

3D Pose Estimation of Cactus Leaves using an Active Shape Model

Thomas B. Moeslund, Michael Aagaard, Dennis Lerche
Laboratory of Computer Vision and Media Technology
Aalborg University, Denmark
E-mail: tbm@cvmt.dk

Abstract

Future Machine Vision applications have to cope with 3D non-rigid objects. One such application is 3D pose estimation of cactus leaves, which is the topic in this paper. We apply an Active Shape Model (ASM) to estimate the 2D pose of a leaf in one image. We post-process the ASM in order to find well-defined characteristics which are located in a second image using correlation. This process provides 3D points which are used to find the 3D pose of the leaf. Tests show that 84.6% of the 3D poses are found correctly.

1. Introduction

In the last decade Machine Vision has been widely used in industry, especially in quality control and to guide robots. Both application domains are usually build around conveyer belts transporting objects that are used in an assembly process or that need to be sorted with respect to quality, size or some other parameters. One thing that traditionally characterizes objects subjected to machine vision is that they are located on a 2D surface, making the vision problem 2D, or if in 3D the objects have a known rigid structure. The next generation of machine vision applications has to cope with non-rigid 3D objects as well. Such applications can e.g., be found in nursery gardens where automatic picking of fruits and leaves is highly desirable in order to reduce production cost and eliminate monotonic and wearing jobs.

At the company "Gartneriet PKM" [9] cactus plants of the type *Schlumbergera Bridesli* (also known as Christmas cactus, see figure 1) are planted, grown and sold. This is done by picking leaves from full-grown plants and re-planting these. The re-planted leaves are automatically watered, fertilized, and given the correct amount of light until they are saleable. Recently the re-planting process has been automated by the "pick and plant" system developed by the company Thoustrup & Overgaard [12]. It operates by detecting the positions and orientations of leaves located on a conveyer belt and guiding a robot to pick up the leaves and



Figure 1. A Christmas cactus. Left: input image. Right: Preprocessed image.

re-planting them. The only sub-process that is yet to be automated in the entire process is the picking of the leaves from the cactus. This paper investigates the possibility of doing so.

1.1. Related Work

Usually a pixel-based or feature-based method is applied in machine vision. But in the case of multiple complex non-rigid biological objects with potential overlap a model-based (also known as shape-based) method might be better as it can incorporate global information and shape variations. Shape-based non-rigid methods can either be free-form deformable templates or parametric deformable templates [7]. The best-known method within the former class is perhaps "snakes" [8] but others also exist. Common for the free-form deformable methods is that they contain little or no a priori knowledge. As the shapes of the leaves of a cactus are not arbitrary, see figure 2.A, parametric deformable templates might be a better choice for detecting cactus leaves. The parametric deformable templates, which are also known as active models, active contours, deformable models, and flexible models, can be divided into analytical form-based methods and prototypical-based methods. As seen in figure 2.A the leaves can have very different shapes and an analytical representation will therefore be very hard, if not impossible, to develop, hence

a prototypical-based method seems to be the correct choice for this application.

One very successful prototypical-based methods is the Active Shape Method (ASM) [3], which has shown its worth in many different application areas, e.g., [6][10][11]. ASM operates using two steps: training and online processing. In the former step a number of representative training images of the object is recorded and aligned. The statistical variation in the training data is found and a shape model of the object is build. In the online step different variations of the model shape is iteratively synthesized into the image for comparison until the best match is found, yielding the final representation of the object in the image. As other iterative methods the ASM requires an initial guess on the location and configuration of the object in the image.

1.2. The Content of This Paper

The approach in this paper is to estimate the 3D pose of a leaf by finding the 2D pose of the leaf using ASM. This is done in two stereo-images and the results are combined using triangulation in order to obtain the 3D pose. The contributions of the paper are investigations of 1) how precise the 3D pose of a leaf can be estimated, 2) how precise the initial guess needs to be for the ASM to produce a useful result. While the former is mostly relevant for the application at hand, the latter issue is rather general and can hopefully provide other researchers with knowledge on the limitations of the ASM. How to achieve an initial guess is not covered in this paper, but rather in [1].

The paper is structured as follows. In section 2 the training of the ASM is described. In section 3 the online part of the ASM is described. In section 4 the estimation of the 3D pose of the leaves is described. In section 5 the test results are presented and finally section 6 contains a conclusion.

