(IJNCAA)

ISSN 2220-9085(ONLINE)
ISSN 2412-3587(PRINT)

INTERNATIONAL JOURNAL OF

NEW COMPUTER ARCHITECTURES AND THEIR APPLICATIONS

Volume 6, Issue 4
2016

SDIWC

www.sdiwc.net

International Journal of
NEW COMPUTER ARCHITECTURES AND THEIR APPLICATIONS

Overview

The SDIWC International Journal of New Computer Architectures and Their Applications (IJNCAA) is a refereed online journal designed to address the following topics: new computer architectures, digital resources, and mobile devices, including cell phones. In our opinion, cell phones in their current state are really computers, and the gap between these devices and the capabilities of the computers will soon disappear. Original unpublished manuscripts are solicited in the areas such as computer architectures, parallel and distributed systems, microprocessors and microsystems, storage management, communications management, reliability, and VLSI.

One of the most important aims of this journal is to increase the usage and impact of knowledge as well as increasing the visibility and ease of use of scientific materials, IJNCAA does NOT CHARGE authors for any publication fee for online publishing of their materials in the journal and does NOT CHARGE readers or their institutions for accessing the published materials.

Publisher

The Society of Digital Information and Wireless Communications
20/F, Tower 5, China Hong Kong City, 33 Canton Road, Tsim Sha Tsui, Kowloon, Hong Kong

Further Information

Website: http://sdiwc.net/ijncaa, Email: ijncaa@sdiwc.net,
Tel.: (202)-657-4603 - Inside USA; 001(202)-657-4603 - Outside USA.

The issue date is December 2016.

IJNCAA

ISSN 2220-9085 (Online)
ISSN 2412-3587 (Print)

Volume 6, Issue No. 4 *2016*

CONTENTS

ORIGINAL ARTICLES

International Journal of New Computer Architectures and their Applications (IJNCAA) 6(4): 114-121
The Society of Digital Information and Wireless Communications, 2016 ISSN 2220-9085 (Online); ISSN 2412-3587 (Print)

Cluster Sampling for the Demand Side Management of Power Big Data

Yongxin Zhang Hong Shen Shitian Xu
School of electrical & electronic engineering, North China Electric Power University
2 Beinong Road, Huilongguan Town, Changping District, Beijing, China
459368011@qq.com

ABSTRACT

In view of the DSM (Demand Side Management) under the big data environment, an improved FCM (Fuzzy C-Mean) clustering with Gauss data preprocessing is proposed, and the daily load curve of the whole study area was obtained with the electricity data. According to the formulation of the TOU (Time of Use) price, which is consistent with the characteristics of local users is given. The electricity suggestions based on the specific user load curve is provided, including the return of the DR (Demand Response). Subsequently, the sampling division is put forward to expand the improved model. Finally, the method is tested by the actual data, and the results show that it has a processing speed 10 times of the direct processing when the data is more than 10000.

KEYWORDS

big data; smart grid; data mining; power demand side management; clustering analysis.

1 INTRODUCTION

With the fast development of information technology, China's power industry has entered the era of big data[1]. At the mean time, power side management has been widely used in many countries, which has achieved some good results[2]. Based on the new electricity reform, the new method of power demand side management under the background of large data is studied. In smart grid, it guides users to participate in power grid interaction, which can not only help enterprises to save energy and reduce the cost of electricity, but also have great significance in many aspects, such as the safe and economical operation, the sustainable development of the country and so on[3]. At present, the application of power big data in China is mainly dependent on the integration of power enterprise group, three operational monitoring (control) center, and the three major power data center.

For power big data processing, the previous study can be divided into three parts: distributed computing, memory computing, stream processing. That is to use the data storage, efficient reading, and real-time processing of the power big data to solve this problem. The scheme and implementation of improving the processing power big data are put forward by the data storage[4]. The method of real time processing of big data in power state detection is given in the literature[5]. However, they are weak in unstructured data processing and data deep mining analysis. What's more, they are difficult to meet the five requirements: large amount of data, high processing speed, large data types, high value and high accuracy of power data[6][7].

We consider electricity suggestion as the research project, which can be applied to the demand side management. And then a solution method under the large data environment is proposed: Firstly, the matrix vector is used to improve the FCM clustering, in order to make it conform to the processing of the data and get the whole power load curve of the study area after data preprocessing. Then according to the rules of TOU electricity price and the whole load curve, we use the method of "midpoint selection", and give TOU electricity price which conforms to the local electricity habits. With regard to the specific user's daily load curve in this region, based on the analysis of the above TOU electricity price, we made suggestions on the way of electricity, and the calculation of the demand side response returns. Secondly, we propose the idea of sampling division to improve the operational efficiency of large-scale data processing. Finally, on the ba-

sis of the actual data obtained by the investigation, the performance of the two methods (the sampling method and the direct clustering method) are compared by the example test, and the results show that the feasibility of sampling partition clustering is proved in large data environment.

2 FCM CURVE CLUSTERING

2.1 Data Preprocessing

The collection of user data is inevitably affected by various noise sources. The inherent volatility of electricity index data will affect the follow-up data compression, repetition and fault diagnosis. Therefore, before processing the data, we use the method called Gauss smoothing filter pretreatment[3] to prepare for the subsequent big data mining.

Data preprocessing steps are as follows:

We consider the original data $\mathbf{X} = [\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_i, \ldots, \mathbf{x}_n]^T \in \mathbf{R}^{n\times p}$ as n user's power consumption data matrix, every user has p data points. After smoothing the data is sorted into $\mathbf{X}_{smooth} = [\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_i, \ldots, \mathbf{x}_n]^T \in \mathbf{R}^{n\times p}$, and $\mathbf{x}_i = [x_{i,1}, x_{i,2}, \ldots, x_{i,m} \in \mathbf{R}^{1\times p}]$. Variance σ and window width w are the adjustable parameter.

Take Gauss kernel function, the expression is:

$$gauss(d, \sigma) = e^{\frac{-d^2}{2\sigma}} \tag{1}$$

Defined window width $w \in Z$, then $\mathbf{D} = [d_1, d_2, \ldots, d_w]_{1\times w}$. Among them,

$$d_i = \frac{w-1}{2} + i - 1 \tag{2}$$

If

$$k_i = gauss(d_i, \sigma) / \sum_{j=1}^{w} gauss(d_j, \sigma) \tag{3}$$

Then coefficient matrix is $\mathbf{K} = [k_1, k_2, \ldots, k_w]_{1\times w}$.

