Bottlenecks and Tradeoffs in High Frame Rate Visual Servoing: A Case Study

Zhenyu Ye, Yifan He, Roel Pieters, Bart Mesman, Henk Corporaal, Pieter Jonker
Eindhoven University of Technology, Eindhoven, the Netherlands
{z.ye, y.he, r.s.pieters, b.mesman, h.corporaal, p.p.jonker}@tue.nl

abstract

Visual servoing applies image sensors instead of mechanical encoders for position acquisition. It can achieve higher resolutions than mechanical encoders at comparable cost. However, visual servoing is computation intensive. Special purpose hardware is often required. This work performs a case study on organic light emitting diode (OLED) manufacturing, a typical industrial application of machine vision. We optimize the vision processing algorithm and implement it on a field-programmable gate array (FPGA). Timing analysis is performed to identify the bottlenecks of the implementation. This work identifies the exposure time and the camera interface as the bottlenecks of the high frame rate visual servoing (above 1000 frames per second). A tracking task is simulated to evaluate the performance of the visual servoing system. We explore the tradeoffs between frame rate, delay, and performance. This work has special emphasis on the time predictability of the visual servoing system, which is a critical requirement for industrial applications. Our implementation exploits the synergy of algorithm, architecture, and interface to guarantee the predictability of the whole system.

1 Introduction

Visual servoing applies image sensors instead of mechanical encoders to acquire the positions of objects being controlled. On the one hand, it reduces the complexity of system by eliminating the number of mechanical encoders. On the other hand, it increases the computation workload due to the vision algorithm. A previous case study on OLED manufacturing [1][2] confirms that visual servoing can achieve better performance than mechanical encoders at comparable cost. However, the computing power of the processing platform limits further improvements of the system.

Special purpose hardware is often used to handle the high demand of computation workload. The field-programmable gate array (FPGA) is one of the most cost effective options. An implementation of high frame rate visual servoing on FPGA was proposed by Komuro, Tabata, Ishikawa [3], Ishii, Taniguchi, Sukendobe, Yamamoto [4] and Iwata, Kagami, Hashimoto [5]. However, several important issues are either not addressed or not analysed in detail by these work. First, the bottlenecks of the implementations are not analysed. Second, the tradeoffs among delay, frame rate, and performance of the system are not explored. Third, the time predictability of the implementations, which is crucial in industrial applications, is not addressed.

Compared to previous work, our work has the following contributions:

1. To the best of our knowledge, this is the first work that identifies the bottlenecks of a high frame rate visual servoing system using detailed timing analysis. The bottleneck analysis of our work indicates that, at such high frame rates, reducing the processing time of the vision algorithms will have diminishing return in terms of both frame rate and delay.

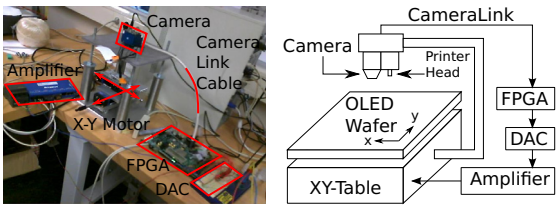
2. This is the first work that quantitatively explores the tradeoffs between frame rate, delay, and performance of the control system in the context of high frame rate visual servoing. The result indicates that, at high frame rate, the delay in the visual servoing system, instead of the frame rate, is the dominating factor on performance.

3. This is the first work to address the issue of time predictability in high frame rate visual servoing system, which is crucial in industrial environments. Besides conditioning the application properly, this work applies time-predictable algorithms, hardware architecture, and camera interface to guarantee the time predictability of the whole system.

This paper is arranged as follows. In section 2, the algorithm of image processing is introduced. In section 3, the architecture of the visual servoing system is described. In section 4, the bottleneck analysis of the visual servoing system is performed. In section 5, the design tradeoffs of the system are explored. In section 6, the method to guarantee the time predictability of the system is described. In section 7, the limitations of this case study are discussed. In section 8, conclusions are drawn with indications for future research direction.