2. Training the ASM

We represent a leaf by a 2D-model, which later can be mapped to 3D as a plane. This can be done as the leaves are almost planar and by doing so the task of collecting data will be simplified considerably.

Before describing the training, the notation used throughout the paper is defined. Data can exist either in the image domain or in the (ASM) model domain. For the former uppercase letters are used and lowercase letters for the latter. A model shape is denoted either \mathbf{x} or \mathbf{X} , an image shape \mathbf{y} or \mathbf{Y} , and a training shape \mathbf{z} or \mathbf{Z} .

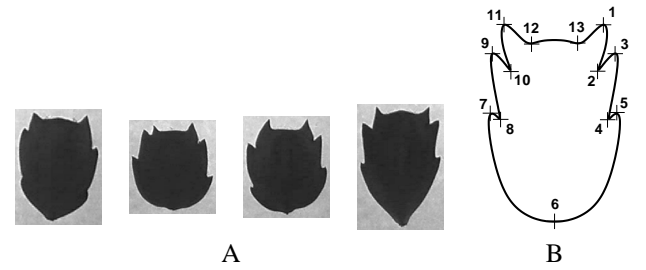


Figure 2. A: Four leaves. B: The distribution of landmarks.

2.1. Data Extraction

Images of 200 representative leaves were recorded and their contours semi-automatically segmented. We did this by manually defining a number of characteristic landmarks, see figure 2.B. Hereafter, we automatically found all intermediate points along the contour that are located between the landmarks. We used more intermediate points in the top of the leaf as more reliable information can be found here. In total 78 intermediate and 13 landmark points were used.

All the extracted points from one leaf are placed in a vector: $\mathbf{Z} = (x_1, x_2, \dots, x_n, y_1, y_2, \dots, y_n)^T$, where n is the number of points describing the shape, i.e., $n = 91$, and x and y are the pixel coordinates in the image domain.

2.2. Aligning Data

After the training shapes have been extracted it is necessary to align them before any statistics can be extracted. We apply the generalized Procrustes Analysis [5], which operates as follows. First all training shapes, \mathbf{Z}_i , are translated so their centers of gravity (CoG) are aligned. Then we take an arbitrary centered \mathbf{Z}_i and convert it into the model domain by scaling it to unit length. We consider this an estimate of the average model shape, $\bar{\mathbf{x}}$. The remaining centered training shapes, \mathbf{Z}_i , are now aligned to $\bar{\mathbf{x}}$ by scaling and rotating them using the analytical solution from [4]. This gives us the training shapes in the model domain, i.e., \mathbf{z}_i , and a new estimate of the average model shape can be calculated as $\bar{\mathbf{x}}_{\text{new}} = \frac{1}{n} \sum \mathbf{z}_i, i \in [1, 200]$. We now calculate how much the average model shape has changed between two iterations: $\epsilon = |\bar{\mathbf{x}}_{\text{new}} - \bar{\mathbf{x}}|^2$. If ϵ is smaller than a predefined constant we conclude that the algorithm has converged. Otherwise we align the centered training shapes, \mathbf{Z}_i , to $\bar{\mathbf{x}}_{\text{new}}$ and recalculate the average model shape and check for convergence, etc. See [1] for more details.

This iterative algorithm converges after only three iterations and the calculated average shape can be seen as the third column in figure 3. All training are were aligned to this mean and used in the calculation of the data statistic.

2.3. Calculating Data Statistic

The training shapes contain a considerable amount of redundant information since each point is highly dependent on the position of its neighboring points. We therefore apply a Principle Component Analysis (PCA) on the aligned data, \mathbf{z}_i , and only keep 14 (out of 182) components corresponding to 98.2% of the variance in the data set. A leaf shape is now represented by the model shape calculated as:

$$\mathbf{x} = \bar{\mathbf{x}} + \hat{\Phi} \cdot \mathbf{b} \quad (1)$$

where $\bar{\mathbf{x}}$ is the average shape of the aligned training data, $\hat{\Phi}$ contains the 14 chosen eigenvectors, and \mathbf{b} denotes the deformation parameters and explains the deviation from the average for each of the principal components.

To give an interpretation of the principal components we show in figure 3 the effect of the first four components by setting the contribution of all other components to zero, i.e., for the first component: $\mathbf{b} = [b_1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0]^T$.

