A Flexible 3D Vision System based on Structured Light for In-line Product Inspection

Øystein Skotheim[a], Jens Olav Nygaard[a], Jens Thielemann[a], Thor Vollset[b]

[a]SINTEF ICT, Høgskoleringen 5, N-7465 Trondheim, Norway
[b]Tordivel AS, Storgata 20, N-0184 Oslo, Norway

ABSTRACT

A flexible and highly configurable 3D vision system targeted for in-line product inspection is presented. The system includes a low cost 3D camera based on structured light and a set of flexible software tools that automate the measurement process. The specification of the measurement tasks is done in a first manual step. The user selects regions of the point cloud to analyze and specifies primitives to be characterized within these regions. After all the measurement tasks have been specified, measurements can be carried out on successive parts automatically and without supervision. As a test case, a measurement cell for inspection of a V-shaped car component has been developed. The car component consists of two steel tubes attached to a central hub. Each of the tubes has an additional bushing clamped to its end. A measurement is performed in a few seconds and results in an ordered point cloud with 1.2 million points. The software is configured to fit cylinders to each of the steel tubes as well as to the inside of the bushings of the car part. The size, position and orientation of the fitted cylinders allow us to measure and verify a series of dimensions specified on the CAD drawing of the component with sub-millimetre accuracy.

Keywords: Structured Light, Shape Measurement, 3D Analysis, Machine Vision, Automation

1. INTRODUCTION

The use of 3D geometric models has become an important tool in the design, development and production of machined and cast parts. Traditionally, coordinate measurement machines (CMMs) have been used to acquire the shape of objects, but these machines are generally slow, expensive and cumbersome to use.

In the recent years, advanced optical 3D digitizers based on *e.g.*, laser scanning, structured light and photogrammetry have emerged and they are slowly coming into use. Examples of manufactureres of such digitizers are GOM mbH, Steinbichler GmbH, Konica Minolta and Metronor. These systems have in common that they are relatively pricy and they are primarily targeted at very advanced end users, such as manufacturers of *e.g.*, cars and airplanes. One example of such a system is the ATOS III digitizer by GOM mbH.

The optical digitizers on the market today are mainly used for reverse engineering or first article inspection. They typically require the object to be pre-treated with white paint or white powder spray, and many require also that markers are attached to the surface under investigation. The digitization is normally carried out by specially trained personell. It is usually a quite tedious process which typically requires alignment and stitching of a series of scans from different angles.

Send correspondence to Ø. Skotheim (E-mail: oystein.skotheim@sintef.no, Telephone: +47 73597047)

After all the point data has been acquired, 3D modelling software systems such as CATIA, Raindrop Geomagic or PolyWorks by Innovmetric are typically used to process the data. The goal is often to convert a point cloud acquired from a 3D digitizer to a CAD model that consists of NURBS surfaces and other simple, geometric primitives. The process of creating a CAD model from a point cloud is usually time-consuming, and it requires manual intervention from the user to *e.g.*, remove noise from the point cloud, fill holes, repair artifacts in the mesh, divide the mesh into subregions and specify boundary conditions.

Another use of such 3D modelling software is to perform comparisons of an object that has been digitized with a pre-existing CAD model. This is also a process that currently consists of several steps and it requires a considerable amount of manual intervention from the user. Typically, the result of such a comparison is a colour-coded plot that is interpreted manually by an operator.

Our ambition has been to create a 3D vision system based on low-cost, off-the-shelf components that is able to acquire 3D point clouds of objects at regular intervals directly in the production line. The system should be applicable for inspection of 100% of the production. This means that the entire process of digitization and subsequent extraction of physical parameters from the point clouds needs to be carried out within a timeframe of very few seconds, and without intervention from an operator.

The idea is to divide the measurement into two phases, whereas the first manual phase consists of a specification of all the parameters that should be measured, and the second phase carries out the actual measurements. The specification of the measurements needs only to be performed once, typically when the system is configured for a new measurement task.

Our measurement tools have been implemented in Scorpion Vision Software® (hereby referred to as Scorpion), which is an independent and open framework for industrial machine vision developed by Tordivel AS. Scorpion has an intuitive user interface, and it contains a set of standard tools to perform typical measurement tasks. Scorpion has traditionally contained tools for machine vision in 2D, but has recently been extended with support for 3D point clouds and a set of 3D measurement tools, such as tools for fitting and characterizing geometric primitives (*e.g.*, cylinders, spheres and planes) and tools for finding relationships between said primitives, such as distances and angles. The tools for fitting and characterization of 3D primitives are based on the results presented in this paper.

Scorpion makes the configuration of the system an easy task, even for production engineers with no specific expertise within the area of image processing or 3D analysis. The configuration is performed by adding tools to a list of operations that is executed in sequential order. Regions-of-interests (ROIs) for the tools can be defined with a relative translation and rotation to each other. This means that, *e.g.*, once a characteristic feature on an object has been localized, subsequent ROIs can be specified with coordinates relative to this feature.

