

High performance visual servoing for controlled μ m-positioning

Citation for published version (APA):

Pieters, R. S., Jonker, P. P., & Nijmeijer, H. (2010). High performance visual servoing for controlled μ m-positioning. In *Proceedings of the 8th World Congress on Intelligent Control and Automation (WCICA 2010)*, 7-9 July 2010, Jinan, China (pp. 379-384). Institute of Electrical and Electronics Engineers.
<https://doi.org/10.1109/WCICA.2010.5554983>

DOI:

[10.1109/WCICA.2010.5554983](https://doi.org/10.1109/WCICA.2010.5554983)

Document status and date:

Published: 01/01/2010

Document Version:

Accepted manuscript including changes made at the peer-review stage

Please check the document version of this publication:

- A submitted manuscript is the version of the article upon submission and before peer-review. There can be important differences between the submitted version and the official published version of record. People interested in the research are advised to contact the author for the final version of the publication, or visit the DOI to the publisher's website.
- The final author version and the galley proof are versions of the publication after peer review.
- The final published version features the final layout of the paper including the volume, issue and page numbers.

[Link to publication](#)

General rights

Copyright and moral rights for the publications made accessible in the public portal are retained by the authors and/or other copyright owners and it is a condition of accessing publications that users recognise and abide by the legal requirements associated with these rights.

- Users may download and print one copy of any publication from the public portal for the purpose of private study or research.
- You may not further distribute the material or use it for any profit-making activity or commercial gain
- You may freely distribute the URL identifying the publication in the public portal.

If the publication is distributed under the terms of Article 25fa of the Dutch Copyright Act, indicated by the "Taverne" license above, please follow below link for the End User Agreement:

www.tue.nl/taverne

Take down policy

If you believe that this document breaches copyright please contact us at:

openaccess@tue.nl

providing details and we will investigate your claim.

High Performance Visual Servoing for Controlled μm -positioning*

Roel Pieters, Pieter Jonker and Henk Nijmeijer

Dynamics and Control Group, Department of Mechanical Engineering
Eindhoven University of Technology, PO Box 513, 5600 MB Eindhoven, The Netherlands
email:{r.s.pieters, p.p.jonker, h.nijmeijer}@tue.nl

Abstract—The research presented in this paper focuses on high performance visual servoing for controlled positioning on micrometer scale. Visual servoing means that the positioning of a x-y table is performed with a camera instead of encoders and one camera image and image processing should determine the position in the next iteration. With a frame rate of 1000 [fps], a maximum processing time of 1 millisecond is allowed for each image (132×45 pixels). The visual servoing task is performed on an OLED substrate, which can be found in displays, with a typical size of 80 by 220 μm . The repetitive pattern of the OLED substrate itself is used as an encoder and thus closing the control loop. We present the design choices made for the image processing algorithms and the experimental setup as well as their limitations.

Index Terms—Visual servoing, image processing, product as encoder.

I. INTRODUCTION

The ability of a robot to perform a task on a moving target has wide applications in industry. This ranges from welding cars on an assembly line to pick-and-place tasks within the food industry. Due to movement of the robot and the object, the position of an object cannot be precisely known. By means of a camera, the robot must track the object at every instance in order to perform its particular task. This controlled positioning is referred to as visual servoing, which is the fusion of many research areas such as image processing, kinematics, dynamics, control theory and real-time computing [2].

