A NETWORKED HIGH-SPEED VISION SYSTEM FOR 1,000-FPS VISUAL FEATURE COMMUNICATION

Shingo Kagami

Graduate School of
Information Sciences,
Tohoku University
6-6-01 Aramaki Aza Aoba,
Aoba-ku, Sendai, Japan.

*Shoichiro Saito**, *Takashi Komuro, and Masatoshi Ishikawa*

Graduate School of
Information Science and Technology,
University of Tokyo
7-3-1 Hongo,
Bunkyo-ku, Tokyo, Japan.

ABSTRACT

A networked high-speed vision system that employs as a vision sensor node a digital vision chip, a CMOS imager that integrates a digital processing element with a photo detector in each pixel is reported. High-speed visual feature information at the frame rate of 1,000 fps is transferred over the standard TCP/IP on 100BASE-TX Ethernet switching network. We evaluated the effectiveness of the system through experiments, and found that the system can convey visual information with sufficiently small latency.

Index Terms— vision chip, real-time network, smart camera, high-speed vision

1. INTRODUCTION

During the last decade, cooperative visual detection and tracking by distributed cameras have been vigorously investigated [1, 2]. Most of the reported systems use standard CCD cameras, which are not fast enough for capturing rapid irregular motion of targets.

We have been engaged in development of a high-speed vision system of which the frame rate is over 1,000 fps [3, 4]. The developed system employs a smart camera approach based on a computational CMOS image sensor called the digital vision chip. In each pixel of the CMOS image sensor, a digital programmable processing element (PE) is integrated with a photo detector (PD). It performs pixel-parallel processing over images immediately after they are captured without time-consuming and power-consuming image transfer from a sensor to a processor. The system offers powerful and flexible image sensing and processing capabilities, and achieves visual processing at a frame rate over 1,000 fps.

In this paper, we report our development for connecting these high-speed vision systems with the standard IP network using off-the-shelf 100BASE-TX Ethernet switches so that they can act as sensor nodes in a high-speed vision network system. By employing the standard IP and Ethernet network, thanks to its versatility and interoperability, high-speed visual information obtained by the vision nodes will be communicated not only to each other, but also to various information appliances that have been already connected with the network and ubiquitously available.

In our system, raw images are not communicated directly, but instead only feature values, such as target positions and shape features, are communicated utilizing the smart camera nature. Thus the most important technical point of this system development is the latency performance of real-time communication instead of the communication bandwidth. For applications that require 1,000-fps visual feedback such as high-speed robot control, the latency on the order of one millisecond needs to be achieved.

In the standard TCP/IP network framework, many schemes for quality of service (QoS) ensuring with respect to, for example, bandwidth or latency have been standardized, including priority control of frame delivery at the Ethernet switches and transport-level real-time protocols [5, 6]. Most of them, however, are designed mainly for multimedia applications such as video and voice communications, and not for hard real time communications with the latency on the order of one millisecond. Hard real time extensions of the Ethernet have been actively developed in the area of industrial field buses [7, 8]. Most of them use special hardwares to guarantee hard real time communication, and/or require special network topology configurations.

Related work on high frame rate visual feature communication can rather be found in the area of the optical motion capture system. In the Vicon system [9], many high frame rate image sensors are connected through Gigabit Ethernet switches and information of marker positions are delivered to the host computer for online computation of motion analysis. This is a good example that shows that Ethernet-based communication systems for high-speed visions are promising. However, to the knowledge of the authors, evaluation of real-

*Shoichiro Saito is currenty with NTT Cyber Space Laboratories, NTT Corporation, 3-9-11 Midori-cho, Musashino-shi, Tokyo, Japan.

