Proc. of 2006 International Workshop on Smart Home (IWSH'06), pp. 308--317 (Nov. 2006).

A Position Detection Mechanism for Location-aware Pin&Play

Yasue Kishino, Tsutomu Terada, and Shojiro Nishio
Graduate School of Information Science and Technology,
Osaka University
1-5 Yamadaoka, Suita, Osaka, Japan
{yasue, tsutomu, nishio}@ist.osaka-u.ac.jp

Nicolas Villar and Hans Gellersen
Computing Department,
Lancaster University
Lancaster LA1 4WA UK
{villar, hwg}@comp.lancs.ac.uk

Abstract

We propose a position detection mechanism for location-aware Pin&Play. Though the positions of inserted pins are important for most applications, they could not be detected in previous systems. The proposed method detects the positions of pins by processing images taken with a camera placed in front of the board. The proposed method has a dynamic adjustment mechanism for image processing parameters to achieve higher accuracy even when the surrounding lighting condition changes. Moreover, it also has an error estimation mechanism to avoid the accumulation of position error. We also clarify the effectiveness of our proposed method through performance evaluations.

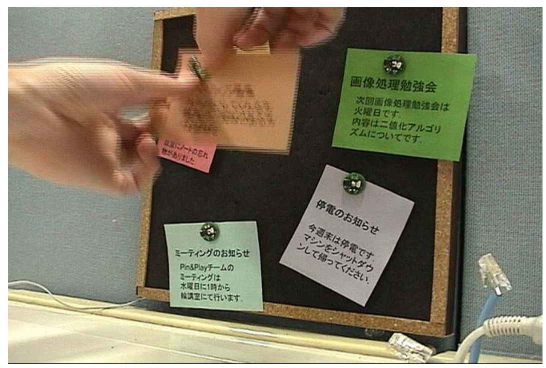


Figure 1. Pin&Play

1 Introduction

In ubiquitous computing environments, various ubiquitous devices collaborate and provide services to support daily life. To realize such environments, secure and robust networks with an efficient power supply are necessary. Networks are categorized into two types: wireless networks and wired networks. In wireless networks, it is necessary to consider security and battery resources for devices, while in wired networks, wiring can cause problems.

We have proposed the Pin&Play system, which uses the surface of a wall as a network infrastructure and controls inserted pin-shaped devices [6]. Since we can find walls almost anywhere, network and power supply through walls solve the problems faced with wired/wireless networks.

In the Pin&Play system, we can easily set ubiquitous devices by just attaching them to a wall to realize ubiquitous computing. We have previously implemented a prototype using a board and pushpins (Fig. 1). However, that system was not able to detect the position of the pushpins; it could only detect whether each pin was inserted since a board is just one large contact in a circuit. The positions of the pins are important in actual use of Pin&Play. For example, we often place notices onto a board according to their level of importance or to a specific timeline. Moreover, in the case where we attach devices on a wall, the behaviors of the devices should depend on their location. Such information on pin positions expands the possible range of applications using Pin&Play.

In this research, we propose a position detection mechanism for Pin&Play that employs image processing. The proposed mechanism has a dynamic adjustment mechanism for image processing parameters and an error estimation mechanism to reduce the degree of position error.

2 Pin & Play

Figure 2 shows the structure of Pin&Play system. It consists of a board that handles communication and the power supply, and pins (Fig. 3) that are inserted into the board. The board consists of five layers. The second and fourth layers are made of conductive fabric while the others are made of insulator. The second layer is connected to the communication and power supply ports of the adapter and the fourth layer is connected to the ground port. Each pin has two ports as shown in Fig. 3(b), and each port is con-

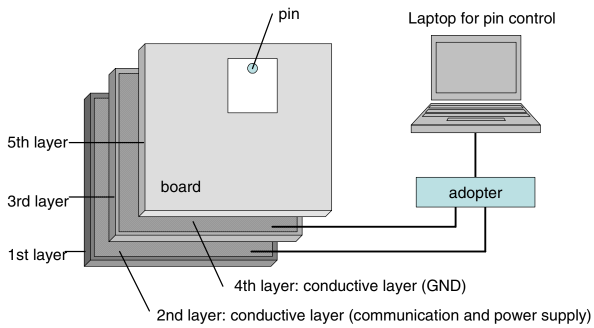


Figure 2. System structure of Pin&Play

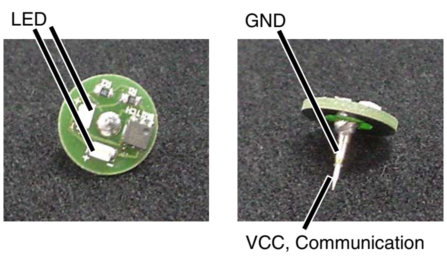


Figure 3. Pins for Pin&Play

nected to each conductive layer when the pin is inserted into the board. Each pin has its own ID and two LEDs.

