



Cloud-based Networked Visual Servo Control

Wu, Haiyan; Lu, Lei; Chen, Chih-Chung; Hirche, Sandra; Kühnlenz, Kolja

Published in:

I E E Transactions on Industrial Electronics

Link to article, DOI:

[10.1109/TIE.2012.2186775](https://doi.org/10.1109/TIE.2012.2186775)

Publication date:

2013

[Link back to DTU Orbit](#)

Citation (APA):

Wu, H., Lu, L., Chen, C-C., Hirche, S., & Kühnlenz, K. (2013). Cloud-based Networked Visual Servo Control. *I E E Transactions on Industrial Electronics*, 60(2), 554 - 566. <https://doi.org/10.1109/TIE.2012.2186775>

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 you believe that this document breaches copyright please contact us providing details, and we will remove access to the work immediately and investigate your claim.

Cloud-based Networked Visual Servo Control

Haiyan Wu, *Member, IEEE*, Lei Lou, *Student Member, IEEE*, Chih-Chung Chen, *Member, IEEE*, Sandra Hirche, *Senior Member, IEEE*, and Kolja Kühnlenz, *Member, IEEE*

Abstract—The performance of vision-based control systems, in particular of highly dynamic vision-based motion control systems, is often limited by the low sampling rate of the visual feedback caused by the long image processing time. In order to overcome this problem, the networked visual servo control, which integrates networked computational resources for cloud image processing, is considered in this article. The main contributions of this article are i) a real-time transport protocol for transmitting large volume image data on a cloud computing platform, which enables high sampling rate visual feedback, ii) a stabilizing control law for the networked visual servo control system with time-varying feedback time delay, and iii) a sending rate scheduling strategy aiming at reducing the communication network load. The performance of the networked visual servo control system with sending rate scheduling is validated in an object tracking scenario on a 14 degree-of-freedom dual-arm robot. Experimental results show the superior performance of our approach. In particular the communication network load is substantially reduced by means of the scheduling strategy without performance degradation.

Index Terms—Networked visual servo control, real-time image transmission, cloud computing, sending rate scheduling, control with time-varying time delay

I. INTRODUCTION

The visual sensor is one of the most essential components for a robot control system. The visual information is expected to be fast, accurate and reliable in providing real-time information of dynamical surroundings including the operating environments, objects, obstacles, and human activities. From 'look-and-move' [1] approaches in open loop fashion to 'visual servoing' [2]–[5] in closed loop fashion, the performance of vision-based control systems benefits from the developments of the sensing technology (e.g. networked cameras [6], camera arrays [7], dense skins of cameras [8]), image processing techniques (e.g. applying image moments [9], Laplacian of Gaussian [10], and wavelets [11] for feature extraction), and different control schemes (e.g. position-based visual servoing (PBVS) and image-based visual servoing (IBVS) [12], 2-1/2D visual servoing [13]). Despite of many advantages,

vision-based control encounters a number of problems, e.g. the choice between using an image-based and position-based architectures, on-line computation of the image Jacobian for IBVS, path-planning combined with image space feature, the stability problem in vision-based motion control, and the low sampling rate of the visual feedback [12], [14]–[16].

The sampling rate of the visual feedback is limited due to the latency induced by the image capturing with a camera, the image transmission from the camera to the processing node, and the image processing itself. In order to achieve a good control performance, a high sampling rate of the visual feedback is required. In the literature there are techniques available, which accelerate image processing, e.g. implementing simplified algorithms and using advanced hardware. In [17], an integral image is used to speed up feature detection. Vision chips, FPGAs (Field-programmable Gate Arrays), and Graphics Cards with parallel computation capability are favored to implement image processing algorithms [18]–[20]. However, the image processing delay still largely exceeds the cycle time of a robot joint position control loop. For example, it takes about 20 ms to extract SURF (Speed Up Robust Features) [17] from an image (640×480 pixels) on graphic card NVIDIA-GeForce 8800, while the sampling interval of the joint position control loop is typically in the range of 1 ms. For the tracking of fast-moving objects, the long image processing delay increases the risk of losing the objects in the field of view, and thus causes control failures. With the recent advances in computation and communication technologies, parallel computation based on networked computational resources (cloud computing) has gained more and more interests for high-performance computing [21]–[25]. With the introduction of networked computation into traditional control systems, the resulting system exhibits several advantages. It features flexible reconfiguration capabilities, e.g. new components can easily be added. Besides, it enables simple maintenance and diagnosis with low wiring effort and digital diagnosis protocols. Standardized components and computational power can be shared among different applications, and there are no such strong constraints on the components in terms of low power and low weight as typical for mobile robotic applications. It allows also for shared sensing concepts, e.g. multiple robots/applications can share sensing resources.

In this article the novel concept of a *networked visual servo control (NVSC) system* is proposed in order to achieve high sampling rate visual feedback by distributed networked computation for highly dynamic vision-based motion control tasks, see Fig. 1 for visualization. In NVSC systems, the components, including image capturing, data processing, control algorithms, and actuators are implemented on spatially distributed processing nodes which are connected by

Manuscript received June 28, 2011; revised September 25, 2011; accepted November 17, 2011. Copyright © 2009 IEEE. Personal use of this material is permitted. However, permission to use this material for any other purposes must be obtained from the IEEE by sending a request to pubs-permissions@ieee.org.

H. Wu has been with the Technische Universität München, Munich, 80290 Germany until 2011. She is now with the Department of Electrical Engineering, Technical University of Denmark, Kgs. Lyngby, 2800 Denmark (e-mail: h.wu@ieee.org).

L. Lou, S. Hirche and K. Kühnlenz are with the Institute of Automatic Engineering, Department of Electrical Engineering and Information Technology, Technische Universität München, Munich, 80290 Germany (e-mail: {lou, s.hirche, kolja.kuehnlenz}@ieee.org).

C.-C. Chen has been with the Technische Universität München, Munich, 80290 Germany until 2010. He is now with ASML, DR Veldhoven, 5504 Netherlands (e-mail: c-c.chen@ieee.org).

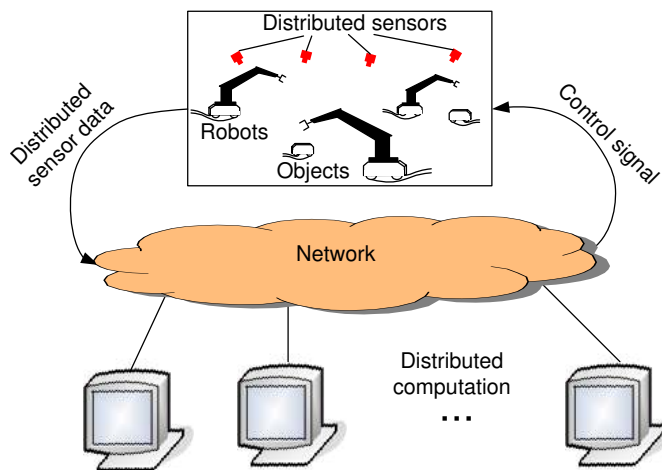


Figure 1: Scheme of a networked visual servo control system with distributed sensors and distributed computation (cloud computing).

a shared (wireless or wired) communication network. By sending sensor data to the computational resources available over the communication network, parallel data processing is carried out to speed up sensor feedback, and thus improve control performance. In spite of significant improvements in image processing capabilities, the system shown in Fig. 1 is confronted with several issues that need to be addressed:

i) In contrast to most conventional networked control systems (NCSs), the data transmitted within a NVSC system are large volume image data. We consider it beneficial to transmit a whole image frame over the communication network as the most time consuming part of image processing is feature extraction from the original images. A camera running at a frame rate of 60Hz and a single channel image of resolution 640×480 pixels (300Kbyte per image) results in a communication network load of 144Mbps. In particular for the image processing algorithms requiring multiple images as input, such as stereo vision algorithm, scene reconstruction algorithm, and image fusion algorithm [26], massive image data have to be transmitted. Therefore, an appropriate real-time transport protocol needs to be designed for large volume image data transmission.

ii) High communication network load results in large communication time delays and high packet drop rates. A communication network load beyond the channel capacity results in a complete communication blackout. The communication constraints are critical in particular in wireless communication networks. Therefore, in the presence of limited communication bandwidth keeping the data transmission over the communication network at a minimum under the constraint of sufficient control performance becomes a challenge task. Smart scheduling strategies are necessary in order to cope with the communication constraints in particular in the case of large volume image data.

iii) In contrast to conventional feedback control loops the transmission of sensor data over a communication network induces uncertainties such as (time-varying random) time delay (in addition to computation induced latency) and packet loss. It is well-known that even a small time delay typically results

in a control performance degradation, and may even lead to unstable closed loop behavior. Accordingly, an appropriate control mechanism able to cope with time-varying random time delay and packet loss needs to be chosen for NVSC systems.

In the most general case of a NVSC system with multiple sensors challenges of sensor synchronization and data fusion are encountered, which, however, is not in the focus of this article. In this article we will investigate the case of a single visual sensor connected to an image processing cloud.

