

A Framework of Networked Visual Servo Control System with Distributed Computation

Haiyan Wu¹, Lei Lou¹, Chih-Chung Chen¹, Kolja Kühnlenz^{1,2}, Sandra Hirche¹

¹*Institute of Automatic Control Engineering (LSR)*

²*Institute for Advanced Study (IAS)*

Technische Universität München

{haiyan.wu, lou, chen, koku, hirche}@tum.de

Abstract—In this paper, a networked visual servo control system with distributed computation is proposed to overcome the low sampling rate problem in vision-based control systems. A real-time image data transmission protocol based on Real-time Transport Protocol (RTP) is developed. The captured images are sent to different processing nodes connected over a communication network and processed in parallel. Thus, a high sampling rate of the visual feedback is achieved under a cloud image processing architecture. The varying image processing delay caused by the varying number of extracted features and the random transmission delay are modeled as a random process with Bernoulli distribution. By using the input-delay approach, the resulted networked visual servo control system is reformulated into a stochastic continuous-time system with time-varying delay. Experiments on two 1-DoF linear motor modules are carried out to validate the proposed approach. A visual servo control system without parallel distributed computation is implemented for comparison. The experimental results demonstrate significant performance improvement by the proposed approach.

I. INTRODUCTION

Benefiting from its accuracy and versatility, visual information obtained from cameras is utilized for closed-loop robot control in unknown environments, which is referred to as visual servo control system. An overview on the advantages and challenges of visual servo systems can be found in [1]. However, the control performance is limited due to the low sampling rate of the visual feedback caused by long image processing delay [2] [3], e.g. only a video rate of 25Hz is achieved for visual servoing in [4], and the visual information is just available at a updating rate of 20Hz in [5].

For a position based visual servoing (PBVS) [6] system considered in this paper, image processing delay mainly consists of several parts: delay caused by feature extraction from the original image, delay due to feature matching and delay resulted from running pose estimation algorithm. To achieve better performance, it is desirable to reduce the latency and sampling interval as far as possible by using simplified algorithms and advanced hardware. For example, a 1ms column-parallel vision system based on S³PE (Simple and Smart Sensory Processing Elements) is applied for a regrasp strategy in [7]. A sampling rate of 120 Hz is reached for visual servoing by reducing the image resolution to 640 × 240 pixels in [8]. However, only simple image processing algorithms are implemented in those studies. For applications in which

more complex image processing algorithms are demanded, e.g. for robust target recognition and object tracking, relative long image processing delay is inevitable. With recent advances in communication and computing technologies, parallel computing based on networked computational resources has gained tremendous attentions and has been proven to be an effective and economical platform for high-performance computing [9] [10]. Particularly for computer vision, the networked distributed computation platform becomes an attractive alternative to traditional supercomputers.

Fig. 1 shows the scheme of networked visual servo control systems (NVSCSs), in which the existing computation resources are utilized for video grabbing, image processing, controller implementation and actuators configuration over a communicational network. The data from distributed sensors (e.g. video cameras, optical sensor, Lidar, etc.) could be processed locally or sent over the communication network to other connected processing nodes. As a result, a parallel data processing architecture is formed and the sampling rate of visual servo control system is increased. Regarding the manner of distributed control architecture, parallel and multiple visual feedback loops are proposed in [11] with the advantages of the reliability and fault-tolerance. A cross-platform with distributed agents is developed for cooperative robot control

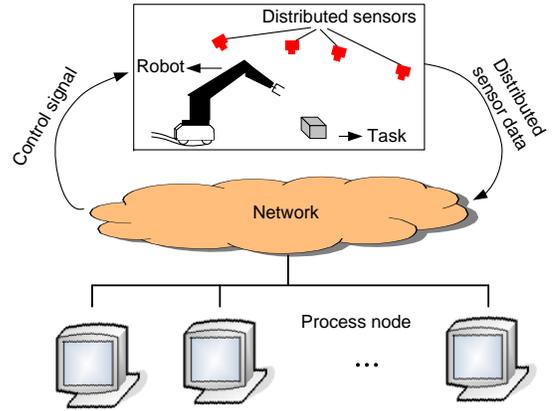


Figure 1: Scheme of networked visual servo control systems with distributed sensors and distributed computations.