When Scorpion has been configured, it can perform repeated measurement tasks automatically for subsequent parts without intervention from the operator. The results of the measurements are continuously logged, and they can also be transferred via popular transport protocols like TCP/IP, RS-232 or Profibus for adaptive process control.

This paper is organized as follows. Section 2 gives a description of our measurement system along with a basic procedure for system calibration. In section 3, our algorithms for geometric modelling and fitting of primitives are presented. The application of our 3D vision system for inspection of a V-shaped car component is given in section 4. Section 5 discusses important topics for further work. Finally, a summary with conclusions follows in section 6.

2. MEASUREMENT SYSTEM

Our 3D camera targeted for automatic product inspection is based on structured light, and has been developed through a series of earlier projects related to product inspection.[1,2,3] Structured light was preferred because of its ability to capture full-field 3D images, its high accuracy and the relatively short measurement time.

We have developed an API for structured light that is very flexible with respect to the choice of hardware and the optical setup. Any commercial multimedia projector with VGA or DVI input can be used to project the structured light patterns, and a standard machine vision camera with *e.g.*, analog or IEEE1394 interface is used to capture the images. The API includes calibration algorithms. This allows the user to use an arbitrary configuration of a projector and a camera as an accurate 3D image acquisition device.

Our structured light system is based on a combination of Gray Code and Phase Shifting fringe projection[1] (hereby referred to as GCPS). The object is first illuminated with a series of shifted cosine patterns. A general N-step algorithm[4] is used to calculate the phase of the projected cosine fringes:

$$\phi(x,y) = \arctan\left(\frac{-\sum_{i=1}^{N} I_i(x,y)\sin[\frac{2\pi}{N}(i-1)]}{\sum_{i=1}^{N} I_i(x,y)\cos[\frac{2\pi}{N}(i-1)]}\right). \tag{1}$$

After the sequence of cosine patterns has been projected, a sequence of Gray code patterns is used to associate a time-encoded binary code with each individual stripe in the structured light pattern. The Gray code sequence is used to eliminate the need of phase unwrapping. For each pixel a Gray code word, $GC(x,y)$, is obtained. The resulting Gray code words, $GC(x,y)$, and the values obtained for the phase, $\phi(x,y)$, can be simply added together to form a continuous function that describes the absolute stripe displacement in each position in the field of view:

$$\delta(x,y) = GC(x,y) + \frac{1}{2\pi}\phi(x,y) \tag{2}$$

2.1 CALIBRATION

2.1.1 Z calibration

Efforts have been made by many researchers[5][6][7] to derive an exact mapping function between the measured stripe displacement, $\delta(x,y)$, and the object height, $Z(x,y)$, in each point in the camera image. These mapping functions quickly become complicated, and they involve system parameters that are difficult to measure, such as the exact position of the exit pupil of the projector as well as the entrance pupil of the camera.

We have observed that the mapping function can be approximated quite accurately by a lower order polynomial,[1]

$$Z(x,y) = c_0(x,y) + c_1(x,y)\delta(x,y) + c_2(x,y)\delta(x,y)^2 + \cdots + c_n(x,y)\delta(x,y)^n, \tag{3}$$

where $\delta(x,y)$ is the measured GCPS value and $c_n(x,y)$ are coefficient matrices that needs to be determined. Typically we have used a polynomial of order 4.

We can estimate a value for the coefficients by minimizing the least squares difference,

$$s(x,y) = \sum_{i=1}^{N} [Z_i(x,y) - h_i(x,y)]^2, \tag{4}$$

where $Z_i(x,y)$ is given by (3), and $h_i(x,y)$ is a set of known height distributions used to calibrate the system. Here, N must be greater or equal to the number of coefficients we want to use in our polynomial approximation (3). An analytical solution that minimizes $s(x,y)$ can be found by differentiating $s(x,y)$ with respect to each of the coefficients, $c_n(x,y)$, and by setting these expressions equal to zero. This results in a possibly overdetermined system of equations that can be solved by Singular Value Decomposition or similar techniques known from linear algebra.

One way of calibrating the system is to use a planar calibration object which is elevated to a set of N constant heights. The height distributions, $h_i(x,y)$, then becomes a set of constant values.

2.1.2 XY calibration

When the coefficient matrices have been determined, we are able to obtain a height map of the scene as seen by the camera. The height map is a floating-point image where each pixel value corresponds to a height above a chosen reference plane. The height map is also sometimes referred to as a 2.5D image of the object.

To be able to measure true 3D coordinates, we need a mapping between pixel indices, (x, y), and coordinates, (X, Y, Z), in millimeters. To obtain this mapping, we have used the excellent MATLAB Camera Calibration Toolbox by Jean-Yves Bouguet[8] to model our camera as a pinhole camera with radial and tangential distortion.