1-4244-1354-0/07/$25.00 ©2007 IEEE

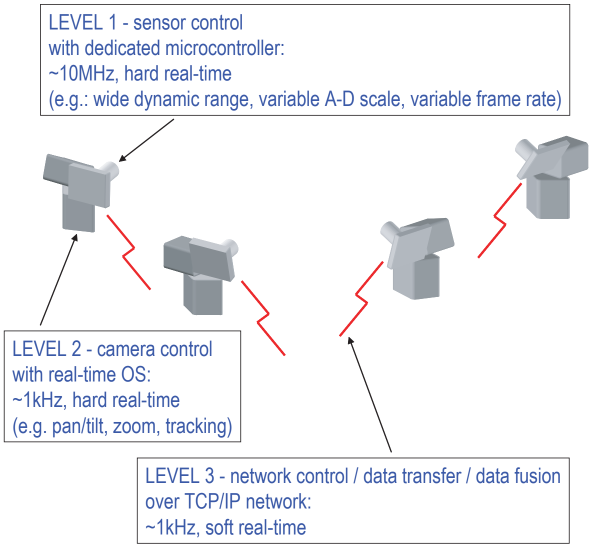
LEVEL 1 - sensor control
with dedicated microcontroller:
~10MHz, hard real-time
(e.g.: wide dynamic range, variable A-D scale, variable frame rate)
LEVEL 2 - camera control
with real-time OS:
~1kHz, hard real-time
(e.g. pan/tilt, zoom, tracking)
LEVEL 3 - network control / data transfer / data fusion
over TCP/IP network:
~1kHz, soft real-time

Fig. 1. Multi-level real-time control structure for high-speed vision network.

time communication latency of the system has not been reported. Actually, unlike visual feedback applications, motion capture applications do not require millisecond-order latency.

We developed a real-time visual processing system in which the above described digital vision chip, its dedicated microcontroller, and an embedded microprocessor with TCP/IP network connectivity are implemented. This multi-level control structure contributes to offering appropriate real-time processing granularity for the corresponding system levels. Packet issuing timing of the real-time visual feature information are controlled by a real-time operation system (RTOS) running on the embedded microprocessor, of which the communication latency is experimentally evaluated.

The rest of the paper is organized as follows: In Section 2, the concept of multi-level real-time control structure is presented. In Section 3, the implementation of the developed vision node is described in detail. Section 4 describes the results of experimental evaluation of the system performance. In Section 5, the conclusion is given.

2. DESIGN CONCEPT OF THE MULTI-LEVEL REAL-TIME CONTROL STRUCTURE

Figure 1 shows the concept of the multi-level real-time control structure for high-speed vision network. From the viewpoint of the granularity of real-time processing, it consists of three deferent levels.

LEVEL 1 – Sensor Control Level: This is the finest-grained level, in which the dedicated microcontroller for the vision chip guarantees real-time operations at the granularity of its instruction cycle. This enables programmable control of such as wide dynamic range imaging, variable photo response curves and variable frame rates.

LEVEL 2 – Camera Control and Image Processing Level: In this second level, real-time processing at the granularity of a millisecond is guaranteed by the RTOS running on the embedded microprocessor. Visual processing tasks for each frame are scheduled at this level. If needed, pan/tilt tracking or zoom control of the active camera should be executed in this level.

In the Level 3 described below, strict hard real time communication is not guaranteed. Therefore, for example in a target tracking application, it may lose track of the target due to the loss of visual feature information packets. Nevertheless this Level-2 hard real time control enables the target to be kept being tracked by individual cameras, and this information can be used to recover the cooperative tracking by multiple cameras.

LEVEL 3 – Inter-Camera Control and Processing Level: This is the top level of cooperative tasks communicating over the TCP/IP network where no strict guarantees of real-time operations are available. Millisecond-order granularity of real-time processing is expected is this level, but unlike the Level 2, it is a soft real time level. Communication and fusion of visual features from multiple cameras take place at this level.

Since there are no hard real time guaranteeing mechanisms, some of visual feature packets might be suddenly delayed or lost. Thus prediction and interpolation of the visual features, or fusion of features from multiple cameras are mandatory. This can be justified by considering the nature of visual processing: Image understanding itself is a hard and complicated job, and always suffers from uncertainty and misrecognition. Therefore these higher-order postprocessing techniques are inherently required.

3. ARCHITECTURE AND IMPLEMENTATION

3.1. Digital Vision Chip

The whole structure of the implemented system is shown in Fig. 2. The digital vision chip contains 64×64 pixels on a 0.35-μ 3-layer metal CMOS chip with 5.4×5.4 mm^2 area [3]. Each PE in a pixel has a bit-serial ALU, composed of a full adder circuit combined with some input/output multiplexers and a carry register, and a 24-bit bit-wise random access local memory. The PE array is controlled in SIMD manner, and pixel-level parallel processing is carried out. As well as the filtering-like local image processing, global image processing such as centroid computation of a target area can be efficiently programmed. Image processing programs can be coded using a high-level language called SPE-C [10], which is a C-like language with variable-bit-length pixel-parallel integer types.

