Paper:

A Pixel-Parallel Algorithm for Detecting and Tracking Fast-Moving Modulated Light Signals

Shingo Kagami, Masatsugu Shinmeimae, Takashi Komuro, Yoshihiro Watanabe, and Masatoshi Ishikawa

Graduate School of Information Science and Technology, University of Tokyo
7-3-1 Hongo, Bunkyo-ku, Tokyo 113-8656, Japan
E-mail: swk@ic.is.tohoku.ac.jp
[Received December 16, 2004; accepted February 19, 2005]

We developed a pixel-parallel image processing algorithm for detecting and tracking temporally modulated light signals for use as optical ID tags. The proposed algorithm requires no wavelength filters and obtains position information on signals at a high frame rate while decoding transmitted data. We implemented the proposed algorithm in a digital vision chip, a pixel-parallel processor embedded in a CMOS image sensor, and proved it to be operationally feasible.

Keywords: ID tag, optical tag, target tracking, high speed vision, vision chip

1. Introduction

Technologies using highly developed information technologies for supporting life and work in the real world are becoming increasingly important. For these technologies to work properly, information systems must recognize the state of the real world accurately and timely. ID tag technologies that apply identification (ID) information to real-world objects and recognize this information are thus attracting attention leading, for example, to the development and standardization of radio-frequency ID (RFID) tags being actively studied by many institutes and companies.

Flexible, natural interaction between information systems and the real world requires accurate position and motion information on objects in the real world. State-of-the-art RFID is not yet mature enough to achieve the required accuracy, however.

Vision effectively obtains position and motion information. In particular, high-speed vision systems that achieve, for example, over 1,000fps track objects moving quickly and irregularly [1]. Determining the correspondence between “what is being watched” and “what it actually is” remains difficult, however, since image-based matching and similar techniques are not always reliable.

We studied temporally modulated light signals as ID information sources recognized using an image sensor and directly associated with visual information, that is, in our case, position and motion in the real world. Examples of light signals discussed here include LED light sources and laser pointer spots temporally modulated to carry information, sometimes called optical tags or optical beacons based on the application [2–6]. These have attracted attention due to their robustness against variations in appearance in contrast to, for example, two-dimensional barcodes.

We propose an image processing algorithm to detect and track fast-moving light signals, and to get position information on them in the image space. We targeted a kilohertz-order frame rate in capturing scenes while locating light signal positions. Huge amount of computation is required to immediately detect and decode light signals that appear anywhere in an image. Our algorithm involves pixel-parallel processing suitable for VLSI implementation, and high-speed processing is achieved using the data parallelism of image processing. We demonstrate and evaluate the proposed algorithm through implementation in a digital vision chip we developed [7], a CMOS image sensor that integrates a digital processing element in each pixel.

Position detection of temporally modulated light signals is more robust and reliable than image-based tracking in passive visual processing. When robust location information is achieved at a high frame rate of 1,000fps and combined with high-speed visual scene information, it synergistically opens up new applications not achieved with existing ID-tag technologies or high-speed vision technologies. For instance, we plan to work on wearable LED tags implemented in wrist watches and rings that assist in gesture recognition, intuitive user interfaces in mobile devices such as PDAs and cellphones based on position and motion, and calibration of mobile cameras using light signals as 3D position markers.

2. Related Work

Much work has been done on modulated light signal detection using image sensors. Limiting our discussion to systems that do not assume synchronization between light sources and sensors raises several prior art applications

[2–6] that are not, however, necessarily applicable to the goal of this paper.

Moore et al. [2] introduced a physical icon, Phicon, that transmits information carried by a blinking pattern of infrared LED light, and proposed a man-machine interface in which a CCD camera reads transmitted information. Information transmitted by Phicon is packetized based on an HDLC-based protocol. Aoki [3] proposed Balloon Tag, which consists of five LEDs arranged geometrically and transmits information carried by their blinking patterns, read by a CCD camera. Since four out of the five LEDs act as clock signals, it has the advantage of not needing to know the transmission rate. Both Phicon and Balloon Tag use conventional CCD cameras, which do not follow quick motion.

Recent advances in CMOS image sensor technology have provided image sensors with higher frame rates, and high-speed sensor applications have emerged [4–6].

