Regular Paper

Worker State Estimation Method with Reduced Manual Task for Teleworking Environment

Kazuyuki Iso[†], Takaya Yuizono[†], and Minoru Kobayashi[‡]

[†]Japan Advanced Institute of Science and Technology, 1-1 Asahidai, Nomi, Ishikawa, Japan
[‡]Meiji University, 4-21-1 Nakano, Nakano-ku, Tokyo, Japan
{iso.kazuyuki, yuizono}@jaist.ac.jp
minoru@acm.org

Abstract - In a teleworking environment, sharing the states of workers is important to facilitate smooth communication; this requires a worker state estimation method to be adaptable to various workspaces. Previous methods have realized high estimation accuracy, but they have needed laborious manual work to suitably tag the learning data. This paper proposes a novel worker state estimation method by reducing the manual task of labeling using three automated processes: sensing, clustering, and selecting. A prototype system was developed and tested in two workspaces for evaluation. The system selected 26.1% and 22.5% of the obtained data for labeling in workspace 1 and workspace 2, respectively. As the data for labeling decreased, the observation time of a worker also decreased. Approximately 75% of the manual work could be reduced. The estimation accuracy was 93.2% in workspace 1 and 76.0% in workspace 2. This method was effective in reducing the manual labor involved in estimation. The estimation accuracy differed depending on the use conditions of the workspace. Methods to improve this metric are also discussed.

Keywords: Worker state estimation, Worker state sharing, Telework, Unsupervised clustering method

1 INTRODUCTION

A broadband network enables internet connectivity and allows the sharing of multimedia content. This allows a social infrastructure to be built, with which people can collaborate using multipoint video conferencing systems, chat systems, e-mails, and telephone conferences. In many companies, workers work remotely from various places, such as their homes. To work in cooperation with an organization, communication between the group members is important. Workers often need to communicate with remote workers; however, this also means that, while communicating, the individual work of the remote worker being contacted must be suspended. If remote workers are interrupted at inappropriate times, their individual work would suffer and, their productivity would decline [1][2]. When working in a common room, there is a shared awareness between the workers; a worker easily notices the states of the other workers. Therefore, a worker can initiate communication at an appropriate time. However, in teleworking, a worker is not able to easily discern the states of other workers, and therefore, abruptly sends them a message, either through a chat system or by a telephone call, thereby interrupting the work of those workers. For effective remote collaboration, recognition of the states of the remote workers is important. Related research [3]–[7] has reported on the significance of this factor.

This research aims to develop a worker state estimation method for use in teleworking. Figure1 shows the concept of the worker state sharing system. A terminal is placed in the room for teleworking. In the figure, co-worker B is a collaborator at another location. The role of this terminal is to estimate the states of the workers in the room and to share these states with other co-workers. The terminal is made aware of the state of a worker, which helps workers to time their communication opportunely. As shown in this figure, the system confirms the state of a worker and improves accuracy in estimation. At this time, the terminal classifies the state of the worker; the role of the worker is to teach the system how to share the current state. Following this process, the terminal does not disturb the worker when crafting, and the worker can appropriately communicate, or be communicated with, using a telephone or messenger, after the crafting activity is completed. The terminal can estimate the state adopted by the worker.

Many studies on human activity estimation have reported a high level of accuracy in estimation by using machine learning [8]. To build an estimator by the conventional method requires enormous amounts of learning data. The learning data are created by labeling supervised information onto data collected from numerous sensors. In many studies, creating this learning data is large and laborious a manual task.

Therefore, we propose a new method to construct an estimator that reduces the manual task. This method has four processes: sensing, clustering, selecting, and labeling. A comparison of the conventional method and the proposed method is shown in Fig. 2. The conventional method manually labels the sensor data and supervised information, and inputs them into the learning process. The target data for labeling include all the sensor data. The proposed method classifies the sensor data using a clustering method. The system selects the target data for labeling based on the results of the clustering process. The number of targets equals the number of clusters. As in the conventional method, the system initially creates an estimator with these four processes. In the proposed method, the process from sensing step to selecting step is automated

ISSN1883-4566 © 2020 - Informatics Society and the authors. All rights reserved.

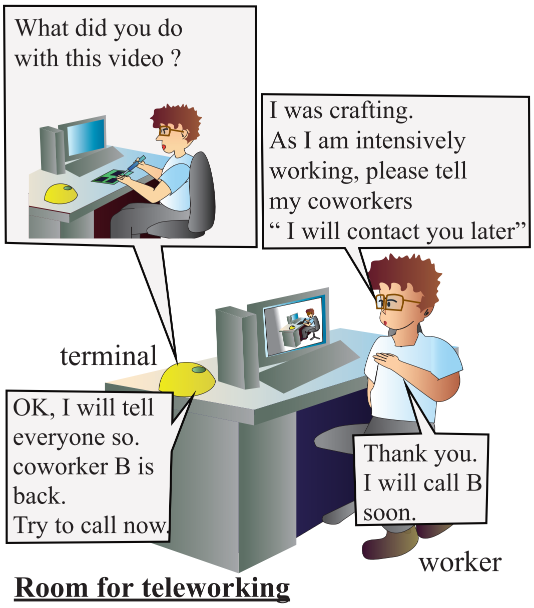

What did you do with this video ?
I was crafting. As I am intensively working, please tell my coworkers " I will contact you later"
terminal
OK, I will tell everyone so. coworker B is back. Try to call now.
Thank you. I will call B soon.
worker
Room for teleworking

Figure 1: Concept of the worker state sharing system. The terminal confirms the state of a worker and develops an estimator.