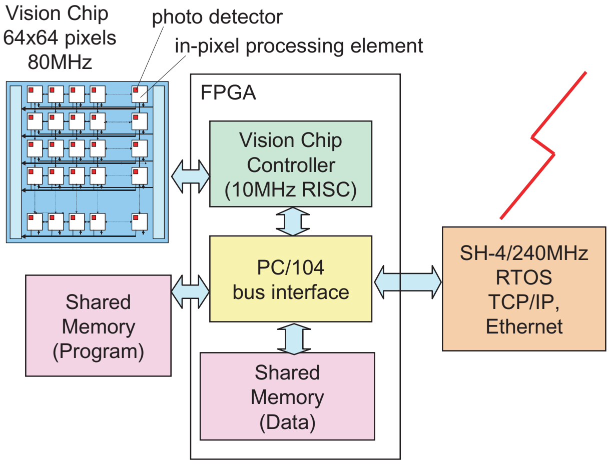

Fig. 2. Structure of the whole system.

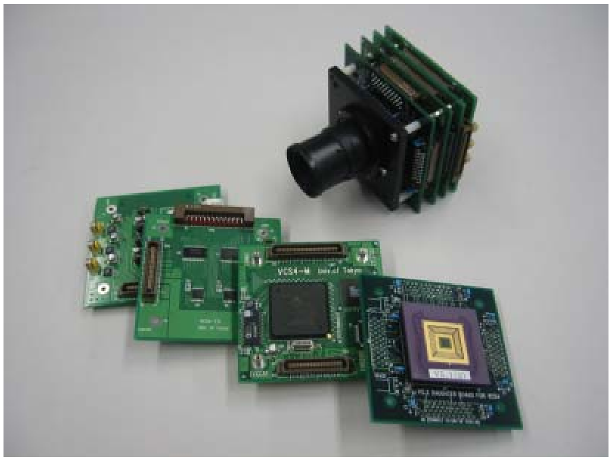

Fig. 3. Implementation of the Level-1 control system.

The PD in each pixel consists of a photodiode, a reset transistor and a comparator, which jointly operate as a programmable in-pixel analog-to-digital converter of incident light amount into digital pixel values when they are supplied with appropriate control signals. For these control signals to be generated on time, fine-grained real-time control structure is needed. The Level-1 real-time control for this purpose is implemented as a dedicated microcontroller.

3.2. Dedicated Microcontroller: LEVEL-1 control

The vision chip controller is a custom RISC processor implemented in an FPGA, in which a dedicated pipeline to control the PD/PE array (called the SIMD pipeline) is integrated with a standard 5-stages integer pipeline [4]. It is designed so that any combinational use of the two pipelines never causes dynamic pipeline stalls, and thus instruction-level real-time operations are guaranteed while maintaining the parallel processing throughput. For the results of programmable imaging, refer to the literature [4].

The FPGA in which the controller is implemented, I/O level conversion circuits for external interfaces, and DC power supply circuits are separately implemented in stackable 76 × 76 mm^2 boards. This structure offers highly flexible expandability of the system because the number of stacks is structurally unlimited. Photograph of the implementation is shown in Fig. 3.

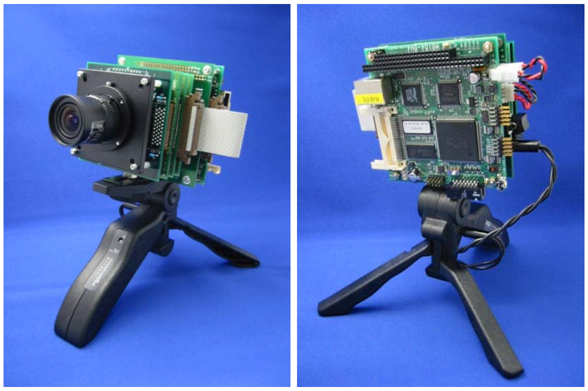

Fig. 4. Photograph of the developed high-speed vision node.

3.3. Embedded Microprocessor: LEVEL-2 control

While the Level-1 controller is effective for control of the vision chip, it is not suitable to cover the camera-level visual processing tasks and network communications, because its computation power besides the SIMD processing is severely limited. We need a more powerful computing processor for higher-level vision tasks and network packet handling.

Although it eventually should be implemented as a 76 × 76 mm^2 board stackable with other components of the system, we decided to employ a commercial embedded processor board at this early development stage in order to test our design. Photograph of the implemented high-speed vision node is shown in Fig. 4.