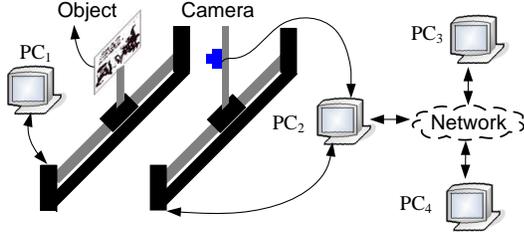


Figure 2: Visual servo control system with distributed computation over the network.

in [12]. However, in those studies enhancing visual sampling rate with distributed computation resources has not been considered, e.g. by distributing captured images to different processing nodes available over the network.

In this paper, a novel framework of NVSCS with distributed computation is proposed to increase the sampling rate of the visual feedback and thus improve the control performance. A real-time transport protocol, *networked control RTP (ncRTP)* based on Real-time Transport Protocol (RTP), is developed for real-time image data transmission. A cloud image processing platform based on ncRTP is proposed for parallel image processing. The random image processing delay and non-ideal image transmission delay are considered in this paper. Delay dependent stability analysis and controller design approach proposed in [13] is applied. A superior performance of the proposed system is validated by the experiments on two 1-DoF linear motor modules. The remainder of this paper is organized as follows: The system scheme of NVSCSs, image processing algorithm and time delay problem of a NVSCS with distributed computation are described in section II. A switching control law is applied to improve the control performance with respect to the feedback delay. In section III, the reformulation of the system into a continuous-time system with time-varying delays is introduced. The data transmission protocol and the cloud image processing architecture are presented in section V. In section VI, the experimental validation and performance comparison are discussed.

II. SYSTEM SCHEME

An example of a simplified NVSCS is shown in Fig. 2. The system consists of two 1-DoF linear motor modules L_1 and L_2 . The linear module L_1 which carries an object is defined as the reference module. A camera is mounted on the other linear module L_2 , which is defined as the slave module. L_1 and L_2 are connected to PC_1 and PC_2 respectively. L_2 is configured to track the object by using the visual feedback. The image acquisition and the controller are implemented on PC_2 . Image data are sent from PC_2 to PC_3 and PC_4 through the network. Image processing algorithm is implemented on both PC_3 and PC_4 . After the image processing on PC_3/PC_4 is finished, the results are sent back to PC_2 for the tracking control.

A. Image Processing

PBVS is considered for object tracking in this paper. The control aim is to minimize the relative pose between the object and the camera mounted on the end-effector. The fundamental problem of PBVS is pose estimation. To achieve robust and accurate pose estimation, Scale-Invariant Feature Transform (SIFT) features [14] are applied as they are well known for the robustness to image rotation, scaling and so on. Moreover, in order to accelerate feature extraction, a Graphics Processing Units (GPU) based computation platform is used. Due to the massive parallel processing capability of GPU, the computational delay of SIFT feature extraction is reduced to about 30 ms (for an image of 640×480 pixels with about 120 features detected, on the graphic card NVIDIA GeForce 8800). The detected features are then matched with the features in the reference image, which is captured at a desired position. A pose estimation algorithm based on virtual visual servoing proposed in [15] is applied. Finally, the relative pose between the camera and the object is obtained.

B. Time delay in the feedback loop

For the system shown in Fig. 2, the feedback delay of the visual information, which are extracted from the image captured at the time instant t_k , consists of three parts: i) transmission delay τ_k^{sp} for transferring image from the sensor to a processing node; ii) computational delay τ_k^c resulted from image processing; iii) transmission delay τ_k^{pc} for transferring image processing results from the process node to the controller. Therefore, the results arrive at the controller with a total feedback delay

$$\tau_k^{c+x} = \tau_k^c + \tau_k^x, \quad k \in \mathbb{N}, \quad (1)$$

where $\tau_k^x = \tau_k^{sp} + \tau_k^{pc}$. The feedback delay τ_k^{c+x} , which may lead to the instability of the system and deteriorate control performance, is considered for the stability analysis. It has to be mentioned that, image processing delay depends on the feature number extracted from the image. The feature number varies from frame to frame due to different viewing angles, illumination condition and noise. Combined with random transmission delay over network, a NVSCS with random feedback delay is resulted.