The model in this toolbox assumes three steps:

1. Rigid-body transformation (rotation + translation) from object coordinates to coordinates centered in the camera's principal point
2. Perspective projection
3. Distortion (radial and tangential)

The parameters used in step 1 of the model are called extrinsic parameters, while the parameters used in step 2 and 3 are called intrinsic parameters. There are a total of about 20 parameters in this model. To solve the equations for the unknowns, a planar calibration object with a chessboard pattern is placed in a series of random positions. A corner detection algorithm is applied to detect the sub-pixel position of the corners of the chessboard squares. The detected points are fitted to the model to obtain estimates for the parameters.

When all the parameters have been estimated, we can use the camera model to trace a ray from a pixel in the camera image to a point on the surface of the object, given by its Z value as obtained with the GCPS method. The result is an ordered point cloud of $N_x \times N_y$ (X, Y, Z) values, where $N_x \times N_y$ corresponds to the resolution of the camera.

3. DATA ANALYSIS

3.1 Modelling

Typical measuring tasks involve measuring properties of or between subparts of the object under inspection. Examples include angles between subparts, or the size of a particular subpart. For example, in the case of the model of figure 3, an important measure is the angle between the axes of the two cylindrical tubes on the V-stay. In order to accomplish this, we need to fit models to the acquired data as accurately as possible. In our case, we have focused on fitting subparts to geometric primitives like cylinders and spheres.

Our desire to automate the measurement process requires a split of the modelling task into two steps:

1. Assisting the user in setting up the measurement task with sufficient precision.
2. Performing the defined measurement in a robust fashion.

3.2 Defining the measurement task

Without using topological information from the whole model, we cannot in general detect and fit primitives or submodels to subparts of the point cloud if the model contains more than just one of these submodels. This makes it necessary to partition the point cloud into ROIs. In our application this can easily be done once for a model to be inspected, and simple rules for this partition can be applied to all subsequent scans and inspections in the production line. The manual ROI selection does not need to be very accurate. There are two

important considerations, first that ambiguities are removed, and second, that most points sampled from the actual submodel are retained in the ROI. The user therefore manually defines the ROI to use for the measurement.

Secondly, the user needs to specify the geometric primitive to find within this ROI. To avoid ambiguities and optimize later measurement speed, the user is also required to provide an initial guess of the whereabouts of this particular geometric primitive. The user is assisted in providing this guess through the use of a MLESAC-inspired algorithm,[9] a robust but rough optimization method. For well-chosen ROIs that really do eliminate ambiguities, the MLESAC-inspired algorithm will find the primitive without further user interaction.

In short, this method works by testing a number of (more or less random) hypotheses, keeping the one which best fits the available data. Measuring how well data fit is typically done by ignoring data points that clearly do not fit with the hypothesis, and calculating a match score based on the remaining points. In the current context, we make use of this algorithm to rapidly locate a geometric primitive in a ROI of a point cloud without any prior information.

This localization can be illustrated by how cylinder candidates are found. To generate a hypothesis, we generate two triples T_1 and T_2 of random points (see figure 1). We restrict the points in each triple to reside within a certain minimum and maximum distance of each other. These parameters depend on the scanning resolution, and the purpose is only to weed out bad hypotheses as early as possible. For each triple we consider the line passing through the mean of the three points, say, M_1 and M_2 respectively, perpendicular to the plane which is spanned by the triples, call these lines L_1 and L_2. There exists a new line A_1 that crosses both L_1 and L_2 at a distance r_1 from M_1 and M_2, as well as being perpendicular to L_1. Similarly, there exists a line A_2 that crosses both L_1 and L_2 at a distance r_2 from M_1 and M_2, as well as being perpendicular to L_2. As our hypothesized cylinder, we consider the cylinder of radius $r = (r_1 + r_2)/2$ with axis A through the points at a distance r from M_1 and M_2 along lines L_1 and L_2, respectively. Again, we speed up the process by discarding hypotheses with radii outside a predetermined interval specified a priori.

To test the hypothesis, we make use of a slightly modified version of the error objective that will be minimized during a subsequent optimization, namely the average squared distance from the points to the geometric surface,

$$\text{error}(P, A, r) = \frac{1}{N} \sum_{i=1}^{N} \left| \boldsymbol{p}_i - \hat{\boldsymbol{p}}_i \right|^2, \tag{5}$$

for the point (sub-)cloud $P = (\boldsymbol{p}_i)_{i=1}^N$, cylinder axis A and radius r, with $\hat{\boldsymbol{p}}_i$ being the projection of point $\boldsymbol{p}_i$ onto the cylinder surface. For the MLESAC hypothesis evaluation, we use

$$\text{MLESAC-error}(P, A, r, \varepsilon) = \frac{1}{N} \sum_{i=1}^{N} \min(\left| \boldsymbol{p}_i - \hat{\boldsymbol{p}}_i \right|, \varepsilon)^2,$$

to define the match score of hypothesis (A, r). Here, the idea is that a point's error is clamped to ε in order not to let adjacent primitives within the same ROI influence the score of the hypothesis too much. To achieve this, we select ε to be approximately the size of the predicted error in the sampling process generating the point cloud.