In literature many different visual servoing techniques are identified, a survey can be found in [6] and [2], a performance review in [4]. Many examples use the so called eye-in-hand approach and are restricted to video frame rates, i.e. not higher than 60 Hz ([3], [9]). More recently, much research is done towards speeding up image retrieval and -processing by FPGA-based methods. Watanabe et al. [20] and Toyoda et al. [19] both present an image processing system with massively parallel co-processor to reach 955-fps and 1000-fps respectively. Jorg et al. presents similar results at lower frame rates (50 Hz) with an FPGA [7]. On the other hand, more accurate robots are being developed by using complex

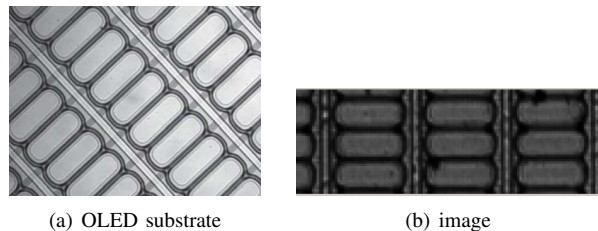


Fig. 1. OLED substrate. High resolution image of OLED substrate (a). The OLED images are rendered in greyscale with, in this case, nine OLEDs per image (b). The size of this image is 132×45 pixels ($W \times H$).

parallel robotic mechanisms which are combined with an integrated camera system for visual feedback ([15], [5]). A higher accuracy and robustness is also obtained by using learning methods. Mansard et al. [11] present a comparison of several Jacobian learning methods with respect to task sequencing. A state-of-the-art approach is presented by Sturm et al. where the robotic body scheme and kinematic model is learned through self-perception, [18].

A drawback is that not much attention (except for the learning methods) is given to measures of accuracy or reproducibility or examples with actual industrial applications. Furthermore, many methods mainly use massive parallel algorithms and rather basic image processing ([12], [17]). Therefore, the combination of intelligent image processing and high performance, high accurate visual servoing in real-time, *without* massively parallel processing and local joint control is rather new. We present in this paper a high performance (1 kHz), highly accurate ($< 10 \mu m$) visual servoing system which uses the repetitive pattern of the product as an encoder for controlled positioning. The product is in our case a substrate (see Fig. 1), consisting of a 2D repetitive pattern of OLED 'wells' or 'cups' which need to be filled (printed) with a polymer substance by an industrial inkjet printer. Each OLED has a typical size of $80 \times 220 \mu m$ that should be filled by a slightly larger droplet with an accuracy of maximum $10 \mu m$. The most limiting factor in this is the balance between a high sampling rate and the size of the image to extract useful information. An equilibrium is found for our OLED application characterized by positioning

*This research is supported by SenterNovem - IOP Precision Technology - Fast Focus On Structures (FFOS).

at micrometer scale and a ROI of 132×45 pixels, whereas 1 pixel equals $6 \mu m$. As the measurements are taken at 1 kHz, this results in a maximum effective computation time of 1 *ms*. The goal of this paper is to give insight into the design objectives regarding high performance visual servoing; in particular, using the product as encoder on micrometer scale. This paper is organized as follows: Section II gives a short overview of the requirements and bottlenecks for visual servoing and our application in particular. Section III and IV present the image processing algorithms and experimental setup respectively to achieve 1 kHz visual feedback. Finally, section V discusses the results obtained from experiments and compares these with former developed methods.

II. HIGH PERFORMANCE VISUAL SERVOING

Since the design choices are known and limited we can use a design space exploration approach to compare different platforms. A brief overview of requirements and their bottlenecks regarding visual servoing is given below. Following, the product and its usage as encoder is explained in further detail.

A. Requirements

When considering using visual servoing as method for controlled positioning, a design trade-off has to be made between flexibility, processing power and delay. This is set out in table I, where a few processors are compared to a number of properties.

Undoubtedly an FPGA will have the most processing power, however its complexity towards software implementation cannot be compared with a standard x86 processor. A main concern is that the platform (i.e. processor) should not only be used for this specific case (OLEDs), but that it is also available to other applications. This flexibility can be achieved relatively easily by a general purpose CPU with a real time operation system (RTLinux). Although the processing power in this is limited and the delay can be significant (i.e. hundreds of microseconds), software for different applications can be written and tested relatively easy.