On account of similarity in the user's electricity data, which shows a strong periodicity in a longer period of time, it is necessary to carry out the complement operations with original data $\mathbf{X}$ to form a new data matrix $\mathbf{X}'$.

Complementary processing is as follows:

Let 1 to $\lfloor w/2 \rfloor$ column and $m - \lfloor w/2 \rfloor + 1$ to m column of $\mathbf{X}$ fill to the head and end of the $\mathbf{X}$, and we can get the $\mathbf{X}'$. The row vector $\mathbf{x}'_i = [x'_{i,1}, x'_{i,2}, \ldots, x'_{i,m-\lfloor w/2 \rfloor+1}]$.

Do the convolution between the x'_i and $\mathbf{K}$:

$$w'_{i,\beta} = \mathbf{x}'_i(\beta) * \mathbf{K}(\beta) = \sum_{j=1}^{\beta} x'_{i,j} k_{\beta-j+1} \tag{4}$$

and $\beta \in [1, m+2\times\lfloor w/2 \rfloor + w - 1], j \in [1, m]$.

We can get matrix $\mathbf{W} = [\mathbf{w}'_1, \mathbf{w}'_2, \ldots, \mathbf{w}'_i, \ldots, \mathbf{w}'_n]^T \in \mathbf{R}^{n\times(m+2\times\lfloor w/2 \rfloor+w-1)}$. We Intercept the $n \times p$ matrix in the $\mathbf{W}$ as the $\mathbf{X}_{smooth}$. The i row and j column of $\mathbf{X}_{smooth}$ is:

$$x_{i,j} = w'_{i,j+\lfloor w/2 \rfloor} \tag{5}$$

We select the power consumption data which has two categories (town,township) from a county area, and the use matlab in drawing the electric power load curves from 200 users, Yuanling County, Hunan Province, China as the figure 1 (the X-axis is time, Y-axis is power).

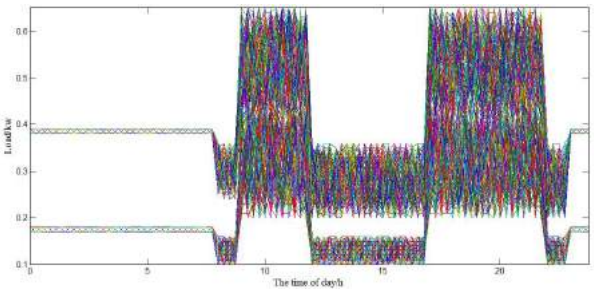


Figure 1. Before preprocess the data

Using Matlab to achieve the above algorithm programming, that is, after the Gauss smoothing filter data preprocessing, the results are shown in figure 2.

It can be seen that the preprocessing algorithm can better identify the inconsistency between the anomalies and the normal data, and the remove outliers. What's more, the measurement data can be very beneficial to retain the data characteristics of the original data.

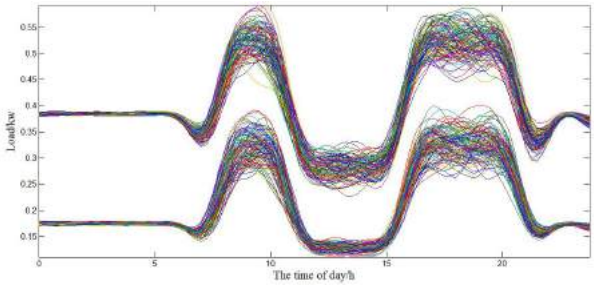


Figure 2. Electric load curve of all users

2.2 Construction of Model

After the data preprocessing, the data can accurately reflect the actual situation, so that it is more convenient for the implementation of clustering.

In order to obtain the user's electricity characteristics and tap the electricity value information further, we take the algorithm called FCM clustering into consideration. FCM clustering is a classical fuzzy clustering algorithm[8], which can be used to estimate the type of data points. Actual user data is often power information on each time node in a period of time, and we need the overall consideration of each time node in this time period, that is to use the user's power curve as the sample unit, clustering a certain number of user data sample unit. Therefore, this paper proposes a clustering model of FCM curve, which is suitable for the estimation of power consumption, and its description is as follows:

$$J(\mathbf{X};\mathbf{M},\mathbf{Z}) = \sum_{i=1}^{c}\sum_{j=1}^{n}(\mu_{ij})^m||\mathbf{x}_i - \mathbf{z}_i||^2 \quad (6)$$

Where, n is the sample number of clusters, c is the number of categories. μ is a kind of fuzzy membership degree of a certain type of samples. $\mathbf{x}_i$ represents the power data of the No.j sample, containing p data. $\mathbf{z}_i$ is a kind of cluster center, $\mathbf{Z} = [\mathbf{z}_1, \mathbf{z}_2, \ldots, \mathbf{z}_c], \mathbf{z}_i \in \mathbf{R}^n$, m is fuzzy index.

Define the distance between the user's electricity data and the cluster center:

$$D_{ij}^2 = ||\mathbf{x}_j - \mathbf{z}_i||^2 \quad (7)$$

The updating equations of membership degree μ and cluster center $\mathbf{Z}$ are respectively:

$$\mu_{ij} = \frac{1}{\sum_{r=1}^{c}(D_{ij}/D_{rj})^{2/(m-1)}}, \quad 1 \le i \le c, 1 \le j \le n \quad (8)$$

$$\mathbf{z}_i = \frac{\sum_{j=1}^{n}\mu_{ij}^m \mathbf{x}_j}{\sum_{j=1}^{n}\mu_{ij}^m}, 1 \le i \le c \quad (9)$$

The specific algorithm steps of fuzzy mean curve clustering:

Step 1: Set the target function precision σ, fuzzy index m (usually take 2), maximum number of iterations T_m;

Step 2: Set membership degree μ;

Step 3: By the (9) we initialize the fuzzy clustering center $\mathbf{Z}$;

Step 4: If $|J(t+1) - J(t) < \sigma|$,or the number of iterations satisfies $t > T_m$, we end the cluster. Otherwise, assign the value of the $t+1$ to the t and go to **step 5**;

Step 5: According the (8) and (9), we update fuzzy membership degree μ and fuzzy clustering center $\mathbf{Z}$, then return **step 4**.