We have proposed several systems for the Pin&Play project [12].

The Reminding Notice Board
This is an augmented notice board. When a new notice is posted on the board, the system indicates the posted time and expiration date. The system also blinks the LEDs of pins that are posting expired notices.

Pin&Play Wallpaper and Light Switch
The position of a switch can be changed by moving it to a more convenient place on the augmented wall while retaining its associations.

A Reconfigurable Musical Interface
In this system we use the input devices shown in Fig. 4. The user places these input devices on a tablet and controls the application. Figure 5 depicts an example of this interface in use. Users can add/remove devices according to their preferences.

The board checks the IDs of inserted pins and detects new inserted pins and removed ones. The system also records when a pin is inserted and removed and it can produce a list of inserted pins. However, it cannot determine pins' inserted positions since whole area of the board has just a single contact with the adapter.

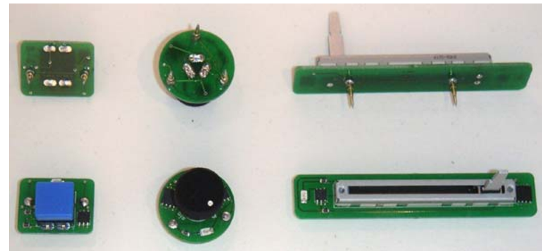


Figure 4. Pin&Play devices (button, dial, and slider)

Figure 5. An example use of Reconfigurable Musical Interface

The positions of devices are important. In the reminding notice board application, positions of posted documents may reflect a level of importance. Moreover, the wallpaper and light switch applications become more convenient though the use of position information. For example, devices in high positions might represent adults and those in low positions might represent children. Furthermore, devices close to a door might control the lights in the room and devices close to a window might control the shutters. In the reconfigurable musical interface, the system can exploit the positions, the directions of devices, and the physical relationships among devices to construct more intuitive interfaces.

3 Position detection mechanism

We propose a position detection mechanism using image processing. In a related work [3], there is a mechanism that detects the positions of pins. Although this research achieves position detection by dividing the board into several parts, it can only capture rough positions, and more complicated hardwirings and management are needed when the number of divisions increases. As another approach, a method to detect the position of an object touched on a computer display has already been realized by using infrared sensors. Though this approach can attach such a function to established public displays, it is limited in the number of detectable objects.

Our position detection mechanism is designed according to the following policies.

Image-based processing
The proposed mechanism is realized by capturing camera images of the board. When a new pin is inserted, it blinks so that its position can be detected by image processing.

No extension to pin and board
Since we use a camera for position detection, we do not need extensions to pins or the board.

Automatic adjustment of image processing parameters
Since this system is assumed to be placed in various lighting conditions, it has a mechanism that adjusts the image processing parameters dynamically.

Error Estimation
Repetition of posting and unsticking of pins increases the amount of position error. By estimating the degree of error, this system avoids the accumulation of it.

We describe the proposed method in detail below.

Figure 6 shows the structure of our system. In our proposed method, the PC makes LEDs on pins blink one by one at a certain interval, extracts the difference images of LEDs captured by a camera, and uses image processing to calculate the position of these LEDs.

3.1 Image processing to detect pins

Our system extracts a different image between the current image and the image of the previous frame, converts it to a gray-scale image, applies *blur* and *maximum filters*, and binarizes it. Next, the program extracts white connected areas, which are pin candidates. When the number of white connected areas and the number of blinking LED(s) are the same, it records their coordinates. If the number of these areas is fewer than that of the blinking LED(s) or these areas are too large not to be assumed as pins, they are ignored. When a pin is found at the same position four times, the system regards the position as one of a blinking pin.

3.2 Conversion of coordination

The system calculates the actual coordinates (X, Y) on the board from the pin position (x, y) on the image by projection conversion. We use the following equations:

$$X = \frac{a_1 x + b_1 x + c_1}{a_0 x + b_0 y + 1}\ ,\ and\ \ Y = \frac{a_2 x + b_2 x + c_2}{a_0 x + b_0 y + 1}.$$