TABLE I
PROCESSOR APPLICABILITY¹

	Nvidia Gforce	IBM Cell processor	FPGA	x86 +RTOS
Delay jitter	- -	- -	++	+
Processing power	++	+	+++	+/-
Processing power scalability	+++	++	+	+
Implementation effort	+/-	-	- -	++
Flexibility	+	-	- -	++
Future proof	+	-	-	++
Price	+	- -	-	++

¹Courtesy of TNO Science & Industry, P.O. Box 6235 5600 HE Eindhoven, The Netherlands

Besides the processor, lighting is also a component that has great influence on the performance of visual servoing. With frame rates as high as 1000 fps, 'normal' lighting no longer suffices since the exposure time of the image sensor is shorter than 1 millisecond. LED lighting is an obvious choice, whether it be high power LEDs or flashing. A drawback of flashing is that this can give synchronization issues which have to be solved separately.

B. Bottlenecks

The requirements stated in the previous section directly influence the bottlenecks of high performance visual servoing. One limitation is that sending data from camera to processing platform can be limited by the bandwidth of the communication protocol. For instance, GigE has a throughput of 116 MB/s while Cameralink Full can reach 765 MB/s. With a frame rate of 1000 fps, the images are easily limited to for instance 300×300 pixels when regarding GigE. A second major limitation is the limited data that can be processed within 1 millisecond if not using an FPGA for parallel processing.

The advantage in our application is that the structures that need to be found are relatively simple, small and repetitive. This makes basic operations and simple geometry on low resolution images possible and results in sub-pixel accurate measurements (1 pixel equals $6 \mu m$).

C. Using the product as encoder

In 'normal' motion control systems the motors are controlled using encoders which are placed inside the motor housing. In our application the repetitive structure of the product is used as encoder which closes the control loop solely on vision. The advantage of using the product as encoder is the fact that motion control is no longer reliable on an accurate (and costly) measurement system. The accuracy of the vision system determines the accuracy of motion control. Besides this, the measurement takes place exactly at the product which takes away the uncertainty of having a difference in measurement and actual position. Therefore, for this visual servoing method to be successful, the main issue is robustness, stability and repeatability of the image processing algorithm. Software should ensure a stable extraction of coordinates for positioning and a safe handling whenever coordinates are missing or out of bound.

D. OLED substrate

An OLED substrate is a small display which can be found in appliances such as cellular phones or digital watches. In a larger format these are usually found in tv's or computer displays. Each (in our case $15 \times 25 \text{ mm}$) display consists of a grid of OLED 'cups' which are stacked by groups of three OLEDs to display the three primary colors: Red, Green and Blue (RGB). Each cup needs to be filled with a polymer substance, which, when applied with a voltage,

emits one of the three colors. To this end, each cup center needs to be found and a nozzle needs to be positioned above it with an accuracy $< 10 \mu m$. Each cup has a typical size of $80 \times 220 \mu m$ and is spaced $80 \mu m$ between long sides and $220 \mu m$ between short sides.

Due to external influences, such as LED lighting (heat), manufacturing errors or vibrations of the setup due to motion, the OLED substrate can have a deformed shape. This leads to a grid of OLEDs which are not properly aligned any more, or OLEDs which are deformed themselves. These difficulties have to be incorporated in the design of a flexible image processing algorithm to detect the center of each OLED.

III. IMAGE PROCESSING

In this section an overview of the used image processing algorithms is given to achieve visual servoing by using the product as an encoder. This involves a calibration step for alignment of the substrate, a homing step to position to the upper left-most OLED and an on-line step where OLED coordinates are extracted from retrieved images at a rate of 1000 fps.

A. Calibration

In order to have a reliable center detection and an as precise as possible accuracy, the orientation of the substrate is calibrated and the upper-left most OLED should be centered in the field of view. This is explained in more detail in [14], however for clarity, we will recall the basic steps.