Finally, the clustering center $\mathbf{Z}$ is the result of the clustering, which can clearly show what types the samples can be divided into.

3 TIME ELECTRICITY PRICE

The clustering intensity is enhanced until all the sample curves are clustered into one class, that is, this unique clustering curve represents the overall electric habit of the sample, and we divides the time interval of the total sample area by this unique clustering curve. According to the load level of the unique cluster curve, the time interval is determined by using the method of the "mid-point selection". The point on the time axis corresponding to the middle point of the peak and valley or between the peak and the flat is the point of the time interval. The $24h$ in this area can be divided into 3 main power use periods: general (flat) period, peak period, and valley period.

We set the price of flat period as p_f, on the basis of it, the peak and valley period under the floating price are respectively:

$$p_p = p_f(1+\mu) \quad (10)$$

$$p_v = p_f(1-\omega) \tag{11}$$

Type μ and ω are floating and floating ratio. Considering the benefits of the power supply side and the user, we define the pull apart as [9]:

$$\Delta = \mu/\omega, W_3/W_1 \leq \Delta \leq 1 \tag{12}$$

W_3 and W_1 express respectively the total electricity consumption in peak and valley period. According to the implementation method of sub-time electricity price and time division above, we get the sub time electricity price which is in line with the local electricity habit.

4 POWER RECOMMENDATION AND RETURN CALCULATION

Power recommendation: Through the ideal "eliminate peak and fill in the valley", the load curve of power system in the region will be a parallel to the time axis line which is the curve most wants to see. For a specific user, the actual load curve and the daily average load curve are used. According to the comparison, it is proposed that the use of electricity should be less in the peak period and be more in the valley period, so that the load curve is similar to that of the daily average load curve.

$24h$ in a region according to the above method is divided into n hours. The electricity price in the $No.i$ period is recorded as p_i. If the electricity consumption of a user in the $No.i$ period in this area is w_i, the user's electricity tariff will be:

$$P = \sum_{i=1}^{n} p_i w_i \tag{13}$$

If the user fully accept the proposal, that is, the total electricity consumption of the user will not change, when the peak power shifts to the valley, the load curve will be idealized into the daily average load curve. Note the length of the $No.i$ period for t_i, after accepting the proposal, the power consumption of the $No.i$ period is:

$$W_i' = (\sum_{i=1}^{n} w_i/24) \times t_i \tag{14}$$

at this time the electricity becomes:

$$P' = \sum_{i=1}^{n} p_i w_i' \tag{15}$$

The savings amount is $\Delta P = P - P'$.

5 REALIZATION IN LARGE DATA ENVIRONMENT: SAMPLING DIVISION

Classical algorithms have different degrees of restrictions on the scale of the data, and the fuzzy curve clustering algorithm introduced in the previous paper is no exception[10][11]. In order to give the realization scheme of FCM curve clustering under the big data environment, the paper proposes the idea of sampling division based on literature[12]: using the correlation between each sampling, not only the independent treatment of each sample can maintain a smaller size, but also the results can reflect the overall processing.

5.1 Sampling

For the small data set extracted, hoping that it can have all of the natural clusters which a large data is contained in, that is, it contains all user types. We give the sampling formula[13]:

$$S = f \times n + c \times \log(\frac{1}{\delta}) + c\times \sqrt{[\log(\frac{1}{\delta})]^2 + 2 \times f \times \frac{n}{c} \times \log(\frac{1}{\delta})} \tag{16}$$

Among them, S is the total number of samples to be taken at least, f is the proportion of extracting to the specified data. n is the data size. c is the class of the number of natural clusters that all the data contains.

Set the total sample number $M = \lceil (S \times c)/n \rceil$, then the sample size of each sample is $k = \lceil S/M \rceil$. The size of the data is often very large, that is, there are $k \times M << n$. The Sample is sampled by this method.

5.2 Initial Position of the Natural Cluster Centroid

We set all sampling to have an ideal coverage of all natural clusters which cover all categories. Natural clusters that are included in

the sample have the approximate centroids like the original data set. We set two steps to determine the initial position of the natural cluster centroid.

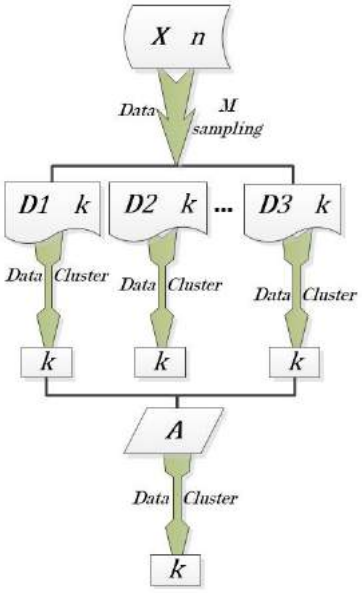


Figure 3. Determine the initial location of the natural cluster centroid

First step: Due to the small size of the sample, we can use the classic algorithm in clustering each sample, so the process of clustering FCM curve will be very fast. It can be implemented in parallel processing which can reduce a lot of running time, for the reason that sampling has the same scale, and the clustering process is independent. Set one sample of the total sample large data set (category number is c) covering c category. Because of the existence of this inequality: $1 \leq c' < c$, the sampling will be clustered into c clusters, which will split a cluster, and that is a more detailed classification situation, which will be revised in the following text.

Second step: By the first step we get $c \times M$ small clusters, and calculate the average value of each small cluster:

$$\alpha_i = \frac{1}{n} \sum_{j=1}^{n_i} \alpha_{ij} \tag{17}$$

n_i is the data size of the cluster $\mathbf{C}_i$, and α_{ij} is the properties of a sample in $\mathbf{C}_i$. Using the mean to represent the cluster, we can get the data set $\mathbf{A}$ whose size is $c \times M$. By clustering $\mathbf{A}$ with improved FCM algorithm, we can get c clusters, then it will make $c \times M$ small clusters into c large clusters. By calculating the mean value of the c clusters, the initial position of the natural cluster centroid can be determined.