ID Cam, proposed by Matsushita et al. [4], based on a 12kHz CMOS image sensor, executes pixel-parallel 4kHz light signal decoding. The sensor integrates an analog circuitry in each pixel that computes interframe difference in incident light, with modulated light signals detected based on this interframe difference, eliminating the need for wavelength filters and capturing both scenes and light signals with a single imager. Transmitted information is packetized into a 22-bit packet with an 8-bit Manchester-encoded payload, and the sensor outputs IDs and their positions at the video frame rate. Since 22-bit time at 4kHz is 5.5ms, a pixel receives the packet if the light signal stays at the pixel for 5.5ms, enabling the system to follow medium-speed motion without dedicated interpixel tracking.

Maximum motion speed depends on the packet length. When a 128-bit payload, which is the size of an IPv6 address, is transmitted, for example, the light signal must stay at a pixel for 65ms, which is too long for the sensor to accept fast motion. Although this can be relaxed if the sensor frame rate is increased, the image readout time from the sensor is limited and sensor SNR deteriorates due to shortened exposure. We believe interpixel tracking is required to solve this problem fundamentally.

Ito et al. [5] and Kato et al. [6] developed light signal detecting systems using partial readout by CMOS image sensors that, instead of reading out the whole image, read only blocks around positions where light signals should lie, markedly reducing image data to be read out from the sensor and achieving high frame rates.

The infrared ID sensor system developed by Ito et al. [5] captures a scene in a frame time of 100ms and recognizes positions of infrared-emitting objects. After image processing such as excluding large area components to avoid misdetection of room lights, it captures images at a frame rate of 400Hz and receives IDs. This means it does not locate IDs at a high frame rate continuously. The interactive optical communication system by Kato et al. [6] reads out signal positions all the time while signals are detected, and thus continuous high frame rate locating of IDs is achieved. A new light signal that appears suddenly cannot be detected immediately, however, because it requires time to scan the entire image area.

Interpixel tracking of light signals in the above two proposals are done by finding the nearest bright points in successive frames. This strategy is, in principle, equivalent to the systems with conventional CCD cameras [2, 3], and regarded as an accelerated version using partial readout. Light signals must have sufficiently high contrast against backgrounds. For realistic implementation, they require infrared-pass wavelength filtering using infrared light signals, which makes some applications that use visible light signals difficult. Since they eliminate background information, additional cameras are required for some applications such as augmented or mixed reality.

In short, combining detection and tracking of modulated light signals at a high frame rate is possible [3, 6] provided that light signals have contrast high enough to be distinguished from the background (or wavelength filters suppress backgrounds). This problem is addressed, for example, by treating obvious obstacles heuristically [5] or by calculating interframe differences in brightness [4]. Both, however, acquire position information on light signals only at the rate of packet arrival, typically around the video frame rate, which is not high enough for our goals.

Our strategy is as follows. First, signal is sensed based on interframe differences as done by ID Cam [4] to receive background scene information and light signals with a single imager. We then construct an algorithm for interpixel tracking of light signals that computes the signal positions at the vision frame rate, which is on the order of kHz.

3. Algorithm

This section discusses an algorithm for detecting and tracking temporally modulated light signals. We refer to both direct light sources, such as modulated LEDs, and indirect light sources, such as modulated laser pointer spots, by the term “temporally modulated light signals” or “light signals.” When we are interested in positions of the light signals in 3D spaces or image spaces, we may call them “light signal sources.” For a direct light source such as an LED, this means the position of the source itself, whereas for an indirect light source such as a laser pointer, it means the position of the reflection point.

When a payload word is transmitted as light signals, just as ID Cam [4] does, a packet is constructed by putting a header and a footer at the beginning and the end of the Manchester-encoded payload, and is transmitted by expressing bits 1 and 0 as light on and off.

The signal receiver captures light with an image sensor, and the image is processed to detect and decode signals. We assume an abstract pixel-parallel image processing architecture consisting of 2D-mesh-connected processing elements, each of which – at least virtually – corresponds to a pixel, controlled in a single instruction stream, multiple data stream (SIMD) manner. In the description of the algorithm below, the term pixel may be used interchangeably to refer to this processing element. The processing element:

1. Executes an arbitrary logical or arithmetic operation for certain data at each pixel.
2. Transmits certain data to a neighboring pixel from each pixel.
3. Generate a binary image where pixels within an arbitrary rectangle (including ones consisting of only a pixel) hold 1 and the other pixels hold 0.
4. Computes the 0th order moment or 1st order moment of a binary image.
5. Reads out certain data at an arbitrary pixel.