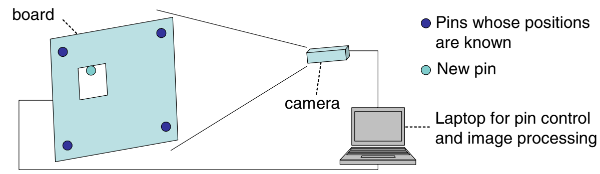


Figure 6. System structure

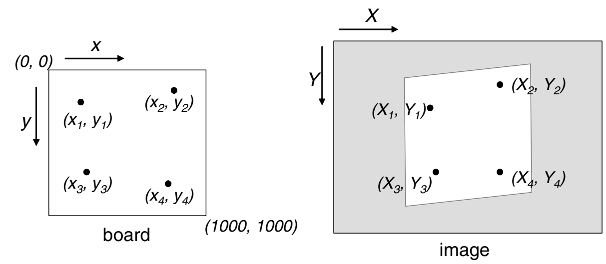


Figure 7. Coordinations on the board and on the camera image

To decide the coordination, the system needs to know the parameters (a_0, a_1, a_2, b_0, b_1, b_2, c_1, c_2). They are calculated using four reference pins whose positions are known ($P_1 : (x_1, y_1), (X_1, Y_1)$, $P_2 : (x_2, y_2), (X_2, Y_2)$, $P_3 : (x_3, y_3), (X_3, Y_3)$, $P_4 : (x_4, y_4), (X_4, Y_4)$) (Fig. 7) by using the following determinant:

$$\boldsymbol{P} \begin{bmatrix} a_0 \\ b_0 \\ a_1 \\ b_1 \\ c_1 \\ a_2 \\ b_2 \\ c_2 \end{bmatrix} = \begin{bmatrix} X_1 \\ Y_1 \\ X_2 \\ Y_2 \\ X_3 \\ Y_3 \\ X_4 \\ Y_4 \end{bmatrix};$$

$\boldsymbol{P}$ is given by the following equation.

$$\boldsymbol{P} = \begin{bmatrix} -x_1X_1 & -y_1X_1 & x_1 & y_1 & 1 & 0 & 0 & 0 \\ -x_1Y_1 & -y_1Y_1 & 0 & 0 & 0 & x_1 & y_1 & 1 \\ -x_2X_2 & -y_2X_2 & x_2 & y_2 & 1 & 0 & 0 & 0 \\ -x_2Y_2 & -y_2Y_2 & 0 & 0 & 0 & x_2 & y_2 & 1 \\ -x_3X_3 & -y_3X_3 & x_3 & y_3 & 1 & 0 & 0 & 0 \\ -x_3Y_3 & -y_3Y_3 & 0 & 0 & 0 & x_3 & y_3 & 1 \\ -x_4X_4 & -y_4X_4 & x_4 & y_4 & 1 & 0 & 0 & 0 \\ -x_4Y_4 & -y_4Y_4 & 0 & 0 & 0 & x_4 & y_4 & 1 \end{bmatrix}$$

Using these equations, the system obtains the positions of pins on the board.

3.3 Adjustment of image processing parameters

Optimum image processing parameters depend on such surrounding situations as room brightness and LED strength. The brightness differs, of course, between a light room and a dark room, and between day and night.

When a new pin is inserted, the system determines the optimum image processing parameters. The system automatically adjusts the following parameters:

- Threshold for binarization

  This depends on the surrounding brightness. A board is divided into several areas, each of which has its own threshold.

- Maximum size of white connected area

  A white connected area means a pin candidate and its size change according to the brightness of the LEDs on pins. If the size is larger than the maximum threshold, it is assumed not to be a pin.

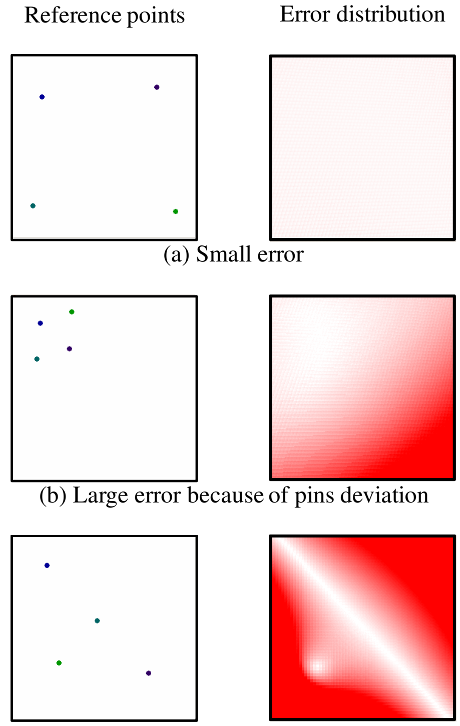
The threshold is automatically adjusted by repeated image processing. Concretely, the system attempts to detect a pin using several thresholds: 2, 4, 6, 8, ... up to 50. When the system succeeds in finding a pin three times in succession, it decides the second value of the three trials as the optimum threshold value. Since this procedure is performed when each pin is newly inserted, each pin has its optimum threshold.

The board is divided into several areas according to the threshold of the nearest pin. When the system binarizes the difference image, it uses thresholds of the area where each bit belongs. The maximum size of a white connected area detected as a pin is twice that of a white connected area when the pin is detected at the first instance. The system records the maximum size for each pin and uses the recorded value when the same pin is inserted again, since this difference depends on the brightness of the LEDs on each pin. If an unknown pin is inserted, a default value is used.