5.3 Means Update

Because the data is local information, the initial position of the cluster centroid often deviates from the cluster centroid of the original data set. Therefore, it is necessary to update the initial position by updating means. According to the distance of the initial centroid, the remaining data samples which are not used to be allocated to the nearest cluster:

$$\lambda = \arg\max_i(-|\alpha_\Delta - \alpha_i|^2) \tag{18}$$

α_Δ are the sample properties whose class is undetermined, λ are the already identified categories.

When a data sample is filled with a cluster $\mathbf{C}_i$ according to the above classification principle, the mean value of the cluster is updated:

$$\alpha_i \Leftarrow \frac{\alpha_i \times n_i + \alpha_\Delta}{n_i + 1} \tag{19}$$

$$n_i \Leftarrow n_i + 1 \tag{20}$$

With the addition of the remaining data, the mean position is constantly updated, and moved to the real center of the natural cluster until the update has completed.

At last, the data set is divided by the above mentioned natural cluster centroid, and the principle of dividing the data is still in the form of the formula (18). That is, according to the distance of the cluster centroid, we can determine the final classification.

6 EXAMPLE ANALYSIS

6.1 Demand Side Power Recommendation

6.1.1 Data Sources and Processing

It can be known from Figure 2, although the individual user's electricity characteristics are not the same, but the county has a total of two

electricity peak (in the morning, afternoon), a period of electricity valley(noon) and a period of electricity flat (night).
Therefore, we set the clustering intensity , and the improved FCM curve clustering is used to cluster the 200 user load curves into a class as shown in Figure 4. The only cluster curve represents the overall power habits and levels of the Yuanling County.

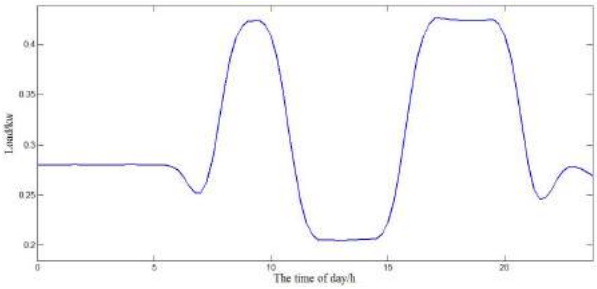


Figure 4. After FCM curve clustering

6.1.2 Tiered Pricing for Electricity

Based on the load level (instantaneous power) of the only cluster curve in Figure 4, and we combine the reference[14][15][16] with the "midpoint method" given in the previous paper to determines the time period. The result is shown in Figure 5. The area $24h$ is divided into 3 parts (5 power period): flat period $(00:00 \sim 8:45)$ & $(21:45 \sim 23:45)$, peak period $(8:45 \sim 11:45)$ & $(16:45 \sim 21:45)$, valley period $(11:45 \sim 16:45)$.

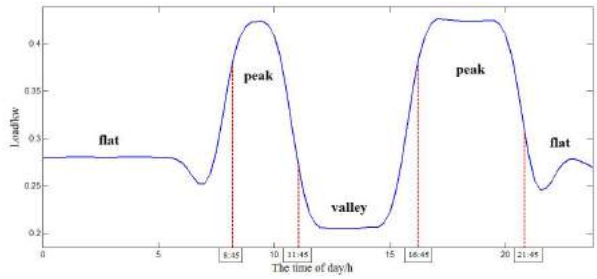


Figure 5. Determine time hours

Combined with *the implementation measures of Hunan power grid peak valley TOU price* and the time division above, the proposed TOU price in line with the local electricity use is shown in table 1.

Table 1. Time sharing electricity price

Period Type	**Range**	
Peak period	$8:45 \sim 11:45 \& 16:45 \sim 21:45$	
Flat period	$00:00 \sim 8:45 \& 21:45 \sim 23:45$	
Valley period	$11:45 \sim 16:45$	
Period Type	**Measures**	**TOU Price**
Peak period	Floating 80%	0.810
Flat period	–	0.450
Valley period	Down 55%	0.202

6.1.3 User Power Recommendation

Calculate and draw the load curve and the daily average load curve of one user, respectively, as shown in Figure 6 in the form of broken line and horizontal line.

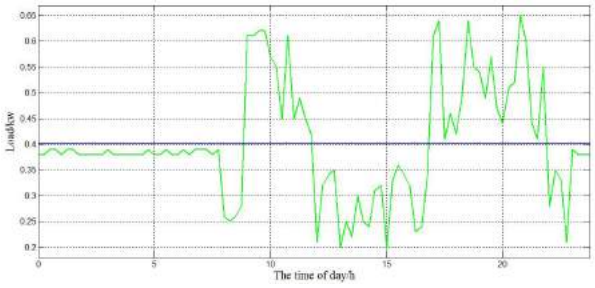


Figure 6. Schematic diagram of power suggestion

According to figure 6, the user's electricity is recommended: Reducing the power consumption according to the daily average load curve in the period when the actual load curve is higher than the daily average load curve. Similarly, the user ought to increase power consumption at a time below the daily average load curve, thus participating in the demand side management. In accordance with the previous return calculation, we can be accurately obtained if the user followed the electricity recommendation, the ideal energy savings is 1.04 yuan per day.

The proposed power recommendation and the amount of return, is a demand side management signal for individual users, and can be more reasonable to strengthen the demand side management[17][18][19].

6.2 Test under the Environment of Large Data

In order to test the performance of sampling partition clustering, the number of users to be processed is increased in turn. The average running time which processing the same user data need are respectively tested by the sampling partition clustering and the direct clustering. We define the processing speed as $v = q/\tau$. q is processed by the number of users. τ is the average running time. Therefore, the lifting speed ratio is $\alpha = v_2/v_1$.
The test and calculation results are shown in Table 2 and figure 7.