The main contribution of this article is to solve distributed image processing and the image data transmission, the control problem, and the optimal scheduling for NVSC systems: i) A novel real-time image data transmission protocol *ncRTP* (*networked control real-time Transport Protocol*) is proposed in this article. Based on *ncRTP*, a cloud computing platform is established for parallel image processing aiming to increase the sampling rate of visual feedback. ii) In order to improve the control performance, a switching control mechanism is proposed which switches with the time delay, and is less conservative than the standard non-switching approach. iii) A switching control and scheduling co-design approach is developed to deal with the trade-off between control performance and communication network load induced by massive image data transmission. A sending rate scheduling with the emphasis on image sending rate scheduling at the streaming server side is proposed. For the tracking problem with NVSC considered in this article, a sending rate scheduling based on the tracking error is proposed.

The remainder of this article is organized as follows: Section II gives an overview of the related work in the field of networked control. The real-time image transmission protocol and the distributed image processing platform for cloud computing are introduced in Section III. The switching controller considering varying feedback delay is presented in Section IV. In Section V, the sending rate scheduling strategy for reducing the communication network load under guaranteed control performance is given. The experimental validation on a 14-DoF dual-arm robot and the performance comparison are presented in Section V with a brief discussion.

II. RELATED WORK

The exchange of data between spatially distributed system components is an essential task in NVSC systems. Aiming at real-time capability a number of specialized network protocols have been developed in the past such as CAN [27] for industrial automation and FieldBus [28] for process control, which however, require specialized hardware. More recently also Ethernet (IEEE 802.3) is used in the field of industrial automation, which we will also adopt in this article. Several Ethernet-based public domain communication protocols have been developed in order to alleviate the non-deterministic behavior of the standard Ethernet to meet the real-time requirements of feedback control systems [29], e.g. switched Ethernet, EtherCAT and PROFINet [30]–[32]. However, these transport protocols are not suitable for transmitting image data due to either dedicated hardware requirements or limited packet size, e.g. UDP cannot transmit data larger than 64 KB

and TCP has a larger absolute value and also larger variance of the transmission time delay due to windowing behavior and re-transmissions. So, currently no real-time transport protocol is available for massive image data transmission over a communication network suitable for visual servo control.

Transmitting images over the communication network results in a large communication network load. To cope with the trade-off between the control performance and the communication network load due to image transmission, many efforts have been made. Scheduling the real-time communication network traffic has been discussed in [33]–[35], e.g. in [33] the node with greatest absolute error wins the right to transmit data. In order to reduce the communication network traffic, the deadband control approach is proposed in [36], [37], which only sends packets when the current value exceeds a given threshold. The benefits of such an event-triggered sampling scheme over a periodic sampling scheme are analytically shown for the first time in [38]. The optimal combination of control and event-triggered sensor transmission is investigated in [39]–[41]. In this article, we will adopt an event-triggered law for image data transmission.

In our specific setting we study the vision-based object tracking problem. The vision system sends images over the communication network considering the tracking error. Assume the tracking error is random. The sending interval, which switches based on the random tracking error, becomes also random. The control signal can only be updated at instants the controller receives information from the vision system, which indicates an event-triggered control scheme. In addition, transmitting data over the communication network introduces transmission time delays into the system. In Ethernet, transmission time delays are generally random. Besides, there are computation time delays due to image processing. Here, the computation delay is modeled random due to data-dependent conditional branches/loops during image processing [42]. For stability analysis and control design, the NVSC system is modeled as a sampled-data system with time-varying delays. For systems with random delays, various control approaches have been proposed in the literature. Approximating a random delay by its upper bound results in a robust but conservative control design [43]. Less conservative control design approaches are based on stochastic analysis [44]–[52], see also [53] for a general overview. A Markov process is used to model the random delays as stochastic process in [44]–[46]. The resulting closed loop system is a Markovian jump system (MJS). However, a constant sampling interval is assumed in most of the existing results, which cannot be applied to the NVSC systems with time-varying sampling intervals.

III. DISTRIBUTED COMPUTATION

To overcome the shortcoming of long image processing time delay, computational power over the communication network is utilized to speed up visual feedback. In this section, a cloud image processing platform shown in Fig. 2 is introduced. Within this platform, images are distributed to different processing nodes for parallel computation. A real-time transport protocol is developed for large volume image data transmission. To further reduce the computation time,

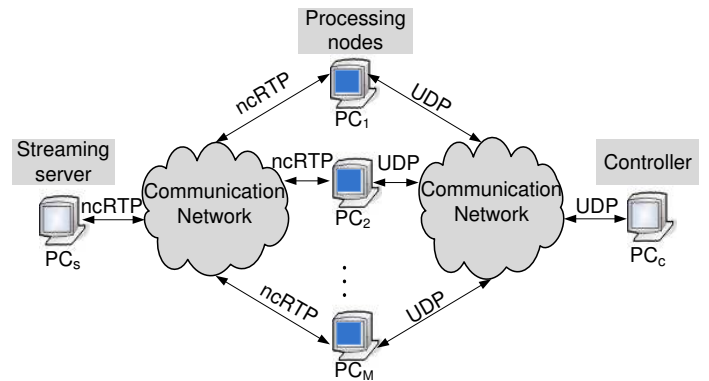


Figure 2: Distributed computation platform with ncRTP for parallel image processing in NVSC systems with one visual sensor. PC_s : image streaming server; PC_c : controller; PC_1, \dots, PC_M : processing nodes.

GPGPU (General-purpose computing on graphics processing units) implementation is adopted for image processing on each processing node. The details of the transmission protocol and the components in the cloud image processing platform are introduced in this Section.

A. Analysis of Image Transmission in NVSC systems

The reason to transmit a whole frame image in vision-based control systems is, that the most time consuming part of image processing is feature extraction from original image sequences. As mentioned in Section I, extracting 100 SURF features from an image takes about 20 ms on the graphic card NVIDIA GeForce 8800, while the followed feature matching, outliers rejection, and pose estimation algorithms take less than 5 ms. Therefore, it is necessary to distribute images to different processing nodes for feature extraction. In such a way, a high-sample-rate visual feedback could be achieved.

Running a camera at 60Hz @ 640×480 pixels requires a bandwidth of 144 Mbps. Transmitting these massive image data over the communication network becomes a challenge task in NVSC systems. As the first step, a real-time transport protocol should be designed. Although there are various transport protocols in the public domain, they are not suitable for real-time image transmission as discussed in Section II. Therefore, an image transmission protocol ncRTP based on real-time Transport Protocol (RTP) is developed for NVSC systems. The second step is to select an appropriate device to build the physical layer. In this work, the 1000BASE-T Ethernet is adopted to transmit image data, while other data link/physical layer devices such as EtherCAT, CAN, SERCOS or WLAN either need dedicated hardware or have limited transmission rates. For example, EtherCAT and RTNET are based on Ethernet and have deterministic time delays. However, the overhead for synchronization (EtherCAT: token ring, RTNET: TDMA) introduces high bandwidth penalty, meaning the mean transmission time 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. With 1000BASE-T Ethernet determined as physical layer, the developed transport protocol ncRTP is briefly introduced below.

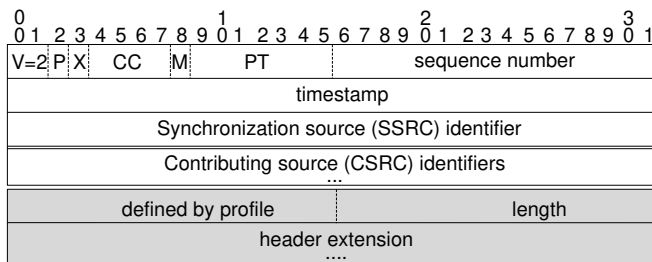


Figure 3: RTP fixed header and header extension (modified from [54]). Upper part: fixed header; lower part: extended header. V: version=2; P: padding; X: extension; CC: CSRC count; M: marker; PT: payload type.

B. Real-time image transmission protocol for NVSC systems

RTP [54] is an application layer protocol for transmitting latency-sensitive data, such as video and audio over 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 article, ncRTP (networked control RTP) is designed and implemented based on GNU ccRTP [55], which supports both UDP and TCP as support layer protocol. Since TCP is considered as unsuitable for real-time applications, UDP is used as the underlying protocol for ncRTP to build a low latency protocol stack providing multicast support. As a support layer protocol, UDP inherits all features of IP protocol, which is characterized as connectionless and unreliable. In comparison with TCP, UDP has no transport level mechanisms to ensure the data integrity, such as flow control and retransmissions, which improve transmission reliability with the payout of extra unpredictable latency. To get a balance between robustness and performance, a subset of TCP features is selectively implemented in ncRTP on the application layer to make it suitable for real-time image transmission. The features of ncRTP are described below:

i) RTP header: The RTP has the header layout shown in Fig. 3 (for details of the meaning of different fields, please refer to [54]). The upper part of Fig. 3 shows the fixed header field of RTP, while the lower part shows the header extension. It has to be mentioned, that PT denotes the payload type supporting the transmission of different image types. The sequence number is assigned to each RTP data packet which can be used by the receiver for both packet restoring and packet loss detection. The header extension here allows carrying of additional information required for individual implementations, e.g. image frame number and timestamp when the image is captured.

ii) Fragmentation mechanism: For transmission of data larger than 64 KB on top of UDP protocol, such as single-channel image with typical size of 300 KB, the data have to be fragmented by the sending module and assembled by the receiving counterpart. However, the fragmentation mechanism is not defined uniformly for different payload types in ccRTP. To support diverse data formats of cameras for NVSC 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 and the extended payload types defined in the header of ncRTP, without data specific implementation in applications.

iii) Congestion control: Most packet loss in Internet is caused by congestion [56], which could lead to congestion collapse. The system time delay increases continuously if data are lost successively. The sliding window mechanism [57] of TCP, by which the transmission stream is expanded to consume all available bandwidth, is not selected as the congestion control strategy for ncRTP. In contrast, the data are transmitted with a certain rate by RTP. Several methods have been proposed for RTP congestion control in [58]. In ncRTP the congestion control is implemented as image resolution/rate adaptation based on control protocol (RTCP) feedback. If there is packet loss, the whole Application Data Unit (ADU), e.g. a frame, is dropped. Instead of retransmitting this frame, the next frame which is available and contains more important information for networked control will be sent in this case. When successive data losses are detected by the receiver, the sender could be notified through RTCP to decrease the sending rate or to reduce the image resolution.

iv) 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 the receiving buffer. Therefore the system works with the same frequency of the grabbing rate of cameras. The Real Time Clock (RTC) of hosts in NVSC systems can be synchronized by Network Time Protocol (NTP) up to 0.1 ms. The RTP header is extended according to [54] 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 μ s. Based on this framework, the control performance could be evaluated with small granularity.

v) System parameter calibration: The parameters of operating system such as Ubuntu Linux are optimized for general-purpose applications. They should be calibrated for large volume data transmission to avoid packet loss. In our experiment, the parameters such as buffer size of IP stack and UDP packet size have been tuned for stable transmission.