The first three operations yield 2D pattern data, and results are stored in the 2D array. Operations 4 and 5 yield scalar data. Once output, scalar results are handled, for example, by an external microprocessor.

For simplicity, we first describe the algorithm assuming that the light signal source does not move. We then add tracking for a moving source and expand it for cases with multiple light signals that sometimes collide.

3.1. Receiving Algorithm

The image sensor captures images at a frame rate of, for example, 1,000fps. We do not assume wavelength filtering and captured scenes including backgrounds are fully available for image processing. Interframe differences in pixel values are computed at every frame for all the pixels, and light signals are detected based on their positive and negative temporal edges. Since all pixels are aware of this interframe difference, light signals are immediately detected and separated from backgrounds without wavelength filtering [4].

When multiple light signals exist in the field of view, the start times of the packets may differ. To cope with this in SIMD parallel processing, we prepare a state machine within each pixel. A pixel receives a packet independently based on the state of the pixel that may be different from other pixels. The state transition diagram is shown in **Fig.1**. A 4-bit header '1001' and a 2-bit footer '00' are assumed in this figure. For an n-bit payload, the number of the states including the initial state is $4n+7$. We need $\lceil \log_2(4n+7) \rceil$ bits of per-pixel memory to maintain the state. Other bits are required for miscellaneous management.

Starting from the initial state, the state machine goes through the header and the payload to the footer while the blinking sequence of the pixel is consistent as a packet. As soon as the blinking sequence is not, the state machine is reset to the initial state.

Each pixel holds a flag bit as a miscellaneous management bit that shows whether the pixel has detected a valid packet. Using this flag bit, we get a binary image in which pixels with light signals are 1, or otherwise 0, at any point in time. We call this binary image a valid flag image.

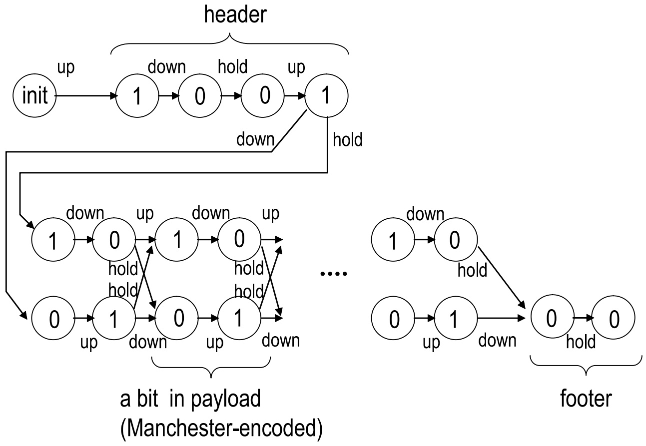

Fig. 1. Transition diagram of the state machine in the pixel. Inputs up, down, and hold stand for cases when incident light gets brighter, gets darker, and remains unchanged. Arcs going back to the initial state are omitted.

When only one light signal exists in the field of view, the light signal appears as one connected component in the image. The 2D position of the light signal is thus obtained by calculating the centroid of the valid flag image. When multiple light signals exist, connected components are labeled and their centroids are sequentially obtained using our multiple target tracking algorithm [8]. Because centroids are obtained at each frame in real time no matter how long the packet is, light signals are located with high temporal resolution.

We have two options in retaining decoded payloads – to store them within the per-pixel memory or to store them outside the processing element array. Storing them in memory is easy to program because we need only store decoded bits with the forward transition of the state machine; this also has the advantage of having processing time independent of the number of light signals in the field of view. These advantages come at the cost of required per-pixel memory to store entire payloads. Since per-pixel memory is usually a finite resource in pixel-parallel architectures, this may cause difficulty in implementation.

Storing them outside the processing element array is more complicated. In each frame, we calculate centroids of connected components in the valid flag image to get payload bits at the frame by reading pixels at coordinates of obtained centroids. These payloads are accumulated outside the array, markedly reducing required per-pixel memory at the cost of processing time that increases in proportion to the number of light signals. Here we focus on storing outside.

3.2. Interpixel Tracking

When a light signal moves quickly, interpixel motion of the signal may interrupt the packet, requiring tracking.