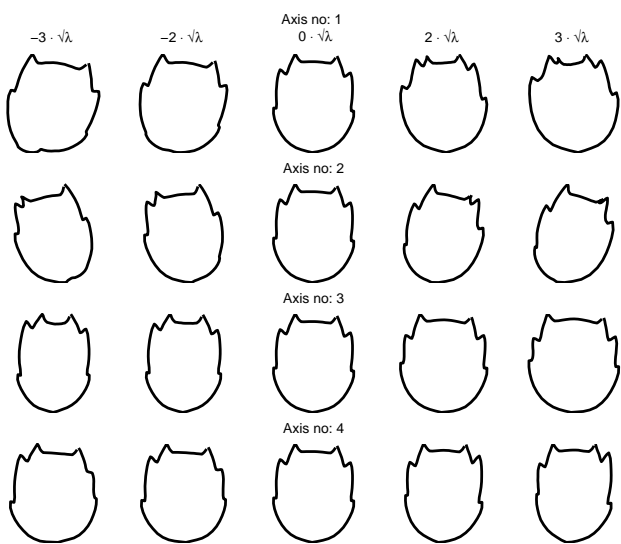


Figure 3. The effect of the four most significant components. λ is the eigenvalue. Note that all shapes in the middle column are identical as $\mathbf{b} = 0$.

Clearly some of the axes describe deformations that are intuitively comprehensible. For example, the third axis reveals something about the width of the leaves and something about the vertical position of landmark number 3 and 9, see figure 2. This is, however, not always the case and especially the less significant axes (not shown here) are often hard to interpret in a physical sense.

3. 2D Shape Estimation

After the ASM has been trained the online part can be applied to locate a leaf. This part is an iterative algorithm that takes an initial guess of a leaf in the image and then iteratively changes the pose parameters and the deformation parameters, \mathbf{b} , until the "true" parameters are found.

The algorithm consists of two steps. The first step is to calculate an image shape in the image. This is done by creating a model shape using the mean shape and the initial pose parameters. The mean shape is placed in the image according to the initial pose parameters and an image shape is calculated from it. The image shape is produced by examining the normals for each point in the model shape. The strongest edge along each normal determines the points in the image shape. In figure 4 this principle is illustrated for a simple geometric shape.

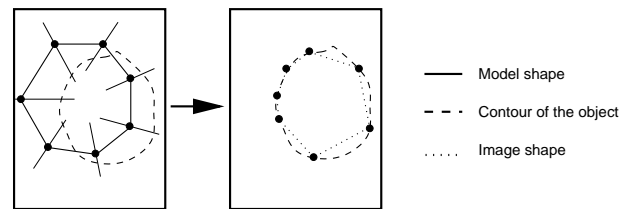


Figure 4. The image shape, \mathbf{Y} , calculated from a model shape, \mathbf{X} .

The second step in the iterative process uses the calculated image shape. This shape is used in an alignment scheme to first recalculate the pose parameters and then the deformation parameters. The new parameters are then used to produce a new model shape that serves as input for the second iteration, etc. The ASM will keep iterating until a convergence criterion is met or a maximum number of iterations has been reached. The pose parameters and deformation parameters for the last iteration determines the 2D pose and shape of the object.

3.1. The Algorithm

Given a certain deformation, \mathbf{b} , the model shape is given in equation 1. Applying the pose parameter, ξ , the model shape is transformed into the image as:

$$\mathbf{X} = T_{\xi}(\bar{\mathbf{x}} + \hat{\Phi} \mathbf{b}) \quad (2)$$

In this work we assume weak perspective transformation and can therefore model 3D shapes by an affine transform. So our pose parameters are, 2D translation in the image plane, rotation in the image plane, scaling in both directions in the image plane, and shear.

Given equation 2 the purpose of each step in the iterative algorithm is to find the pose parameters and the deformation parameters which best maps the mean model shape to the current image shape, or in mathematical terms:

$$\arg \min_{\mathbf{b}, \xi} |\mathbf{Y} - \mathbf{X}|^2 = \arg \min_{\mathbf{b}, \xi} |\mathbf{Y} - T_{\xi}(\bar{\mathbf{x}} + \hat{\Phi}\mathbf{b})|^2 \quad (3)$$

The algorithm which implements this minimization is shown below.