C. Cloud-based Image Processing

Based on ncRTP introduced above, a cloud image processing platform is established to achieve high-speed visual feedback, see also Fig. 2. The cloud image processing platform utilizes existing computing resources, and is flexible and economical for NVSC systems. It consists of a streaming server, several processing nodes, and a controller. To further accelerate image processing on each single processing node, a GPGPU implementation is applied by exploiting its massive parallel computation capacity developed in recent years.

The features of the streaming server, the processing nodes and the controller in the cloud image processing platform shown in Fig. 2 are given below:

i) Streaming server on PC₁: The streaming server is equipped with a high-speed camera providing image se-

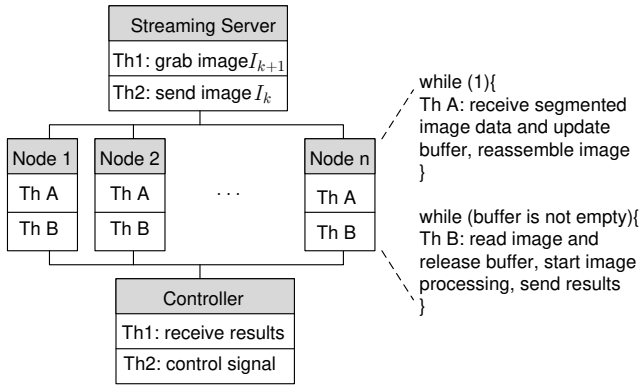


Figure 4: Thread scheduling on streaming server, processing nodes and controller in NVSC systems. I_k is the image captured at time instant t_k .

quences. The captured images are sent through the sending process scheduled by the real-time RTAI kernel. To reduce the latency, two real-time tasks run in parallel as shown in Fig. 4: 1) One task polls the image from the camera frame buffer periodically, which is necessary especially for cameras not supporting external trigger mode; 2) The other one is triggered when the first task has received a whole frame from the camera. Then it immediately sends the image data together with a frame index through ncRTP. Multi-streaming is also supported by ncRTP. To reduce the communication network load, sending rate scheduling algorithm considering control error and/or network status could also be implemented on the streaming server.

ii) Processing nodes PC_3, \dots, PC_n : The ncRTP protocol is installed on the nodes, and is optimized for receiving large volume image data. Similar to the steaming server, two real-time tasks run in parallel on each processing node: one for receiving image data and image reassembling, and the other one for image processing and sending processing results, see also Fig. 4. Moreover, fast, accurate and robust image processing is required with respect to a good control performance, in particular for high dynamic motion control tasks. Therefore, the GPGPU implementation of image processing algorithm is adopted in this article. Each processing node in this platform is equipped with a graphic card. For position-based visual servoing (PBVS) applied in this article, the image processing algorithm consists of feature extraction, feature matching, outliers rejection, and pose estimation. The output of image processing gives the pose (incl. the translation and the rotation) of the object relative to the camera. The image processing algorithm is scheduled as soon as an image has been reassembled in the receiving buffer.

iii) Controller on PC_2 : the controller could be implemented on another node. It receives image processing results (< 64 KB) as well as the frame index from the processing nodes through UDP. At the controller side, the received data are pre-filtered by sorting it along the frame index. Considering the real-time requirements, the data with smaller frame index compared with the one of the latest received data is dropped. Thus, the packet disorder problem is prevented. Finally, the image processing results are used to determine the configuration signal for the manipulator to realize the control task.

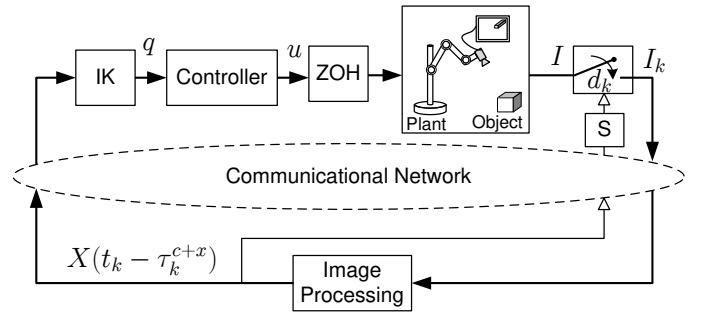


Figure 5: System diagram of NVSC system for object tracking. I : image sequence; $\mathbf{X} \in \mathbb{R}^6$: pose of the end-effector relative to the reference object in Cartesian space; τ_k^{c+x} : feedback time delay consisting of image processing time delay τ_k^c and data transmission time delay τ_k^x ; \mathbf{q} : joint displacement; \mathbf{u} : control signal; IK: inverse kinematics; d_k : sending interval; t_k : sampling instant; ZOH: zero-order hold; S : scheduler.

With the platform introduced above, a parallel image processing architecture is established, and thus high-speed visual feedback is achieved for NVSC systems. Another important issue of NVSC systems is controller design which is introduced in the following section.

IV. CONTROL DESIGN FOR NVSC SYSTEMS

A 3D tracking problem with NVSC based on the cloud computing platform developed above is studied in this article. Position-based visual servoing (PBVS) is applied and a camera-in-hand structure is selected for object detection and localization. Several aspects towards NVSC systems are discussed in this section, including feedback time delay analysis, system formulation, and control design.

A. Feedback Time Delay Analysis

The block diagram of the NVSC system with 3D tracking is shown in Fig. 5. In order to approximate an event-triggered vision system, a camera running at a high image frame rate, e.g. ≥ 400 Hz, is selected in this article. Compared to the time delays of data transmission and image processing typically of ≥ 30 ms, the image sampling interval, here ≤ 2.5 ms, is negligible. Besides, the image transmission over the communication network is triggered by a sending rate scheduler S . Thus, the vision system is approximately considered as an event-triggered sampling system.

The image sequence captured by the eye-in-hand camera is denoted by I , and the image captured at sampling time t_k is denoted as I_k . The captured images are sent to the processing nodes over the communication network with the sending interval d_k . The NVSC system becomes a sampled-data system with zero-order hold (ZOH). After running the image processing on the cloud computing platform, the pose of the end-effector $\mathbf{X}(t_k - \tau_k^{c+x}) \in \mathbb{R}^6$ relative to the object is available to the controller with the time delay

$$\tau_k^{c+x} = \tau_k^c + \tau_k^x,$$

where τ_k^{c+x} denotes the feedback time delay for the t_k -th image consisting of the image processing time delay τ_k^c and the data transmission time delay τ_k^x . Through the robot inverse

kinematic model, the Cartesian relative pose \mathbf{X} is mapped on a joint displacement $\mathbf{q}(t_k - \tau_k^{c+x})$ in the joint space. Then, the control signal \mathbf{u} is calculated in the joint space, and applied to the individual motors in the joints driving the robot motion. The feedback $\mathbf{X}(t_k - \tau_k^{c+x})$ is also utilized to adjust the sending interval d_k with the scheduler S in Fig. 5. The policy of sending interval scheduling is presented in Section V. For system modeling and control design, the properties of the image processing time delay, the data transmission time delay, and the sending interval are discussed below:

i) Image processing and transmission time delays: As discussed in Section III, the image processing time delay is caused by feature extraction, feature matching, and pose estimation. As each image contains a different scene and therefore different features, the feature number varies from frame to frame due to different view angle, illumination conditions, and random noise. In this article we model the varying feature number over time as a random independent and identically distributed (i.i.d.) sequence resulting in a random i.i.d. image processing time delay τ_k^c . Since the i.i.d. process is popular model in the communication community [59], we also consider an i.i.d. transmission time delay τ_k^x in this article. Furthermore, the communication network / computational load are assumed such that the transmission/computation time delay is lower and upper bounded: $\tau^x \in [\underline{\tau}^x, \bar{\tau}^x] / \tau^c \in [\underline{\tau}^c, \bar{\tau}^c]$.