Depth of focus: A Laplacian sharpness-of-focus measure is used to determine if the depth-of-focus should be corrected [16]. When the sum of the Laplacian deviates too much from a given threshold, the height of the camera should be adjusted. This could be done manually or by means of a routine which searches for the highest focus measure.

Shading correction: The shading is estimated e.g. by morphological filtering where a smoothed version (Gaussian) of the input image is subtracted from the original input image. This smoothed version is the estimate of the background. The standard deviation (sigma) of the 9×9 Gaussian smoothing kernel is determined from the standard deviation of the input image.

Orientation calculation: The orientation of the complete OLED substrate / complete display should be determined with an accuracy of $\pm 1^\circ$ to ensure a roughly correct alignment with respect to the initial camera frame. The OLED orientation is determined by calculating the probabilistic Hough Transform (PHT, [8]) from the first derivative (Sobel) in x-direction of the input image. When the calculated orientation is larger than a predefined value (i.e. $|\theta| > 1^\circ$), the camera orientation should be adjusted or the OLED structure should be repositioned.

Homing: Prior to actual visual servoing and the processing of individual OLED cups, a homing procedure has to be performed to position the end effector (i.e. the camera) above the upper left-most OLED. The procedure for this is as follows: The OLED cups are separated from their surroundings with a Difference of Gaussians filter [10]. After thresholding and morphological operations, which are equal to the operations performed in the on-line step, the upper left-most OLED is found by the smallest euclidean distance between the origin of the image (i.e. upper left-most pixel, $p_0 = [0, 0]$) and each found OLED (i.e. the euclidean norm):

$$\begin{aligned} q(x, y)_{i+1} &= \min_{0 \leq n \leq n_t} \left(\sqrt{(p_n(x, y) - p_0)^2} \right) \\ &= \min_{0 \leq n \leq n_t} \left(\sqrt{p_{n,x}^2 + p_{n,y}^2} \right) \end{aligned} \quad (1)$$

with n the number of found OLEDs in iteration i , n_t the total number of found OLEDs in iteration i and $q(x, y)$ the target position for the table controller.

This algorithm is performed on an as large as possible image (VGA) with an accompanying frame rate (i.e. as high as the computations allow). From the found OLED centers, the target OLED is set as new position for the table controller. This routine is executed until

$$q(x, y)_i = \sqrt{(p_{c,x}^2 + p_{c,y}^2)} \quad (2)$$

with $p_{c,x}$ and $p_{c,y}$ the width and height of the input image respectively, meaning that the upper left-most OLED is in the center (p_c) of the retrieved image.

B. Center detection

1 kHz visual feedback means that each retrieved image has a maximum processing time of 1 millisecond. To reach the required sampling rate, the images are limited to 132×45 pixels, which contains nine OLEDs stacked 3×3 .

Otsu's Method: The main problem in detecting the OLEDs is separating the foreground (OLEDs) from the background. Since the images are in greyscale and two main intensity distributions can be distinguished, the solution is to find the most optimum threshold value. Otsu's method is an Optimum Global Thresholding technique which computes a global threshold value which is optimum in the sense that it maximizes the between-class variance, a well-known measure used in statistical discriminant analysis. In addition to its optimality, Otsu's method has the advantage that it is based entirely on computations performed on the histogram of an image, a 1-D array [13].

Mathematical morphology: The binary image has OLED cups as foreground pixels (i.e. having value '1') and the surroundings as background (i.e. having value '0'). From this point, two different methods for center calculation are employed. The first does boundary following and center-of-gravity calculation and is explained in detail in [14]. The second method is as follows:

From the binary image containing OLED blobs the boundary is followed and a bounding box is set around it. The two vertical edges of this bounding box are known and used to determine a pixel accurate x -position for each of the four points (see Fig. 2):

$$\begin{aligned} hx[1] &= box.x + 0.2 * box.width \\ hx[2] &= box.x + 0.8 * box.width \end{aligned} \quad (3)$$