3.3 Optimal fitting

With a proper ROI, initial guess, and error-function (all described for the cylinder above), a standard minimization method like *e.g.*, conjugated gradients[10] or Nelder-Mead[11] has a good chance at converging on a (global) minimum. Note however, that it is very likely that the ROI still contains points that are not sampled from the geometry to which we attempt to fit the points. Such points may be sampled from another adjacent part of the geometric primitive or from some background, like a table or other fixture. They may also be artifacts, *e.g.*, produced by reflections. The consequence of this is a suboptimal fit, because the fitting procedure may try to accomodate these points along with the ones actually sampled from the primitive. Just as we took measures to eliminate these during MLESAC, we try to mask them out from the final fitting.

3.3.1 Decimation and re-fitting

We cannot simply minimize the error function (5), and then remove points further away than a given distance, for example the previously used distance, ε, since outliers may be sufficiently dominating so as to skew the minimum enough to make us discard the wrong set of points. Neither can we remove points far from the MLESAC-result, as this is just an inaccurate estimate of the global minimum, only suitable for an initial guess of a better optimization.

To overcome this, we use the following procedure. We iteratively fit and remove the worst outliers, a small fraction at a time. Doing this, we should get a very small average error when only the points sampled from the geometry are left. However, in the process of getting to this point, we will most likely have "lost" a lot of points which should really have been retained. (Points sampled from the geometry, being discarded early in the process when the geometry best fitting the point cloud is far from the final and correct solution.)

To remedy this, we only record the point in this iteration where the error *decline* drops markedly, and the solution at this stage (see figure 2). We find this point by intersecting two straight lines fitted by linear regression to the first and last 10% of the points on these curves. Now we assume that *this* solution will be close to the final solution. We then start over again with the full point cloud and this intermediate solution, and decimate the point cloud by discarding all points outside a layer around this solution. Finally, we do a refitting of the remaining points to the geometry primitive we seek.

4. APPLICATION

The test case for the project is a V shaped car component, with dimensions approximately 700x700mm. The component consists of two steel tubes attached to a central bushing with two additional bushings clamped to its ends (see figure 3). A series of dimensions given on the CAD drawing of the component should be inspected and verified. Examples of such dimensions are: the distance between the centers of the cylindrical bushings, the diameters of the tubes, the angle spanned by the two tubes and the angle of the bushing with respect to the baseplane of the item. The geometry of the component makes it impractical to perform this verification with only 2D inspection tools.

4.1 V-stay inspection cell

A measurement cell based on structured light (as described in section 2) was built to acquire the 3D shape of the V-stay. A Plus U5-632h multimedia projector was used together with a Sony XCD-SX910 machine vision camera with SXGA resolution (1280x960 pixels) and an IEEE1394 (FireWire) interface. The inclination of the camera and the projector was chosen carefully to provide a high number of data points on the inside of each of the bushings as well as on the surface of the steel tubes. The field of view for the measurement cell is around 1200x900mm, and the system was calibrated for a Z range from 0mm to 355mm.

4.2 Calibration of measurement cell

Our calibration object has been a 1200x900x6 mm glass plate. The glass plate was first painted matte white, and afterwards a chessboard pattern was applied onto the surface by silkscreen printing. The same calibration object was used both for the Z calibration and for the XY calibration.

We have also fabricated a set of 6×3 spacers with accurately known height. This allowed us to place the calibration plate in six different known positions, $z = \{h_1, h_2, h_3, h_4, h_5, h_6\}$.

During the calibration process, a GCPS measurement is performed on the glass plate in each of the six positions. Because reliable values for $\delta(x, y)$ can only be obtained for the white squares of the chessboard pattern, the result is interpolated by least-squares fitting to a sixth order polynomial surface in x and y (given by 28 coefficients). The interpolation also has the nice side effect of suppressing high-frequency noise in the measurements. The result of the interpolation is a set of six smooth distributions of GCPS values, $\delta_i(x, y)$.

Dimension	Nominal	1	2	3	4	5	6	7	8	μ	σ
Cylinder Ø	49.8	49.8	50.0	50.2	50.2	50.2	49.7	49.9	49.6	49.95	0.24
Sphere 1 Ø	50.01	50.2	50.4	50.2	50.5	50.5	50.1	50.7	49.9	50.31	0.26
Sphere 2 Ø	50.01	50.1	49.9	50.3	50.6	50.5	50.0	50.1	49.9	50.18	0.27
$C_{S1} - C_{S2}$	500.47	499.6	499.7	499.9	499.7	499.9	499.8	500.3	499.9	499.85	0.21

Table 1. Nominal and measured values for calibration test object

Estimates for a set of 5 coefficient matrices, $\{c_n(x,y) \mid n \in [0,4]\}$, are found by minimization of the equation given in (4).