ii) Sending interval: The sending interval between the t_k -th and the t_{k+1} -th sampling is denoted by d_k . For the tracking problem studied in this paper, the tracking error is utilized for sending rate scheduling, which will be introduced in Section V. While the relative pose $X(t)$ between the end-effector and the object measures the tracking error when tracking a moving object, it is utilized as tracking error for performance evaluation. Assume the tracking error $\mathbf{X}(t)$ is random i.i.d. Accordingly, we model the sending interval as i.i.d variable $d_k(\mathbf{X}(t))$. For simplicity of analysis we assume the sending interval takes values in a finite set

$$d_k \in D = \{d^1, d^2, \dots, d^M\}, \quad (1)$$

where $d^m > d^{m+1}$, $m = 1, \dots, M-1$. In order to determine the values of d^1, \dots, d^M , we need to consider about the available computational resources and the computational complexity of the image processing. For example, with only one processing node, in order to guarantee it finishes image processing before receiving the next image, the worst case processing time $\bar{\tau}^c$ is selected as the smallest sending interval. With more processing nodes available, a smaller sending interval can be assigned, e.g. with two processing nodes a sending interval $\frac{\bar{\tau}^c}{2}$ can be achieved. Assume there are a maximum of G processing nodes available for image processing. The lower bound of the sending interval becomes $\underline{d} = d^M = \frac{\bar{\tau}^c}{G}$. The upper bound of the sending interval relies on a heuristic choice as the upper bound of the computation time delay $\bar{d} = d^1 = \bar{\tau}^c$ in this work. Therefore, the sending interval d_k is bounded by $d_k \in [\underline{d}, \bar{d}]$, where

$$\underline{d} = \frac{\bar{\tau}^c}{G}, \quad \bar{d} = \bar{\tau}^c. \quad (2)$$

The values of d^2, \dots, d^{M-1} can be arbitrarily selected. In this work, we associate them with the number of processing nodes

in the way

$$\left\{d^2 = \frac{\bar{\tau}^c}{2}, d^3 = \frac{\bar{\tau}^c}{3}, \dots, d^m = \frac{\bar{\tau}^c}{g}, \dots, d^{M-1} = \frac{\bar{\tau}^c}{G-1}\right\}, \quad (3)$$

where g denotes the the number of necessary processing nodes for the sending interval d^m . The policy of selecting the appropriate sending interval d_k from (1) and the associated number of processing nodes g will be discussed in Section V.

B. System Modeling

For the preliminary study of NVSC systems with time delays, we consider a locally linearized version of the robot dynamics by the computed torque feed-forward approach [60] resulting in the linear time-invariant continuous-time plant

$$\dot{\mathbf{x}}(t) = \mathbf{A}\mathbf{x}(t) + \mathbf{B}\mathbf{u}(t),$$

with initial condition \mathbf{x}_0 , where $\mathbf{x}(t) = [\mathbf{q} \ \dot{\mathbf{q}}]^T \in \mathbb{R}^n$ is the state vector and $\mathbf{u}(t) \in \mathbb{R}^n$ is the control input; \mathbf{q} denotes the joint angles; \mathbf{A}, \mathbf{B} are constant matrices of appropriate dimensions. Assuming sampling of the states at the instant t_k the state feedback controller is given by

$$\mathbf{u}(t) = \mathbf{K}\mathbf{x}(t_k), \quad t \in [t_k, t_{k+1}), \quad \forall k \in \mathbb{N},$$

and the closed loop system is derived as

$$\dot{\mathbf{x}}(t) = \mathbf{A}\mathbf{x}(t) + \mathbf{B}\mathbf{K}\mathbf{x}(t_k), \quad t \in [t_k, t_{k+1}). \quad (4)$$

For feedback systems with periodic sampling and ideal data transmission channels, the closed loop system (4) is equidistantly updated by

$$h = t_{k+1} - t_k.$$

In this case, the lifting technique [61] is applied to derive stability conditions and appropriate control algorithms. However, for NVSC systems the feedback time delays depend on data transmission time delay, the image processing time delay, and the sending interval, which are random and time-varying.

As introduced in Section IV-A, the value of the state vector $\mathbf{x}(t_k)$ arrives at the controller side with a feedback time delay τ_k^{c+x} consisting of the image processing and data transmission time delays, as shown in Fig. 6. At the controller side the holding interval between two consecutive updates at time instants $t_k + \tau_k^{c+x}$ and $t_{k+1} + \tau_{k+1}^{c+x}$ is denoted by h_k

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

Throughout this article, we assume that packets do not overtake each other, meaning that packets arrive at the controller according to their sending order.

As a result, the closed loop system (4) becomes a sampled-data system with the feedback delay τ_k^{c+x}

$$\dot{\mathbf{x}}(t) = \mathbf{A}\mathbf{x}(t) + \mathbf{B}\mathbf{K}\mathbf{x}(t_k), \quad t \in [t_k + \tau_k^{c+x}, t_{k+1} + \tau_{k+1}^{c+x}). \quad (6)$$

The control problem to be addressed in this article is formulated as follows:

Control problem: Given a system with random feedback time delays, develop a control algorithm such that the closed loop system in (6) is exponentially mean-square stable (EMSS), i.e. satisfying

$$\mathbb{E}\{\|\mathbf{x}(t)\|^2 | \mathbf{x}_0\} \leq b \|\mathbf{x}_0\|^2 e^{-\rho(t-t_0)},$$

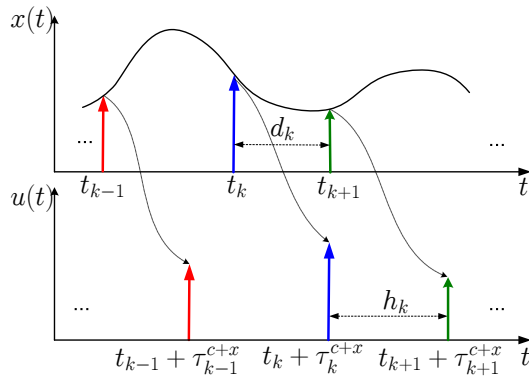


Figure 6: Time delay diagram of NVSC systems. τ_k^c : computation time delay for image I_k sampled at time instant t_k ; τ_k^x : transmission time delay for image I_k from sensor to processing node; $\tau_k^{c+x} = \tau_k^c + \tau_k^x$; d_k : sending interval between the t_k -th and t_{k+1} -th samplings; h_k : holding interval between two consecutive updates at the controller side.

where $b > 0$, $\rho > 0$ are real numbers and \mathbf{x}_0 is the initial condition.

For stability analysis and control design, the discrete-time system in (6) is further reformulated into a continuous-time system.

C. System Reformulation for Control Design

In this section, the NVSC system with random sending interval, random computation and transmission time delays is reformulated into a time-varying delay system by means of the input-delay approach [62]. In order to achieve stability and high control performance, a switching control mechanism is proposed which switches depending on the value of the overall time delay.

Reconsider the sending instant t_k as

$$t_k = t - (t - t_k) = t - \tau(t), \quad t \in [t_k + \tau_k^{c+x}, t_{k+1} + \tau_{k+1}^{c+x}),$$

where $\tau(t)$ is the overall time delay for the processed sensory feedback $\mathbf{x}(t_k)$ during the period $[t_k + \tau_k^{c+x}, t_{k+1} + \tau_{k+1}^{c+x})$ at the controller side, see Fig. 7. The overall feedback time delay $\tau(t)$ consists of three parts: the computation time delay τ_k^c , the transmission time delay τ_k^x , and the holding time delay $h(t)$

$$\begin{aligned} \tau(t) &= \tau_k^c + \tau_k^x + h(t) \\ &= \tau_k^{c+x} + h(t), \quad t \in [t_k + \tau_k^{c+x}, t_{k+1} + \tau_{k+1}^{c+x}), \end{aligned} \quad (7)$$

where $h(t) \in [0, h_k)$, $t \in [t_k + \tau_k^{c+x}, t_{k+1} + \tau_{k+1}^{c+x})$, with h_k given by (5). Between two consecutive updates at the controller side the holding delay $h(t)$ has the slope $\dot{h}(t) = 1$ as shown in Fig. 7. Combining (2), (5) and (7), the overall time-varying delay is bounded by $\underline{\tau} \leq \tau < \bar{\tau}$ with

$$\underline{\tau} = \underline{\tau}^c + \underline{\tau}^x, \quad \bar{\tau} = \bar{d} + \bar{\tau}^c + \bar{\tau}^x = 2\bar{\tau}^c + \bar{\tau}^x.$$

Substituting $\mathbf{x}(t_k) = \mathbf{x}(t - \tau(t))$ into the closed loop system in (6) yields

$$\begin{aligned} \dot{\mathbf{x}}(t) &= \mathbf{A}\mathbf{x}(t) + \mathbf{B}\mathbf{K}\mathbf{x}(t - \tau(t)), \quad t \in [t_k + \tau_k^{c+x}, t_{k+1} + \tau_{k+1}^{c+x}), \\ \mathbf{x}_0 &= \mathbf{x}(\theta), \quad \theta \in [-\bar{\tau}, 0], \end{aligned} \quad (8)$$

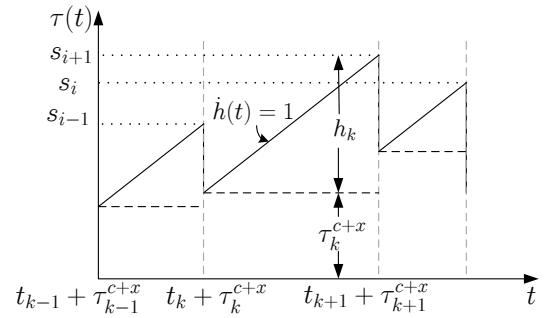


Figure 7: The evolution of time-varying delay $\tau(t)$.

where \mathbf{x}_0 is the initial condition. Hence, the sampled-data system with time-varying sampling intervals is reformulated into a continuous-time system with time-varying feedback time delay $\tau(t)$.