The horizontal edge line position given from the y -position of the bounding box gives a pixel accurate edge point. This is the start-point to localize the maximum edge gradient position in sub-pixel accuracy in y -direction. A local maximum is interpolated using five neighboring points and calculating their gradient norm. Simply said, this is fitting 3 points (i.e. representing the edge gradient) to a quadratic equation and finding its maximum:

$$y(x) = a + bx + cx^2 \quad (4)$$

and

$$\frac{dy}{dx} = 0 \quad (5)$$

with x the integer x -coordinates of the points and $y(x)$ the corresponding gradient norm values. With $L = \{l(j) | j \in \mathbb{N}\}$ an infinite line of pixels with a peak at coordinate 0 corresponding to the middle pixel $l(0)$ and ∇ a general derivation operator (symmetric) the gradient norm becomes:

$$\begin{aligned} a &= |\nabla l(-1)| \\ b &= |\nabla l(0)| \\ c &= |\nabla l(1)| \end{aligned} \quad (6)$$

The maximum of the parabola (i.e. the highest slope in intensity) passing through $(-1, a)$, $(0, b)$ and $(1, c)$ is now found by:

$$y_m = \frac{a - c}{2(a - 2b + c)} \quad (7)$$

This maximum y_m is calculated for $hx[x]$ and its direct neighbors, i.e. $hx[x] - 1$ and $hx[x] + 1$, and the average of these three values is then passed as local y -maximum (see Fig. 3). From the four found sub-pixel accurate points, a line is fit from the two pairs of opposite points on both vertical and horizontal OLED edges. The crossing of these lines determine the final center coordinates of the OLED structure (Fig. 2).

C. OLED center passing

Since the stability of the motion system depends on a continuous stream of correct OLED coordinates, it is of critical importance that fault or missing positions are dealt with in a smart manner. From the repetitive pattern we know that OLEDs are always stacked next to each other (i.e. a rectangular grid). When in an image OLEDs are missing this information is then used to virtually determine the missing

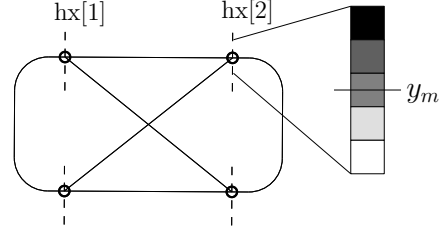


Fig. 2. Outline of OLED with sub-pixel accurate points to determine center point. $hx[1]$ and $hx[2]$ are determined from the vertical edges of the OLED structure (bounding box). The right grid shows the intensity value changes for the edge and the point where the slope has a maximum (y_m).

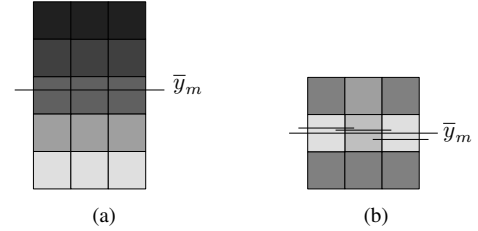


Fig. 3. Fig. 'a' shows the intensity values for each vertical edge line ($hx[i]$, $hx[i - 1]$ and $hx[i + 1]$) and points. The long horizontal line represents the mean highest slope of all three vertical pixel-lines. Fig. 'b' shows the derivative (1D symmetric) of (a) in vertical direction. The short lines represent the maximum of each vertical grid line, the long line represents the mean of the three short lines.

position(s). First, all found OLEDs are ordered in a compass-wise manner. Locations that remain zero (i.e. no found OLED) are filled with neighboring information. This means that in theory a minimum of three OLEDs would be sufficient to determine all other, missing center coordinates (i.e. one value on each row and column, zero otherwise). The idea is that in any way there should be an OLED position (even if this is erroneous), to ensure stability of the control loop. As extra step, a software flag could be set that ensures no printing is performed on that location.