4.3 Performing the measurements

In this particular setup, a set of 18 patterns was used for the GCPS measurements. This corresponds to 4 phase-shifted sine patterns, 6 Gray code patterns, 6 inverse Gray code patterns and uniform black and white patterns (for normalization and thresholding purposes). A measurement is performed in about 5 seconds and results in an ordered point cloud with 1280x960 (X, Y, Z, q, I) values, where (X, Y, Z) are coordinates in millimetres, q is a contrast value and I is the intensity obtained from uniform illumination of the object. The contrast value, q, is obtained by calculating the standard deviation of the perceived intensity in each pixel during the phase stepping process. That means that pixels that reside in the shadow can easily be eliminated by throwing away points with a contrast value smaller than a certain threshold.

4.4 Verification of measurement accuracy

To verify the accuracy of the calibration, we have fabricated a precise test object. The test object consists of a cylinder with accurately known radius. The cylinder has spheres attached on either side. Various dimensions for our verification object are given in table 1. These dimensions were obtained by a coordinate measurement machine (CMM) with a mechanical probe.

The test object was placed in eight random positions in the measurement volume, and a set of eight point clouds was obtained by our structured light system. In measurements 1-4, the test object was placed in various positions in a plane near $Z = 0$mm. In measurements 5-6, the test object was located near $Z = 155$mm. In measurements 7-8 one end of the test object was elevated around 200mm whilst the other end resided in $Z = 0$mm.

To eliminate possible influences of inaccuracies in our fitting algorithms, the commercial software package Geomagic was used to fit two spheres and one cylinder to manually selected ROIs for each of the eight measurements. The results of the measurements are given in table 1.

As can be seen, the parameters vary within a few tenths of a millimeter. The standard deviation is around 0.25mm for all the parameters, corresponding to around 1 part in 5000 of the field of view of the camera (1200x900mm). Since values do not have a mean value identical to the mechanically measured value, they all seem to have a certain bias. The spheres and the cylinders tend to have slightly larger radii in our measurements compared to the CMM results, while the distance between the spheres actually appear to be, in average, 0.6mm smaller than the values obtained with the CMM.

The dominant systematic errors in the system is believed to arise from inaccuracies during the system calibration. Our current calibration procedure requires the calibration plate to be located in six distinct, accurately known positions. Effects such as small displacements of the projector and/or camera relative to the calibration object, environmental vibrations, deflection of the plate because of gravity etc. are likely to introduce unacceptable inaccuracies. The fact that the field-of-view and the calibration plate for the V-stay inspection cell is very large (1200x900mm) makes it difficult to achieve a high absolute accuracy with the current calibration procedure.

We have also observed that inaccuracies are introduced by the interpolation of the GCPS values with a 6th order polynomial surface, especially when regularly spaced samples are used, and when periodical, systematic noise is

Dimension	C1	C2	C3	S1	S2	S3	S4	S5	μ	σ
R(Cyl1)	26.45	26.35	26.2	26.18	26.19	26.21	26.21	26.18	26.19	0.0165
R(Cyl2)	26.45	26.35	26.2	26.21	26.23	26.24	26.18	26.20	26.21	0.0218
Angle	56.74			56.48	56.47	56.45	56.47	56.45	56.47	0.0119

Table 2. Extraction of dimensions on the V-stay. C1-C3: Measurements performed with callipers on three different positions on the tube (C1: near central hub, C2: center of tube, C3: end of tube). S1-S5: fitting of a single cylinder to five different measurements of same V-stay.

present on the GCPS data.

We are currently working on new approaches for calibrating the system that have the potential to yield a higher degree of absolute accuracy. This is discussed further in section 5.

4.5 V-stay measurements

Three critical dimensions for the V-stay are the radii of each of the two tubes and the angle spanned by the tubes. To extract these parameters, we performed cylinder fits on each of the two tubes. A pair of 2D rectangular ROIs was defined in the xy-plane that roughly covered a set of approximately 12 thousand points each on the surface of the two arms of the V-stay.

The two cylinder fit minimizations were done with the conjugated gradient method, Polak-Ribiere variation,[10] used in the multipass framework described in section 3. The initial guess was supplied by the MLESAC-variation also described in section 3. In average around 11000 points were used as final input to each of the two cylinder fits, and these fits were performed in around 56 ms each on a stationary x86 desktop PC. For the multipass fitting, we used 50 passes. We briefly note that focus has not been on minimizing the speed of these fittings, but on the automation and robustness of the procedure. For speeding up the procedure both the number of passes can be reduced, and the point clouds can be decimated.