For our goals, we extended a pixel-parallel target tracking algorithm called self windowing [9], suitable for high-speed vision. Self windowing tracks a region in two steps

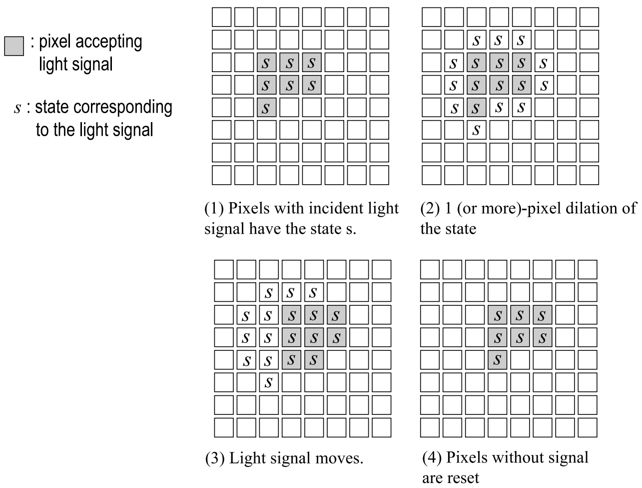

Fig. 2. Interpixel tracking.

at frame: morphological dilation of the target region of the previous frame, and pixel-wise ANDing of the dilated previous target and the current frame image. It takes advantage of the fact that images in two successive frames differ little at a high frame rate.

Our extension to the algorithm handles the "state machine image" instead of a binary image, as shown in **Fig.2**. A pixel not in the initial state copies its state to neighboring pixels – an extension of morphological dilation to state image processing. If the image of a light signal moves to neighboring pixels, state machines at new pixels continue while they are within the radius of copying. If a pixel receiving a copy of the state from its neighbor does not actually have the light signal on it, the state machine at the pixel is immediately reset, just as with pixel-wise ANDing in self windowing. Repeating this state copy and the state machine transition for each frame enables simultaneous tracking and decoding of light signals.

Depending on the relation between the motion of light signal sources and the frame rate of the receiver, 1-pixel dilation may not be sufficient. Since the state copying procedure may be repeated just as morphological dilation may, faster motion is followed by increasing the number of repeats. The greater the number of repeats, the more likely multiple signals collide with each other. The number of repeats should thus be selected appropriately.

3.3. Handling Multiple Light Signals

As stated, multiple light signals in the field of view are detected based on the multiple target tracking algorithm [8], summarized as follows:

The receiver lists light signals being tracked, including decoded payloads and their current centroid positions. For each centroid listed, a connected component including the centroid position is extracted from the valid flag image at the current frame using area filling as follows: generating a binary image in which only the specified centroid pixel is 1, dilation of the pixel-wise ANDing of this centroid point image and the current valid flag image, and repeating this pixel-wise ANDing and this dilation until the size of the resulting area is unchanged. The centroid of this extracted area is calculated and the payload at the centroid pixel is read out and accumulated. When the area filling results in nothing (an area with the size of 0), the light signal is removed from the tracked signal list.

Signals detected since the previous frame or earlier are thus extracted. Next, new signals that appear in the current frame must be extracted. We can generate a binary image in which all areas being tracked have pixel values of 0 and the other pixels have 1 because all tracked areas were extracted as described in the previous paragraph. Applying pixel-wise ANDing to this image and the current valid flag image yields a binary image that includes only new light signal areas. Connected component labeling [8] is executed for this image because the image may have multiple new areas. By repeatedly pixel-wise ANDing an image of interest and an image with an arbitrary rectangle area of 1s, generated with the assumed architecture, we find a pixel included in a new area in a 2D binary search fashion. Area filling with this pixel as a seed extracts a connected component including the pixel. By repeating this, all light signals in the field of view are extracted.

When multiple light signals are tracked, processing time is proportional to the number of light signals. Although the number of signals read out within the frame time is thus limited, light signals in the tracked signal list are preferentially tracked with the algorithm above, so a packet once tracked is not lost by the emergence of too many new signals. Instead, excessive new signals are deleted.

3.4. Collisions

What happens when different light signals collide in the image space. We consider two cases: one in which multiple light signals fall onto a pixel and one in which multiple light signals fall onto adjacent pixels.

When multiple light signals fall onto a pixel, if one of the signals is sufficiently stronger than the others, this is received. If not, no signals are received normally. Given that payloads are Manchester-encoded and disturbance by multiple signals only adds incident light, i.e., changing bit 0 to 1, disturbed signals are discarded in most cases without generating a garbled signal. This is not always guaranteed, however, and redundant encoding is needed for higher reliability.