2 Case Study: OLED Manufacturing

Due to the wide variety of visual servoing applications, this paper focuses on a typical industrial application, which is OLED manufacturing. An experimental setup, as shown in Fig. 1a, is constructed to replicate the OLED manufacturing machine used in an industrial environment [6]. The system architecture of the setup is shown in Fig. 1b. In the experimental setup, the camera and the lights are fixed rigidly at the top. In the industrial machine, an inkjet printing head, which is not present in the experimental setup, is also fixed rigidly at the top, next to the camera. The XY-table, which moves on a 2D plane, holds the OLED wafer which is to be manufactured. The inkjet printing head has to shoot chemical materials at the center of each OLED structures while the OLED wafer is moving with the XY-table.



(a) The experimental setup. (b) The system architecture.

Figure 1: The experimental setup and its system architecture.

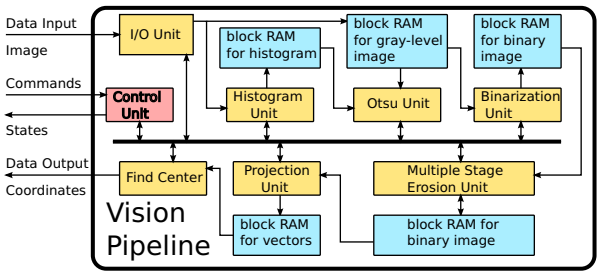


Figure 3: Hardware diagram of the vision pipeline implemented on the FPGA.

The vision pipeline is described in Fig. 2. The input of the pipeline is a region of interest (ROI) image with typical size of 120 × 45. The output of the pipeline are the coordinates of the centers of OLED structures. The front end of the pipeline applies Otsu optimum threshold [7] to segment the OLED structures from the background. The binary image is then eroded multiple times to remove the noise in the image. By utilizing the characteristics of the repetitive structures, reductions of the segmented OLED structures into two vectors are performed by horizontal and vertical projection. The centers of the OLED structure are found by searching the two vectors reduced from projection. Every stage of the pipeline has a predictable execution time, which leads to the predictability of the whole vision pipeline.

3 Visual Servoing System

The system architecture of the visual servoing system is shown in Fig. 1b. An SVS-VISTEK svs340MUCP camera [8] is connected to the FPGA board via a CameraLink interface. The camera is capable of sending a region of interest (ROI) image of size 120 × 45 via the CameraLink at a rate of 1000 frames per second (fps). The vision processing and the control algorithm run on the FPGA. The FPGA actuates the motor of the XY-table via power amplifiers. Due to the time-predictable implementation of the CameraLink interface and the motor interface, together with a time-predictable pipeline, the whole visual servoing system is also time-predictable.

The implementation of the vision pipeline in the FPGA is shown in Fig. 3. Each stage of the vision pipeline is implemented in a dedicated module. The pipeline runs as a synchronous systolic array. Image is streamed through the pipeline under the coordination of the control unit. To buffer the image stream at each stage, the block Random Access Memory (RAM) is used and customized to meet the need of each stage.

The timing breakdown of the whole visual servoing system is shown in Fig.4. The timing breakdown consists of four components: exposure of the image sensor, data read out from the image sensor, computer vision pipeline, and control algorithm. The required exposure time is measured on a real setup with the OLED structure. The exposure time depends on the lighting condition and the type of surface of the plate. It can vary from 10 μs for paper surface [1] to 400 μs for glass surface [2]. The image read out time is measured on the CameraLink interface. The vision processing time is further broken down into kernels, which correspond to the pipeline stages in Fig. 2. The data flow of the vision pipeline is analysed to overlap independent kernels for parallel execution. We also split a kernel to refine the granularity of the overlap. For example, the Otsu optimum threshold kernel is split into multiple sub-components such that parts of the kernel can be executed earlier. The control algorithm takes a relatively small amount of time, which is typical in industrial applications.