1. Initialize the deformation parameters, \mathbf{b} and \mathbf{b}_{old} to zero and the pose parameters, ξ , to the initial pose.
2. Generate the current model shape: $\mathbf{x} = \bar{\mathbf{x}} + \hat{\Phi}\mathbf{b}$
3. Map the model shape to the image: $\mathbf{X} = T_{\xi}(\mathbf{x})$
4. Calculate image shape \mathbf{Y} by searching along the normals of \mathbf{X}

This principle is shown in figure 4. To find a strong edge along the normal, the gradients of the intensity values are used. As the interior of a leaf can contain many edges depending on the lighting, we use the pre-processing approach suggested in [10]. The approach is to convert the color image into a greyscale image by using two times the green channel subtracted by the blue channel and setting all saturated values to 255. By using this approach the highlights in the leaves will be damped while the green plant material will be accentuated. The leaves get a smooth appearance with less internal intensity changes. The effect is illustrated in figure 1.

5. Find the pose parameters ξ which best align \mathbf{x} to \mathbf{Y} in a least square sense. We apply the analytical solution described in [4].
6. Transform the image shape into the model domain using the pose parameters found in the previous step: $\mathbf{y} = T_{\xi}^{-1}(\mathbf{Y})$
7. Find the deformation parameters which best explain the model data calculated in step 6: $\mathbf{b} = \hat{\Phi}^T(\mathbf{y} - \bar{\mathbf{x}})$
8. Adjust the deformation parameters, \mathbf{b} , by applying constraints.

If \mathbf{b} is allowed to take arbitrary values, then some very peculiar shapes can appear! The parameters therefore need to be constrained. We do this using the Mahalanobis distance:

$$D^2 = \mathbf{b}^T \mathbf{C}_b^{-1} \mathbf{b} = \sum_{i=1}^{14} \frac{b_i^2}{\lambda_i} \quad (4)$$

where \mathbf{C}_b is the covariance matrix of the deformation parameters calculated during training and λ_i is the i 'th

eigenvalue. If the calculated distance D is larger than a maximum distance D_{max} , then the b_i values are constrained by:

$$\mathbf{b}_{con} = \mathbf{b} \cdot \frac{D_{max}}{D} \quad (5)$$

and we update the deformation parameters as $\mathbf{b} = \mathbf{b}_{con}$. The threshold D_{max} is chosen by using the χ^2 distribution with 14 degrees of freedom, corresponding to the 14 eigenvectors.

9. Compute a convergence criterion: $C = |\mathbf{b}_{old} - \mathbf{b}|^2$
10. Set $\mathbf{b}_{old} = \mathbf{b}$
11. Check for convergence: $C <$ predefined error value, i.e., no significant changes in the deformation parameters are present. If the algorithm has not converged, then return to step 2.

A multi-resolution approach is applied in order to reduce the overall number of required iterations. We use a Gaussian image pyramid (to avoid aliased patterns) and start the algorithm at the coarsest level. This allows for a faster convergence of the translational pose parameters. For more information see [1].

4. 3D Shape Estimation

Having estimated the 2D shape of a leaf in one image (from the left camera) our initial idea was to use this 2D shape to define an initial guess in the other image (from the right camera). Then we would run the ASM in the other image and end up with two 2D shape models which could be used to calculate the 3D pose of the leaves. However, as we started to test the ASM we learned that the ASM in general has problems finding the fine details of the leaves. This means that no single landmark (see figure 2) is always found precisely at its correct position. We therefore decided to postprocess the image from the left camera in order to find some distinct features and then use template matching to find the corresponding features in the second image.

4.1. Calculating 3D Points

The only features that could be found reliably and did not change too much between image views were the landmarks: 1,3,9,11, see figure 2. These four spikes were therefore found by applying a corner algorithm [2] in regions around the position of the spikes found by the ASM. The templates are defined as 11x11 rectangles centered in the corners of the each spike. Large enough to capture the structure of the spike and small enough to avoid too much influence of the background. The search region was constrained by the use of distance constrained epipolar geometry.

When a corresponding spike can be found in the right image we use triangulation to calculate a 3D point for each spike. Before using the 3D points to calculate the 3D pose of the leaf we evaluate whether we have sufficient information to do so. First of all we need three or four spikes to be able to calculate 3D pose parameters. Secondly, we test whether the 3D points are co-linear. If so, the procedure used to estimate the 3D pose will be too sensitive to noise. We calculate the co-linearity by first spanning a 3D line between the two 3D points farthest apart and then calculate the perpendicular distance from the remaining point(s) to this line. If these distances are too small we ignore this leaf. The third evaluation is to calculate the internal distances between all 3D points. These values are compared to numbers obtained during training, where the topology of the leaves in general and the spikes in particular were investigated. If the topology of the current 3D points is too different from the leaves observed during training we ignore this leaf.