3.4 Error estimation

The system should select better reference pins to obtain more accurate positions. In selecting four reference points for projection conversion, our proposed method estimates the degree of error for each set of reference points and selects the most appropriate set.

The degree of position error depends on the positional relationship among reference pins as shown in Fig. 8. In Fig. 8(a), since the four pins are placed in each corner uniformly, the degree of error negligible. On the other hand,



(a) Small error

(b) Large error because of pins deviation

(c) Large error due to difficulty of projection conversion

Figure 8. Relationship between placement of reference points and error distribution (dark color: large error)

if the pins are concentrated in one corner, the degree of error becomes large in the opposite corner as shown in (b). In (c), since three pins are in a straight line, the calculation of projection conversion includes large errors. These mean irrelevant selections of reference pins may cause large errors. We call this type of error the *error from the positional relationship* (E_p).

Errors occur with not only the positional relationship among reference points but also image processing to detect LEDs and low camera resolution. For example, when the board is far from the camera and a 10-mm^2 square area of the board is captured on one pixel of the image, the result is estimated to include approximately 10-mm error even if image processing is perfect. We call this error *image processing error* (E_i).

Moreover, when reference points have existing error, this error is propagated to the new pin. We also consider this error as *error from the reference points* (E_r).

Thus, when a new pin is inserted, the estimated error is calculated as the sum of the three types of error for all possible combinations of points for which positions are known. The set of points with the smallest estimated error is selected as the reference points for the new inserted pin. In the following, we explain how to estimate position error in detail.

E_p: Error from the positional relationship

This error is estimated by calculating the degree of error supposing the camera is placed in an ideal position to detect pins correctly. Concretely, the coordinates of the new pin on the camera image (x, y) are transformed to location (X, Y) on the board. Then, (X, Y) is transformed to location (x', y') on a virtual camera image by affine conversion, which assumes the camera is placed in the ideal position. After that, the location (X', Y') on a virtual board is calculated by projection formation from (x', y'), and the tentative error (e) is calculated with the following equation.

$$e = \sqrt{(X - X')^2 + (Y - Y')^2}$$

Since position (X, Y) on the board may include error, the system also calculates the error at eight surrounding points, which are $\pm\frac{e}{2}$ distance from (X, Y). The average of these errors is set to E_p.

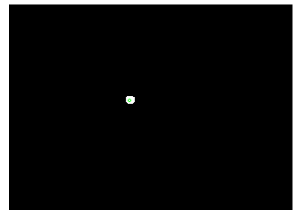
E_i: Image processing error

The distance between the upper-left and lower-right corners of the board is calculated by using inverse projection conversion, and the distance on the board in one pixel on the image is set to E_i.

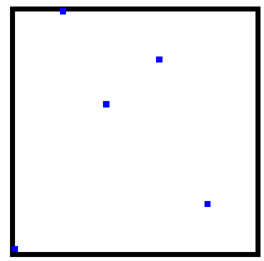
E_r: Error from the reference points

The average of estimated errors of reference points is set to E_r.

(a): Camera image



(b): Difference image



(c): Calculated position

Figure 9. Procedure of image processing

4 Implementation

4.1 Implementation of Pin&Play

The prototype system of Pin&Play was implemented using a 1-Wire network (Dallas Semiconductor/MAXIM Corporation). In this network we can control devices via two wires. We used a DS2406 which controls the ON/OFF status of output ports and obtains the ON/OFF status of input ports. We use blue and green LEDs and resistance (6.8k ohm), which adjusts the electric current. The network is controlled by an RS-232C via DS9097U-E25 adapter (Phillips Corporation).

4.2 Implementation of the proposed method

We implemented the proposed method with Microsoft Visual C++ on a laptop (NEC Lavie LL750/4!$CPU: 1.5 GHz!$RAM: 752MB!$OS: Windows XP Home Edition). We used a USB camera (Logicool Qcam for Notebooks Pro) to capture the board. The system obtains camera images with DirectShow and we used IPL library (Intel) for gray-scale conversion, the blur filter, and the maximum filter. The pins' LEDs blink at 200-ms intervals since the implemented program operates at 30 ms/frame. Furthermore, it tolerates ± 2 position error in camera images on finding LEDs, and the default threshold of binarization is 10. As Fig. 7 shows, we defined the top-left corner as (0, 0) and the bottom-right as (1000, 1000). The default value of the maximum size of white connected areas, which is assumed to be a pin, is 1000 pixels, and the minimum size is fixed to 4 pixels.