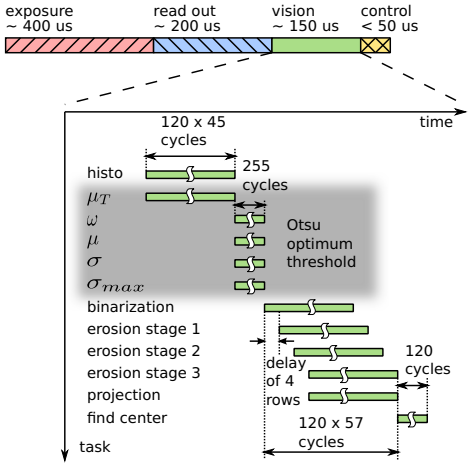


Figure 4: Timing breakdown of the whole visual servoing system for images size of 120×45. The detail of the sub-components of the Otsu algorithm is described in [7]. The vision pipeline runs at $100MHz$. The clock cycle time in this figure is $10ns$.

The breakdown of resource utilization for each component of the vision pipeline is shown in Fig. 5. The vision pipeline is implemented on a Xilinx XC2VP30 board containing a Virtex II-Pro FPGA. The resource utilization of the FPGA is less than 20% in this implementation. The vision pipeline can be further accelerated by utilizing more hardware resources of the FPGA. However, the bottleneck analysis in the next section shows that further acceleration of the vision pipeline can not increase the frame rate and has a diminishing return in reducing the delay.

4 Bottleneck Analysis

The bottlenecks of the visual servoing system could either be the exposure time or image read out time, depending on the image size, as shown in Fig. 6. The exposure time and image read out time both scale linearly with the image size. For center detection of repetitive structures, the processing time of the vision

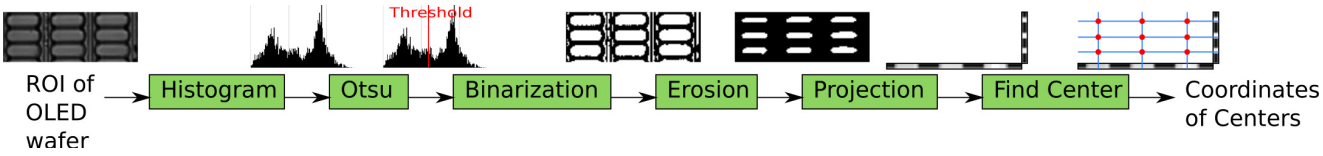


Figure 2: Vision pipeline of OLED center detection.

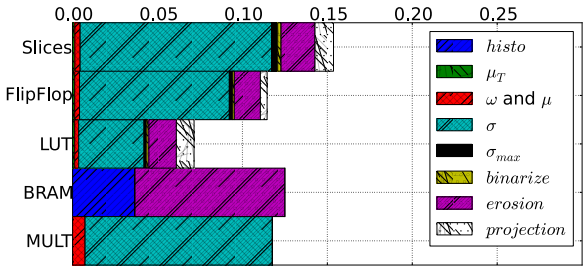


Figure 5: Resource utilization breakdown for the vision pipeline implemented on the FPGA. The numbers on the horizontal axis represent the percentages of hardware resources being used by the vision pipeline. The resource utilization of the Otsu algorithm is broken down into sub-components, which are described in [7].

pipeline also scales linearly with the image size. Because the vision pipeline is implemented on the FPGA, we can further reduce its processing time, if needed, at the cost of utilizing more hardware resources of the FPGA. Therefore, the processing time of the vision pipeline will not dominate the delay of the whole system. Moreover, while the computing power grows according to Moore's Law, the exposure time of the image sensor and the speed of the camera interface improve at a relatively slow pace, which will only enlarge the gap between them over time.

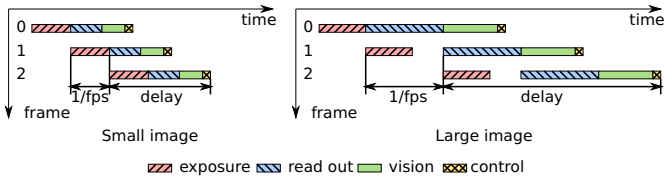


Figure 6: The bottlenecks of frame rate (1/fps) and delay for small and large image sizes.