When multiple light signals fall onto adjacent pixels, receiving within a pixel is normally executed. A problem occurs in signal area extraction, however. A connected component in the valid flag image is treated as a signal area. Light signals adjacent in the image space are thus recognized as a single signal.

To avoid this, a modified valid flag image is used for the area extraction procedure instead of the valid flag image itself. If the state of a pixel differs from neighbor pixel states, the pixel is set to 0 in this modified valid flag image – a type of morphological erosion with dependence on the state machine, separating clinging light signals.

This separation is not perfect because it is based on the comparison of states. When the clinging light signals are coincidently synchronized, i.e., packet start times are the same, they cannot be separated. If per-pixel memory is available, storing the payload or part of it within the pixel and comparing them improves separability.

4. Implementation

We implemented and evaluated the proposed algorithm. As a pixel-parallel processing system, we used VCS-IV, a visual processing system based on a digital vision chip [7, 10].

The digital vision chip [7] is a CMOS image sensor in which each pixel integrates a programmable digital processing element. It is a sensor and a pixel-parallel image processing system at the same time. The system we used for implementation consists of 64×64 pixels (pixel pitch: 64μm), and each pixel has a 24-bit local memory and a bit-serial ALU. All pixels are controlled in SIMD via a dedicated microcontroller that conducts 32-bit scalar processing in addition to the SIMD instruction delivery and program control.

Items in Section 3 stated as required for the abstract pixel-parallel architecture are met by this system. Items 1 and 2, which are purely pixel-parallel processing, are executed ideally for the processing speed. The system also has common data buses that provide per-column or per-row common data, a unit for image-wide summation of all pixel values, and a unit for image-wide ORing of all pixel values to generate rectangles (item 3) and calculate moments (item 4).

Dedicated functions to access the value of a pixel, corresponding to item 5, is not implemented in the system, but since generation of a point image, in which only one pixel of 1 exists, is possible as a special case for rectangle generation, calculating image-wide ORing of pixel-wise ANDing of the point image and image of interest gives a result equivalent to single pixel readout.

VCS-IV [10] is shown in **Fig.3**. A CS mount lens with a focal length of 6 mm was used in a setup we call a "camera." We used red LEDs as light signal sources. Modulation frequencies and transmitted payloads were controlled by a Tektronix data generator DG2020, which directly drove LEDs.

4.1. Communication Between Transmitter and Receiver

We put only one LED in the camera field of view, and confirmed that the transmitted light signal was received normally. The cycle of light blinking (shortest time between light on and off) was set to 1ms, and a packet with a payload of 8-bit data – i.e., 16-bit time due to Manchester encoding – was transmitted repeatedly. The whole packet, including the header and footer, was 22-bit time, that is, 22ms.

All received packets were correctly read out and no garbled data were generated, but periods occurred when no packets were received every several tens of thousands of frames (several tens of seconds). This was caused by accumulated small disagreement between the transmitter's blinking cycle and the receiver's frame time. This problem is solved by making the receiver's frame rate higher than the transmitter's blinking rate and adjusting the phase of a bit time at the start time of each packet as done by ID Cam [4].

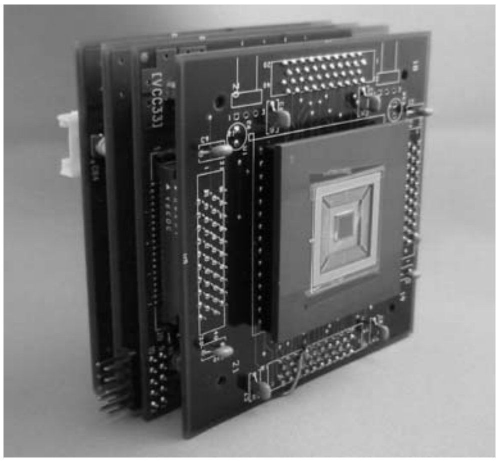

Fig. 3. VCS-IV, a pixel-parallel visual processing system in which the proposed algorithm is implemented [10].

None of the experiments in the rest of this paper required continuous observation for a long time, so we left this problem unsolved and used only periods when packets are received correctly.