We employed an Alpha Project MS104-SH4 board, in which a Renesus SH-4 240-MHz processor (SH7750RF240) and an SMSC Ethernet chip LAN91C111 are implemented. A Mispo NORTi 4 with the TCP/IP stack, which is an RTOS complying with the μITRON 4.0 specification, is installed. Although the default OS tick time of the NORTi for this board was 10 ms, it was tuned up to 1 ms. The tasks running on this processor can communicate with the vision chip controller through the PC/104 bus interface implemented in the vision chip controller FPGA. From the software viewpoint, the memory space of the vision chip controller is mapped to an area of the SH-4 main memory, and can be used as a shared memory.

Our hierarchical control design does not depend on the

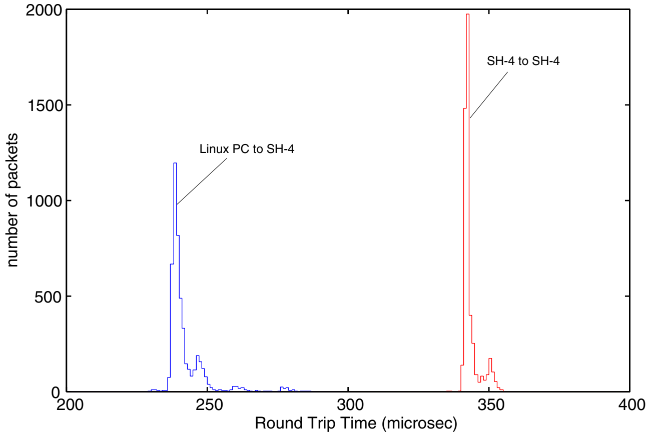

Fig. 5. Histograms of the round-trip times.

processor architecture and devices that are currently used. It will be possible to implement the proposed control design in more compact forms by employing, for example, microprocessors embedded in FPGAs in future work.

3.4. UDP/IP processing on RTOS: LEVEL-3 control

The Level-3 control is also implemented as a task set on the RTOS. Non real time communications such as program loading or configuration messages are implemented using TCP connections. On the other hand, visual feature communications are implemented as series of UDP packets to eliminate the overhead of the TCP protocol. A vision node can communicate with multiple peers including other vision nodes and host computers. After the communication is set up, a packet-sending task is waked up every one millisecond, and the packets are issued periodically.

At the receiving side, a receiving task is prepared, and waked up upon the receipt of a UDP packet from a sending peer. Received data are stored in a buffer space in the memory allocated for the corresponding peer, and then simply relayed to the next hop or aggregated with other data using some data fusion methods. In the current implementation, data fusion processing is executed within the sending task, which reads out the visual feature information from the local sensor, fuses it with the data in the reception buffers, and sends it to the next hop.

4. EVALUATION

4.1. Communication Latency

In order to estimate the communication latency, round-trip times were evaluated. Two developed vision nodes and a host computer were connected to an Allied-Telesis FS708XL 100BASE-TX 8-port switching hub. The host computer is a PC with a 850-MHz Pentium III processor running the Vine Linux 2.1.

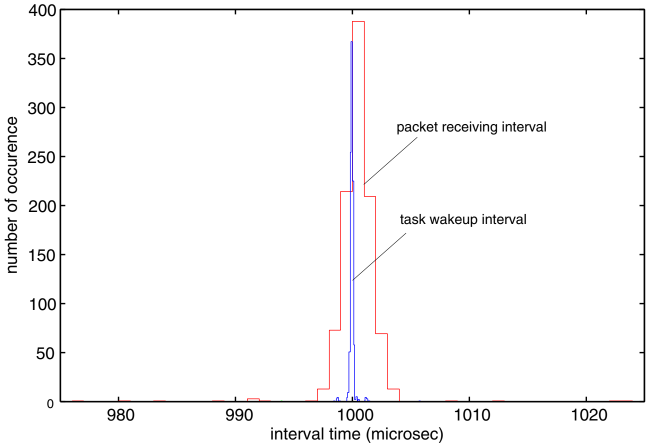

Fig. 6. Histogram of the sending task wakeup interval and the receiving interval.

The round-trip time of a 64-Byte UDP packet between the developed vision nodes, and the one between the vision node and the host computer were measured 5,000 times for each.

