of the object. For the image acquisition of seedlings, a controlled environment with light cabinet is used. Therefore, the segmentation step is straightforward. Below, we discuss the calibration procedure and the cone intersection algorithm.

Calibration

The purpose of a camera calibration is to determine the mapping between the pixel coordinates of features in the camera image $[u,v]^T$ and the real world coordinates of the location of that feature $[X,Y,Z]^T$. This mapping is given by Equation 1.

$$s[u,v,1]^T = \mathbf{K}[\mathbf{R}|\mathbf{t}]\ [X,Y,Z]^T \quad (1)$$

Where s is a scaling factor, K is a matrix of the intrinsic camera parameters, and R and t are the rotation and translation of the camera.

Our calibration technique builds on the Halcon single camera calibration code (http://www.mvtec.com/halcon/overview/calibration.html). This code is widely used in commercial applications and is fully supported. The calibration routine uses a simple pattern, which must be placed in a number of different positions. For a straightforward positional calibration this pattern would have to be visible to all the cameras in all the positions, then R and t could be estimated relative to the same real world origin for all cameras. However this is not possible in our setup. Instead we place the pattern in a variety of different positions such that each camera has a sufficient number of good views of the calibration pattern, even though it cannot view the pattern in every position. Each camera's position is thus calibrated relative to a different origin. We use the overlap in camera views to calculate the projection between cameras and relate their positions to the same real-world origin. First, the calibration pattern is moved through a variety of positions j in the capture volume to ensure that each camera has a minimum number of good views of the pattern in different positions. Then for each position the algorithm determines which cameras had a good view of the pattern. Positions in which the pattern was visible to multiple cameras are then used as stepping-stones to move from camera to camera. At each step the projection between the two camera images is calculated, $proj_{i,i+1}$ (Figure 1).

For any given camera i there are multiple choices for the next camera i+1, and multiple choices for the calibration pattern position j to use in stepping to that camera. As a result there are multiple

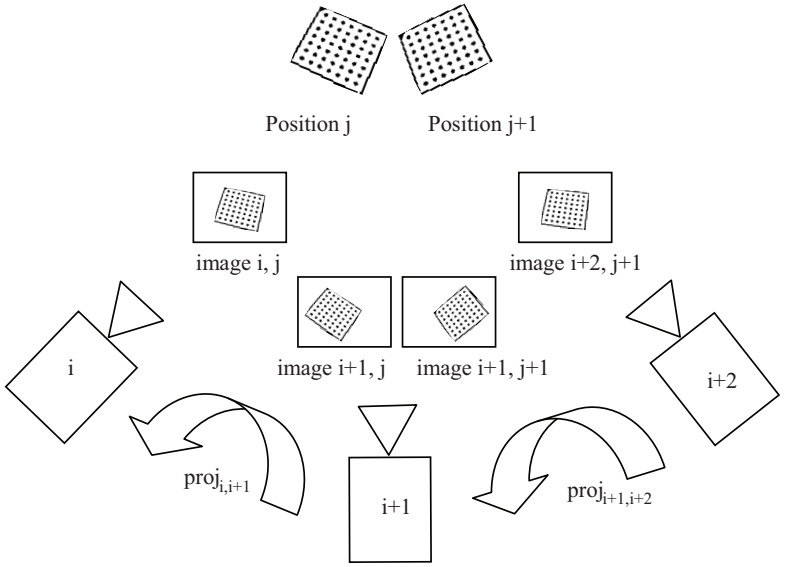

Figure 1. Cameras i and i+1 can both see position j. Cameras i+1 and i+2 can both see position j+1. Via the images of the calibration pattern (i,j) and (i+1, j) we can calculate the projection between cameras 1 and 2. The same can be done for cameras i+1 and i+2.

possible paths to each camera in the model. For each camera the calibration error is estimated (Equation 2).

$$error_i = \frac{\sqrt{e_{i,x}^2 + e_{i,y}^2}}{\left(\frac{Area_i}{BoundingBoxArea_i}\right)} \tag{2}$$

Where $error_i$ is the calibration error in the ith camera calibration relative to the local origin, $e_{i,x}^2$ and $e_{i,y}^2$ are the calibration errors in the x and y direction as calculated by the Halcon module, $Area_i$ is the area of the calibration pattern as viewed in the camera image, and $BoundingBoxArea_i$ is the area of the bounding box of the pattern in the camera image. Dividing the error by the ratio of the two areas increases the error in images where the pattern is viewed from an angle (giving a less reliable calibration). This error is used to calculate the total path error for each possible path (Equation 3). We can then choose the path with the lowest total error.

$$error_k = \sqrt{\sum_{i=1}^{n} error_i^2} \tag{3}$$

Where $error_k$ is the error for path k, $error_i$ is the error for camera i and n is the total number of cameras. Finally, the calculated projections along the path are multiplied together (Equation 4) and used to calculate R and t for each camera relative to the same real-world origin.

$$proj_{1,N} = \prod_{1}^{N} proj_{i,i+1} \tag{4}$$

where $proj_{1,N}$ is the projection between camera 1 and camera N. The intrinsic parameters K of each camera can be calculated via the single-camera calibration as normal.