IV. EXPERIMENTAL SETUP

The camera and lens used for experiments are standard of-the-shelf industrial components: a Prosilica GC640M camera with a frame rate of 197 fps full frame (near VGA, 659×493 pixels) combined with a standard 1.5x magnifying lens results in a pixels size of $6 \mu m$. The selected region of interest (132×45 pixels) then contains nine OLEDs to obtain 1000 fps. The camera is connected via a Gigabit Ethernet interface (GigE Vision) to a standard notebook with 2 GB of RAM and 2.4 GHz Intel Core 2 Duo CPU.

Coaxial lighting is applied which has the advantage that the light that enters the camera sensor is reflected mainly from axial illumination. This is due to the use of a beam-splitter which directs illumination from a power LED source downwards onto the OLED substrate which subsequently is reflected up into the camera.

The on-line algorithm to detect OLED centers in real time is controlled by the timer of the camera, which is more stable (i.e. less jitter) than a standard linux (or Xenomai) timer. The camera is set to stream images at a fixed rate (1 kHz). Whenever a frame is read and send to the computer, a callback is send which enables immediate processing on the obtained image. In this way the delay between image retrieval and processing is set to a minimum.

The platform with OLED substrate is a planar xy-table (2 DOF) actuated by 2 linear motors. The camera and the lighting itself are fixed. An electro-mechanically controlled height and orientation motion is not yet present. The actual control methodology is based on prediction with a steady state Kalman filter and can be found in [1].

V. RESULTS

A. Calibration

The orientation calibration has an accuracy of 0.15° due to the discrete nature of the pixel grid. This is, however, accurate enough for reliable center detection. For the homing procedure an accurate OLED position is not necessary. The only requirement from homing is that the resulting substrate position should be correctly aligned with the first region of interest in the on-line center detection step. This transition, having nine full OLEDs in field of view, is satisfied well.

B. Center detection

Experiments have been carried out with several algorithms for comparison. The algorithm presented in [14], an algorithm which calculates a center only with obtained bounding box and our 'ruler' method are compared with respect to accuracy, reproducibility and timing performance. Fig. 4 shows the output after thresholding and the bounding boxes which are used for optimal edge detection.

Table II gives an overview of the three tested algorithms. The bounding box method is disregarded directly due to its low accuracy and reproducibility caused by the pixel accurate bounding box. The center-of-gravity approach from [14] has advantages towards accuracy and reproducibility. However, this is only in an ideal situation when the complete OLED structure has uniform intensity. Whenever there is slight lighting variation within an OLED, this affects the outcome and reliability of the algorithm (see Fig. 1b and Fig. 6).

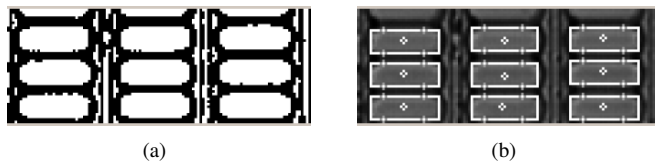


Fig. 4. Output of 'ruler' algorithm. Fig. 'a' shows the output after thresholding. Fig. 'b' shows the found OLEDs outlined with a rectangle. On horizontal lines the points are shown where the optimal vertical edges are searched.

TABLE II
ALGORITHM COMPARISON

	Bounding box	Ruler	Center of Gravity[14]
Accuracy	0.5 [px] 3 [μm]	0.2 [px] 1.2 [μm]	0.1 [px] 0.6 [μm]
Reproduc. (3σ)	0.5 ± 0.5 [px] 3 ± 3 [μm]	0.36 ± 0.06 [px] 2.16 ± 0.36 [μm]	0.24 ± 0.06 [px] 1.44 ± 0.36 [μm]
Speed	0.50 [ms]	0.60 [ms]	0.55 [ms]
Robustness	mid	high	low



Fig. 5. Image with scratch in OLED substrate surface. Fig. 'a' shows the output after thresholding. Fig. 'b' shows the found OLEDs outlined with a rectangle. Note that OLEDs with scratch are disregarded, but should be dealt with appropriately.