D. Switching Control Approach for NVSC

In order to stabilize the closed loop system and at the same time provide high control performance, a time delay-dependent switching control approach is proposed, in which the state feedback control gain switches according to the current time delay value. The varying time delay is categorized by boundary values s_i into n finite intervals

$$S_i = \{\tau | s_{i-1} \leq \tau < s_i\}, \quad i = 1, \dots, n$$

where $0 < s_i < s_{i+1}$, for $i = 1, \dots, n-1$, and $s_0 = \underline{\tau}$ and $s_n = \bar{\tau}$, see also Fig. 7. The delay distribution is discretized to reduce the computational complexity for the control design, as for each discrete value of time delay a different control gain needs to be computed. For the stability analysis and control design the discrete time delay values then act as upper bounds for their corresponding discretization. A higher resolution of the discretization results in lower conservatism of the control design regarding the performance. More details on the selection of appropriate boundary values of the bins s_i is presented in [63].

The switching control law is defined by

$$\mathbf{u}(t) = \mathbf{K}_{\sigma(\tau(t))}\mathbf{x}(t - \tau(t)), \quad (9)$$

where $\mathbf{K}_{\sigma(\tau(t))}$ denotes the state feedback controllers switching depending on the value of the time delay $\tau(t)$, and $\sigma(\tau(t))$ is a delay-dependent indicator function defined by

$$\sigma(\tau(t)) = i, \quad \text{if } \tau(t) \in S_i, \quad i = 1, \dots, n.$$

Accordingly, the system in (8) can be rewritten as

$$\dot{\mathbf{x}}(t) = \mathbf{A}\mathbf{x}(t) + \mathbf{B}\mathbf{K}_{\sigma(\tau(t))}\mathbf{x}(t - \tau(t)), \quad (10)$$

The resulting system (10) is a continuous-time system with a controller switching the feedback gains depending on the time-varying feedback time delay.

Compared with a conventional non-switching control approach which applies a constant gain designed for the highest appearing time delay $\bar{\tau}$, the proposed switching control is less conservative with respect to control performance [63]. There is a systematic way to derive the stabilizing control gains using linear matrix inequalities (LMIs). Due to space constraints the

reader is referred to [63]–[65], where the stability analysis and control design algorithms for this system class are described in detail.

V. DATA SCHEDULING

In this section, the communication network load problem resulting from large volume image data transmission in NVSC systems is discussed. The goal of this section is to find the trade-off between the control performance and the communication network load of the NVSC system. A scheduling mechanism is proposed to reduce the communication network load. Within the context of this paper, sending rate scheduling means to i) determine an appropriate sending interval d_k at time instant t_k based on the tracking error, and ii) to achieve desired optimal occurrence probabilities of different sending intervals. The details of the designed sending rate scheduling meeting these two requirements are introduced below.

A. Cost Function

For further technical development, assume each sampling interval in (1) has the occurrence probability

$$P\{d^m\} = p_m^d, \quad \sum_{m=1}^M p_m^d = 1. \quad (11)$$

To determine the optimal occurrence probabilities of $p_m^d, m = 1, \dots, M$, we consider here a cost function which jointly optimizes the communication network load and the control performance

$$J = \lim_{T \rightarrow \infty} \mathbb{E} \left\{ \int_0^T \mathbf{x}^T(t) R \mathbf{x}(t) dt + \frac{1}{T} \int_0^T C(d_k(t)) dt \right\}, \quad (12)$$

where $R \in \mathbb{R}^{n \times n}$, $R > 0$, and $C(d_k(t))$ denotes the communication network cost associated with the sending interval

$$C(d_k(t)) = C_m, \quad \text{if } d_k(t) = d^m, \quad m = 1, \dots, M,$$

where C_m is the communication network cost factor associated with d^m , $C_m \in \mathbb{R}$. Here we assume that the communication network cost is a monotonically decreasing function of the sending interval. With a smaller sending interval d^m , a higher communication network cost factor C_m is assigned.

The first term on the right-hand side in (12) represents the control performance. Given the boundary values s_i of the delay intervals, and assume the probability distributions of transmission and computation delays are constant and known, the expected value of the control performance is shown to be bounded by a positive scalar $\bar{L}_{\text{per}}(p_1^d, \dots, p_M^d)$ in [66]. The second term denotes the communication network cost over the run time T . With the occurrence probability of each sending interval defined in (11), the expected value of the normalized communication network usage becomes

$$\lim_{T \rightarrow \infty} \mathbb{E} \left\{ \frac{1}{T} \int_0^T C(d_k(t)) dt \right\} = \sum_{m=1}^M p_m^d C_m.$$

Then the stochastic cost function in (12) is bounded from above by a deterministic function

$$J(p_1^d, \dots, p_M^d) \leq \bar{L}_{\text{per}}(p_1^d, \dots, p_M^d) + \sum_{m=1}^M p_m^d C_m, \quad (13)$$

depending on the probability distribution of the sampling intervals. Minimizing (13) results in an optimal trade-off between the control performance and the communication network cost.

Note that the stability for the system in (10) depends also on the probability distributions of the sending interval $p_m^d, m = 1, \dots, M$, as derived in [63]. Therefore, stability constraints have to be considered in the optimization problem. It becomes a static optimization problem with linear matrix inequality constraints, see details in [63]. A local minimum can be found by the function `fmincon` from the optimization toolbox in Matlab.

B. Policy of sending rate scheduling

Instead of selecting the sending interval d randomly during the procedure, the tracking error is considered for sending rate scheduling. Generally speaking, a smaller sending interval should be assigned when the tracking error becomes larger. This is a reasonable heuristics for a visual servo control system with a camera-in-hand structure, because a large tracking error may result in losing the object in the image plane. However, by minimizing the cost function (13) in Section V-A, the occurrence probability of each sending interval is predetermined, which places additional constraint on the scheduling. Therefore, for sending rate scheduling both the tracking error and the optimal sending rate probability are taken into consideration.

The proposed sending rate scheduling algorithm is shown in Fig. 8. There are two steps in sending rate scheduling: the first step is to determine the sending interval d_k at time instant t_k , while the second step is to distribute images to processing nodes with determined sending interval. The details of the two steps are introduced below.

Step 1) Determination of the sending interval: In order to reduce the communication network load due to the image data transmission, the tracking error \mathbf{X} in Cartesian space is considered in selecting an appropriate sending interval at the steaming server side. According to (1), there are M sending intervals available for scheduling. To select the sending interval d_k according to the tracking error, the tracking error \mathbf{X} is categorized also into M intervals

$$r_{m-1} \leq |\mathbf{X}| < r_m, \quad m = 1, \dots, M, \quad (14)$$

where $0 < r_{m-1} < r_m$ are the bounds of the intervals, $r_0 = 0$ and $r_M \rightarrow \infty$.

As pointed out previously, both the tracking error X and the optimal occurrence probabilities $p_m^d, m = 1, \dots, M$, are utilized for selecting the sending interval d_k at time instant t_k . Assume the tracking error falls into the interval $r_{m-1} \leq |\mathbf{X}| < r_m$. If till the time instant t_k the measured occurrence probability of the sending interval d^m , denoted by $p_m^d(t_k)$, is smaller than the pre-determined optimal occurrence probability p_m^d , the sending interval d^m is selected. Otherwise, another sending interval should be chosen from the sending intervals, whose optimal occurrence probabilities are not reached till the time instant t_k . Denote the set consisting of these sending intervals with \underline{D} , and it becomes a subset of D in (1): $\underline{D} \subset D$. To select an

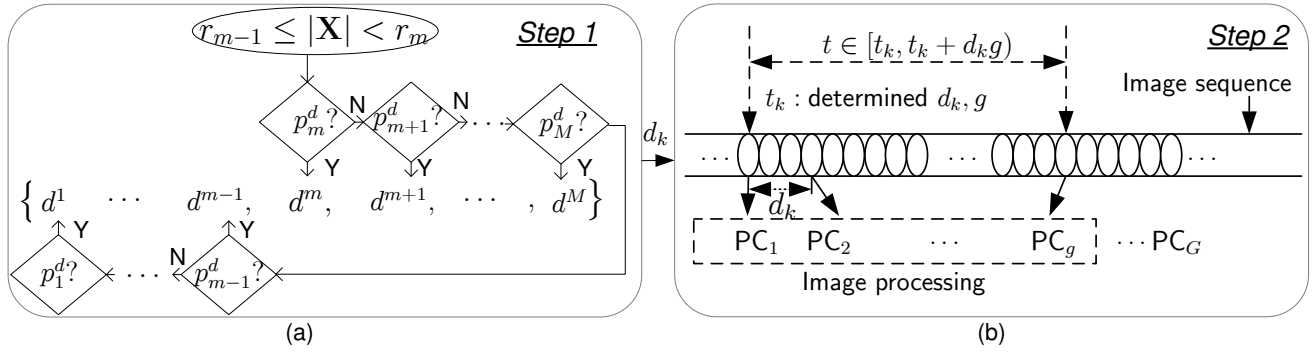


Figure 8: Sending rate scheduling at the streaming server side. a): d_k determination considering the tracking error \mathbf{X} and occurrence probability p_m^d at time instant t_k . The tracking error \mathbf{X} is categorized into m intervals with r_m denoting the bounds of the intervals. b): image distribution with determined sending interval d_k , and the number of processing node is denoted by g .

appropriate sending interval, the set \underline{D} is further divided into two groups

$$\begin{aligned} \underline{D}_1 &= \{d^m | d^m \in \underline{D}, d^m < d^m\}, \\ \underline{D}_2 &= \{d^m | d^m \in \underline{D}, d^m > d^m\}, \end{aligned}$$

where d^m represents the element in \underline{D} . The sending intervals in \underline{D}_1 and \underline{D}_2 are assigned with different priorities:

$$\begin{aligned} \text{Pri}(\underline{D}_1) &> \text{Pri}(\underline{D}_2), \\ \text{Pri}(d^m) &< \text{Pri}(d^{m+1}), \quad d^m \in \underline{D}_1, \\ \text{Pri}(d^m) &> \text{Pri}(d^{m+1}), \quad d^m \in \underline{D}_2 \end{aligned}$$

where $\text{Pri}(\cdot)$ denotes the priority of the set or the element during the selection of the sending interval, see also Fig. 8(a) for visualization. The rule of assigning the priority above is reasonable, since for a tracking problem studied in this work, the first task is to keep the object stay in the field of view.