Though the image processing delay τ_k^c is reduced by GPU implementation, the sampling rate of the visual feedback (e.g. 30 Hz) is still much lower than the linear motor module with 1 KHz. In order to achieve better control performance, a higher sampling rate is desirable. Distributed computation over network is thus taken into consideration.

C. High Sampling Rate of Visual Feedback

We assume in this paper that the computational delay τ_k^c resulted from image processing is lower and upper bounded by $\underline{\tau}^c \leq \tau_k^c \leq \bar{\tau}^c$. $\underline{\tau}^c$ is the minimum image processing delay and $\bar{\tau}^c$ is the maximum image processing delay. In addition, the packet disorder caused by delays is assumed to be excluded in this paper. For a visual servo control system without distributed computation, the camera is configured to run at a

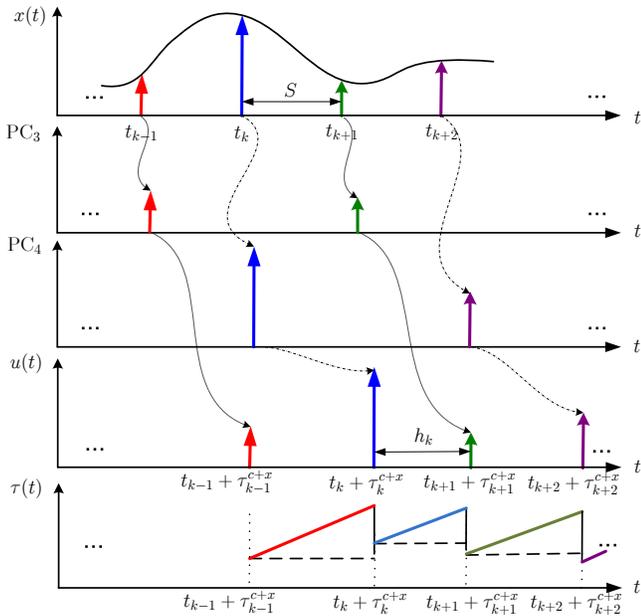


Figure 3: Timing diagram of a visual servo control system with distributed computation on two processing nodes.

sampling rate of $R = 0.9/\bar{\tau}^c$ (remain 10% free through output) to ensure each captured image is processed. However, the control performance is limited by this low sampling rate.

A parallel computation manner with distributed processing nodes is proposed in this paper to increase the sampling rate. Fig. 3 shows the timing diagram of a NVSCS with distributed computation on two processing nodes. Images are periodically captured by the camera running with a sampling interval of $1/R$. At time instant t_{k-1} , an image I_{k-1} is captured and sent to PC₃. While PC₃ processes the image I_{k-1} , another image I_k is available at t_k and is sent to PC₄. PC₃ and PC₄ run the image processing algorithm in parallel. Hence, a visual servo control system with distributed computation can achieve a higher sampling rate $\hat{R} = 0.9/\bar{\tau}^c > R$. For such a system, the image processing delay and transmission delay is not reduced. However, a higher sampling rate \hat{R} is achieved with distributed computation.

It has to be mentioned that, if more computational resources are available and can be integrated into the system, e.g. process nodes $n \geq 2$, a higher sampling rate is possible. However, a high network load at the sending side is induced by increasing processing nodes. In addition, the packet disorder problem at the controller side cannot be ignored when more processing nodes are added to the system. In the future, a strict mechanism for synchronizing a large number of processing nodes will be considered.

III. SYSTEM MODELLING

For the special case of visual servo control shown in Fig. 2, the linearized system dynamics with identified parameters m , c is given by

$$\dot{x}(t) = Ax(t) + Bu(t), \quad (2)$$

where $A = \begin{bmatrix} 0 & 1 \\ 0 & -c/m \end{bmatrix}$ and $B = \begin{bmatrix} 0 & 0 \\ 1/m & 0 \end{bmatrix}$. For a sampled-data system with zero-order hold (ZOH), the controller is

$$u(t) = Kx(t_k), \quad t_k \leq t < t_{k+1}, \quad (3)$$

where t_k denotes the sampling instant. Consider the controller law in (3) and the feedback delay τ_k^{c+x} defined in (1), the system in (2) becomes

$$\dot{x}(t) = Ax(t) + BKx(t_k), \quad t_k + \tau_k^{c+x} \leq t < t_{k+1} + \tau_{k+1}^{c+x}, \quad (4)$$

with update interval

$$h_k = t_{k+1} + \tau_{k+1}^{c+x} - t_k - \tau_k^{c+x} = S + \tau_{k+1}^{c+x} - \tau_k^{c+x}, \quad (5)$$

where S is the periodic sampling interval, see also Fig. 3.