Through the use of a turntable, the calibration pattern can be automatically moved through the required range of positions. As such the entire calibration procedure can be carried out without user intervention.

Intersection

With the 24 images from the calibrated cameras segmented, 24 silhouettes of the seedling under inspection are taken from different viewpoints. In order to ensure that the processing of these silhouettes takes place in an efficient way, we aim to quickly reduce the number of voxels in memory. Instead of processing the recorded silhouettes one by one, we process the datacube voxel by voxel. This process has proven to be a crucial step in the quick rendering of the 3D model. To quickly decide whether a voxel has to be kept or can be discarded, we determine wether the corresponding pixels in each of the camera images is part of the object or of the background (see Figure 2). If the pixel is background we discard the voxel directly and move on to the next one.

In more detail, let f_c be the focal point of camera c. Let i_c be the 2D image recorded by camera c. This image plane is defined by the normal $n = (x_n,y_n,z_n)$. Let (t,u) be the coordinate system of i_c. Let $p_{t,u}$ be a pixel in i_c. Let V be the voxel cube consisting of voxels $v_{i,j,k}$. For each voxel $v_{i,j,k}$ in V, the projection of this voxel onto each camera image i_c can be determined. This projection lies both on the line through f_c and $v_{i,j,k}$ and on the image plane i_c. In the first step of the intersection algorithm, the silhouette in the first image is processed in the voxel cube and all background voxels are carved away. This initial step results in an immediate reduction of the remaining voxels of approximately 80%. Then, for each remaining voxel, the value of the corresponding pixels in each of the camera images is intersected. When this intersection leads to a background value it is discarded. Hence, only the object voxels are kept as part of the 3D model. When each voxel has been processed, the solid 3D model has been created. Then, a second processing step takes place to ensure that only the boundary voxels are kept and the interior voxels are discarded. In this step, the number of neighbouring voxels is counted for each voxel in the 3D model. When a voxel has too few neighbours, it is considered as a boundary voxel. Otherwise, the voxel is an interior voxel and is discarded. The result is a 3D model of the object surface.

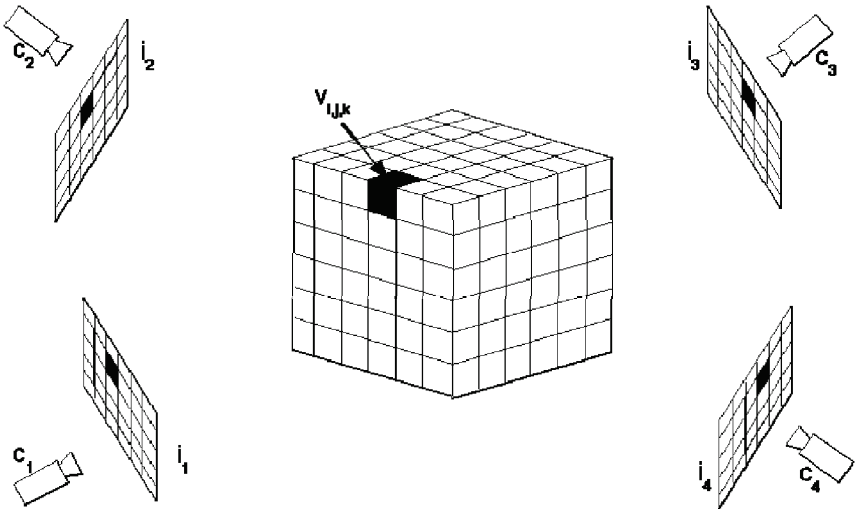

Figure 2. Schematic representation of camera viewpoints with respect to the voxel cube. The voxel $v_{i,j,k}$ can be seen from each of the camera's. The algorithm processes the datacube voxel by voxel.

Results

The acquisition and calibration methods have both been tested with a limited number of cameras. These tests have yielded the following preliminary results, which are promising.