Finally, the selection algorithm is summarized as

$$d_k = \begin{cases} d^m, & \text{if } p_m^d(t_k) < p_m^d, \\ d^m, & d^m \in \underline{D}, \text{ otherwise,} \end{cases} \quad (15)$$

where the sending interval d^m has the maximum priority of being selected among the elements in the set \underline{D} . After d_k is determined, the number of necessary processing nodes g can be found according to (2)(3).

Step 2) Image distribution: With determined sending interval d_k from Step 1), the streaming server starts to send images to processing nodes. The details of image distribution are illustrated in Fig. 8(b) as well as in Algorithm 1: a) First, at the instant t_k , the number of necessary processing nodes g are determined according to d_k ; b) Then, for the time period $t \in [t_k, t_k + d_k g)$, the image sequence with sampling interval d_k is sent to the processing nodes available over the communication network; c) After sending the images, steps a) and b) are repeated from the instant $t_k + d_k g$ and so on.

With the algorithm introduced above, the communication network load caused by image transmission is effectively reduced. Meanwhile, an acceptable control performance is maintained.

Remark 1: Note that in the current work, only the tracking error and the optimal occurrence probabilities of the sending interval are considered in the sending interval determination. In the future, network congestion and the packet loss rate will be integrated in design of sending rate scheduling as well.

Algorithm 1 Image distribution at streaming server

Require: d_k

Task: send image I_t (image sampled at time instant t , $t \in [t_k, t_k + d_k g)$) to processing node with determined d_k

sorting $|\mathbf{X}|$ with (14) \rightarrow determine d_k according to (15) \rightarrow determine g according to (2)(3).

for $t = [t_k, t_k + d_k g)$ **do**

$t_k = t$;

send I_{t_k} to PC, (PC $\in \{PC_1, \dots, PC_g\}$ represents the processing node available at t_k);

$t = t_k + d_k$;

end for

VI. EXPERIMENTS

In order to validate the efficacy of the proposed sending rate scheduling algorithm, the proposed approach is compared with the standard non-scheduling approach. Experiments for a 3D tracking problem with a 14-DoF dual-arm [67] are conducted. The end-effector of the right arm (7-DoF) is equipped with a high-speed camera (Mikrotron MC1363), while the object is mounted on the end-effector of the left arm (7-DoF), as shown in Fig.9.

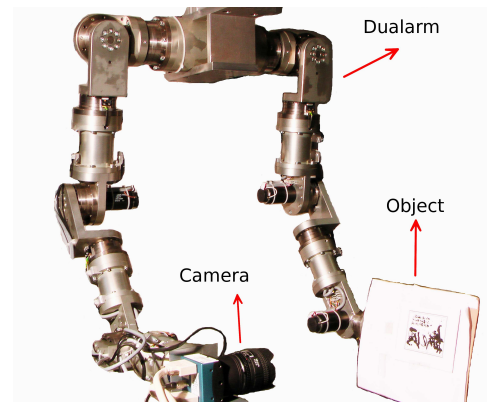


Figure 9: Experimental setup with 7-DoF robot arm, high-speed camera and moving object.

In the experiments, the camera runs at a high sampling rate of 400Hz@640 \times 480 pixels approximating an event-

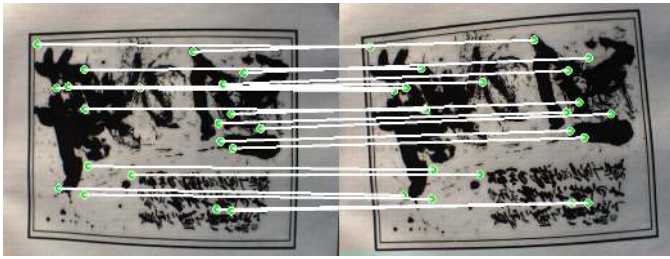


Figure 10: Image processing results (feature matching). Left side: reference image; right side: current image. Green point: SURF feature; white line: matched feature pair; number of matched feature pair: 20; image processing time: 18.3 ms.

based sampling system. Two sending rates $\{40 \text{ Hz}, 80 \text{ Hz}\}$ are selected for sending rate scheduling. To determine the intrinsic parameters (e.g. focal length, principal point, and distortion coefficients) of the camera, it is calibrated off-line before carrying out the experiments. The dual-arm is connected to a PC running real-time RTAI/Linux. The control loop is implemented in MATLAB/SIMULINK blocksets. Standalone real-time code is generated directly from the SIMULINK models. Two other PCs, PC₁ and PC₂ run image processing in parallel.

A. Image Processing

On the processing nodes, image processing algorithms for 3D pose estimation are implemented. First, the image features are extracted from the image. To increase the accuracy of the pose estimation and speed up visual feedback, a GPU implementation of SURF [17] is applied by exploiting its massive parallel processing capability. After feature matching [68], RANSAC (RANDOM Sample Consensus) [69] is selected for outliers rejection. Fig. 10 shows an example of a matching result during the experiment. Finally, the relative pose between the camera and the object is obtained by calculating the extrinsic parameters (relative translation and rotation) of the camera based on the matched feature pairs.

B. Control Design

After linearization of the manipulator equations through computed torque control the system can be represented by seven decoupled subsystems. For each joint, the overall time delay τ is categorized into two intervals with $s_1 = 45 \text{ ms}$. By optimizing the cost function (13) with the stability constraints derived in [63], the occurrence probabilities of the two sending intervals are obtained: $p_{40\text{Hz}}^d = 48\%$, $p_{80\text{Hz}}^d = 52\%$. The associated stabilizing state feedback gains for each joint with $\tau \leq s_1$ are $K_{p1} = [120, 120, 60, 60, 30, 30, 30]$, $K_{d1} = [2, 2, 2, 2, 1, 1, 1]$. The gains for $\tau > s_1$ are $K_{p2} = [96, 96, 48, 48, 24, 24, 24]$, $K_{d2} = [2, 2, 2, 2, 1, 1, 1]$. The tracking error threshold $r_1 = 0.01 \text{ m}$ is heuristically selected.

To compare the control performance and the network load, a standard networked visual servo control system without sending rate scheduling is implemented. Both PC₁ and PC₂ run at 40 Hz, resulting in high communication network load. The occurrence probabilities of the sending intervals become: $p_{40\text{Hz}}^d = 0\%$, $p_{80\text{Hz}}^d = 100\%$. The

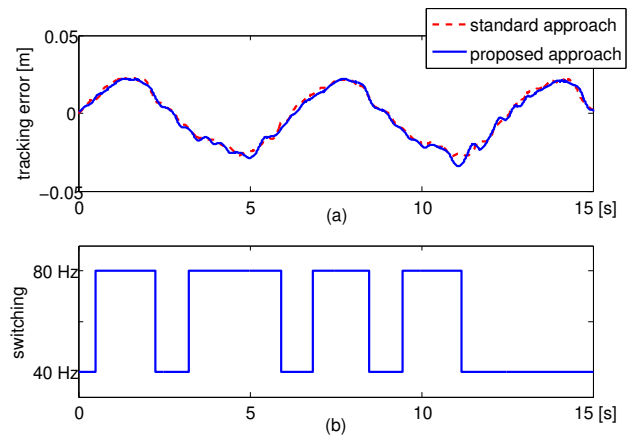


Figure 11: Tracking results. a): the control error evolution of the proposed approach with data scheduling (solid line) and the standard approach without data scheduling (dash line). b): Sending rate switching between 80 Hz and 40 Hz for the proposed approach in the experiment. 80 Hz: 52%, 40 Hz: 48%.

overall time delay is categorized only into one interval. The controller with gains $K_{p1} = [120, 120, 60, 60, 30, 30, 30]$, $K_{d1} = [2, 2, 2, 2, 1, 1, 1]$ is applied.

C. Sending Rate Scheduling

According to the scheduling policy described in Section V, if at time instant t_k the tracking error $|\mathbf{X}| > r_1$ and $p_{80\text{Hz}}^d(t_k) < p_{80\text{Hz}}^d$, the nRTP steaming server sends the images at 80 Hz to PC₁ and PC₂. Otherwise, it sends the images at 40 Hz to only one PC. With this approach, the sampling rate of the feedback is bounded by [40 Hz, 80 Hz]. The operating frequency of different component in NVSC systems is listed in Table I.