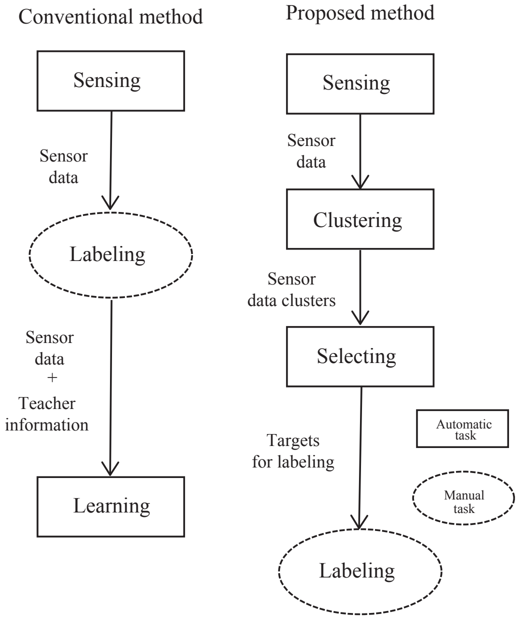


Figure 2: Comparison of the differences in the proposed method and conventional method

task. This process of selection decreases the manual tasks by reducing the number of targets for labeling. The labeling process is carried out only once after multiple targets are selected by the selection process. The worker observes the target scenes selected by the selection process and indicates the worker's state. An estimator is created by running these four processes only once. When estimating the worker's state from new sensing data, the system calculates the cluster containing the sensing data. At the time of collection of the sensing data, the worker's state is predicted by using the label of the cluster.

A prototype system was developed to evaluate the proposed method, and experiments were performed in two workspaces. The evaluation experiments confirmed the reduction in manual work. The rest of this paper is organized structure as follows. Chapter 2 describes related research on human activity estimation. Chapter 3 describes the proposed method in detail. Chapter 4 describes the experimental prototype system. Chapter 5 describes the experimental methods and their results. Chapter 6 provides a discussion of the results. Chapter 7 presents the conclusions.

2 RELATED RESEARCH

2.1 Human Activity Estimation

Numerous reports on human behavior estimation employ machine learning [8]. Avrahami et al. [9] reported on the estimation of the behavior of convenience store clerks and desk workers. They used a support vector machine (SVM) and the K-nearest neighbor (KNN) to learn the states of store clerks and office workers. A large amount of supervised data is required for learning using SVM- or KNN-based systems. In addition, Laput et al. [10] reported on a technological solution to estimate the activity in a house, and also used an SVM as the learning method. Both studies reported a high level of accuracy in estimation, but they required a large amount of learning data with supervised information to be generated.

Each telework environment is different in terms of the size and structure of the room, as well as the size and arrangement of the furniture. These conditions affect the data input from the sensor. Collecting data under all conditions is difficult, and learning data must be generated for each work environment. Reducing the magnitude of the task of generation of the learning data is important.

2.2 Sensor for Human Activity Estimation

Previous studies have reported that various sensors can be used to a estimate human activity. These sensors are classified into three types, as shown in Fig. 3.

Murao et al. [11] used wearable sensors to estimate the state of a remote worker. A wearable sensor can typically sense the worker to which it is attached. A wearable sensor is battery-powered, and therefore, requires regular charging. This charging is inconvenient for workers.

A personal computer (PC) is a well-known tool for teleworking. Hashimoto et al. [12] obtained useful information about remote workers by maintaining an operational log on the PCs of the workers. This method only estimates the work done using a PC.

Other methods for estimation have been proposed that use ambient sensors. Laput et al. [10] used a sensor module that combined multiple sensors in a room to estimate human activity. Avrahami et al. [9] installed a radio frequency (RF)-radar under a desk to sense human motion without interfering the work. However, these studies have a heavy manual tasks load of creating a large amount of learning data by labeling super-

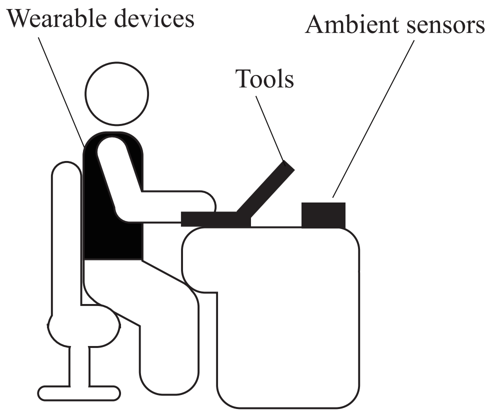


Figure 3: Classification of the sensors that collect the data for human activity estimation

vised information to sensor data.

Ambient sensors can be installed at locations that do not interfere with the actual work, and where a stable source of power is available. An ambient sensor can also collect data when the worker is not using the PC. Such ambient sensors are suitable for use in teleworking environments. The proposed method uses an ambient sensor installed in a telework environment. Our previous research [13][14] reported experiments to collect data using ambient sensors for classifying workers' states. The new proposed method creates an estimator using four processes including the selecting process and the labeling process.