Model acquisition

We have implemented the acquisition method described above and tested it firstly with a prototype environment consisting of a single camera which can be moved to multiple viewpoints to simulate a multi-camera environment, and secondly with a six-camera setup. For the first test, images of a watering can were captured from 24 different viewpoints. For the second test images of an artificial seedling were captured by the six cameras. Each camera image had a resolution of 743×552 pixels. The 3D models were processed in a voxel cube with 128 voxels on each side. In Figure 3, four of the seedling images are shown together with the created 3D model.

The 3D voxel model was constructed by first processing the silhouette of one camera (see Figure 4a), thereby discarding most of the background pixels. Next, for each voxel, its value as background voxel, interior voxel or boundary voxel was determined. The voxels were discarded or kept accordingly. This procedure results in the 3D model displayed in Figure 4b. In Figure 5, we have displayed only half of the watering can, since the hollow interior is more clearly visible for this object than for a seedling. This figure shows that the interior voxels of the object are discarded and only boundary voxels are kept.

Calibration

The calibration was tested in the prototype environment, with five cameras simulated. Acquisition of the calibration images took approximately five minutes, while the calibration algorithm itself required approximately thirty seconds. Further tests will be carried out once the full 24 camera test environment is constructed, as calibration accuracy cannot be adequately tested in the prototype environment, since errors in placing the camera in the different viewpoints could be wrongly attributed to calibration errors.

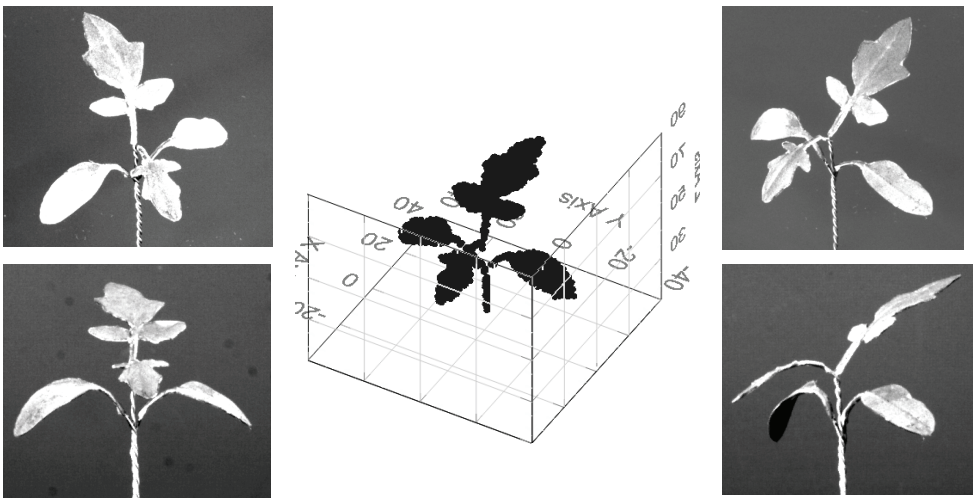

Figure 3. A 3D-model of a recorded artificial but realistic seedling model based on volumetric intersection with six cameras. Four of the camera images are displayed.

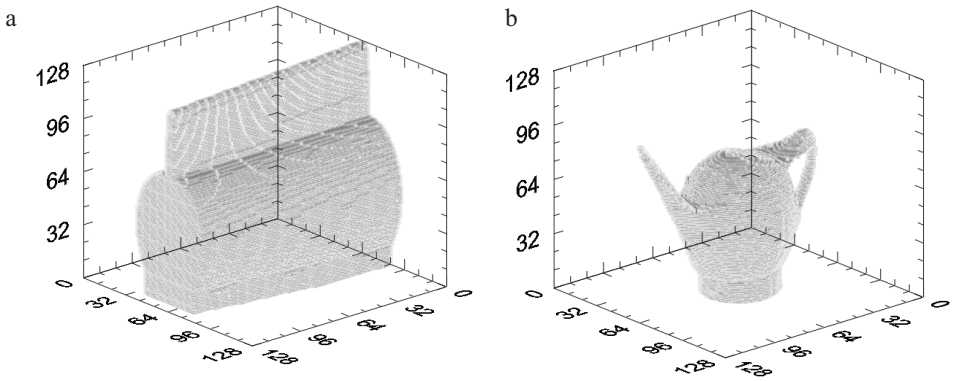

Figure 4. Illustration of the volumetric intersection process. In image (a) the first camera viewpoint has been processed, in image (b) the 3D representation of a watering can is depicted.