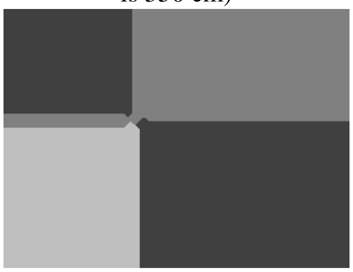
Figure 9 shows an actual example of the procedure. Figure 9(a) is a captured image and Fig. 9(b) is the difference image between Fig. 9(a) and the previous frame. Figure 9(c) shows the result.

5 Performance evaluation

5.1 Evaluating the performance of image processing

Next we evaluate the effectiveness of our position detection mechanism. We used a board measuring approximately

(a): A captured image (distance between camera and board is 350 cm)



(b): Distribution of the threshold on (a)
(dark gray: 10; gray: 8; light gray: 6)

Figure 10. Example of the threshold distribution

25 cm x 25 cm and six pins. The system manages coordination on the board as integers. We input the coordinates of four pins and evaluated the accuracy of position detection by calculating coordination of the other two pins. In addition, we set a time restriction since the program sometimes fails to find pins. If the program could not find any pins beyond the timeout, it abandoned the detection of the pin and tried to find another one.

Metrics in this evaluation are as follows.

- Position error

  This is the error between the calculated and the actual position. When the calculated result lies outside of the board, we ignore it and do not include it with the results.

- Time period for LED detection

  This is the time period between LED switching on and the program finding the pin. In the best case it takes 1 sec on average.

First, we evaluated the relationship between distance and performance. Next, we determined the effectiveness of the proposed method in various conditions: the case of a cross-shot, the case in a dark room, and the case with vividly colored pieces of paper. Figure 10 shows an example of the threshold distribution calculated in this evaluation.

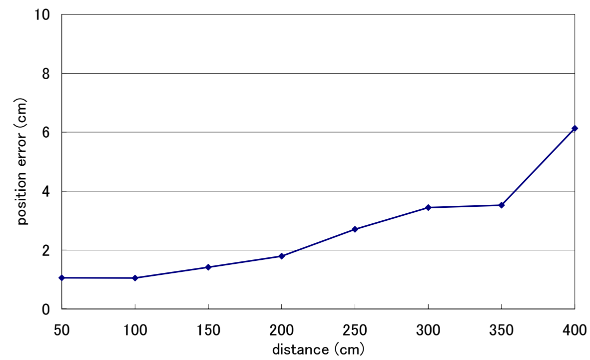


Figure 11. Performance vs. distance

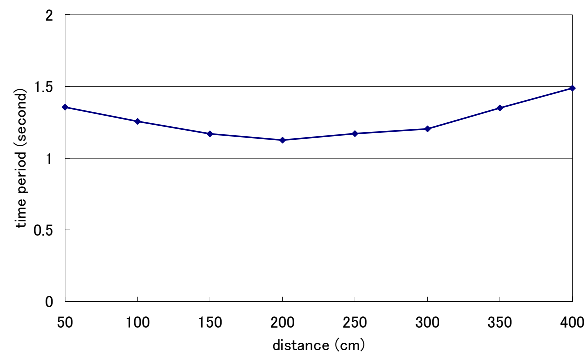


Figure 12. Time period to detect a pin vs. distance

5.1.1 Distance vs. performance

We evaluated the system's performance when the distance between the camera and the board was incrementally changed from 50 cm to 400 cm in steps of 50 cm. The detection of two pins was repeated 20 times at each position.

Figures 11 and 12 display the results of this evaluation. In Fig. 11, the vertical axis represents the distance between the camera and the board while the horizontal axis denotes the degree of position error. In Fig. 12, the vertical axis is the time period between LED switching to detect a pin, not including the case of a timeout. In addition, Table 1 shows the number of timeouts, the terminations of detection, and cases where the calculated result was outside the board.

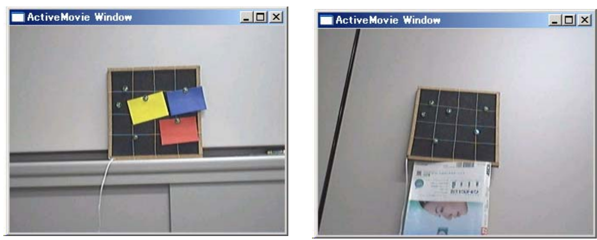
The following results are derived from this evaluation.

- The program can calculate the position of a pin at less than 250 cm, when the position error is less than 4 cm and timeout rarely occurs.

- At the distances of 350 cm and 400 cm, the numbers

Table 1. Ratio of errors

Distance (cm)	Timeout before paramater adjustment	Timeout after parameter adjustment	Aborted of pin detection	Outside of the board
50	0.08	0	0	0
100	0	0	0	0
150	0	0	0	0
200	0	0	0	0
250	0.02	0	0	0
300	0.019	0.25	0	0
350	0.27	0.55	0.15	0.03
400	0.34	0.53	0.7	0.4



(a) With vividly colored pieces of paper

(b) Cross-shot

Figure 13. Evaluation environment

of timeouts and terminations increase. This means we cannot place the camera at more than 350 cm from the board, though this depends on the size of the board.