3 PROPOSED METHOD

The proposed method consists of four processes: sensing, clustering, selecting, and labeling. The process from sensing to selecting is automated. Only labeling is the manual task. When developing an estimator with these four processes, the system records a video simultaneously along with the collection of sensor data. This video is used to observe the state of a worker in the labeling process. After clustering the sensor data, the system selects from the entire video the scene to be observed. This scene selection process shortens the observation time. The four processes are described in detail in the following paragraphs.

3.1 Sensing

The sensing process uses a microphone and distance sensor. The microphone detects signals based on the behavior of the worker (e.g., worker voice, keystroke sound, and door opening and closing sounds). The distance sensor detects the area where the worker is present. Our previous research [13][14] reported that the states of a worker can be clustered using vibration and distance sensors. In this research, we replaced the vibration sensor with a microphone, which can sense voices and sounds all around in a workspace.

The feature quantity, as shown in Fig. 4, is determined at fixed time intervals from each sensor data. The microphone data are Fourier transformed, and the distance sensor data are averaged.

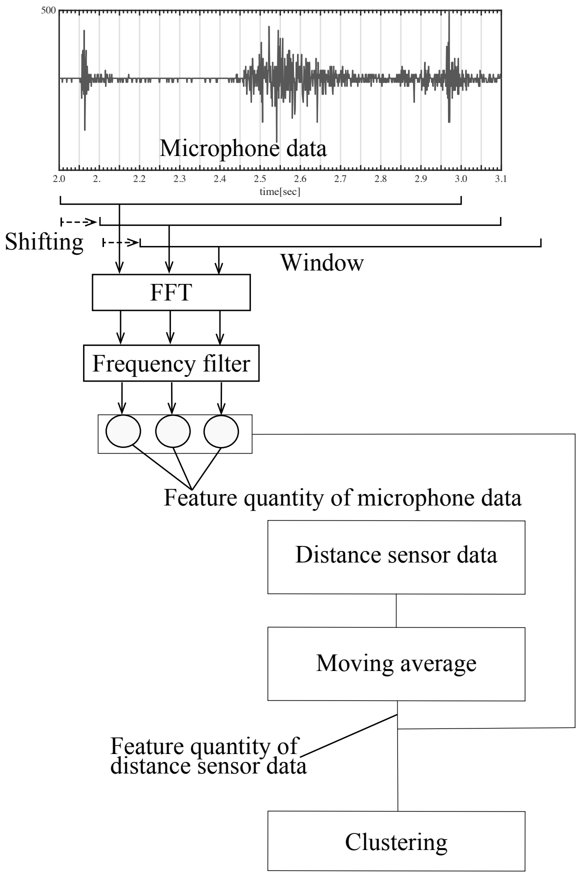


Figure 4: Sensing of the data from a microphone and distance sensor and calculation method of the feature quantity.

3.2 Clustering

This process uses an unsupervised clustering method. There are many clustering methods [15]. We compared the following four methods in terms of the accuracy and calculation time: k-means method, Gaussian mixture model (GMM) method, mean-shift method, and spectral clustering method (Table 1). For this evaluation, data were collected in the workspace for teleworking. The microphone and distance sensor data collected in the sensing process were used. When data collection was conducted, the worker's states were classified into the following four types: "Meeting via video conference.", "Using a PC on a desk", "Crafting on a desk", and "Leaving the seat". The data were collected for one day only. The calculation time ratio in Table 1 is a ratio for calculating time using the k-means method.

The proposed method uses the k-means method. The two methods, mean-shift and spectral clustering, slightly improve the accuracy compared to the k-means method, but the calculation time is more than 18 times, which is extremely long. There is nearly no difference between the accuracies of the GMM and k-means methods; however, the calculation time

Table 1: Accuracies and calculation times of the clustering methods

Method	Accuracy rate [%]	Calculation time ratio
k-means	89.4	1.0
GMM	89.6	1.3
Mean-shift	91.0	21.8
Spectral clustering	91.6	18.4

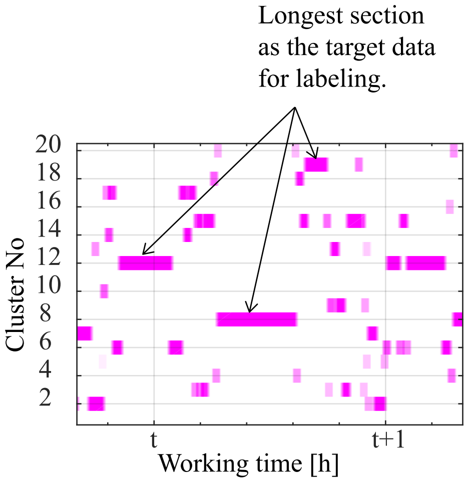


Figure 5: Example of arranging the sensor data in a time series and selecting the longest section as the target data

for the latter is short.

The k-means method can be used for clustering with sensors, even on small computers. Miniaturization of a terminal facilitates installation in the workspace.