4.2. 3D Pose Estimation

Given the 3D points the task is now to find the 3D pose of the leaf with respect to a world coordinate system defined when calibrating the cameras.

From the 3D points a plane is estimated either analytically or by a least square method, depending on whether three or four points are present. The 3D position of the leaf is defined as the intersection of the estimated plane and a 3D line. The line is spanned by the optical center of the camera and the CoG of the model shape estimated in the left image and transformed into the image using the affine transformation in equation 2.

The three rotations of the leaf with respect to the world coordinate system is described as a 3x3 matrix containing three orthogonal vectors. The first vector is equal to the unit normal vector of the estimated 3D plane. The second is defined in the following way. We take the intermediate point in the estimated shape model which is the middle point between landmark number 12 and 13, see figure 2. As was done for the CoG, we map this point to a point on the plane, denoted P_{top} . The 3D CoG and P_{top} span a 3D line which is normalized and used as the second vector in the rotational matrix. The last vector is found by calculating the cross product between the two other vectors.

5. Results

As described in the Introduction two issues were addressed in this paper: the accuracy of the estimated 3D pose of the leaves and the required accuracy of the initial guess. In this section we evaluate these two issues.

5.1. Required Accuracy of the Initial Guess

The evaluation of the required accuracy of the initial guess was conducted by first - manually - finding the "correct" initial guess of a number of leaves on a number of different plants. Next we used the same leaves, but changing the initial guesses and visually inspected when the model was no longer aligned correctly. The results were not surprisingly found to be dependent on the application, for example the image resolution and the clutter around the investigated leaves. However, some general observations were made. First of all, the initial rotation has a large effect on the performance of the ASM. In fact, when the initial rotation is more than $\pm 10^\circ$ from the correct rotation, the ASM in general starts to produce erroneous results. Secondly, the initial scaling guess was in general found to have very little effect on the ASM. Thirdly, the initial translation (of the CoG) was in general starting to have an effect when it was more than ± 10 pixels from the correct CoG.

5.2. The Accuracy of the Estimated Pose

Given that the initial guess for the ASM is within the above limits, the next issue is how precise the pose of the leaves can be found. We did three tests to assess this matter, two best-case tests and a real plant test.

In the best-case tests a number of individual leaves were placed, one at a time, in a gripping arm that could be translated and rotated. We found the "correct" initial parameters manually and had the system estimate the pose of each leaf. These poses were then superimposed onto the two camera images for a qualitative test. All pose parameters were visually found to be correct.

In order to get quantitative results we did a repeatability test by varying the initial parameters with respect to the "correct" initial parameters of each leaf within $\pm 0.15 \text{ rad}$ (8.6°) for the rotation and ± 10 pixels in both directions for the translation. We did this in steps of 0.05 rad and 5 pixels, respectively, yielding 175 different combinations for each leaf. For each of these combinations we let our system estimate the 3D pose parameters and compare them to the "correct" pose parameters. The results were that 95% of all initial guesses resulted in 3D pose parameters that were within 3 mm and within 10° of the "correct" pose parameters. For the remaining 5% of the initial guesses up to twice these values was seen. It was observed that for virtually all of the remaining 5% only three spikes were found in each leaves and it should therefore be considered whether only to accept leaves where four spikes are detected. The drawback of this is that correctly estimated leaves will then be rejected.

The third test was conducted on images of different plants, see figure 1 for one example. Good initial guesses