A. Continuous-time System with Varying Feedback Delay

According to [16], a sampled-data system with varying feedback delay can be reformulated into a continuous-time system by using input-delay approach. The overall varying feedback delay including computation, transmission delays τ_k^{c+x} and holding delay $\tau^h(t) \in [0, h_k]$ ($\dot{\tau}^h(t) = 1$) is assumed to be bounded

$$\begin{aligned} \tau(t) &= \tau_k^{c+x} + \tau^h(t), \quad t \in [t_k + \tau_k^{c+x}, t_{k+1} + \tau_{k+1}^{c+x}), \\ \underline{\tau} &= \min_{k \in \mathbb{N}} \{ \tau_k^{c+x} \}, \quad \bar{\tau} = \max_{k \in \mathbb{N}} \{ \tau_k^{c+x} + h_k \}. \end{aligned} \quad (6)$$

It is seen from (6) and (5) that, the overall delay $\tau(t)$ is related with the sampling interval S . That means, the maximum over all delay $\bar{\tau}$ is reduced with high sampling rate $1/S$ when more processing nodes are available for parallel image processing. Then the system (4) is reformulated into continuous-time system as

$$\begin{aligned} \dot{x}(t) &= Ax(t) + BKx(t - \tau(t)), \quad t_k + \tau_k \leq t < t_{k+1} + \tau_{k+1} \\ x_0 &= x(\tau_0), \quad \tau_0 \in [-\bar{\tau}, 0]. \end{aligned} \quad (7)$$

B. Switching Control Law

To achieve better control performance regarding the varying feedback delay, a switching control law proposed in [13] is applied to the system (7), which results in

$$\dot{x}(t) = Ax(t) + \sum_{i=1}^n \beta_i BK_i x(t - \tau(t)), \quad (8)$$

where β_i is the indicator function

$$\beta_i = \begin{cases} 1, & s_{i-1} \leq \tau(t) < s_i, \quad i = 1, \dots, n, \\ 0, & \text{otherwise,} \end{cases} \quad (9)$$

with $s_0 = \underline{\tau}$, $s_n = \bar{\tau}$ and $s_i > s_{i-1} > 0$, $i = 1, \dots, n-1$. Thus the overall delay is categorized into n intervals.

For further analysis we assume that the computation delays τ_k^c and transmission delays τ_k^x are modeled by i.i.d. process and take values in finite sets

$$\begin{aligned} \tau_k^c &\in \mathbb{T}^c = \{T^{c1}, T^{c2}, \dots, T^{cp}\}, \quad p \in \mathbb{N}, \\ \tau_k^x &\in \mathbb{T}^x = \{T^{x1}, T^{x2}, \dots, T^{xq}\}, \quad q \in \mathbb{N}. \end{aligned}$$

Define $\tau_k^a = \tau_k^c + \tau_k^x + h_k$ and according to (5), it becomes as

$$\tau_k^a = \tau_{k+1}^{c+x} + S, \quad \tau_k^a \in \mathbb{T}^a = \{\mathbb{T}^c + \mathbb{T}^x + S\}.$$

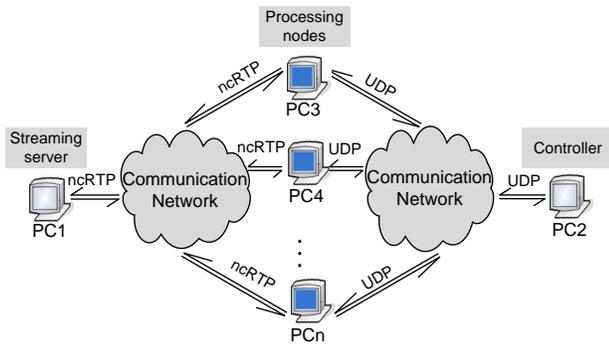


Figure 4: Cloud image processing architecture with ncRTP. PC₁: image streaming server; PC₃, ..., PC_n: processing nodes; PC₂: controller.