4.2. Interpixel Tracking

To evaluate the effectiveness of the proposed tracking algorithm, we compared results obtained with tracking enabled and disabled. A LED transmitter was moved in a plane nearly parallel to the image plane to draw the Japanese syllable "Ru" at about 20cm from the camera. The blinking cycle was set to 1ms, and the payload was 8-bit data. The time from the beginning of writing to the end was about 2.0s (2,000 frames).

Figure 4 plots the trajectory of the light signal in the image space. In sections where light signal sources moved relatively slowly, including start and end points, and corner sections, packets were received correctly with or without tracking enabled. In sections where sources moved fast, the receiver with tracking disabled hardly detected the light signal. The receiver with tracking enabled always detected the light signal correctly.

For the light signal to be detected completely when tracking is disabled, the whole packet from header to footer must stay within at least a pixel, which is likely to be met, for example, in sections such as corners or end points. In sections where the light signal source moves straight and fast, this condition is rarely met and tracking was effective. The position of the light signal was obtained at each image frame regardless of the packet length,

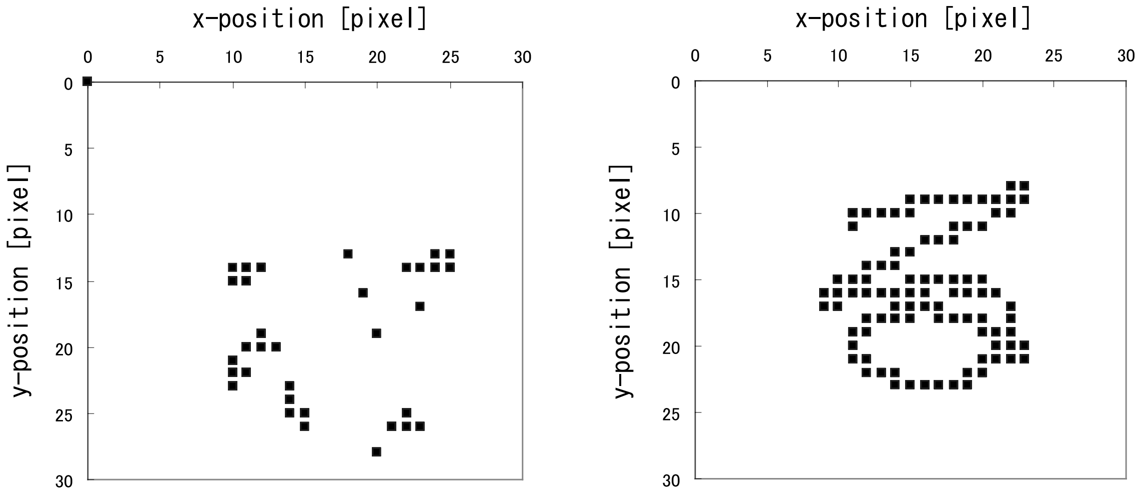

Fig. 4. Experimental results of moving light signal detection. Left: tracking disabled. Right: tracking enabled.

and our proposal is particularly effective when target light signals move quickly or when trajectories with high temporal density must be observed.

Note that coordinates of centroids were calculated in integer precision in experiments, so trajectories of light signals are shown in pixel precision. Trajectories are obtained in sub-pixel precision by computing the centroid, which is the 1st order moment divided by the 0th order moment, in floating-point precision.

4.3. Tracking Multiple Light Signals

We also implemented multiple target tracking. The maximum number of light signals tracked simultaneously was 4, and the blinking cycle was 2ms. Two LED transmitters with different payloads were put 10cm to 20cm from the camera. Starting times of the two light signals were not synchronized. Both light signals were detected and decoded correctly.

To evaluate separation in multiple light signal collision, we conducted an experiment with two LEDs overlapped in the image space. One LED transmitted a payload of "94" expressed in binary, and the other "183." The first LED remained stationary and the second was moved to approach the first. After images of both LEDs were half overlapped in the image space of the receiver, the second was moved away from the first. The area of a light signal in the image space was about 30 pixels.

Figure 5 shows the time evolution of *x* coordinates of the two LEDs. When one LED approached the other, the light signal transmitted by the stationary LED moved in the image space as if it was pushed by the moving light signal. The two light signals were detected and decoded separately without disappearing or generating garbled data.

Where the two light signals were overlapped, the one closer to the camera was received under this experimental condition, meaning the two different light signals were adjacent in the image space when they were half overlapped. Nevertheless, signals were successfully separated and our proposal turned out to be effective.