3.3 Selecting

The system selects the target data so as to reduce the manual task of labeling. In the labeling process, a worker observes the video of the sensed time of the target data, determines the worker state, and labels the target data. The system selects the target data from the clustering result and cuts out the video scene captured when the selected data are sensed. The state of a worker may change frequently in a short time or remain the same for a long time. When a scene that changes frequently is selected, the length of the video to be observed is reduced, but the determination of the state becomes difficult. When a scene in a specific state continues to be selected, the video scene to be observed becomes longer, but the determination of the state is smooth. This process arranges the data into the cluster in a time series and selects the longest section as the target data for labeling (Fig. 5). The system cuts out the video scene from the time when the target data are sensed and presents the video scene to the worker. The system selects the data of one section from one cluster.

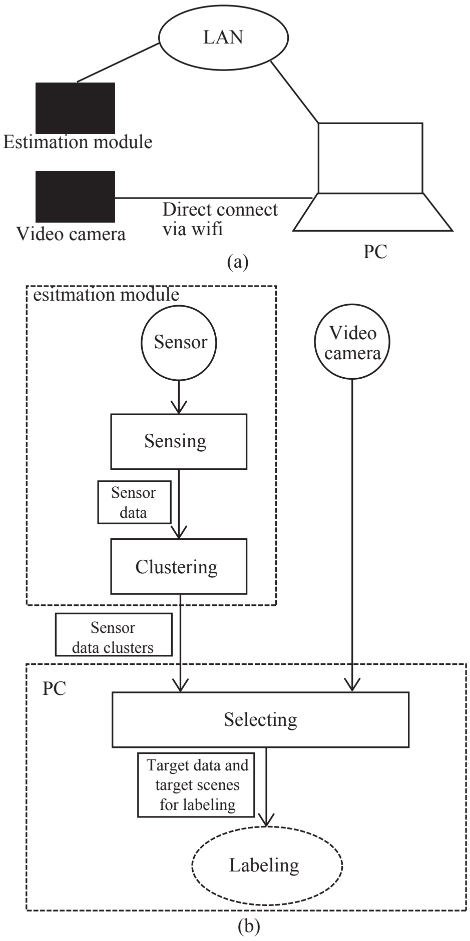


Figure 6: Prototype system. (a) Hardware configuration. (b) Flow of the four processes in the prototype system.

3.4 Labeling

The worker observes the video scene selected by the system and determines the worker state. The worker returns the determined state to the system. The number of video scenes observed in this process is the same as the number of clusters. The length of the video scene is shorter than that in the selecting process. The observation time for labeling is shorter than in the case of observing all the video scenes.

The system provides a complete estimator, with the workers labeling each cluster. When new sensing data is provided to this estimator, the distance between the data and the center of each cluster is calculated; the cluster was created by the k-means method. The closest cluster is selected, and the worker's state is predicted by using the label of the cluster.

4 Prototype system to develop estimator using four processes

As shown in Fig. 6, a prototype system is developed to evaluate this method. The prototype terminal consists of a sensor and a small computer, and executes the processes of sensing and clustering. The timing of the prototype terminal and video camera are synchronized. The prototype terminal is described in detail in the next section.

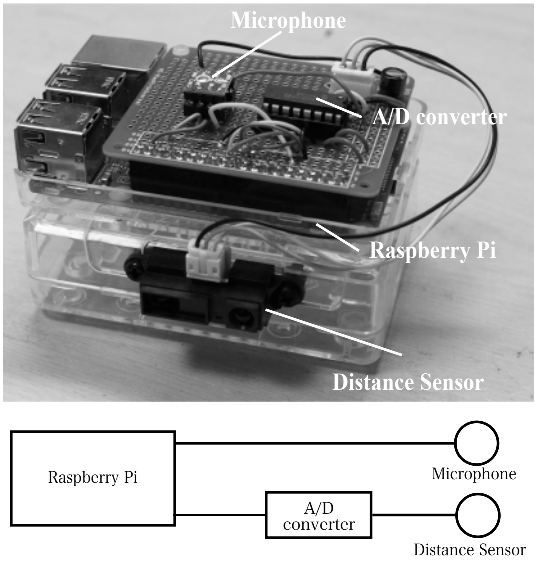


Figure 7: Prototype terminal with a combination of a microphone and distance sensors.

The video camera of this system is GoPro5. The prototype terminal and the PC are connected by the same LAN. Videos from the camera are transmitted directly to the PC. The clustering results are transmitted from the prototype terminal to the PC.

In the selecting process, the prototype program on the PC selects the target data, cuts out the video, and presents the video scenes to the worker. In the labeling process, the worker observes the video and labels the target data on the same PC.

The prototype terminal implements the sensing and clustering processes (Fig. 7). The terminal is used on a work desk. This terminal receives data from the microphone and distance sensors synchronously using a microcontroller. The microcontroller uses Raspberry Pi3 Model B. The microphone used is ADMP441, manufactured by Analog Devices. The distance sensor used is GP2Y0A710K, manufactured by SHARP. The direction of the distance sensor of the terminal is adjusted to the location where a worker sits. The sensing data from the two sensors are clustered by the k-means method in this terminal.

5 EVALUATION TEST USING PROTOTYPE SYSTEM

5.1 Test Procedure

The terminals were positioned at two workspaces for the evaluation test. The first workspace (WS1) is a private room used by a worker at his home, that the worker used for teleworking once or twice in a week. Other co-workers never enter the room. When working in WS1, the worker in this room talks to other co-workers by videoconference. The estimator for WS1 was created by collecting data for one day at this workspace.