The histograms of the measured round-trip times are shown in Fig. 5. The blue line (on the right side) shows the round-trip time between the vision node and the host computer, and the red line (on the left side) shows the one between the two vision nodes. It should be noted that the former blue histogram (between the vision and the host) has several outliers around 260 μs and 275 μs, while the latter (between the two visions) has no outliers. In the both cases, the worst-case time fell within one millisecond with sufficient margins.

It is notable that the jitter of the round-trip time between the two visions is smaller than that of the one between the vision and the host, while the average round-trip time between the visions is longer. Because the computing power of the embedded microprocessor is weaker than that of a PC, it is not advantageous for the average performance. However, thanks to the priority control of the tasks by the RTOS, fairly small jitter of the communication latency is achieved.

4.2. Periodic Packet Delivery

Experiments of delivering visual feature information every one millisecond from the developed vision node to the Linux host computer were carried out. The sending task in the vision node was waked up by the timer event handler every one millisecond, and the task read out an image feature from the vision chip controller. Then it sent a UDP packet conveying the feature information to the host.

First, the elapsed time between the wakeup of the sending task and the UDP packet sending was measured 1,000 times. The elapsed time was 5.6 μs at minimum and 6.7 μs

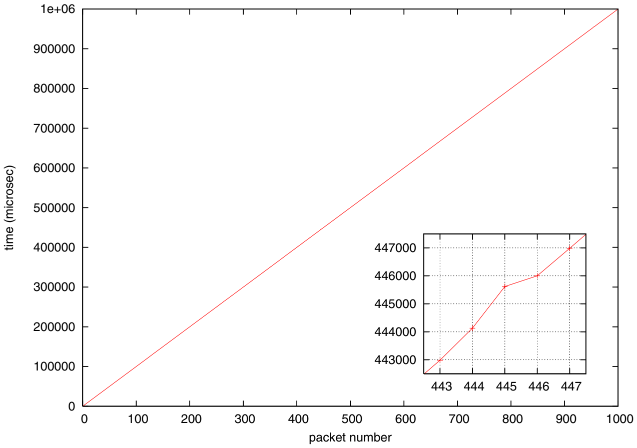

Fig. 7. Measured receiving time for each packet. Data around the maximum measured delay is enlarged and shown in the box at the lower right.

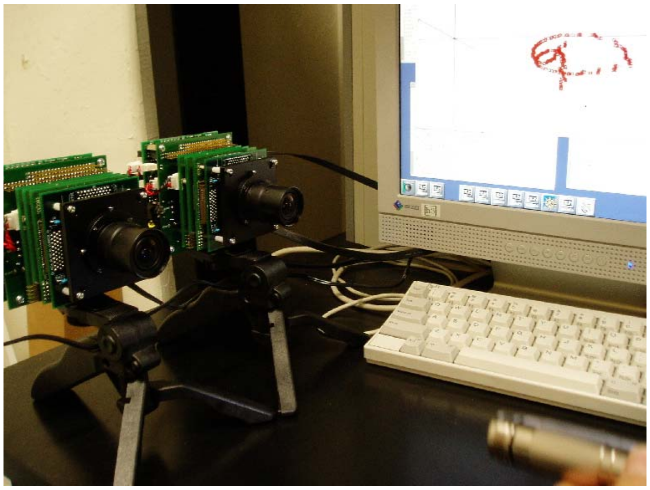

Fig. 8. Experiment setup.

at maximum. Thus the time required for accessing the vision chip controller memory can be negligible. (Note that image capturing and low-level image processing are executed in the vision chip without consuming the processing time of the embedded processor.)

The wakeup interval of the sending task at the vision node, and the packet receiving interval at the Linux host were also measured 1,000 times. The histograms are shown in Fig. 6. The blue narrower peak is the histogram of the wakeup intervals, and the red wider one is the histogram of the receiving intervals at the host.

It can be seen that the wakeup of the sending task is, and thus the packet sending is precisely periodic. On the other hand, the receiving intervals more widely varies. While most of the measured intervals fell within 1 ms ± 25 μs, 7 out of the 1,000 measured intervals fell out of this region. The interval was 380 μs at minimum and 1,605 μs at maximum. These outliers were always "one-shot," and did not affect their successive packets. For example, the minimum interval 380 μs occurred immediately after the maximum interval 1,605 μs as shown in Fig. 7.

4.3. Target Tracking Example

As a simple application example, target tracking by two high-speed vision nodes were implemented. Figure 8 shows the experimental setup. Because photo sensitivity of the implemented vision chip was not high enough, a pen light was used as a target, and moved in front of the two vision nodes.