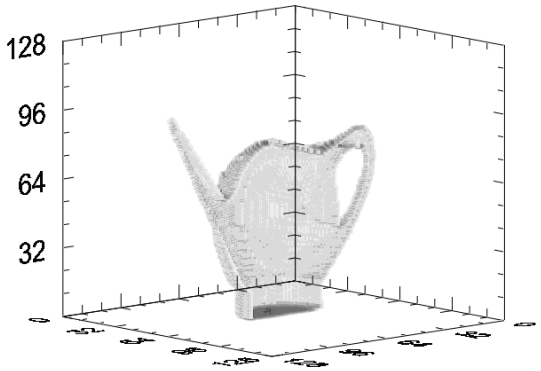

Figure 5. When we show only half the watering can, it is clearly visible that only the outer layer of voxels is part of the model.

Evaluation

For the acquisition of a seedling we use a volume of 10×10×10 cm^3. The resulting volume is divided up into a block of 128×128×128 voxels, giving a voxel size of 1.1×1.1×1.1 mm^3. It is possible to increase the voxel resolution to 256×256×256 without memory issues, which would improve the accuracy by a factor of 2. The 1 mm accuracy demanded by the application is therefore within the reach of our software. Processing a 128×128×128 voxel cube for the test seedling requires 20 ms on a 2.8 GHz Intel Q9550 processor. Doubling the resolution in each direction would theoretically increase the required time eight-fold to 160 ms. To attain the desired capacity of 10,000 plants per hour, each plant must be processed in 360 ms. The speed achieved therefore meets the system requirements.

The construction of the watering can model from the simulated 24-camera environment demonstrates that the approach also works for a larger number of cameras. The next step will be to construct the 24-camera environment for further testing of calibration and acquisition.

Conclusion

We have shown that our acquisition software is capable of constructing models with the required accuracy and speed for a commercial seedling classification machine. The calibration routine we have described is based on proven commercial code, and is automatable, which will contribute greatly to the user-friendliness of the machine.

References

Eisert, P., 2005. Reconstruction of volumetric 3D models, 3D Videocommunication. Wiley, pp. 133-150.

Grau, O., 2004. 3D sequence generation from multiple cameras, BBC R&D White Paper WHP 102.

Grau, O., 2005. A 3D production pipeline for special effects in TV and film, BBC R&D White Paper WHP 108.

Kehl, R., Bray, M., and Van Gool, L., 2005. Full Body Tracking from Multiple Views Using Stochastic Sampling, In: Proceedings of CVPR'05, 2, June 20-25, 2005. San Diego, CA, USA. pp. 129-136.

Koenderink, N.J.J.P., Top, J.L., and van Vliet, L.J., 2006. Supporting Knowledge-Intensive Inspection Tasks with Application Ontologies, International Journal of Human-Computer Studies, 64 (10): 974-983.

Kurillo, G., Zeyu, L., and Bajcsy, R., 2008. Wide-area external multi-camera calibration using vision graphs and virtual calibration object. In: Second ACM/IEEE International Conference on Distributed Smart Cameras, IEEE, September 7-11, 2008. Stanford, CA, USA. pp. 1-9.

Kutulakos, K. N. and Seitz, S.M., 2000. A Theory of Shape by Space Carving, International Journal of Computer Vision 38(3): 199-218.

Matusik, W., Buehler, C., McMillan, L., and Gortler, S., 2002. An Efficient Visual Hull Computation Algorithm, MIT LCS. Technical Memo 623.

Rocchini, C., Cignoni, P., Montani, C., Pingi, P., and Scopigno, R., 2001. A low cost 3D scanner based on structured light, Eurographics 20(3): 299-308.

Sakamoto, N., Kukimoto, N., Yasuhara, Y., Ebara, Y. and Koyamada, K., 2005. 3D Modelling and Displaying System for Volume Communication, JSME Int. J., 48(2): 247-251.

Sebe, I. O. and Chen, G.Q., 2002. Multi-camera Calibration. STMicroelectronics Technical Report, STMicroelectronics.

Svoboda, T., Martinec, D., and Pajdla, T., 2005. A convenient multi-camera self-calibration for virtual environments, Presence:Teleoperators and Virtual Environments 14(4): 407-422.

Zeng, G., Paris, S., Quan, L., and Sillion, F., 2006. Accurate and scalable surface representation and reconstruction from images, IEEE Transactions on Pattern Analysis and Machine Intelligence 29(1): 141-158.

Zhang, S. and Huang, P.S., 2006. High-resolution, real-time three-dimensional shape measurement. Optical Engineering 45(12): 123601.