5 Exploring Design Tradeoffs

This work explores the tradeoffs between delay, frame rate and performance of the visual servoing system. To evaluate the performance of the visual servoing system, a tracking task is simulated. The performance is measured in terms of the tracking error. In the simulation, the camera is assumed to be rigidly fixed. The OLED wafer performs a sinusoidal movement in one dimension, as shown by Equation 1, where A and P are the amplitude and period of the sinusoidal movement.

$$x = A \cdot sin(2\pi \cdot t/P) \quad (1)$$

To avoid loss of generality, various values of A and P have been applied in the simulation. According to the simulation, while different values of A and P resulted in different performances in the tracking task, they have similar tradeoffs in the design space. Due to page limit, this paper only presents the design space obtained at $A = 0.02m$ and $P = 0.04s$, as shown in Fig. 7.

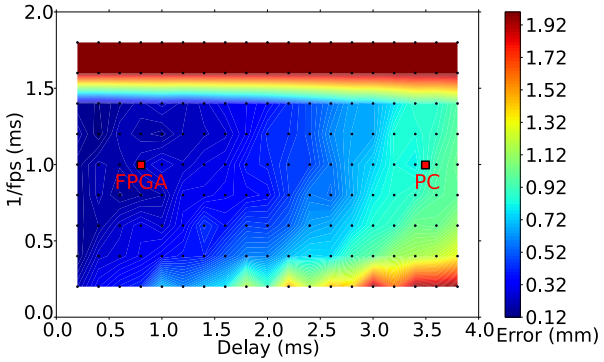


Figure 7: Design space exploration for the tradeoffs between frame rate (1/fps), delay and performance of the system, which is measured by the tracking error. The design points obtained on the FPGA based implementation of this work and on the PC based implementation of previous work [1] are annotated in this figure.

The methodology of design space exploration used in this work is as follows. First, a certain movement of the OLED wafer is assumed, which is unknown by the tracker. Second, the tracker predicts the position of the center of the moving OLED structure. The tracker runs at the frequency of the frame rate with a delay that simulates the delay from exposure to actuation. An $\alpha - \beta$ predictor, tuned to work at a specific frame rate, is used by the tracker to compensate the delay. Third, the predicted position of the center of the OLED structure is compared with the real position of the center according to the trajectory of the movement. The tracking error is defined as the maximum error between the predicted position and the real position during a certain period of simulation. Finally, the procedure described above is performed repeatedly for different frame rates and delays.

According to the exploration of the design space, if the delay of the system is much larger than the reversed of the frame rate ($1/fps$), the delay has a much larger impact on the performance than the frame rate. While the PC based implementation can achieve a comparable frame rate as the FPGA based implementation, it has a large delay compared to its $1/fps$. On the other hand, the FPGA based implementation can dramatically reduce the delay due to the fine-grained and low-overhead pipelining of the exposure, image read-

out, vision processing, and control.

6 Guarantee of Time Predictability

The time predictability is a critical requirement of visual servoing in industrial environments. Synergy of algorithm, hardware implementation, and camera interface is required to guarantee the time predictability of the overall system. We tackle all three aspects in our design of the system.

At the algorithmic level, we select a predictable vision pipeline, which has a static computation workload regardless of the input. For example, contour tracing, often used in the detection of repetitive structures, is not time-predictable. Therefore, we choose to use projection, which has a comparable accuracy as contour tracing but is time-predictable.

At the level of hardware implementation, we customize the components and their communication protocols to allow cycle-accurate estimation of the overall timing. Non-predictable factors, such as hardware cache, task scheduling of operating system, communication over Ethernet, etc. are eliminated in the implementation.

On the aspect of the camera interface, the CameraLink interface is chosen due to its time-predictable property. Other camera interfaces either fail to meet the delay and bandwidth requirements or have non-predictable timing properties [9].