Select s_i in (9) as any subset T , $s_i \in T \subset T^a$. Due to the i.i.d. assumption on τ_k^c and τ_k^x , s_i is also i.i.d [17]. As a result, the occurrence probability p_i of each delay interval satisfies

$$\Pr\{\beta_i = 1\} = p_i, \quad \sum_{i=1}^n p_i = 1.$$

Remark 1: See stability analysis and controller design algorithm in [13].

IV. DISTRIBUTED COMPUTATION PLATFORM

In order to speedup visual feedback, a distributed computation platform by utilizing processing nodes across the network is proposed in this paper. In this section, the transport protocol developed for real-time image transmission in NVSCS and a parallel image processing architecture will be introduced.

A. Real-time Transport Protocol

Transmitting massive image data over networks becomes a challenge for NVSCSs. For example, a single channel image with a resolution of 640×480 pixels has a size of about 300 KB. For a sampling rate larger than 60 Hz, the network load is more than 144 Mbps. Although there are various transport protocols in the public domain, they are not suitable for image transmission, e.g. UDP cannot transmit data larger than 64 KB and TCP has a large variance of the transmission delay due to retransmissions. Therefore, a real-time image transmission protocol based on Realtime Transport Protocol (RTP) is developed.

Besides, the 1000BASE-T Ethernet is used to build the physical network. Other data link/physical layer devices such as EtherCAT, CAN, SERCOS or WLAN either need dedicated hardwares or have limited transmission rates. For example, EtherCAT and RTNET are based on Ethernet and have deterministic delays. However, the overhead for synchronization (EtherCAT: token ring, RTNET: TDMA) will introduce high bandwidth penalty, meaning that the average transmission delay is much larger. By WLAN, the theoretical speed of 802.11n is 300 Mbps. However, the maximum real-world speed is only about 100 Mbps.

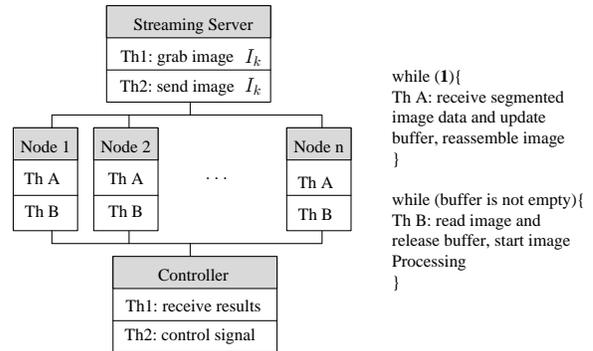


Figure 5: Thread scheduling on streaming server, processing nodes and controller in NVSCSs. I_k is the image captured at time instant t_k .

ncRTP for NVSCSs

RTP [18] is an application layer protocol for transmitting latency-sensitive data, such as video and audio on the Internet. It is the foundation of many Voice over IP and media streaming systems. RTP uses a separate channel for monitoring and adjusting the data delivery. In this paper, ncRTP (networked control RTP) is designed and implemented based on GNU ccRTP [19], which supports UDP as transport layer protocol. The algorithms of ncRTP are described in the following.

1) *Fragmentation Mechanism:* To support diverse data formats of cameras for NVSCS applications, a transparent data fragmentation/reassembly layer is built in ncRTP. Large frames are truncated as a series of data blocks patched with an extended identification number. For example, an image with a size of about 300 KB is segmented into five data blocks with each block smaller than 64 KB. The receiving stack can detect and reassemble the frame according to the identification number. If there is any packet losses, the whole image frame will be dropped, instead of retransmitting it.

2) *Synchronization and Timing:* The distributed computation system is designed as an event-driven system. Data processing threads are triggered when new frames have been assembled in receiving buffer. Therefore the system works with the same frequency of the grabbing rate of cameras. The Real Time Clock (RTC) of hosts in NVSCS can be synchronized by Network Time Protocol (NTP) up to 0.1 ms. The RTP header is extended according to [18] and the RTC time stamp is appended in the extension. The streaming server is designed on RTAI real-time kernel and the sending process is scheduled periodically within a jitter of $10 \mu s$.

B. Cloud Image Processing

Since most image processing algorithms are time-consuming, e.g. the feature detection and matching algorithms, the computational capacity becomes a bottleneck for NVSCSs that need high frame rate. With the development of GPGPU (General-purpose computing on graphics processing units) technique in recent years, we have built a GPGPU cluster in a LAN based on ncRTP, see Fig. 4.