A series of five measurements was performed on one V-stay. The component was lifted up and lowered again onto a black cardboard background between each of the measurements. Three pieces of aluminium were used as indicators to guide the V-stay roughly into place. The expected displacement in the xy-direction was in the order of a few millimeters between each of the measurements. Definitions of ROIs were done on the first V-stay, and subsequent cylinder fits and angle computations were done automatically for successive V-stay measurements.

For our test measurements, approximately 5 seconds were spent on performing the image acquisition and about 3 seconds on the cylinder fits and extraction of measurement parameters per V-stay.

The results of a series of measurements is given in table 2. Note that the tubes are not perfect cylinders, and that the radii vary along the tubes. The angle displayed in the first column is the nominal value, taken from the CAD drawing of the component. The ROIs used for automatic measuring were placed around the outer half of the tubes.

As can be seen in table 2, the values obtained have a smaller standard deviation than the results in table 1. This is expected, because the V-stay was located in roughly the same position for all the five measurements, whereas our test object was located in a series of arbitrary positions and orientations within the entire measurement volume. Apart from small systematic errors due to miscalibration, variations present on the different parameters arise when slightly different subregions of the point clouds are used as ROIs for the fitting procedures.

5. FURTHER WORK

The results presented in this paper are preliminary results from an ongoing, three-year research project. Increasing the speed, accuracy and robustness of the system is an important challenge in our further work. As has been briefly discussed earlier, we are working with methods for calibrating our structured light system which does

not require exact positioning of the calibration object in the measurement volume, and hence have the potential to yield a higher degree of absolute accuracy. One approach that is being evaluated is to use the calibrated camera to estimate the orientation and pose of the calibration object (instead of using spacers with a precicely known height), similar to what is described by Vargas *et. al.*[12] Another approach is to model the projector as an "inverse camera" and use techniques known from stereo photography and photogrammetry to calibrate the system, such as described by *e.g.*, Brenner *et. al.*[13] and Zhang *et. al.*[14]

When it comes to the data analysis algorithms, we are focusing on the development of fitting functions for more geometric primitives, as well as more general geometries such as CAD models. Efficient use of CAD models for the specification of measurement tasks and for comparing measurements with its nominal values is also an important task.

The system is currently being trialled for random bin picking applications, where 3D point clouds of an unorganised crate full of items are being acquired at regular intervals. Algorithms are under development for detection and localization of known 3D objects in the crate.

6. SUMMARY AND CONCLUSIONS

We have presented our 3D vision system suitable for automatic in-line product inspection. The system is based on structured light with low cost, off-the-shelf projectors and cameras. Software tools have been implemented that allow the user to specify and carry out measurements on 3D shape data, such as the fitting of primitives (planes, cylinders, spheres etc.) to regions of point clouds, and extraction of parameters such as lengths, directions, angles and radii. The specification of measurements is done in a first manual step. Afterwards, successive measurements can be carried out automatically and without supervision.

Finally, our 3D vision system has been demonstrated for inspection of a V-shaped car part, consisting of steel tubes with cylindrical bushings. Critical dimensions, such as the diameters of the tubes and the angle that is spanned by the tubes, is extracted automatically from the point clouds by utilising a robust cylinder fitting algorithm.

The API for structured light shape acquisition, as well as tools for fitting primitives to subregions of point clouds, have been integrated into Scorpion Vision Software®, a commercial software package for industrial machine vision.

We believe that our system will bridge the gap between high-end 3D digitizers and complex CAD systems on one side, and flexible and user-friendly 2D machine vision tools on the other side, by enabling the use of 3D vision for product inspection directly in the production line. Eventually, efficient and automatic 3D inspection will become as pervasive as 2D machine vision solutions are today.

ACKNOWLEDGMENTS

This work has been supported by the Norwegian Research Council through the Auto3D project grant. We also wish to thank Kongsberg Automotive AS for their contribution to the funding of the project.