- The number of timeouts decreased after dynamic parameter adjustment.

5.1.2 Various environments

Next, we evaluated the performance in various environments. The camera was set at 100 cm from the board. In the following four environments, we evaluated the performance using a dynamically adjusted threshold and the static threshold values of 4, 10, and 16.

- A lit-up room (normal)
- A dark room
- Cross-shot

  The camera was placed 35 cm lower than the board. Figure 13(b) shows a captured image of the situation.
- The board with vividly colored pieces of paper

  Red, yellow, and blue notices were tacked onto the board as shown in Fig. 13(a).

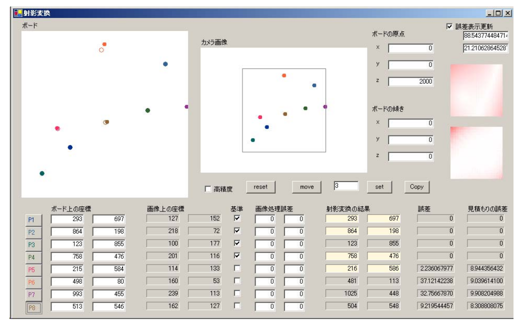
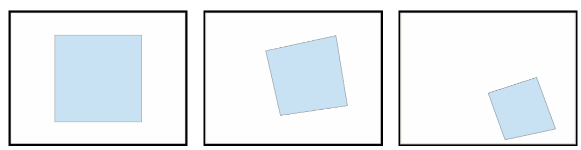


Figure 14. A screenshot of the application for determining the degree of error



Position 1 Position 2 Position 3

Figure 15. Position of board on camera image in each camera position

Table 2 shows the result. Though the performance of the adjusted threshold is behind the static threshold in some cases, the adjusted threshold achieves sufficient detection capability in all environments.

5.2 Evaluation the performance of selecting reference pins

Here we evaluate how the proposed method of selecting reference points improves error accumulation. We implemented a simulator to measure error accumulation as shown in Fig. 14. In the evaluation, we placed

Table 2. Performance in various environments.

Normal

Threshold	Error (cm)	Time of switching (sec.)	Rate of timeout
auto	0.417	1.11	0
4	0.337	1.54	0.16
10	0.772	1.28	0.04
16	0.225	1.21	0.12

Cross-shot

Threshold	Error (cm)	Time of switching (sec.)	Rate of timeout
auto	0.46	1.39	0.22
4	2.60	1.37	0.14
10	2.82	1.29	0.04
16	0.97	1.45	0.46

Dark room

Threshold	Error (cm)	Time of switching (sec.)	Rate of timeout
auto	1.15	1.51	0.08
4	-	-	-!Ævaluation impossible!K
10	1.92	1.31	0.02
16	1.48	1.25	0.08

Vividly colored paper

Threshold	Error (cm)	Time of switching (sec.)	Rate of timeout
auto	0.75	1.17	0
4	0.69	1.45	0.04
10	0.48	1.14	0
16	0.08	1.18	0.02

four pins (P_1, P_2, P_3, P_4) randomly as the default reference points. After that we placed four new pins (P_5, P_6, P_7, P_8) randomly, and repeatedly removed one of six pins $(P_3, P_4, P_5, P_6, P_7, P_8)$ and inserted it to another random position.

We evaluated the proposed method in comparison with a simple method in which the oldest four pins are selected as reference points. The evaluation was performed with three camera positions. In position 1 the camera was placed plumb in the face of the board. In position 2 the camera was placed diagonal to the front of the board, and in position 3 it was farther from the board than positions 1 and 2. Figure 15 shows images captured in these positions. The upper-left corner of the boards in the images is $(0, 0)$. In each round of measurement we repeated the relocation of pins 20 times, after which we calculated the average error of the six pins $(P_3, P_4, P_5, P_6, P_7, P_8)$. We repeated the measurements ten times, and the results are shown in Table 3, where the average errors are expressed as a ratio to the length of the side of the board. The results reveal that in all camera positions the proposed method obtained less error than the simple method, confirming the effectiveness of the proposed method.

Table 3. Performance of the reference pins selection method

	Average error in proposed method (%)	Average error in simple method (%)
Position 1	3.22	28.42
Position 2	9.69	41.26
Position 3	11.75	23.60

Figure 16 shows the results of error accumulation when we repeated the pin relocation 100 times in camera position 1. We evaluated two placements of default reference points: placement 1 with a small error and placement 2 with large error. The graph shows that the error does not accumulate in the proposed method; it is especially low in placement 1. On the other hand, in the simple method, the error accumulates after the relocations and it too large to utilize in practices. This means the proposed method can avoid sets of reference points that have an undesirable positional relationship (as shown in Fig. 8) and also avoid accumulation of error.