The second workspace (WS2) is a university office and is the room of an instructor. WS2 is primarily used by one worker, and other co-workers may enter it. In addition, the worker in this room talks to co-workers via a video conference. The estimator for WS2 was created from three-day sensor data, including in-room conferences and video conferences. The sensor data of the different workdays were collected and evaluated.

Table 2: Sensing result: collecting time and number of sensor data

Workspace	Total time	Number of data
WS1	12h 53min	44,654
WS2	16h 11min	59,105

Table 3: Selecting result: selecting time and number of sensor data. The last column is the number of video files created by the selecting process

Workspace	Total time	Number of selected data	Number of movie files
WS1	3h 21min	12,097	20
WS2	3h 38min	13,095	30

5.2 Test Result

Table 2 lists the collection time and number of sensor data determined using the prototype terminal.

The total time and number of data after the selecting process are listed in Table 3. The last column is the number of video files created by the selecting process. The system selects 26.1% of the data for labeling in WS1 and 22.5% data in WS2. The number of video scenes to observe is small: 20 scenes in WS1 and 30 scenes in WS2. As the target data for the labeling decreases, the observation time of a worker also decreases.

The four states of the workers in each workspace are observed in the video after the selecting process.

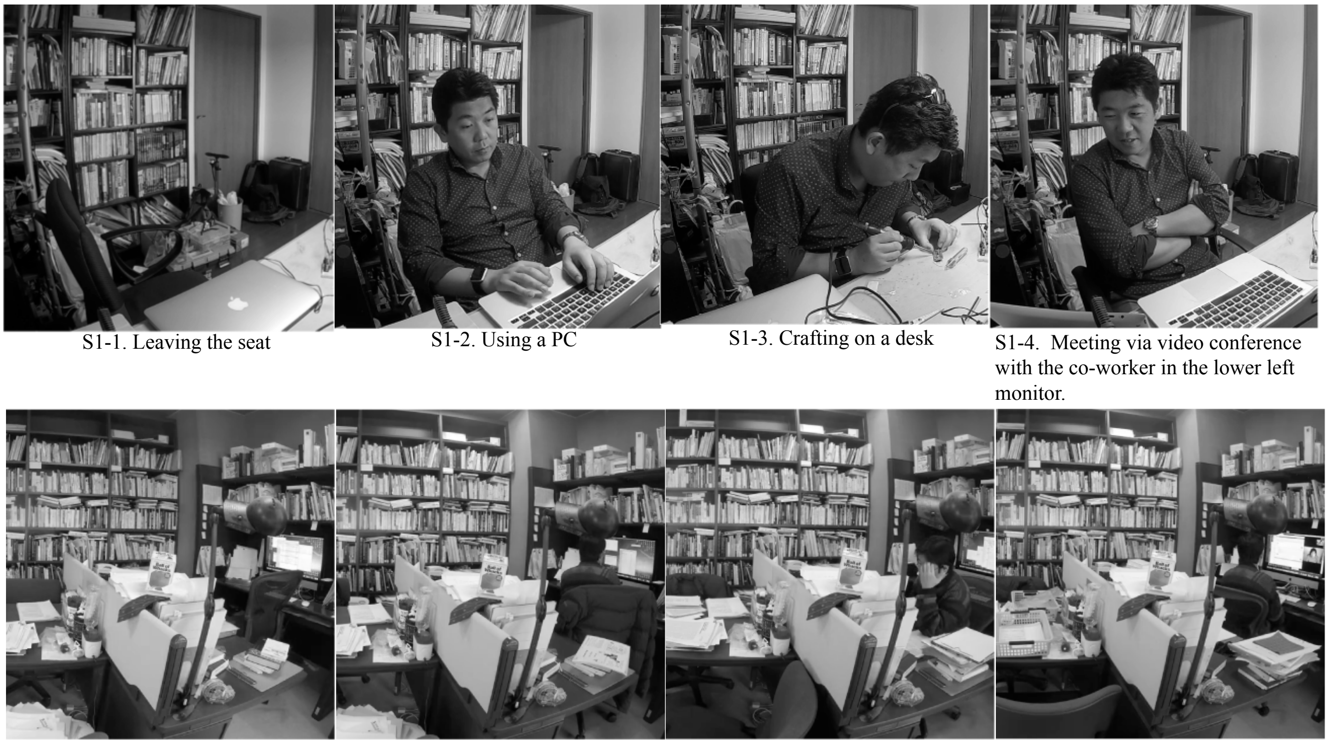
The states of a worker in WS1 are:

S1-1: Leaving the seat.
The worker leaves the seat and moves to the next room.

S1-2: Using a PC on a desk
For example, the worker creates documents using a PC on the desk or browses the web.

S1-3: Crafting on a desk
The worker works without using the PC. In this case, the worker crafts an electric circuit (soldering, cable making, and circuit assembly).

S1-4: Meeting via video conference.
Workers in both the workspaces use a PC to conduct a video conference. The workers talk to a remote worker. The worker in WS1 talks to other people in the same room.



S1-1. Leaving the seat

S1-2. Using a PC

S1-3. Crafting on a desk

S1-4. Meeting via video conference with the co-worker in the lower left monitor.

S2-1. Leaving the seat

S2-2. Using a PC on the wall side desk

S2-3. Writing or reading on the central desk

S2-4. Talking with other co-worker

Figure 8: States of a worker in workspace 1 and workspace 2

The states of a worker in WS2:

S2-1: Leaving the seat.
The worker leaves the seat and moves to the next room.