REFERENCES

1. Ø. Skotheim and F. Couweleers, "Structured light projection for accurate 3d shape determination," in *Advances in Experimental Mechanics*, C. Pappalettere, ed., *Proceedings of the 12th International Conference on Experimental Mechanics*, pp. 536–541, (Bari, Italy), 2004.
2. F. Couweleers, Ø. Skotheim, H. Schulerud, and K. Kaspersen, "In-line geometric fault detection in car parts based on structured light projection and image processing," in *Optical Measurement Systems for Industrial Inspection III*, W. Osten, M. Kujawinska, and K. Creath, eds., *Proc. SPIE* **5144**, pp. 697–706, (Munich, Germany), 2003.
3. F. Couweleers, Ø. Skotheim, and B. W. Tveiten, "Optical measurement of weld toe geometry with structured light," *Proceedings of 8th International Symposium on Measurement and Quality in Production*, (Erlangen, Germany), 2004.
4. C. Reich, R. Ritter, and J. Thesing, "3-d shape measurement of complex objects by combining photogrammetry and fringe projection," *Optical Engineering* **39**(1), pp. 224–231, 2000.
5. K. Gåsvik, *Optical Metrology*. John Wiley and Sons, third ed., 2002.
6. G. Sansoni, M. Carocci, and R. Rodella, "Three-dimensional vision based on a combination of Gray-code and phase-shifting light projection: analysis and compensation of systematic errors," *Applied Optics* **38**, pp. 6565–6573, 1999.
7. L. Salas, E. Luna, J. Salinas, V. Garcia, and M. Servin, "Profilometry by fringe projection," *Optical Engineering* **42**(11), 2003.
8. J.-Y. Bouguet, "Camera calibration toolbox for matlab." `http://www.vision.caltech.edu/bouguetj/calib_doc/`.
9. P. Torr and A. Zissermann, "MLESAC: a new robust estimator with application to estimating image geometry," *Computer Vision and Image Understanding* **78**(1), pp. 138–156, 2000.
10. S. A. Teukolsky, W. T. Vetterling, and B. P. Flannery, *Numerical Recipes in C*. W. H. Press, Cambridge, second ed., 1992.
11. K. I. M. McKinnon, "Convergence of the Nelder-Mead simplex method to a nonstationary point," *SIAM Journal on Optimization* **9**, pp. 148–158, 1999.
12. J. Vargas and J. A. Quiroga, "Flexible calibration procedure for fringe projection profilometry," *Optical Engineering* **46**(2), 2007.
13. C. Brenner, J. Böhm, and J. Guhring, "Photogrammetric calibration and accuracy evaluation of a cross-pattern stripe projector," in *SPIE Conference on Videometrics*, **3641**, pp. 164–172, (San Jose, California), Jan 1999.
14. S. Zhang and P. S. Huang, "Novel method for structured light system calibration," *Optical Engineering* **45**(8), 2006.

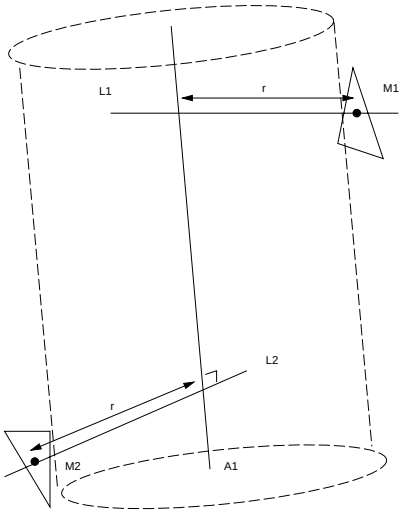

Figure 1. Cylinder MLESAC

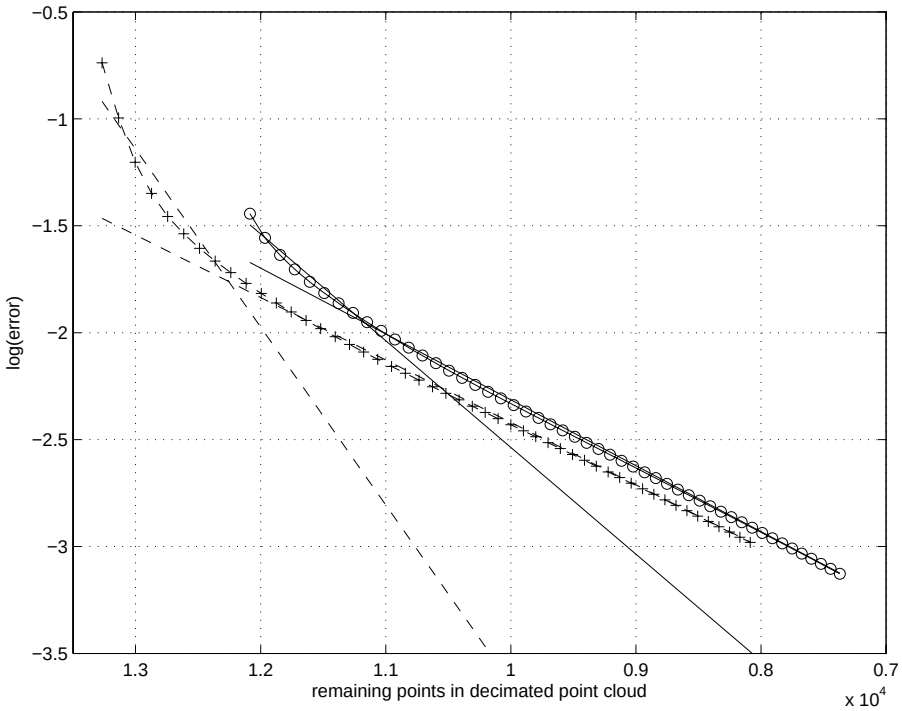

Figure 2. Plots of log(error) for the error (5) against the number of remaining points in an iteratively decimated point cloud for two different ROIs during the multipass cylinder fitting. The original point cloud is the same, but drawn solid and with 'o's we have a ROI with points mainly sampled from the geometry, and with dashed lines and '+'s we have a ROI with approximately 30-40% of the points belonging to the end bushing of the V-stay. The straight lines and corresponding intersections indicate chosen solutions for classification of points.