Table 2. Example contrast test

A[1]	B[2]	C[3]	D[4]	E[5]	F[6]
1000	7.34	1.44	5.11	136.19	695.78
2500	36.23	3.36	10.78	69.00	744.00
5000	74.20	7.50	9.89	67.39	666.43
7500	109.96	11.08	9.92	68.20	676.91
10000	122.35	13.98	8.75	81.73	715.41
13000	166.83	18.57	8.98	77.93	700.09
16000	233.37	23.19	10.06	68.56	690.03
20000	296.54	27.84	10.65	67.44	718.28
30000	428.04	42.94	9.97	70.09	698.69

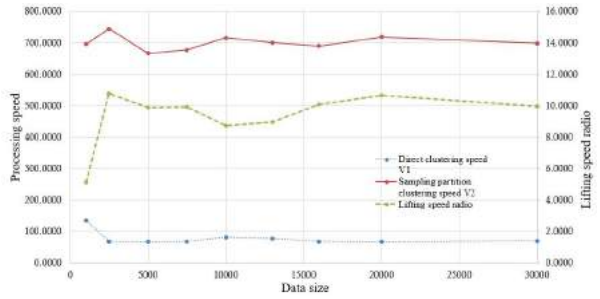


Figure 7. Relationship between direct clustering speed, clustering speed, lifting speed ratio and data size

From Figure 2 and table 7, we can see that the processing speed of sampling partition clustering is always higher than that of direct clustering in the range of the data used in the experiment. That is to say, the drawback of direct clustering: low speed affects the rapidity of the power system analysis.
In addition, when the data quantity is higher than 10000, the lifting speed ratio between the sampling partition clustering and the direct clustering is stable at about 10. In terms of large-scale data, the sampling partition clustering can still maintain a good processing speed, which proved its good processing performance.

7 CONCLUSIONS

Based on FCM clustering and sampling algorithm, the electricity recommendation model is proposed for the demand side. Through the actual data matlab simulation test, it is proved that in the electric power big data environment the sampling partition clustering has a faster processing speed. With the increase of the amount of data, it has obvious advantages compared with the direct clustering. In addition, the power recommendation (TOU price, electrical mode, demand side response, etc.). what the model give is clear and accurate to meet the user's electricity habits and characteristics.
For the future work ,we will further improve the efficiency of the algorithm. Try to build a real time data processing platform, which is used in the field of power system planning, load forecasting, customer type classification and so on.
The technology can be applied to the power demand side management in the background of intelligent power grid with big data. It can not only mobilize the enthusiasm of the power demand side response, ease the power shortage pressure, and reduce the supply and consume power cost, but also improve the system's overall energy utilization efficiency. So it can achieve the long-term goal of environmental protection and sustainable development.

[1] **A** means "Quantity"
[2] **B** means "Average time of direct clustering", **time measurement: s**
[3] **C** means "Average time of sampling partition clustering"
[4] **D** means "Lifting speed ratio"
[5] **E** means "Speed of direct clustering"
[6] **F** means "Speed of sampling partition clustering"

REFERENCES

[1] Zhang Pei, "Current Situation and Prospect of Power Big Data Applica-

tion," Electrical age, 2014(12):14.

[2] Zeng Ming, "Power Demand Side Management," Beijing: China Electric Power Press, 2001:1-8.

[3] Nai Jiawen, Ms D Thesis, "Research and Development of Customer Power-expenditure Information Intelligent Analysis System," Guangdong University of Technology, 2014.

[4] Mao Yufeng, "Design and implementation of massive power data analysis system based on Cloud Computing," Beijing Jiaotong University, 2015.

[5] Zhou Guoliang, et al, "Application of real time big data processing technology in the field of state monitoring," Chinese Journal of electrical engineering, 2014, 29(S1): 432-437.

[6] Tang Ruiwei, "Application Status and Development Prospect of Electric Power Big Data," China Electric Power Enterprise Association Science and Technology Development Service Center: 2014:7.

[7] Yan Longchuan, Li Yaxi, Li Binchen, "Opportunities and Challenges Faced by Power Big Data," Electric power informatization: 2013,11(4):1-4.

[8] Balazs B, Janos A, Balazs F, "Fuzzy clustering and data analysis toolbox [EB]," 2006[2015-10-01].

[9] Liu Xiaocong, Wang Beibei, Li Yang, "Day-ahead Generation Scheduling Model Considering Demand Side Interaction under Smart Grid Paradigm," Chinese Journal of Electrical Engineering: 2013,01:30-38.

[10] Li Yin. Ms D Thesis, "Research and Application of Clustering Algorithm," Jiangnan University, 2013.

[11] Li Ziliu, Ms D Thesis, "Research on Real-time Data Streams Clustering Framework," Zhongshan University, 2013.

[12] Lu Zhimao, Feng Jinmei, Fang Dongmei, "Novel Partition Clustering Algorithm for Large Data Processing," Systems Engineering and Electronics, 2014,36(5):1010-1015.

[13] Sudipto G, Rajeev R, Kyuseok S, "Cure: an efficient clustering algorithm for large databases," Information Systems, 2001,26(1):33-58.

[14] Zhu Keding, Song Yihang, Tan Fuzhong, "Design Optimization Model for Tiered Pricing of Household Electricity Consumption," East China Electric Power, 2011,06:862-867.

[15] Li Yuan, Luo Qin, Song Yiqun, "Study on tiered level determination of TOU & tiered pricing for residential electricity based on demand response," Power System Protection and Control, 2012,18:65-68+74.

[16] Song Yihang, Ms D Thesis, "Optimization model for the design of tiered pricing for household electricity in china," North China Electric Power University (Beijing), 2011.

[17] Cai Pei, Weng Huiying, Yu Bin, "Research on Smart Demand Side Management System in Low-Carbon Economy," Power System Technology, 2012,10:11-16.

[18] Wang Beibei, Li Yang, "Demand side management planning and implementation mechanism for smart grid," Electric Power Automation Equipment, 2010,12:19-24.

[19] Wang Mingjun, "Load Management and Demand Side Management in Electricity Market Environment," Power System Technology, 2005,05:1-5.