In summary, these three aspects should be considered as a whole in an early design phase to guarantee the time predictability of the whole system. While previous work provide timing measurements of their systems, the time predictability of these systems is often unknown. In this paper we show that time predictability is achievable in a non-trivial visual servoing system.

7 Limitations

The methodologies applied in this work are not without limitations.

First, this case study is performed on a well conditioned industrial application. It eases the design of time-predictable algorithms. However, there are visual servoing systems that work in ill-conditioned environments, for example, varying lighting condition, occlusion, vibration, etc. In these ill-conditioned environments, designing a time-predictable system is challenging, if not impossible. These designs often apply a worst case scenario to bound the execution time, which is predictable but could be far worse than the actual performance.

Second, the computation workload of the vision pipeline in this case study is relatively low, compared to state-of-the-art machine vision algorithms. For vision pipelines that involve a heavy computation workload and are difficult to be exploited by accelerators, the processing of the vision pipeline could be the bottleneck of the system.

Third, the mechanical system and the electronic system are relatively simple in this case study. On the one hand, as the complexity of the mechanical system grows, building a dynamic model of the mechanical system becomes more difficult. This implies that the design space exploration is difficult to be both precise and comprehensive. On the other hand, as the complexity of the electronic system grows, it is difficult to avoid using non-predictable components or protocols. This again will make the design of a time-predictable system challenging.

Despite these limitations, the methodologies of this work are generic enough to be applied to a wide range of visual servoing applications.

8 Conclusions

The conclusions of this case study are multifold. First, the exposure time and image read out time are identified as bottlenecks of high frame rate visual servoing in a typical industrial application, despite that a high speed camera and its interface are used. Moreover, the improvement of computing power grows at a faster pace than that of the camera and its interface. These bottlenecks will only be exacerbated in the future. Second, the delay of the whole visual servoing system, which is seldom addressed in previous work, is shown to have a great impact on the performance. This work makes a case for taking the delay of the whole system into account in the context of high frame rate visual servoing. Third, this work implements a visual servoing system with time predictability guarantee, which is a critical requirement of industrial environment. The methodology of building a predictable visual servoing system is described and discussed.

We also identify several limitations of our methodologies. Our future research will tackle these challenges by extending and generalizing the methodologies proposed in this paper.

References

[1] J.J.T.H. de Best, M.J.G. van de Molengraft, and M. Steinbuch. Direct dynamic visual servoing at 1 khz by using the product as 1.5d encoder. In *ICCA*, pages 361 –366, dec. 2009.

[2] R. Pieters, P. Jonker, and H. Nijmeijer. Real-Time Center Detection of an OLED Structure. In *ACIVS*, page 400. Springer, 2009.

[3] T. Komuro, T. Tabata, and M. Ishikawa. A reconfigurable embedded system for 1000 f/s real-time vision. *Circuits and Systems for Video Technology, IEEE Transactions on*, 20(4):496 –504, april 2010.

[4] I. Ishii, T. Taniguchi, R. Sukenobe, and K. Yamamoto. Development of high-speed and real-time vision platform, h3 vision. In *IROS*, pages 3671 –3678, oct. 2009.

[5] N. Iwata, S. Kagami, and K. Hashimoto. A dynamically reconfigurable architecture combining pixel-level simd and operation-pipeline modes for high frame rate visual processing. In *ICFPT*, pages 321 –324, 12-14 2007.

[6] OTB Display. Pcap20 oled processing line. `http://www.otbdisplay.com/`.

[7] N. Otsu. A threshold selection method from gray-level histograms. *Systems, Man and Cybernetics, IEEE Transactions on*, 9(1):62 –66, 1979.

[8] SVS-VISTEK. svs340mucp. `http://www.svs-vistek.com/`.

[9] M.A. Azzopardi, I. Grech, and J. Leconte. A high speed tri-vision system for automotive applications. *European Transport Research Review*, 2(1):1–21, March 2010.