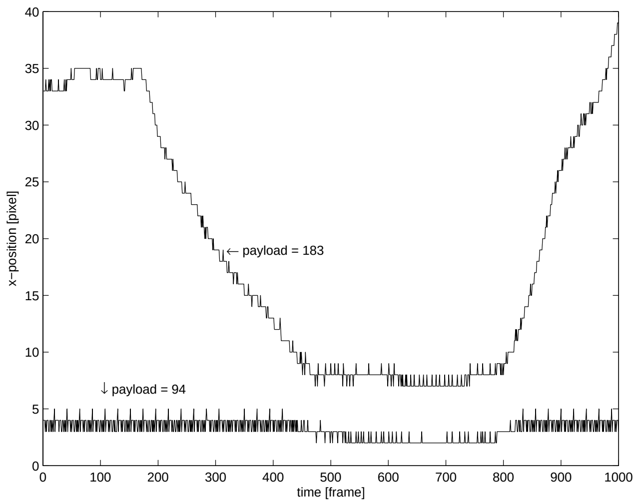

Fig. 5. Experimental results of light signal collision.

This separation, however, was not perfect. When the image of a light signal is too small, for example, tracking sometimes failed, probably due to elimination of an area by state-dependent erosion and by rounding of centroid coordinates in integer precision that led to failure to point inside the corresponding area in the next frame as an area-filling seed. This limitation will be relaxed with improvement of the spatial resolution of the image sensor. For basic solutions, we must improve the algorithm itself and introduce redundancy in encoding light signal packets.

5. Conclusions

We have developed a pixel-parallel image processing algorithm for detecting and tracking temporally modulated light signals. This algorithm enables both locating and decoding of signals at a high frame rate. While based on interframe difference in detecting signals for robustness against background scenes, issues of fast-moving signals and their collision are also addressed. The proposed algorithm was implemented in a digital vision chip and proved to be operationally feasible. Projected work includes improving the algorithm and encoding for further robustness and constructing actual applications.

References:

[1] M. Ishikawa, and T. Komuro, "Digital vision chips and high-speed vision systems," In Digest of Technical Papers of 2001 Symposium on VLSI Circuits, pp. 1-4, 2001.

[2] D. J. Moore, R. Want, B. L. Harrison, A. Gujar, and K. Fishkin, "Implementing phicons: Combining computer vision with infrared technology for interactive physical icons," In ACM Symposium on User Interface Software and Technology (UIST99), pp. 67-68, 1999.

[3] H. Aoki, "Infrared identification system with tags and video camera," Interactive Systems and Software, Vol.VIII, pp. 131-136, 2000 (in Japanese).

[4] N. Matsushita, D. Hihara, T. Ushiro, S. Yoshimura, J. Rekimoto, and Y. Yamamoto, "ID Cam: A smart camera for scene capturing and ID recognition," In The Second IEEE and ACM International Symposium on Mixed and Augmented Reality, p. 227, 2003.

[5] S. Ito, Y. Sumi, and K. Mase, "Interaction capturing device with an infrared ID sensor," In IPSJ SIG Technical Report, 2003-HI-104, pp. 13-18, 2003 (in Japanese).

[6] K. Kato, S. Tanaka, and I. Ishii, "Interactive optical communication based on high-speed vision," In Vision Engineering Workshop 2003 (ViEW2003), pp. 189-194, 2003 (in Japanese).

[7] T. Komuro, S. Kagami, and M. Ishikawa, "A dynamically reconfigurable SIMD processor for a vision chip," IEEE Journal of Solid-state Circuits, Vol.39, No.1, pp. 265-268, 2004.

[8] Y. Watanabe, T. Komuro, S. Kagami, and M. Ishikawa, "Real-time visual measurements using high-speed vision," In Optics East 2004, 2004.

[9] I. Ishii, Y. Nakabo, and M. Ishikawa, "Target tracking algorithm for 1ms visual feedback system using massively parallel processing," In Proc. 1996 IEEE Int. Conf. Robotics and Automation, pp. 2309-2314, 1996.

[10] S. Kagami, T. Komuro, and M. 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, pp. 439-443, 2004.