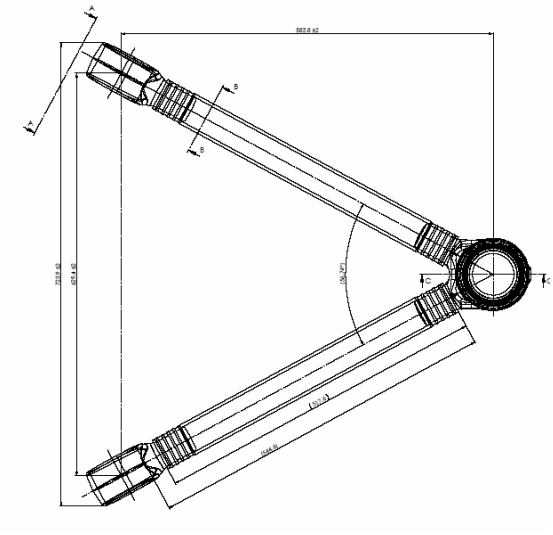

Figure 3. CAD drawing of V-stay

Figure 4. Measurement cell for inspection of V-stay

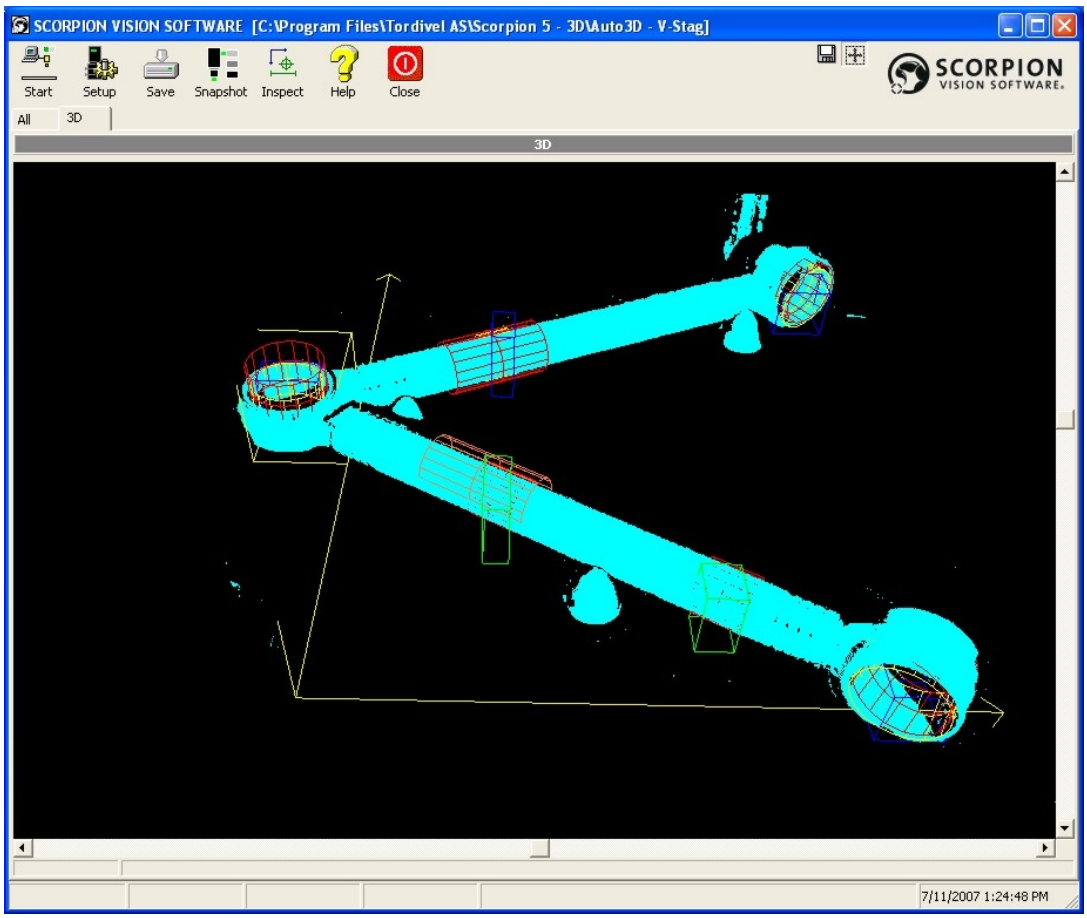

Figure 5. Screenshot from SCORPION. A point cloud of a V-stay is shown with 6 ROIs (shown as cuboids) and 6 cylinders fitted to the points within the ROIs.