International Journal of New Computer Architectures and their Applications (IJNCAA) 6(4): 122-131
The Society of Digital Information and Wireless Communications, 2016 ISSN 2220-9085 (Online); ISSN 2412-3587 (Print)

Adaptability to Periodic Variable Disturbance using Probabilistic State-Action Pair Prediction

Masashi Sugimoto[1*] Naoya Iwamoto[1] Robert W. Johnston[1]
Keizo Kanazawa[1] Yukinori Misaki[2] Kentarou Kurashige[3]
[1]National Institute of Technology, Kagawa College
[2]Toyohashi University of Technology
[3]Muroran Institute of Technology
Email(*: corresponding author): sugimoto-m@es.kagawa-nct.ac.jp

ABSTRACT

When operating a robot in a real environment, its behavior is probabilistic because of slight transition of the robot's state or error in the action taken at a given time. In this case, it is difficult to operate the robot using rule-based-like action decision methods. Therefore, ad-hoc-like action decision methods are needed. A method is proposed for deciding on future actions based on a robot's present information. The state-action pair prediction method has been reported; it links the state and future actions of a robot using internal information. A statistical approach to state-action pair prediction has been introduced previously, in which the existence probability of a state and action in the future is calculated according to the normal distribution. This paper considers the situation where a command input is sent to an inverted pendulum. Based on this command input, the shape of the floor is changed from flat to undulating. The results of verification experiments confirm that the proposed method can adjust the shape of the floor autonomously.

KEYWORDS

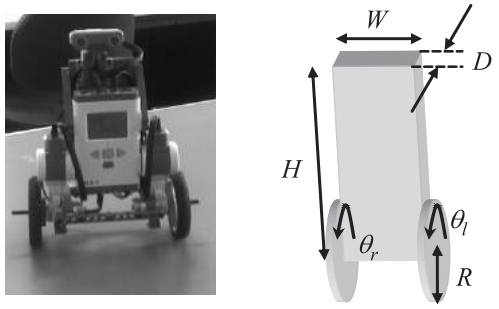
State-Action Pair Prediction, Decision for Optimal Action, Prediction Distribution, Online SVR, Normal Distribution, Online State Prediction

1 INTRODUCTION

In controlling robots in dynamic environments, an action may be chosen and adopted based on current state and action results by predicting a future state from previous actions and states [1]. In operating a robot in a real environment, its behavior changes probabilistically due to slight fluctuations in the robot's state or errors in the actions taken at a given time. In this case, a stochastic technique is necessary for handling problems with unknown disturbances [2, 3, 4, 5, 6, 7].

From this standpoint, a stochastic technique that features a state and action pair prediction method is effective and necessary. In a previous study [8], the statistical approach was introduced to state-action pair prediction. In particular, a method that decides which future action to take was proposed given the current action. In the proposed method, the existence probability of a state and action in the future is calculated according to the normal distribution. In a previous study [8], an inverted pendulum was kept balanced while given a predictable unknown disturbance using the proposed method. This study considers the situation where a command input is sent to an inverted pendulum; from this command input, the shape of the floor is changed from flat to undulating.

This paper is organized as follows: in section 2, motivation is provided for using the state-action pair prediction technique for deciding the optimal action for a robot to apply, introducing a stochastic technique. Furthermore, details of stochastic state-action pair prediction are stated. In section 3, a verification experiment configuration using variable floor shape is described. Finally, section 4 presents the conclusions of this study.



(a) Structure (b) 3D Model

Figure 1. The Structure of "NXTway-GS."

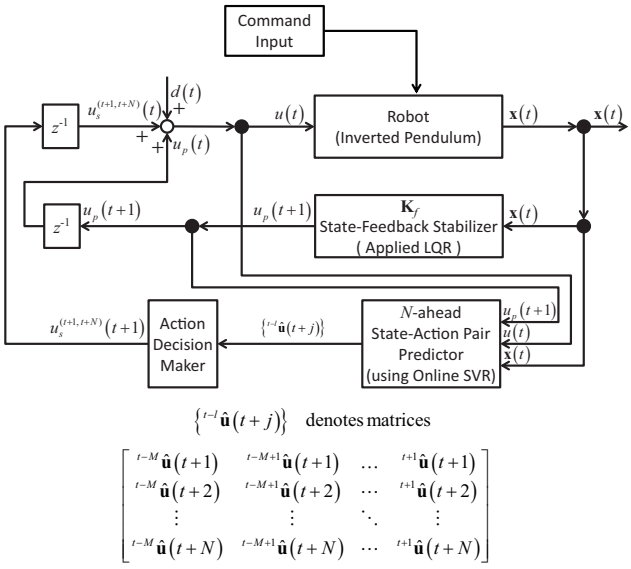


Figure 2. Outline of deciding the optimal action for a robot using the prediction of a stochastic state-action pair.

2 AN APPROACH FOR INTRODUCING THE NORMAL DISTRIBUTION FUNCTION INTO STATE-ACTION PAIR PREDICTION

In this study, we consider determining the optimal action to minimize the body pitch angle of an inverted pendulum (shown in Fig. 1) under a predictable unknown disturbance given continuously using the state-action pair prediction proposed by our former studies [9, 10]. Therefore, in this paper, the system in Fig. 2 based on the former studies is considered. As shown in Fig. 2, ${}^{t-l}\hat{\mathbf{u}}(t+j)$ describes the prediction result of the control input $\mathbf{u}(t+j)$ when this input is predicted at time $(t-l)$. Therefore,

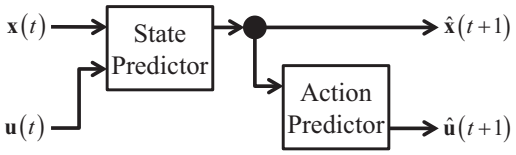
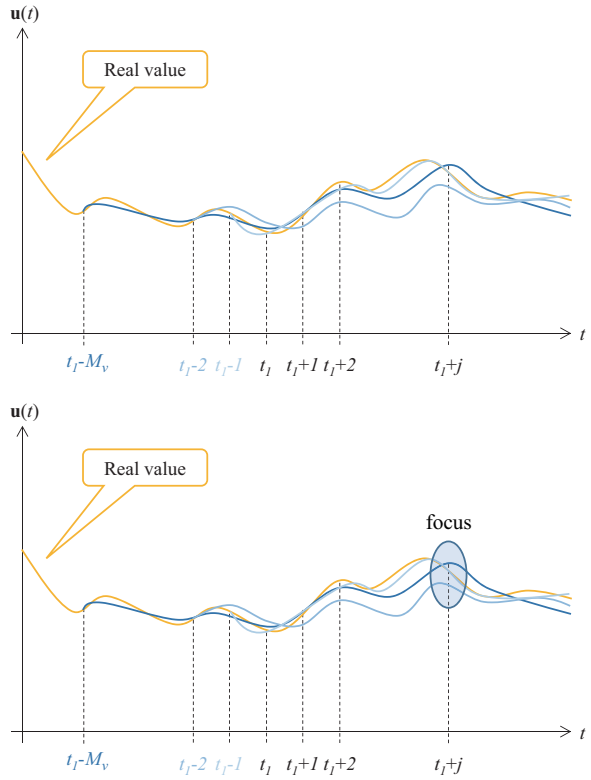


Figure 3. Structure of the N-ahead state-action pair predictor [1].