S2-2: Using a PC on the wall side desk
The worker uses a desktop PC on the wall desk. For example, the worker creates documents using a PC on the desk or browses the web.

S2-3: Writing or reading on the central desk
The worker works without using the PC. For example, the worker writes or reads paper documents. The worker works quietly and concentrates on the task.

S2-4: Talking with other co-workers
The worker talks with other co-workers in the same room. He also talks with a remote co-worker using a PC to conduct a video conference.

The estimator is created when a labeling result corresponds to a clustering result. The sensor data for the evaluation were collected on workdays different from the days when the estimator was made. The estimation accuracy at this time was 93.2% for workspace 1 and 76.0% for workspace 2. The estimation results for WS1 are shown in Fig. 9 and are shown in Fig. 10 for WS2.

The conventional method[9] using a single terminal could estimate the office workers' states with an accuracy of approximately 90%, and the same level of accuracy will be required in a telework environment. The result of successful estimation in WS1 was 93.2% and was very close to the workers' states observed in the video for several hours. The worker's states can be shared with an accuracy of 93.2%, and the effect of reducing unnecessary interruptions will be high. This result would be sufficient for meaningful sharing with remote co-workers.

In WS2, the estimation result and the workers' states showed a similarity of 76.0%. However, for the time segment T1 in Fig. 10, the estimation results for state "S2-4: Talking with other co-workers" were mostly incorrect for a duration of 25 minutes. The estimation accuracy was lower than that of WS1 and was insufficient. In the future, the effects of estimation errors need to be examined.

6 DISCUSSION

6.1 Reduction in Manual Tasks by Proposed Method

The video scene for labeling was reduced by more than 70%. At this time, each video contains the same work scene, and there is little change. The proposed method reduced the time required for observation and enabled the efficient determination of the state of a worker.

The labeling process was short and smooth, and estimators could be developed for each workspace. The worker state sharing system using these estimators could be constructed, and the sharing information was defined according to the task of each worker and the structure of the workspace. This method enabled co-workers to share each others state, and is good for creating communication opportunities according

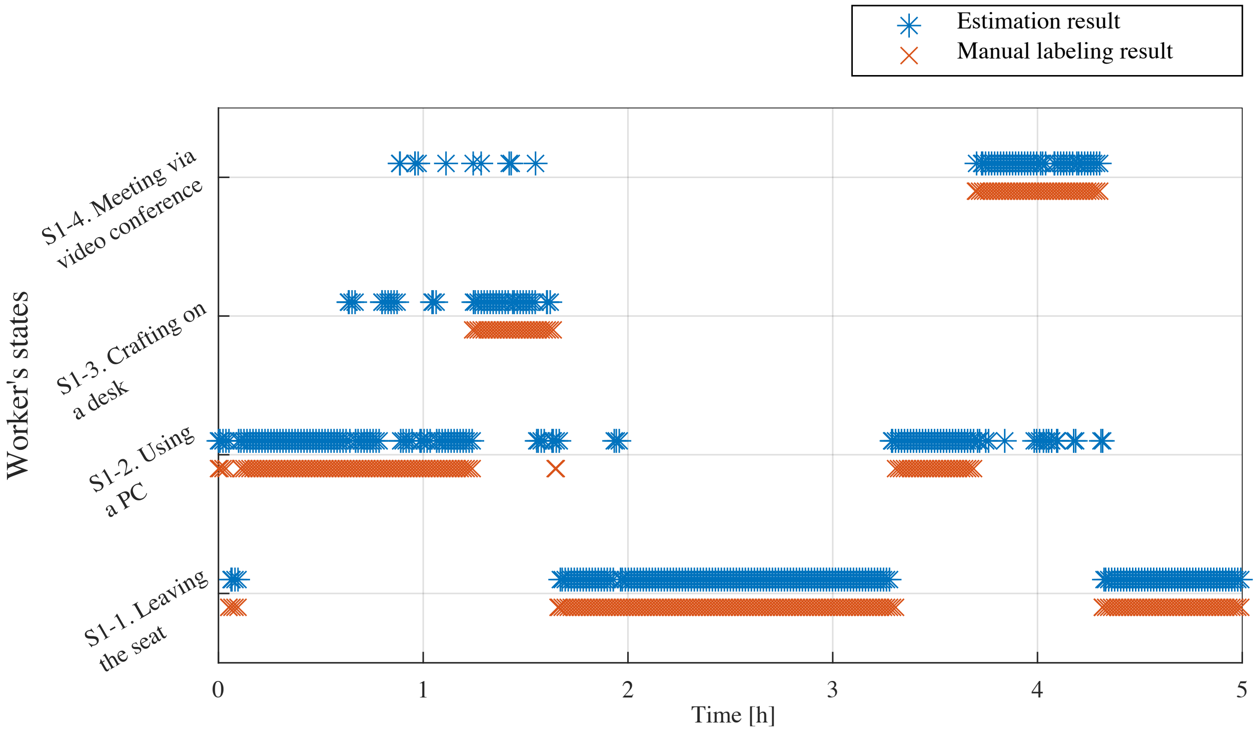


Figure 9: Estimation result for 5 h from the start of the work in workspace 1.