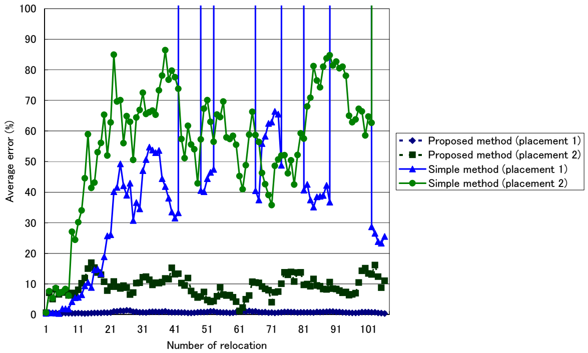


Figure 16. Accumulation of error

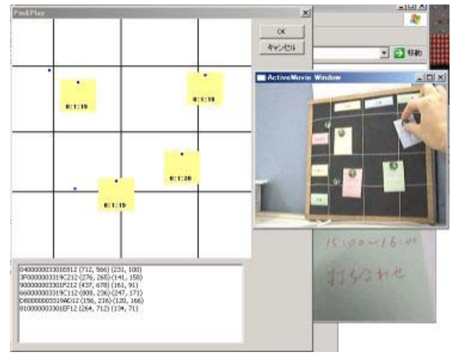


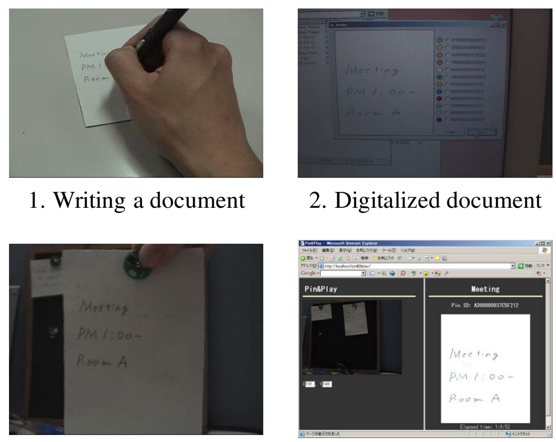
Figure 17. Screenshot of the schedule managing/sharing application

6 Applications

Boards and pins are common instruments, and it easy to grasp the entire situation from boards with attached actual papers. Conventional computer systems, however, have difficulty in realizing this characteristic. Our position detection method can be used to produce novel types of applications combining the merits of paper, which give us a large field of view with merits of digitization, with allows users to share information remotely and manage it flexibly. We have developed two applications that employ our method.

Figure 17 shows a screenshot of a schedule managing/sharing application for several users in a group, such as a research group or a family, on a board. To add a new schedule, a user writes the schedule contents on a note; the system captures an image of the note and the user is pinning it on the board. The board is divided into several blocks, the positions of which denote time and author. After detecting the position of an inserted pin, the system records the captured image with time and author information. To delete this schedule, he has only to remove the note. The notes of current schedules are emphasized by turning on their respective pins'LED. If schedules conflict, the blinking of another colored LED indicates the confliction. Users can also check their schedule by PC and cellular phone remotely via http connection. The system automatically updates schedule data when it detects such changes as addition and removal of a pin. The location information of inserted pins helps the simple interaction that enables user can to obtain detailed information by just clicking a note on a picture. Conventionally, even if people frequently check each others' schedules, problems may arise if for example two or more are set for the same time. This application will alleviate such problems.

Figure 18 shows an example using our document management board application. The application manages posted documents in the same manner as the Reminding Notice



1. Writing a document
2. Digitalized document
3. Posting the document
4. Checking Documents

Figure 18. Document management board

Board. Moreover, user can check the board remotely. A user writes a document by hand and posts it to the board. The application records the handwritten information, inserted time, its position, and current camera image. By using such information the application can show the board's current status via a Web browser.

7 Related works

PushPin Computing is a related work that uses pin-shaped devices [7]. A position-detection method for PushPin Computing has also been proposed [2]. In this method, positions of pins are detected by using a device called a pinger. The pinger simultaneously sends light and ultrasonic waves, and pins calculate their position autonomously from the time difference between the light and ultrasonic waves. Though its error is about 4 cm and it is reasonably accurate, it needs extension to pins and new devices are required for only position detection.