Table I: Frequencies of components in NVSC system.

Camera	PC ₁	PC ₂	Visual feedback	Controller
400 Hz	[0, 40Hz]	[0, 40Hz]	[40Hz, 80Hz]	1000 Hz

D. Experimental Results

A sinusoidal function, which has an amplitude of 0.15 m and a frequency of 1 rad/s, serves as the desired trajectory \mathbf{X}_d for the object moving along one axis at the Cartesian space. The tracking error \mathbf{X} between the object motion and camera motion is discussed for performance evaluation.

The results are shown in Fig. 11, and the distribution of delays behaved as desired. As shown in Fig. 11(a), a comparable control performance is achieved with and without sending rate scheduling. The proposed approach with sending rate scheduling has a mean tracking error $|\mathbf{X}|_{\text{mean}} = 1.73 \text{ cm}$, which is similar to the mean tracking error of the high sampling rate design approach ($|\mathbf{X}|_{\text{mean}} = 1.67 \text{ cm}$), see Table II. However, the communication network load is decreased by 24% compared to the high sampling rate design approach.

The experimental results demonstrate a comparable control performance of the proposed approach with lower communication network load than that of the non-scheduling approach.

Table II: Control performance and communication network load.

	$ \mathbf{X} _{\text{mean}}$ [cm]	$ \mathbf{X} _{\text{var}}$ [cm ²]	Communication network load [Mbps]
with scheduling	1.73 (+4%)	0.09	146 (-24%)
high sampling rate	1.67	0.018	192

VII. CONCLUSION

This paper presents a novel analysis and design approach for networked visual servo control systems with cloud computing and sending rate scheduling. Based on a real-time transport protocol nCrTP for image data transmission, a GPGPU cluster is built for cloud image processing which enables high-sampling-rate visual feedback. A switching control law is applied considering the varying feedback time delay caused by image processing and data transmission. Moreover, a sending rate scheduling at the streaming server in NVSC systems is designed based on the tracking error to reduce the communication network load caused by large volume image data transmission. The proposed approach is validated by experiments on a 14-DoF dual-arm. The results demonstrate comparable control performance of the proposed approach with lower communication network cost than the non-scheduling counterpart.

The future work is concerned with optimizing the sending rate scheduling by considering not only tracking error, but also network congestion, packet loss and motion dynamics of the object.

VIII. ACKNOWLEDGMENTS

This work is supported in part by the German Research Foundation (DFG) within the 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, the Priority Program SPP 1305 "Control Theory of Digitally Networked Dynamical Systems", and the Institute for Advanced Study (IAS), Technische Universität München, see also www.tum-ias.de.

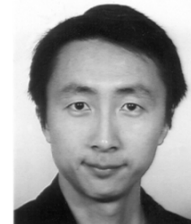
REFERENCES

- [1] Y. Shirai and H. Inoue, "Guiding a robot by visual feedback in assembling tasks," *Pattern Recognition*, vol. 5, no. 2, pp. 99–106, 1973.
- [2] L. Weiss, A. Sanderson, and C. Neuman, "Dynamic sensor-based control of robots with visual feedback," *IEEE Journal of Robotics and Automation*, vol. 3, no. 5, pp. 404–417, 1987.
- [3] K. Hashimoto, *Visual servoing: real-time control of robot manipulators based on visual sensory feedback*. World scientific, 1993.
- [4] P. Corke, "Visual control of robot manipulators – a review," in *Visual Servoing*, pp. 1–31, World Scientific, 1994.
- [5] F. Chaumette and S. Hutchinson, "Visual servo control. I. Basic approaches," *Robotics & Automation Magazine, IEEE*, vol. 13, no. 4, pp. 82–90, 2006.
- [6] S. Funiak, C. Guestrin, M. Paskin, and R. Sukthankar, "Distributed localization of networked cameras," in *Proc. of the 5th inter. conf. on Information processing in sensor networks*, pp. 34–42, ACM, 2006.
- [7] B. Wilburn, N. Joshi, V. Vaish, E. Talvala, E. Antunez, A. Barth, A. Adams, M. Horowitz, and M. Levoy, "High performance imaging using large camera arrays," *ACM Transactions on Graphics (TOG)*, vol. 24, no. 3, pp. 765–776, 2005.
- [8] B. Weber and K. Kühnlenz, "Visual servoing using triangulation with an omnidirectional multi-camera system," in *Proc. of the International Conference on Control, Automation, Robotics and Vision (ICARCV)*, pp. 1440–1445, 2010.
- [9] F. Chaumette, "Image moments: a general and useful set of features for visual servoing," *IEEE Trans. Robot.*, vol. 20, no. 4, pp. 713–723, 2004.
- [10] D. Lowe, "Distinctive image features from scale-invariant keypoints," *Inter. journal of computer vision*, vol. 60, no. 2, pp. 91–110, 2004.
- [11] M. Nixon and A. Aguado, *Feature extraction and image processing*. Academic Press, 2008.
- [12] S. Hutchinson, G. Hager, and P. Corke, "A tutorial on visual servo control," *IEEE Trans. Robot. Automat.*, vol. 12, no. 5, pp. 651–670, 1996.
- [13] E. Malis, F. Chaumette, and S. Boudet, "2 1/2 d visual servoing," *IEEE Trans. Robot. Automat.*, vol. 15, no. 2, pp. 238–250, 1999.
- [14] P. Corke and M. Good, "Dynamic effects in visual closed-loop systems," *IEEE Trans. Robot. Automat.*, vol. 12, no. 5, pp. 671–683, 1996.
- [15] N. Gans and S. Hutchinson, "Stable visual servoing through hybrid switched-system control," *IEEE Trans. Robot.*, vol. 23, no. 3, pp. 530–540, 2007.
- [16] H. Wu, T. Zhang, A. Borst, K. Kühnlenz, and M. Buss, "An explorative study of visual servo control with insect-inspired Reichardt-model," in *Proc. of IEEE Inter. Conf. Robot. Auto. (ICRA)*, pp. 345–350, 2009.
- [17] H. Bay, T. Tuytelaars, and L. Van Gool, "Surf: Speeded up robust features," *Computer Vision–ECCV 2006*, pp. 404–417, 2006.
- [18] A. Namiki, Y. Nakabo, I. Ishii, and M. Ishikawa, "High speed grasping using visual and force feedback," in *Proc. IEEE Inter. Conf. on Robot. Auto. (ICRA)*, vol. 4, pp. 3195–3200, 1999.
- [19] T. Zhang, H. Wu, A. Borst, K. Kühnlenz, and M. Buss, "An FPGA implementation of insect-inspired motion detector for high-speed vision systems," in *Prod. IEEE Inter. Conf. Robot. Auto. (ICRA)*, pp. 335–340, 2008.
- [20] T. Xu, T. Pototschnig, K. Kühnlenz, and M. Buss, "A high-speed multi-GPU implementation of bottom-up attention using CUDA," in *Proc. IEEE Inter. Conf. Robot. Auto. (ICRA)*, pp. 41–47, 2009.
- [21] V. Sunderam *et al.*, "PVM: A framework for parallel distributed computing," *Concurrency Practice and Experience*, vol. 2, no. 4, pp. 315–339, 1990.
- [22] D. Henrich and T. Honiger, "Parallel processing approaches in robotics," in *Proc. IEEE International Symp. Indus. Elec. (ISIE'97)*, vol. 2, pp. 702–707, 1997.
- [23] R. Buyya, "High performance cluster computing: Architectures and systems, volume 1," *Prentice Hall PTR*, vol. 82, pp. 327–350, 1999.
- [24] X. Li, B. Veeravalli, and C. Ko, "Distributed image processing on a network of workstations," *International Journal of Computers and Applications*, vol. 25, no. 2, pp. 136–145, 2003.
- [25] H. Wu, L. Lou, C. Chen, S. Hirche, and K. Kühnlenz, "A framework of networked visual servo control system with distributed computation," in *Proc. of the 11th Inter. Conf. on Cont. Auto. Robot. & Visi. (ICARCV)*, pp. 1466–1471, 2010.
- [26] H. Li, B. Manjunath, and S. Mitra, "Multisensor image fusion using the wavelet transform," *Graphical models and image processing*, vol. 57, no. 3, pp. 235–245, 1995.
- [27] G. Cena and A. Valenzano, "An improved CAN fieldbus for industrial applications," *IEEE Transactions on Industrial Electronics*, vol. 44, no. 4, pp. 553–564, 1997.
- [28] E. Tovar and F. Vasques, "Real-time fieldbus communications using profibus networks," *IEEE Transactions on Industrial Electronics*, vol. 46, no. 6, pp. 1241–1251, 1999.
- [29] T. Yang, "Networked control system: a brief survey," in *IEE Proc. - Control Theory Appl.*, vol. 153, pp. 403–412, IET, 2006.
- [30] J. Wang and B. Ravindran, "Time-utility function-driven switched ethernet: Packet scheduling algorithm, implementation, and feasibility analysis," *IEEE Tran. on Para. and Dist. Syst.*, pp. 119–133, 2004.
- [31] D. Jansen and H. Buttner, "Real-time Ethernet: the EtherCAT solution," *Computing and Control Engineering*, vol. 15, pp. 16–21, 2004.
- [32] J. Feld, "Profinet-scalable factory communication for all applications," in *Proc. IEEE Inter. Workshop on Fact. Comm. Syst.*, pp. 33–38, 2004.
- [33] G. Walsh and H. Ye, "Scheduling of networked control systems," *Control Systems Magazine, IEEE*, vol. 21, no. 1, pp. 57–65, 2001.
- [34] D. Henriksson and A. Cervin, "Optimal on-line sampling period assignment for real-time control tasks based on plant state information," in *Proc. 44th IEEE Conf. Deci. Cont. (CDC)*, pp. 4469–4474, 2005.
- [35] M. Gaid, A. Çela, and Y. Hamam, "Optimal integrated control and scheduling of networked control systems with communication constraints: Application to a car suspension system," *IEEE Transactions on Control Systems Technology*, vol. 14, no. 4, pp. 776–787, 2006.
- [36] P. Otanez, J. Moyne, and D. Tilbury, "Using deadbands to reduce communication in networked control systems," in *Proceedings of the American Control Conference*, vol. 4, pp. 3015–3020, 2002.