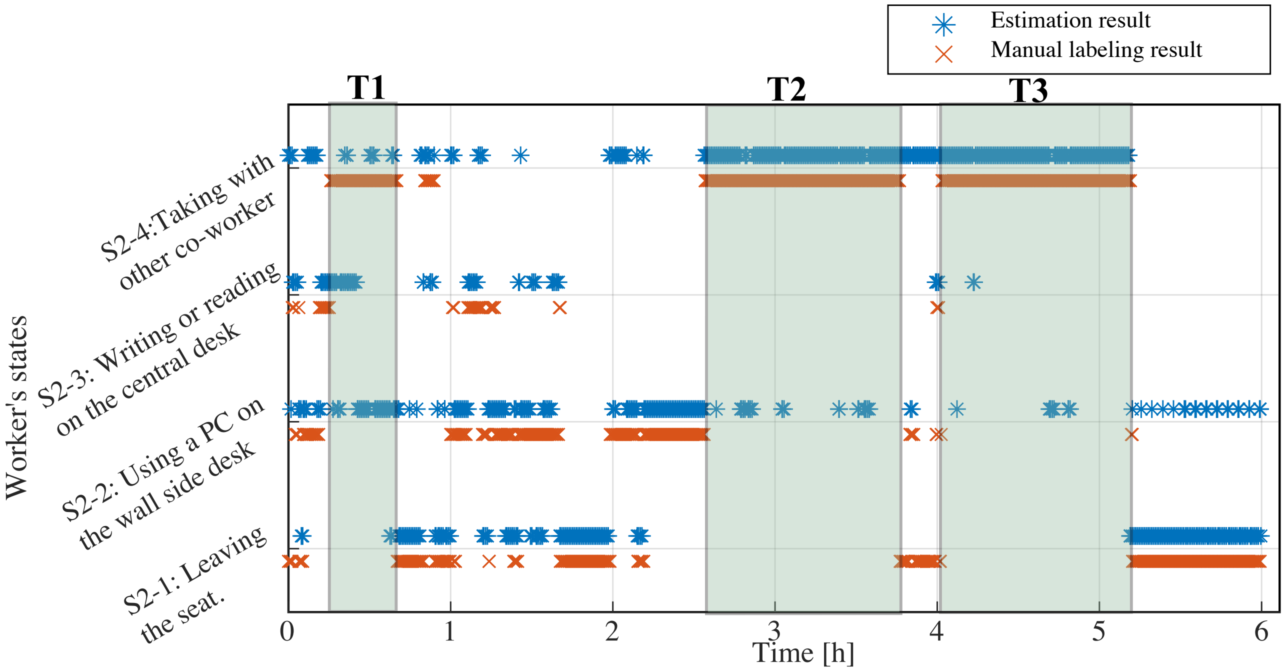


Figure 10: Estimation result for 6 h from the start of the work in workspace 2.

to the states of each worker.

Occasionally, even if a worker uses the same room, the features of the work may change, and the tools used may differ accordingly. In such cases, the estimator may have to be recreated to account for such changes. We expect the idea of the present method also contributes to reduce the amount of manual work to recreate the estimator. To develop a more robust method, we plan to examine the effectiveness of the present method to recreate the estimator based on a large amount of new sensor data generated when the features of the work has been modified.

6.2 Number of Workers in Workspace and Configuration of Sensors

The estimation accuracy of WS1 was high at approximately 90%, but the accuracy of WS2 was lower than that of WS1. One of the differences between the two workspaces was the number of workers. In time segment T1 in Fig. 10, another worker entered WS2, and the two workers talked. In addition, the two workers sometimes quietly browsed documents or searched for books on a bookshelf. In time segment T2 and T3 in Fig. 10, a worker in WS2 and a remote worker were talking to each other via video conference, and there was almost no quiet time. The estimation accuracies were different for these three segments because of the amount of voice data from the microphone. When designing this system, we assumed that more voice information could be obtained if there were two workers in the same room.

Only one distance sensor was attached to the terminal, and the system could not classify the differences in the states depending based on the number of workers. To classify the states of multiple workers, the terminal should have multiple distance sensors connected pointing in multiple directions, as the accuracy would then increase if the sensor data changed.

We will continue to improve the sensors on the terminals while investigating different conditions of the teleworking environment. Furthermore, we will install improved estimators in many telework environments and also evaluate the effect on remote collaboration work.

7 CONCLUSION

The sharing of the states of a worker with other workers is important to facilitate effective communication in teleworking. In this study, we developed a new method that required less manual work to develop a worker state estimator. We tested the new method for estimation in two workspaces. The system selected 26.1% of the data for labeling in workspace 1 and 22.5% data in workspace 2. As the target data for the labeling decreased, the observation time of a worker also decreased. Thus, 70% or more of the manual work could be reduced. The estimation accuracy at this time was 93.2% in workspace 1 and 76.0% in workspace 2. The proposed method significantly reduced the manual tasks. However, the estimation accuracy differed depending on the use conditions at the workspace. This difference may have been caused by presence of other co-workers. The accuracy could be improved by adding additional distance sensor(s) to consider other co-workers.

In future, the prototype system will be extended to remote collaboration to evaluate the effect of this method on smooth communication in this aspect also.

ACKNOWLEDGMENTS

We would like to thank Editage (www.editage.jp) for English language editing.