The cloud image processing platform shown in Fig. 4, which utilizes existing computational resources, is flexible and

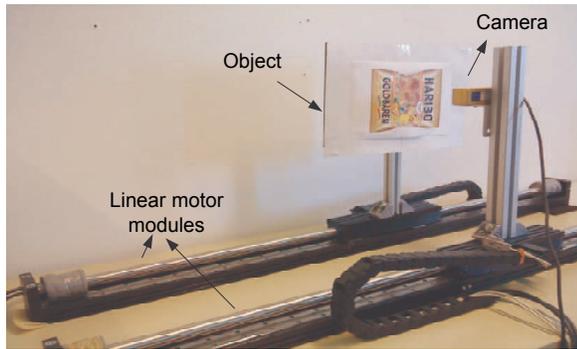


Figure 6: Experimental setup of a NVSCS with distributed computation

economical for NVSCSs. It consists of a streaming server, several processing nodes, and a controller. The thread scheduling running on streaming server, processing nodes and controller are illustrated in Fig. 5.

- Streaming server on PC_1 : The streaming server is equipped with a high-speed camera and the sending process is scheduled by the real-time RTAI kernel. To reduce the latency, two real-time tasks run in parallel. One polls the camera frame buffer periodically, since not all cameras support external trigger mode. Another is triggered when the first task has received a frame from the camera. Then it immediately sends the image data together with a frame index through ncRTP. The jitter of the streaming server is smaller than 0.1 ms. Multi-streaming is also supported by ncRTP.
- Processing nodes PC_3, \dots, PC_n : Each processing node in this platform is equipped with a Nvidia GeForce 8800 GTX graphic card. The ncRTP protocol is installed on the nodes and is optimized for receiving large volume image data. The image processing algorithm is scheduled as soon as an image has been reassembled in the receiving buffer.
- Controller on PC_2 : the controller is implemented on another node, which receives image processing results (< 64 KB) from the processing nodes through UDP. The image processing results are used to determine the configuration signal for the manipulator.

With the platform introduced above, a parallel image processing architecture is established. Thus, high-speed visual feedback is achievable.

V. EXPERIMENTS

The NVSCS with distributed computation based on ncRTP proposed in this paper is tested in this section. Experiments are carried out on two 1-DoF linear motor modules as shown in Fig. 2 and Fig. 6. As illustrated in section II, the control objective is to drive the linear motor module which carries a camera to track the moving object on the reference linear motor module. Both modules are controlled through MATLAB/SIMULINK blocksets on two standalone PCs with a sampling rate of 1KHz. The reference module controlled by

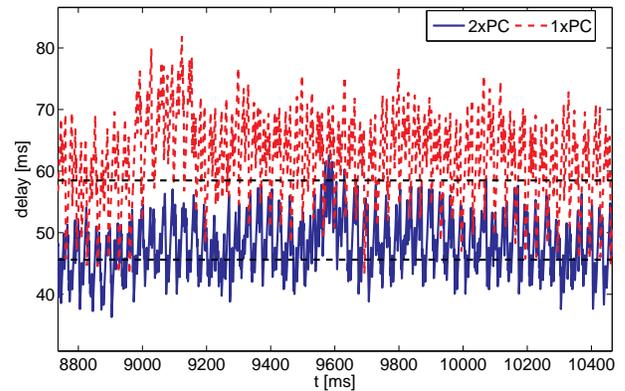


Figure 7: Feedback delay in visual serving system with two PCs for parallel image processing (solid line) and only one PC for image processing (dash line).

PC_1 (i686, AMD, Athlon(tm), Processor 3000+) is assigned with the following signal:

$$x_r = 0.3 \sin(1.05t).$$

The camera (Mikrotron EoSens MC1363) is connected to PC_2 (X86-64, AMD, Athlon(tm) 64X2 Dual Core Processor) with the controller implemented on the same PC. The camera runs at a frame rate of 60fps with a resolution of 640×480 pixels. Other two PCs, PC_3 (i686, AMD, Phenom(tm), 9850, Quad-Core Processor) and PC_4 (i686, AMD, Phenom(tm) IIx4 945 Processor) equipped with GeForce 8800 (GTX) seraphic cards of NVIDIA, are selected for image processing. PC_3 and PC_4 are assigned to deal with images with the frame index of

$$PC_3 : n_1 \in \{1, 3, 5, 7, \dots\},$$

$$PC_4 : n_2 \in \{2, 4, 6, 8, \dots\}.$$

The image data transmission is realized based on ncRTP developed in Section IV.