- [37] S. Hirche, P. Hinterseer, E. Steinbach, and M. Buss, "Towards deadband control in networked teleoperation systems," in *Proc. Int. Fed. Autom. Control (IFAC), Prague, Czech Republic*, 2005.
- [38] K. Åström and B. Bernhardsson, "Comparison of Riemann and Lebesgue sampling for first order stochastic systems," in *Proc. of the 41st IEEE Conference on Decision and Control*, vol. 2, pp. 2011–2016, 2002.
- [39] A. Molin and S. Hirche, "On LQG joint optimal scheduling and control under communication constraints," in *Proc. of the 48th IEEE Conf. on Decision and Control (CDC)*, pp. 5832–5838, 2009.
- [40] A. Molin and S. Hirche, "Structural characterization of optimal event-based controllers for linear stochastic systems," in *Proc. of the 49th IEEE Conf. on Decision and Control (CDC)*, pp. 3227–3233, 2010.
- [41] A. Molin and S. Hirche, "On the optimality of certainty equivalence for event-triggered control systems," *IEEE Transactions on Automatic Control*, (2012), to appear.
- [42] K. Shin and X. Cui, "Computing time delay and its effects on real-time control systems," *IEEE Transactions on Control Systems Technology*, vol. 3, no. 2, pp. 218–224, 1995.
- [43] T. Chen and B. Francis, *Optimal sampled-data control systems*, vol. 124. Springer London, 1995.
- [44] L. Xiao, A. Hassibi, and J. How, "Control with random communication delays via a discrete-time jump system approach," in *Proceedings of American Control Conference*, vol. 3, pp. 2199–2204, 2000.
- [45] L. Zhang, Y. Shi, T. Chen, and B. Huang, "A new method for stabilization of networked control systems with random delays," *IEEE Trans. Automat. Control*, vol. 50, no. 8, pp. 1177–1181, 2005.
- [46] C. Chen, S. Hirche, and M. Buss, "Controller design and experimental validation for networked control systems with time-varying random delay," *Journal of the Society of Instrument and Control Engineers*, vol. 47, no. 8, pp. 676–685, 2008.
- [47] S. Hirche, C. Chen, and M. Buss, "Performance oriented control over networks: switching controllers and switched time delay," *Asian Journal of Control*, vol. 10, no. 1, pp. 24–33, 2008.
- [48] C. Chen, A. Molin, and S. Hirche, "Guaranteed cost control over quality-of-service networks," in *Proc. of the European Control Conference, Budapest, Hungary*, pp. 2355–2360, 2009.
- [49] C. Chen, S. Hirche, and M. Buss, "Stability, stabilization and experiments for networked control systems with random time delay," in *Proc. of the American Control Conference*, pp. 1552–1557, 2008.
- [50] C. Chen, S. Hirche, and M. Buss, "Sampled-data networked control systems with random time delay," in *Proc. the 17th World Congress the International Federation of Automatic Control*, pp. 11594–11599, 2008.
- [51] S. Hirche, C. Chen, and M. Buss, "Performance oriented control over networks - switching controllers and switched time delay," in *Proc. 45th IEEE Conf. Decision and Control, San Diego*, pp. 4999–5005, 2006.
- [52] C. Chen, S. Hirche, and M. Buss, "Towards quality-of-service control of networked control systems: a switched time delay systems approach," in *Proc. IEEE Inter. Conf. on Cont. Appl. (CCA)*, pp. 1325–1330, 2006.
- [53] R. Gupta and M. Chow, "Networked control system: overview and research trends," *IEEE Transactions on Industrial Electronics*, vol. 57, no. 7, pp. 2527–2535, 2010.
- [54] H. Schulzrinne, S. Casner, R. Frederick, and V. Jacobson, "RFC3550: RTP: A transport protocol for real-time applications," *RFC Editor United States*, 2003.
- [55] <http://www.gnu.org/software/ccrtp/>.
- [56] W. Stevens, "TCP congestion control," *Internet Engineering Task Force, RFC2581*, 1999.
- [57] S. W. R. Wright, Gary R., *TCP/IP illustrated. Vol.2: The implementation*. Addison-Wesley Professional Computing Series, 1995.
- [58] H. Schulzrinne, "Rtp profile for audio and video conferences with minimal control," *Internet Engineering Task Force, RFC3551*.
- [59] L. Montestruque and P. Antsaklis, "Stability of model-based networked control systems with time-varying transmission times," *IEEE Transactions on Automatic Control*, vol. 49, no. 9, pp. 1562–1572, 2004.
- [60] L. Sciavicco and B. Siciliano, *Modelling and control of robot manipulators*. Springer Verlag, 2000.
- [61] T. Chen and B. Francis, *Optimal Sampled-Data Control Systems*. Berlin: Springer-Verlag, 1995.
- [62] E. Fridman, A. Seuret, and J.-P. Richard, "Robust sampled-data stabilization of linear systems: an input delay approach," *Automatica*, vol. 40, pp. 1441–1446, 2004.
- [63] C. Chen, H. Wu, K. Kühnlenz, and Hirche, "Switching control for a networked vision-based control system," *Automatisierungstechnik*, vol. 59, pp. 124–133, 2010.
- [64] H. Wu, C. Chen, J. Feng, K. Kühnlenz, and S. Hirche, "A switching control law for a networked visual servo control system," in *Proc. IEEE Inter. Conf. Robot. Auto. (ICRA), Anchorage*, pp. 5556–5563, 2010.
- [65] H. Wu, L. Lou, C. Chen, S. Hirche, and K. Kühnlenz, "Performance-oriented networked visual servo control with sending rate scheduling," in *Proc. IEEE Inter. Conf. Robot. Auto. (ICRA)*, pp. 6180–6185, 2011.
- [66] H. Wu, L. Lou, C. Chen, K. Kühnlenz, and S. Hirche, "Distributed computation and data scheduling for networked visual servo control systems," in *Proceedings of IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 6216–6221, 2010.
- [67] B. Stanczyk, *Development and Control of an Anthropomorphic Telerobotic System*. PhD thesis, Technische Universität München, 2006.
- [68] M. Muja and D. G. Lowe, "Fast approximate nearest neighbors with automatic algorithm configuration," in *Proc. Inter. Conf. Comp. Visi. Theo. and Appl. (VISSAPP'09)*, pp. 331–340, INSTICC Press, 2009.
- [69] M. A. Fischer and R. C. Bolles, "Random sampled consensus: A paradigm for modeling fitting with applications to image analysis and automated cartography," in *Communications of the ACM*, vol. 2, pp. 381–395, 1981.



Haiyan Wu was a PhD student at the Institute of Automatic Control Engineering at Technische Universität München. She is now an external lecturer at Department of Electrical Engineering, Technical University of Denmark. Her research interests include networked visual servo control systems, high-speed vision-based robot control and bio-inspired vision.



Lei Lou is a Ph.D. student at the Institute of Automatic Control Engineering at Technische Universität München. His research interests include visual servoing, Image feature extraction and 3D reconstruction.



Chih-Chung Chen received his PhD in Electrical Engineering from Technische Universität München in 2010. He is now a design engineer with ASML. His research interests include networked control systems, stochastic time delay systems and stochastic hybrid systems.



Sandra Hirche is a professor heading the Associate Institute for Information-oriented Control in the Department of Electrical Engineering and Information Technology, Technische Universität München. She received the diploma engineer degree in Mechanical Engineering and Transport Systems in 2002 from the Technical University Berlin, Germany, and the Doctor of Engineering degree in Electrical Engineering and Information Technology in 2005 from the Technische Universität München, Munich, Germany.

In 2005 she has been awarded a PostDoc scholarship from the Japanese Society for the Promotion of Science (JSPS) for two years at the Fujita Lab at the Tokyo Institute of Technology, Tokyo, Japan. Her research interests include networked dynamical systems, control over communication networks, cooperative and distributed control, event-triggered control, human-in-the-loop control, haptics, physical human-robot interaction, multimodal telepresence



Kolja Kühnlenz Kolja Kühnlenz is a Senior Lecturer at the Institute of Automatic Control Engineering (LSR) and Carl von Linde Junior Fellow at the Institute for Advanced Study (IAS) at Technische Universität München (TUM). He is also chairman of the CoTeSys Graduate Center within and member of the Extended Board of the DFG cluster of excellence "Cognition for Technical Systems - CoTeSys". His research interests include computer vision, robotics, and human-robot interaction.