Various conventional methods to detect positions of devices using image processing have already been proposed. There are several methods such as those to detect objects just using image processing, methods using paper-printed markers, and methods using tags with infrared LEDs or other lighting devices. Papier-Mache [5] is a tool kit for realizing tangible interfaces using image processing and barcodes. Methods just employing image processing need complete image processing however it is difficult to recognize similar-shaped objects. Famous examples of paper-printed markers to obtain three-dimensional positions are ARToolKit [11], CyberCode [9], and TRIP [4]. There are many studies on this type of image processing, which can detect objects and recognize each one more easily. However, markers measuring several square centimeters must be attached to every object. This can be troublesome, especially with small objects. In the proposed method, we use LEDs, which have already been fitted to certain devices, and we use the LEDs for pin detection only during operation. Thus, our method can solve these problems.

Examples of methods using tags are a system using the blinking of infrared tags [1] and a system that captures blinking beacons with a special camera [8]. Since the systems in these studies cannot estimate when devices will appear in their captured images, they use infrared fields or high-speed blinking to find the devices robustly. In our research, the image processing program can use the LED status via a network and the system can have LEDs continue blinking until it finds them with certainty.

ALTAIR [10] is a system for controlling infrared LED tags via network and obtains positions of tags by image processing. This system needs exclusive infrared tags in order to detect tags' position. Although this is different from our proposed method, which uses LEDs fitted to the devices, it can control tags just like the proposed method. These methods can control the blinking of LEDs and attempt image processing again to obtain more accurate positions if they estimate large error. Moreover, they can retry detection of LEDs later if LEDs are presently occluded.

8 Conclusion

In this paper, we proposed a position detection mechanism using a camera for Pin&Play. Our proposed mechanism has two characteristics: adaptation of image-processing parameters and error estimation. We could expand the application of Pin&Play by utilizing information on pin positions. We also evaluated the performance and implemented two applications.

We plan to extend the mechanism to deal with multiple cameras and boards and realize a framework to develop application that controls cameras and pins remotely. Moreover, we are also considering other applications.

Acknowledgement

This research was supported in part by "The 21st Century Center of Excellence Program" and Grant-in-Aid for Scientific Research (A) (17200006) of the Japanese Ministry of Education, Culture, Sports, Science and Technology.

References

[1] H. Aoki and S. Matsushita. Balloon tag: (in)visible marker which tells who's who. In *Proc. of the Fourth International Symposium on Wearable Computers (ISWC '00)*, 2000.

[2] M. Broxton, J. Lifton, and J. A. Paradiso. Localizing a sensor network via collaborative processing of global stimuli. In *Proc. of the Second European Workshop on Wireless Sensor Networks*, pages 321–332, Jan./Feb 2005.

[3] F. Helin, T. Hoglund, R. Zackaroff, M. Hakansson, S. Ljungblad, and L. E. Holmquist. Supporting collaborative scheduling with interactive pushpins and networking surfaces. In *UbiComp 2004 Adjunct Proceedings*, Sept. 2004.

[4] D. L. Ipina, P. Mendonca, and A. Hopper. Trip: a low-cost vision-based location system for ubiquitous computing. *Personal and Ubiquitous Computing journal*, 6(3):206–219, May 2002.

[5] S. R. Klemmer, J. Li, J. Lin, and J. A. Landay. Papier-mache: toolkit support for tangible input. In *Proc. of Conference on Human Factors in Computing Systems (CHI2004)*, pages 399–406, Apr. 2004.

[6] K. V. Laerhoven, N. Villar, A. Schmidt, H.-W. Gellersen, M. Hakansson, and L. E. Holmquist. Pin&play: The surface as network medium. *IEEE Communications Magazine*, 41(4):90–96, Apr. 2003.

[7] J. Lifton, D. Seetharam, M. Broxton, and J. A. Paradiso. Pushpin computing system overview: a platform for distributed, embedded, ubiquitous sensor networks. In *Proc. of Pervasive 2002*, pages 139–151, Aug. 2002.

[8] N. Matsushita, D. Hihara, T. Ushiro, S. Yoshimura, Rekimoto, and Y. Yamamoto. Id cam: A smart camera for scene capturing and id recognition. In *Proc.of ACM International Symposium on Mixed and Augmented Reality (ISMAR 2003)*, pages 227–236, 2003.

[9] J. Rekimoto. Matrix: A realtime object identification and registration method for augmented reality. In *Proc. of Asia Pacific Computer Human Interaction 1998 (AOCHI'98)*, 1998.

[10] M. Sakata, H. Sasaki, M. Imura, Y. Yasumuro, Y. Manabe, and K. Chihara. Active ir-tag user location system for mr collaborative environment. In *Proc. of 3rd CREST/ISWC Workshop on Advanced Computing and Communicating Techniques for Wearable Information Playing*, pages 90–93, Oct. 2004.

[11] *ARToolKit*. http://www.hitl.washington.edu/artoolkit/.

[12] *Pin&Play*. http://ubicomp.lancs.ac.uk/pin&play/.