An image captured by the camera at time t_k is packetized with a timestamp when the image is buffered and sent from PC_2 to either PC_3 or PC_4 depending on the frame number. The image processing results together with the timestamp are sent from both PCs to the controller on PC_2 through network. The timestamp is then extracted from the received data. By comparing it with the current time the overall feedback delay including data transmission, image processing and holding delays is obtained.

For comparison of the control performance, the standard visual servo control approach with one computational resource for image processing is implemented, i.e. camera runs at the framerate of 30 fps and image processing algorithm is only implemented on PC_3 . For both approaches, the NVSCS with distributed computation on two PCs and the standard visual servo control system with one PC for image processing, the experiments are repeated for 20 times with the same initial conditions of both modules. The resulted feedback delay is shown in Fig. 7. For the system with distributed computation on two PCs, the feedback delay is bounded by [32.62 ms, 86.57 ms] with a mean value of 45.63 ms, see

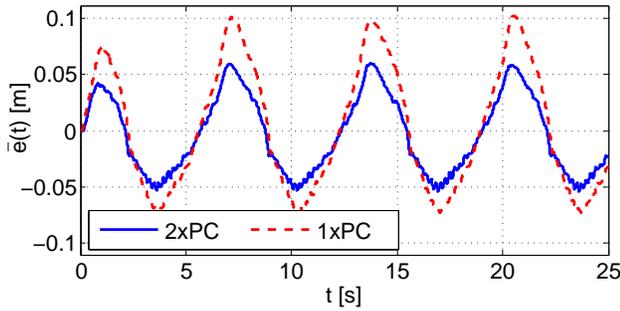


Figure 8: Mean control error evolution of the system with two image processing nodes (solid line) and one image processing node (dash line).

Table I. It is smaller than the feedback delay of the system with only one PC for image processing, whose delay is bounded by [35.80 ms, 105.40 ms] with a mean value of 58.48 ms.

Using the controller design approach in [13], the feedback gain K in (7) for each approach is selected as

$$K_{2PC} = 1200, \quad K_{1PC} = 750,$$

where K_{2PC} is the feedback gain for the system with two processing nodes, and K_{1PC} is the feedback gain for the system with one processing node. The control error is defined by

$$\bar{e}(t) = x_r(t) - x_c(t),$$

where $x_r(t)$ denotes the position of the reference module, and $x_c(t)$ is the position of the controlled module. The evolution of mean control error is shown in Fig. 8.

Table I: Feedback delay and performance comparison.

	τ [ms]	$\bar{\tau}$ [ms]	τ_{mean} [ms]	$ \bar{e} _{\text{max}}$ [cm]	$ \bar{e} _{\text{var}}$ [cm ²]
2xPC	32.62	86.57	45.63	6.24	12
1xPC	35.80	105.40	58.48	9.7	30

The proposed system with parallel image processing has maximal tracking error $|\bar{e}|_{\text{max}} = 6.24\text{cm}$ and variance of the tracking error $|\bar{e}|_{\text{var}} = 12\text{cm}^2$, while the maximum tracking error of system without parallel image processing is $|\bar{e}|_{\text{max}} = 9.70\text{cm}$ (+12.18%) and the variance $|\bar{e}|_{\text{var}} = 30\text{cm}^2$. The experimental results show that the NVSCS system with parallel image processing enables a better control performance compared to the conventional visual servo control system.

VI. CONCLUSION

This paper presents a framework of a networked visual servo control system with distributed parallel image processing on process nodes connected over a communicational network. A real-time image data transmission protocol based on RTP is developed to establish the distributed computation platform. It enables a high sampling rate of visual feedback for a NVSCS by which complex image processing algorithm is required. The proposed approach is validated by comparing a NVSCS on two 1-DoF linear motor modules with a standard visual servo control system without parallel image processing over network. The experimental results demonstrate the superior

performance of the proposed approach over the conventional visual servo control system. The future work is concerned with extending the approach to more general nonlinear visual servoing system with 6-DoF.