This robustness is however present in the 'ruler' approach, which only uses edge information to calculate the center of an OLED. For these reasons, and because the algorithm is well within specifications (see table II), the most desired method is the 'ruler' algorithm.

When an OLED center for any reason is missed (i.e. scratch in surface, dust or lighting faults), this should not affect the outcome of the complete algorithm. Fig. 5 shows how a scratch in the OLED substrate can cause the missing of OLEDs. The solution to use neighboring OLED x - and y -positions ensures that the measurement loop can continue although no actual printing on broken OLED cups will be carried out (see Fig. 6). The accuracy of the algorithm is not affected by this, since for the broken OLEDs no task has to be executed.

C. Bottlenecks

With an image size of 132×45 pixels and a frame rate of 1000 fps, the network load is 6.1 MB/s. Since GigE allows 116 MB/s, bigger images could be acquired or higher frame rates could be achieved. The maximum that can be reached



Fig. 6. Image with broken OLED structures. Fig. 'a' shows the output after thresholding. Fig. 'b' shows the found OLEDs outlined with a rectangle. The coordinates of the OLEDs that are broken are determined from surrounding OLEDs. This is necessary to ensure a continuous stream of (nine) coordinates for stable motion control. Note that some OLEDs which are broken are still found (east and south-west OLED), however, the centers are not calculated correctly.

is limited by the processing power of the CPU and thus indirectly by the size of the image and the frame rate.

Another bottleneck is that the algorithm is limited to rectangular products. However, with minor changes this could be adapted, as long as the object that has to be detected has a closed contour with distinguishable edges.

VI. CONCLUSIONS AND FUTURE WORK

We present a framework for using visual servoing as encoder-based control methodology, in which the encoders and low-level joint controllers present in the motors are disregarded. The repetitive pattern of the product is used as encoder grid, thus closing the control loop solely on vision. An image processing algorithm, which calibrates the visual setup, initiates a homing procedure and extracts center coordinates of OLED structures at 1 kHz, is presented and compared with two different methods regarding accuracy, reproducibility and processing time. The presented algorithm is not more accurate, however favorable over others (i.e. a simplified version of the presented algorithm and an earlier developed algorithm presented in [14]) due to its advantage in using edge information instead of incorporating the complete OLED surface [14] which can be non-uniform in intensity. Within the field of view (132×45 pixels) nine OLED centers can be found with $1.2 \mu\text{m}$ accuracy and 0.60 ms delay. This is well within specification (i.e. $< 10 \mu\text{m}$, $< 1 \text{ ms}$) and sufficient to close the motion control loop at 1 kHz. For future research, an industrial variant will be implemented on an FPGA to cope with delay and to obtain even faster computation.