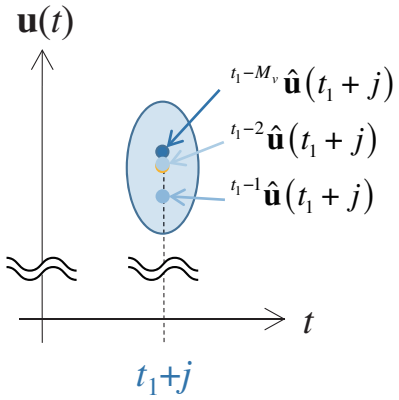
this proposed method attempts to compensate for a current action in progress using a combination of optimal control and state-action pair prediction. A structure of prediction of a state-action pair called an "N-ahead state-action pair predictor" is the internal structure described in Fig. 3. In this case, if N is larger than the current time t, then the prediction error rate is proportional to N; moreover, the cumulative prediction error cannot be ignored in a prediction depending on time t. In case of stochastic probability for ordinary robot control, probabilistic curves have certain peaks. However, in this study, the maximum peak of probability for a robot action is the focus. From this viewpoint, the normal distribution is applied in this study, and the existence probability of a state and action in the future, according to the normal distribution, is calculated. Moreover, future actions are revised based on the state and action that have the highest existence probability. In addition, this study considers the increasing influence of an action at the latest time t, focused on the time at which prediction starts [11].

On the basis of this idea, the distribution of existence probability of a prediction series is calculated based on the time at which a prediction is started. In addition, the focus is on the average and standard deviation. The predictor makes predictions continuously at each point in time. Therefore, from this mechanism, the values of prediction results are obtained at each sampling time. Future values at each sampling time are predicted and obtained by a stochastic state-action pair prediction. These results are illustrated in Fig. 4(a). Note the "focus" area in fig. 4(a).

Here, the prediction results at time (t_1+j) can be obtained from each previous time at (t_1-i) (the range of i is given by $(0 \ll M_v \leq i \leq N)$)



(a) Series of state-action pair predictions



(b) Focus on variations of predicted values at past times (from Fig. 4(a)).

Figure 4. Variations of predicted series using a state-action pair.

as illustrated in Fig. 4(b). In this zone at $(t_1 + j)$, the prediction results are distributed. To quantitatively measure these variations, the average and the standard deviation are emphasized. In brief, the average $\mu_j^{\mathbf{u}}$ and the standard deviation $\sigma_j^{\mathbf{u}}$ of the control input at time (t_1+j) using the values predicted at each $(t_1 - i)$ (as past times) can be derived as follows:

$$\mu_j^{\mathbf{u}} = \mu\left(\frac{w_M}{w_\Sigma}{}^{t_1-M}\hat{\mathbf{u}}(t_1+j), \frac{w_{M+1}}{w_\Sigma}{}^{t_1-M+1}\hat{\mathbf{u}}(t_1+j), \ldots, \frac{w_1}{w_\Sigma}{}^{t_1+1}\hat{\mathbf{u}}(t_1+j)\right) \tag{1}$$

$$\sigma_j^{\mathbf{u}} = \sigma\left(\frac{w_M}{w_\Sigma}{}^{t_1-M}\hat{\mathbf{u}}(t_1+j), \frac{w_{M+1}}{w_\Sigma}{}^{t_1-M+1}\hat{\mathbf{u}}(t_1+j), \ldots, \frac{w_1}{w_\Sigma}{}^{t_1+1}\hat{\mathbf{u}}(t_1+j)\right) \tag{2}$$

Similarly, for the state $\mathbf{x}(t_1 + j)$, the average $\mu_j^{x_k}$ and standard deviation $\sigma_j^{x_k}$ are derived. Here, $k \in \dim \mathbf{x}$.

$$\mu_j^{x_k} = \mu\left(\frac{w_M}{w_\Sigma}{}^{t_1-M}\hat{x}_k(t_1+j), \frac{w_{M+1}}{w_\Sigma}{}^{t_1-M+1}\hat{x}_k(t_1+j), \ldots, \frac{w_1}{w_\Sigma}{}^{t_1+1}\hat{x}_k(t_1+j)\right) \tag{3}$$

$$\sigma_j^{x_k} = \sigma\left(\frac{w_M}{w_\Sigma}{}^{t_1-M}\hat{x}_k(t_1+j), \frac{w_{M+1}}{w_\Sigma}{}^{t_1-M+1}\hat{x}_k(t_1+j), \ldots, \frac{w_1}{w_\Sigma}{}^{t_1+1}\hat{x}_k(t_1+j)\right) \tag{4}$$

Hence, $\mu(\cdot)$ denotes the average of $(\cdot)$ and $\sigma(\cdot)$ denotes the standard deviation of $(\cdot)$. Moreover, $w_M, w_{M+1}, \ldots, w_1$ denote weights of the weighted moving average, and w_Σ denotes the sum of these weights. From the above expression, the normal distribution function $N_j^{x_k}$ at time $t_1 + j$ and the function $N_j^{\mathbf{u}}$ for the control input $\mathbf{u}$ are derived, as shown in the

equations below.

$$N_j^{x_k}(X_k) = \frac{1}{\sqrt{2\pi\left(\sigma_j^{x_k}\right)^2}} \exp\left\{-\frac{\left(X_k - \mu_j^{x_k}\right)^2}{2\left(\sigma_j^{x_k}\right)^2}\right\} \tag{5}$$

$$N_j^{\mathbf{u}}(U) = \frac{1}{\sqrt{2\pi\left(\sigma_j^{\mathbf{u}}\right)^2}} \exp\left\{-\frac{\left(U - \mu_j^{\mathbf{u}}\right)^2}{2\left(\sigma_j^{\mathbf{u}}\right)^2}\right\} \tag{6}$$