VII. ACKNOWLEDGMENTS

This work is supported in part by the DFG excellence initiative research cluster *Cognition for Technical Systems - CoTeSys*, see also www.cotesys.org, the BMBF Bernstein Center for Computational Neuroscience Munich, see also www.bccn-munich.de, and the Institute for Advanced Study (IAS), Technische Universität München, see also www.tum-ias.de.

REFERENCES

- [1] S. Hutchinson, G. Hager, and P. Corke, "A tutorial on visual servo control," *IEEE transactions on robotics and automation*, vol. 12, no. 5, pp. 651–670, 1996.
- [2] P. I. Corke, *High-performance visual closed-loop robot control*. PhD thesis, Department of Mechanical and Manufacturing Engineering, University of Melbourne, 1994.
- [3] M. Vincze, "Dynamics and system performance of visual servoing," in *Proceedings of IEEE Conference on Robotics and Automation*, pp. 14–35, 2000.
- [4] E. Malis, F. Chaumette, and S. Boudet, "2 1/2 D visual servoing," *IEEE Transactions on Robotics and Automation*, vol. 15, no. 2, pp. 238–250, 1999.
- [5] R. Kelly, R. Carelli, O. Nasisi, B. Kuchen, and F. Reyes, "Stable visual servoing of camera-in-hand robotic systems," *IEEE/ASME transactions on mechatronics*, vol. 5, no. 1, pp. 39–48, 2000.
- [6] W. Wilson, C. Hulls, and G. Bell, "Relative end-effector control using Cartesian position based visual servoing: Special section on vision-based control of robot manipulators," *IEEE Transactions on Robotics and Automation*, vol. 12, no. 5, pp. 684–696, 1996.
- [7] A. Namiki, Y. Imai, M. Ishikawa, and M. Kaneko, "Development of a high-speed multifingered hand system and its application to catching," in *Proceedings of the 2003 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 2666–2671, Citeseer, 2003.
- [8] J. Gangloff and M. de Mathelin, "High-speed visual servoing of a 6-dof manipulator using multivariable predictive control," *Advanced Robotics*, vol. 17, no. 10, pp. 993–1021, 2003.
- [9] R. Buyya *et al.*, "High Performance Cluster Computing: Architectures and Systems, Volume 1," *Prentice Hall PTR*, vol. 82, pp. 327–350, 1999.
- [10] V. Sunderam *et al.*, "PVM: A framework for parallel distributed computing," *Concurrency Practice and Experience*, vol. 2, no. 4, pp. 315–339, 1990.
- [11] G. DeSouza and A. Kak, "A subsumptive, hierarchical, and distributed vision-based architecture for smart robotics," *IEEE Transactions on Systems, Man, and Cybernetics-Part B: Cybernetics*, vol. 34, no. 5, 2004.
- [12] E. Cervera, "Distributed visual servoing: A cross-platform agent-based implementation," in *Proc. International Conference on Intelligent Robots and Systems*, pp. 3676–3681.
- [13] H. Wu, C. Chen, J. Feng, K. Kühnlenz, and S. Hirche, "A switching control law for a networked visual servo control system," in *Proceedings of IEEE International Conference on Robotics and Automation, Anchorage, Alaska*, 2010.
- [14] D. G. Lowe, "Distinctive image features from scale-invariant keypoints," in *International Journal of Computer Vision*, 2004.
- [15] E. Marchand and F. Chaumette, "Virtual visual servoing: A framework for realtime augmented reality," *Computer Graphics Forum*, vol. 21, no. 3, pp. 289–298, 2002.
- [16] H. Gao, J. Wu, and P. Shi, "Robust sampled-data H_∞ control with stochastic sampling," *Automatica*, vol. 45, pp. 1729–1736, 2009.
- [17] R. M. Gray and L. D. Davisson, *An Introduction to Statistical Signal Processing*. Cambridge, UK: Cambridge University Press, 2004.
- [18] H. Schulzrinne, S. Casner, R. Frederick, and V. Jacobson, "RFC3550: RTP: A Transport Protocol for Real-Time Applications," *RFC Editor United States*, 2003.
- [19] <http://www.gnu.org/software/ccrtp/>.