In this example, data fusion by computing the 3-D position is demonstrated. From one of the vision nodes, the target position in its image coordinates was communicated to the other vision node every one millisecond. The receiving vision node combined its own 2-D target position with it, computed the 3-D position of the target assuming the baseline stereo, and sent it to the Linux host computer every one millisecond. Figure 9 shows the computed 3-D trajectory of the target. Because this is just for a demonstration of the capability of visual feature communication at 1,000 fps with data fusion, the 3-D position accuracy and the end-to-end system delay are not evaluated.

5. CONCLUSION

Design and implementation of a networked high-speed vision system have been presented. The system employs a multi-level real-time control structure by which appropriate real-time processing granularity for the corresponding system levels is offered. At the level of visual feature communication, we employed the standard IP network using off-the-shelf Ethernet switches. Although this communication network does not guarantee hard real time operations, the experimental results show that the sufficiently low latency is available for 1,000-fps visual feature communication for over 99% of the delivered packets.

The results presented in this paper were obtained under the circumstances where no interfering traffic exists. Evaluation of the effect of other traffic will be done in future work, where Ethernet switches with QoS mechanism [5] should be introduced. Future work will also include designing more sophisticated software framework and developing actual application systems.

6. REFERENCES

[1] R. T. Collins, Lipton A. J., T. Kanade, H. Fujiyoshi, D. Duggins, Y. Tsin, D. Tolliver, N. Enomoto, O. Hasegawa, P. Burt, and L. Wixson, "A system for

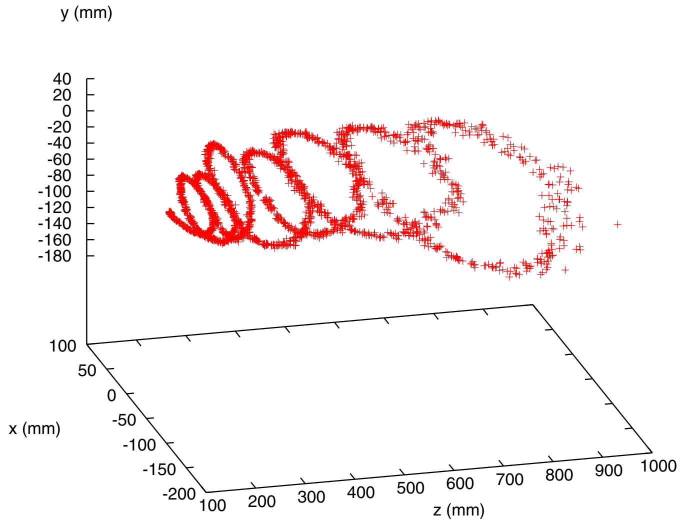

Fig. 9. Estimated 3-D trajectory of the target.

video surveillance and monitoring," Technical Report CMU-RI-TR-00-12, Carnegie Mellon University, 2000.

[2] Takashi Matsuyama and Norimichi Ukita, "Real-time multitarget tracking by a cooperative distributed vision system," *Proceedings of the IEEE*, vol. 90, no. 7, pp. 1136–1150, 2002.

[3] Takashi Komuro, Shingo Kagami, and Masatoshi Ishikawa, "A dynamically reconfigurable SIMD processor for a vision chip," *IEEE Journal of Solid-state Circuits*, vol. 39, no. 1, pp. 265–268, 2004.

[4] Shingo Kagami, Takashi Komuro, and Masatoshi Ishikawa, "A high-speed vision system with in-pixel programmable ADCs and PEs for real-time visual sensing," in *8th IEEE International Workshop on Advanced Motion Control*, 2004, pp. 439–443.

[5] "IEEE standard for local and metropolitan area networks, Media Access Control (MAC) Bridges," IEEE 802.1D, June 2004.

[6] H. Schulzrinne, S. Casner, R. Frederick, and V. Jacobson, "RTP: A Transport Protocol for Real-Time Applications," RFC 3550 (Standard), July 2003.

[7] "PROFINET," http://www.profibus.com/pn/.

[8] "EtherCAT technology group," http://www.ethercat.org/.

[9] "Motion capture systems from Vicon," http://www.vicon.com/.

[10] Takashi Komuro, Shingo Kagami, Masatoshi Ishikawa, and Yoshio Katayama, "Development of a bit-level compiler for massively parallel vision chips," in *7th IEEE International Workshop on Computer Architecture for Machine Perception*, 2005, pp. 204–209.