REFERENCES

[1] D. D. Salvucci, N. A. Taatgen, and J. P. Borst, “Toward a unified theory of the multitasking continuum: From concurrent performance to task switching, interruption, and resumption,” in *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI ’09, (New York, NY, USA), pp. 1819–1828, ACM (2009).

[2] G. Mark, V. M. Gonzalez, and J. Harris, “No task left behind?: Examining the nature of fragmented work,” in *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI ’05, (New York, NY, USA), pp. 321–330, ACM (2005).

[3] P. Dourish and V. Bellotti, “Awareness and coordination in shared workspaces,” in *Proceedings of the 1992 ACM Conference on Computer-supported Cooperative Work*, CSCW ’92, pp. 107–114 (1992).

[4] T. Rodden, “Populating the application: A model of awareness for cooperative applications,” in *Proceedings of the 1996 ACM Conference on Computer Supported Cooperative Work*, CSCW ’96, pp. 87–96 (1996).

[5] C. Gutwin and S. Greenberg, “Design for individuals, design for groups: Tradeoffs between power and workspace awareness,” in *Proceedings of the 1998 ACM Conference on Computer Supported Cooperative Work*, CSCW ’98, pp. 207–216 (1998).

[6] C. Gutwin and S. Greenberg, “A descriptive framework of workspace awareness for real-time groupware,” *Comput. Supported Coop. Work*, vol. 11, pp. 411–446 (2002).

[7] N. Romero, G. McEwan, and S. Greenberg, “A field study of community bar: (mis)-matches between theory and practice,” in *Proceedings of the 2007 International ACM Conference on Supporting Group Work*, GROUP ’07, (New York, NY, USA), pp. 89–98, ACM (2007).

[8] A. Bulling, U. Blanke, and B. Schiele, “A tutorial on human activity recognition using body-worn inertial sensors,” *ACM Comput. Surv.*, vol. 46, pp. 33:1–33:33 (2014).

[9] D. Avrahami, M. Patel, Y. Yamaura, and S. Kratz, “Below the surface: Unobtrusive activity recognition for work surfaces using rf-radar sensing,” in *23rd International Conference on Intelligent User Interfaces*, IUI ’18, (New York, NY, USA), pp. 439–451, ACM (2018).

[10] G. Laput, Y. Zhang, and C. Harrison, “Synthetic sensors: Towards general-purpose sensing,” in *Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems*, CHI ’17, (New York, NY, USA), pp. 3986–3999, ACM (2017).

[11] K. Murao and T. Terada, "A combined-activity recognition method with accelerometers," *Journal of Information Processing*, vol. 24, no. 3, pp. 512–521 (2016).

[12] S. Hashimoto, T. Tanaka, K. Aoki, and K. Fujita, "Improvement of interruptibility estimation during pc work by reflecting conversation status," *IEICE Transactions on Information and Systems*, vol. E97.D, no. 12, pp. 3171–3180 (2014).

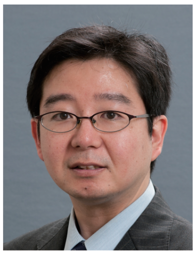
[13] K. Iso, M. Kobayashi, and T. Yuizono, "A method for estimating worker states using a combination of ambient sensors for remote collaboration," in *Collaboration Technologies and Social Computing*, pp. 22–28, Springer International Publishing (2017).

[14] K. Iso, M. Kobayashi, and T. Yuizono, "A trial of ambient sensor method for worker states classifier," in *Proceedings of International Workshop on Informatics*, pp. 289–294 (2017).

[15] "scikit-learn Machine Learning in Python." `https://scikit-learn.org/`, accessed on May 10 (2019).

(Received June 10, 2019)
(Revised Jan. 14, 2020)

Kazuyuki Iso received a B.E. from Gunma National College of Technology,in 1998, and an M.S. from Japan Advanced Institute of Science and Technology, in 2000. He joined Nippon Telegraph and Telephone (NTT) Corporation, NTT Cyber Space Laboratories in 2000, where he has been engaged in research and development of communication systems using virtual reality technology. He is currently a Senior Research Engineer at NTT Media intelligence Laboratories, and a Doctor-course student at Japan Advanced Institute of Science and Technology. He is a member of IEEE, VRSJ, and IPSJ.



Takaya Yuizono received the B.E., M.E., and Dr. of Engineering from Kagoshima University, in 1994, 1996, 1999, respectively. He was a research associate in Kagoshima University, a lecturer and an assistant professor in Shimane University, respectively. He has been an associate professor in Japan Advanced Institute of Science and Technology since 2006. His research interests include in Groupware, CSCW, Creativity, and Education with interdisciplinary approach. He received best paper award in KES2005, NLP-KE2012, KICSS2016, and best paper award of journal of Japan Creativity Society in 2013, 2015, 2016, 2018. He is a member of ACM, IEEE, IPSJ, and IEICE.

Minoru Kobayashi is a professor at Meiji University, Tokyo Japan. His research interests include human-computer interaction, computer-supported cooperative work, and the design of IoT devices to enrich human-human communication. Kobayashi received a B.E. and an M.S. from Keio University, an M.S. from the Massachusetts Institute of Technology, and a Ph. D. in instrumentation engineering from Keio University. He is a member of ACM, IEEE, IPSJ, IEICE, and VRSJ.