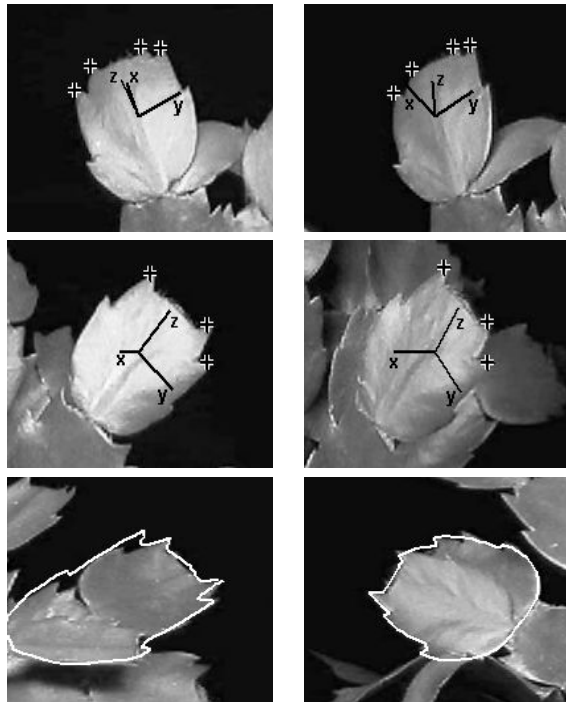


Figure 5. The top four figures show examples of the estimated pose parameters superimposed onto the images. The crosses illustrate the detected spikes. The two other figures show examples of incorrect shapes estimated by the ASM algorithm. Notice that two leaves are overlapping in the bottom right figure.

were found manually for a number of leaves in each image and provided to the system which calculated the 3D pose parameters of each leaf. Of these leaves 40.9% were rejected due to one of the constraints mentioned in the previous sections. Of the accepted leaves the pose parameters were superimposed onto the images, as done above, and visually inspected. It was found that the pose parameters of 84.6% of the leaves were correctly estimated, see figure 5 for examples. The primary reason for the 15.4% errors is that the ASM is aligned incorrectly, as for example seen in figure 5.

6. Discussion

We were surprised to see how sensitive the ASM method was with respect to rotation and that it was not capable of estimating all details (spikes) in the object. The former suggests that a good initial guess is required and the latter suggests that post-processing is a necessity if 3D pose data are to be extracted.

Whether the accuracy of our pose estimation is sufficient for the application of automatic picking of leaves, depends on the robot and its picking tool. However, we anticipate that the accuracy is sufficient.

What remains to be done, besides the robotic part, is a scheme to find the best leaf within the cameras' field-of-views. In general an image contains several leaves that potentially can be pose estimated by the system. However, it might be that some leaves cannot be pose estimated correctly (or at all!), but if just one leaf can be picked by the robot for each image then the system is successful (as the next image differs due to the picked leaf and/or because the plant or cameras are rotated to provide a new point-of-view). So a scheme is required which evaluates the different leaves against each other. This will probably be based on a combination of the different error and convergence measures used in the system, e.g., the error when fitting the 3D points to a plane.

References

- [1] M. Aagaard and D. Lerche. 3D Pose Estimation of Cactus Leaves using Active Shape Models. Technical report, Lab. of Computer Vision and Media Technology, Aalborg University, Denmark, 2004.
- [2] D. Chetverikov and Z. Szab. Detection of High Curvature Points in Planar Curves. In *Workshop of the Austrian Pattern Recognition Group*, pages 175–184, 1999.
- [3] T. Cootes and C. Taylor. Active shape models. In *British Machine Vision Conference*, 1992.
- [4] T. Cootes and C. Taylor. Statistical Models of Appearance for Computer Vision. Technical report, University of Manchester, October 2001.
- [5] I. Dryden and K. Mardia. *Statistical Shape Analysis*. John Wiley & Sons, 1998. ISBN 0-471-95816-6.
- [6] T. Hutton, P. Hammond, and J. Davenport. Active Shape Model for Customised Prosthesis Design. In *Joint European Conference on Artificial Intelligence in Medicine and Medical Decision Making*, 1999.
- [7] A. Jain, Y. Zhong, and S. Lakshmanan. Object Matching Using Deformable Templates. In *IEEE Transactions on pattern analysis and machine intelligence*, volume 18, pages 267–278. March 1996.
- [8] A. W. M. Kass and D. Terzopoulos. Snakes: Active contour model. *International Journal of Computer Vision*, 1:321–331, 1987.
- [9] G. PKM. <http://www.gartneriet-pkm.dk/>.
- [10] H. Sogaard and T. Heisel. Weed classification by active shape models. In *Automation and Emerging Technologies (AgEng'02)*, Budapest, Hungary, 2002.
- [11] H. Thodberg and A. Rosholm. Application of the Active shape model in a commercial medical device for bone densitometry. In *British Machine Vision Conference*, 2001.
- [12] Thoustrup and Overgaard. <http://www.tando.dk/>.