REFERENCES

- [1] J. J. T. H. de Best, M. J. G. van de Molengraft, M. Steinbuch, "Direct Dynamic Visual Servoing at 1 kHz by Using the the Product as One Dimensional Encoder," 7th IEEE International Conference on control and Automation, New Zealand, 2009, in press.
- [2] P. I. Corke, "Visual Control of Robot Manipulators A Review," in K. Hashimoto, ed., Visual Servoing, Robotics and Automated Systems, Vol. 7. World Scientific, Singapore, pp. 1-31, 1993.
- [3] W. Czajewski and M. Staniak, "Real-time Image Segmentation for Visual Servoing," in B. Beliczynski et al. (Eds.): ICANNGA 2007, Part II, LNCS 4432, pp. 633-640, 2007.
- [4] N. R. Gans, S. A. Hutchinson, P. I. Corke, "Performance Tests for Visual Servo Control Systems, with Application to Partitioned Approaches to Visual Servo Control," in International Journal of Robotics Research, 22, p.p. 955-981, 2003.
- [5] R. Garrido, A. Soria, G. Loreto, "Visual Servoing of a Planar Overactuated Parallel Robot," in SPIE, International society for optical engineering, vol. 6719-03, 2007.
- [6] S. A. Hutchinson, G. D. Hager, P. I. Corke, "A Tutorial on Visual Servo Control," in Robotics and Automation, IEEE Trans., 12(5): pp 651-670, 1996.
- [7] S. Jorg, J. Langwald, M. Nickl, "FPGA based Real-time Visual Servoing," in Proc. of 17th International Conference on Pattern Recognition, pp. 749-752, 2004.
- [8] N. Kiryati, Y. Eldar, A. M. Bruckshtein, "A Probabilistic Hough Transform," in Pattern Recognition 24 (4), pp. 303-316, 1991.
- [9] C. Lazar, A. Burlacu, "Performance Evaluation of Point Feature Detectors for Eye-in-hand Visual Servoing," in Proc. of 5th IEEE International Conference of Industrial Informatics (1), pp. 497-502, 2007.
- [10] D. G. Lowe, "Distinctive Image Features from Scale-Invariant Keypoints," in International Journal of Computer Vision, Vol. 60, No. 2, pp. 91-110, 2004.
- [11] N. Mansard, M. Lopes, J. Santos-Victor, F. Chaumette, "Jacobian Learning Methods for Tasks Sequencing in Visual Servoing," in International Conference on Intelligent Robots and Systems, IEEE/RSJ, pp.4284-4290, 2006.
- [12] Y. Nakabo, M. Ishikawa, H. Toyoda, S. Mizuno, "1 ms Column Parallel Vision System and It's Application of High Speed Target Tracking," in Robotics and Automation, Proc. IEEE Int. Conf., vol. 1, pp 650-655, 2000.
- [13] N. Otsu, "A Threshold Selection Method from Gray-level Histograms," in IEEE Transactions on Systems, Man and Cybernetics, vol. 9 (1), pp. 62-66, 1979.
- [14] R. S. Pieters, P. P. Jonker, H. Nijmeijer, "Real-Time Center Detection of an OLED Structure," in ACIVS 2009, LNCS, vol. 5807, pp. 400-409, 2009.
- [15] Z. Qi, J. E. McInroy, "Nonlinear Image based Visual Servoing Using Parallel Robots," in IEEE Int. Conf. on Robotics and Automation, pp. 1715-1720, 2007.
- [16] M. Riaz, S. Park, M. B. Ahmad, W. Rasheed, J. Park, "Generalized Laplacian as Focus Measure," in ICCS Part I, Lecture Notes in Computer Science, pp. 1013-1021, 2008.
- [17] K. Shimizu, S. Hirai, "CMOS-FPGA Vision System for Visual Feedback of Mechanical Systems," in Proceedings of the IEEE International Conference on Robotics and Automation, pp. 2060-2065, 2006.
- [18] J. Sturm, C. Plagemann, W. Burgard, "Unsupervised Body Scheme Learning through Self-perception," in Proceedings - IEEE International Conference on Robotics and Automation, art. no. 4543718, pp. 3328-3333, 2008.
- [19] H. Toyoda, M. Takumi, N. Mukozaka, M. Ishikawa, "1 kHz Measurement by Using Intelligent Vision System - Stereovision Experiment on Column Parallel Vision System: CPV4," in Proc. of SICE 2008, Int. Conference on Instrumentation, Control and Information Technology, pp. 325-328, 2008.
- [20] Y. Watanabe, T. Komuro, M. Ishikawa, "955-fps Real-time Shape Measurement of a Moving/deforming Object Using High-speed Vision for Numerous-point Analysis," in Proc. of 2007 IEEE International Conference on Robotics and Automation, pp. 3192-3197, 2007.