Name:
Shingo Kagami

Affiliation:
Research Associate, Department of Information Physics and Computing, Graduate School of Information Science and Technology, University of Tokyo

Address:
7-3-1 Hongo, Bunkyo-ku, Tokyo 113-8656, Japan

Brief Biographical History:
2003 Received Ph.D. degree in Mathematical Engineering and Information Physics from the University of Tokyo, Tokyo, Japan
2003 Research Fellow, Japan Science and Technology Corporation
2003- Research Associate, Department of Information Physics and Computing, University of Tokyo

Main Works:
- "A Sensor Selection Method Considering Communication Delays," Proc. IEEE Int. Conf. Robotics and Automation, pp. 206-211, Apr. 2004.
- "A Software-Controlled Pixel-Level A-D Conversion Method for Digital Vision Chips," Proc. IEEE Workshop on Charge-Coupled Devices and Advanced Image Sensors, May 2003.
- "A Real-Time Visual Processing System using a General-Purpose Vision Chip," Proc. IEEE Int. Conf. Robotics and Automation, pp. 1229-1234, May 2002.

Membership in Learned Societies:
- The Robotics Society of Japan (RSJ)
- The Institute of Image Information and Television Engineers (ITE)

Name:
Masatsugu Shinmeimae

Affiliation:
Master Course Student, Department of Information Physics and Computing, Graduate School of Information Science and Technology, University of Tokyo

Address:
7-3-1 Hongo, Bunkyo-ku, Tokyo 113-8656, Japan

Brief Biographical History:
2005 Received B.E. degree in Mathematical Engineering and Information Physics from the University of Tokyo, Tokyo, Japan
2005- Master Course Student, Department of Information Physics and Computing, University of Tokyo

Name:
Takashi Komuro

Affiliation:
Research Associate, Department of Information Physics and Computing, Graduate School of Information Science and Technology, University of Tokyo

Address:
7-3-1 Hongo, Bunkyo-ku, Tokyo 113-8656, Japan

Brief Biographical History:
2001 Received Ph.D. degree in Mathematical Engineering and Information Physics from University of Tokyo, Tokyo, Japan
2001-2002 Research Fellow, Japan Science and Technology Corporation
2002- Research Associate, Department of Information Physics and Computing, University of Tokyo

Main Works:
• "Dynamically Reconfigurable SIMD Processor for a Vision Chip," IEEE Journal of Solid-State Circuits, Vol.39, No.1, pp. 265-268, 2004.
• "A Digital Vision Chip Specialized for High-speed Target Tracking," IEEE transaction on Electron Devices, Vol.50, No.1, pp. 191-199, 2003.
• "Device and System Development of General Purpose Digital Vision Chip," Journal of Robotics and Mechatronics, Vol.12, No.5, pp. 515-520, 2000.

Membership in Learned Societies:
• The Robotics Society of Japan (RSJ)
• The Society of Instrument and Control Engineers (SICE)
• The Institute of Image Information and Television Engineers (ITE)

Name:
Masatotshi Ishikawa

Affiliation:
Professor, Department of Information Physics and Computing, Graduate School of Information Science and Technology, University of Tokyo

Address:
7-3-1 Hongo, Bunkyo-ku, Tokyo 113-8656, Japan

Brief Biographical History:
1979- Senior Researcher, Industrial Products Institute
1989- Associate Professor, University of Tokyo
1999- Professor, University of Tokyo

Main Works:
• "A CMOS Vision Chip with SIMD Processing Element Array for 1ms Image Processing," Proc. IEEE Int. Solid-State Circuits Conference (ISSCC'99), pp. 206-207, Feb. 1999.

Membership in Learned Societies:
• The Robotics Society of Japan (RSJ)
• The Society of Instrument and Control Engineers (SICE)
• The Institute of Electronics, Information and Communication Engineers (IEICE)

Name:
Yoshihiro Watanabe

Affiliation:
Doctor Course Student, Department of Information Physics and Computing, Graduate School of Information Science and Technology, University of Tokyo

Address:
7-3-1 Hongo, Bunkyo-ku, Tokyo 113-8656, Japan

Brief Biographical History:
2004 Received M.E. degree in Department of Information Physics and Computing from the University of Tokyo, Tokyo, Japan
2004- Doctor Course Student, Department of Information Physics and Computing, University of Tokyo
2004- Research Fellow, Japan Society for the Promotion of Science

Main Works:
• "Real-time Visual Measurements using High-speed Vision," Optics East 2004 / Machine Vision and its Optomechatronic Applications, Proceedings of SPIE, Vol.5603, pp. 234-242, 2004.

Membership in Learned Societies:
• The Robotics Society of Japan (RSJ)