Here, X_k, U are random variables:

$$X_k = {}^{t-M}\hat{x}_k\left(t_1+j\right), \ldots {}^{t-2}\hat{x}_k\left(t_1+j\right), {}^{t-1}\hat{x}_k\left(t_1+j\right) \tag{7}$$

$$U = {}^{t-M}\hat{\mathbf{u}}\left(t_1+j\right), \ldots {}^{t-2}\hat{\mathbf{u}}\left(t_1+j\right), {}^{t-1}\hat{\mathbf{u}}\left(t_1+j\right) \tag{8}$$

From this normal distribution function, values larger than the threshold ϑ are treated as follows:

$$\Pr\left(N_j^{x_k} > \vartheta\right) = N_{j,l}^{x_k} \tag{9}$$

$$\Pr\left(N_j^{\mathbf{u}} > \vartheta\right) = N_{j,l}^{\mathbf{u}} \tag{10}$$

In these equations, l denotes the number of states or actions larger than the threshold ϑ. Therefore, the compensation action regarded in the future can be obtained if an action corresponds to the disturbance input before a future action is created based on probability. In other words, the compensation action for the normal distribution function $N_{j+1,l}^{\mathbf{u}}$ is as follows:

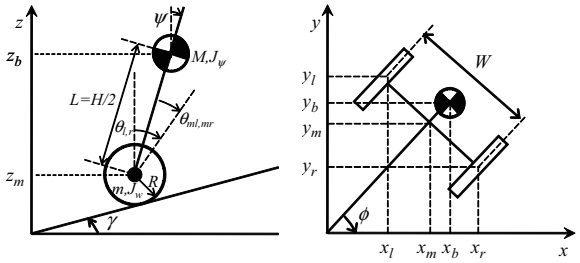
$$N_{j+1,l}^{\mathbf{u}} = -\mathbf{k}_f \cdot N_{j,l}^{\mathbf{x}} \tag{11}$$

In the above equation, $x_k \in \mathbf{x}, 1 \leq l \leq k$. From this equation, a future action is compensated continuously using the largest existence probability as the compensation action at each given time.

3 VERIFICATION EXPERIMENT - COMPUTATIONAL SIMULATION USING THE PROPOSED METHOD

3.1 Experimental Outline

In this verification experiment, the posture of a two-wheeled inverted pendulum "NXTway-GS" is considered as an application. A computer simulation indicated its stabilization in a verification experiment. Here, we obtained training sets for postural control. The training sets contain states and actions of NXTway-GS. In this experiment, stabilizing the inverted pendulum is considered between using stochastic state-action prediction and an ordinary linear quadratic regulator (LQR) controller. The response of the proposed method using stochastic state-action prediction was compared with the control response of the conventional method using LQR. The experiment included 270 steps (the actual stochastic predictable range was 3.00 [s] to 14.00 [s]).



(a) Side model view (b) Plane model view

Figure 5. Model side and plane view for NXTway-GS [12]

3.2 Configuration of Simulation 1 for the NXTway-GS model.

As shown in Fig. 5, "NXTway-GS" can be described as a two-wheeled inverted pendulum model. The coordinate system used in section 3.3 is described in Fig. 5. Moreover, in Fig. 5, ψ denotes the body pitch angle and $\theta_{l,r}$ denotes the wheel angle (l and r indicate *left* and *right*, respectively. Furthermore, $\theta = 1/2 \cdot (\theta_l + \theta_r)$, and $\theta_{ml,mr}$ denotes the DC motor angle (l and r indicate *left* and *right*, respectively). The NXTway-GS's physical parameters are listed in Table 1.

3.3 Configuration of Simulation 2 – for NXTway-GS modeling.

NXTway-GS's motion equations can be derived as shown in Fig. 5. If the direction of the model is the positive x-axis direction at

$t = 0$, the equations of motion for each coordinate can be given as follows [12]:

$$\left[(2m+M)R^2 + 2J_w + 2n^2 J_m\right]\ddot{\theta} + \left(MLR - 2n^2 J_m\right)\ddot{\psi} - Rg(M+2m)\sin\gamma = F_\theta \quad (12)$$

$$\left(MLR - 2n^2 J_m\right)\ddot{\theta} + \left(ML^2 + J_\psi + 2n^2 J_m\right)\ddot{\psi} - MgL\psi = F_\psi \quad (13)$$

$$\left[\frac{1}{2}mW^2 + J_\phi + \frac{W^2}{2R^2}\left(J_w + n^2 J_m\right)\right]\ddot{\phi} = F_\phi \quad (14)$$

Here, $\mathbf{x}_1$ and $\mathbf{x}_2$ represent state variables. In addition, $\mathbf{u}$ denotes input:

$$\mathbf{x}_1 = \begin{bmatrix}\theta & \psi & \dot{\theta} & \dot{\psi}\end{bmatrix}^\top \quad (15)$$

$$\mathbf{x}_2 = \begin{bmatrix}\phi & \dot{\phi}\end{bmatrix}^\top \quad (16)$$

$$\mathbf{u} = \begin{bmatrix}v_l & v_r\end{bmatrix}^\top \quad (17)$$

From the above equations, the state equations of NXTway-GS can be derived using eqs. (12), (13), and (14).

$$\frac{d}{dt}\mathbf{x}_1 = \mathbf{A}_1\mathbf{x}_1 + \mathbf{B}_1\mathbf{u} + \mathbf{S} \quad (18)$$

$$\frac{d}{dt}\mathbf{x}_2 = \mathbf{A}_2\mathbf{x}_2 + \mathbf{B}_2\mathbf{u} \quad (19)$$

In this verification experiment, only the state variable $\mathbf{x}_1$ is used because $\mathbf{x}_1$ contains the body pitch angles as variables ψ and $\dot{\psi}$, which are important for self-balancing. This why planar motion ($\gamma_0 = 0, \mathbf{S} = \mathbf{0}$) is not considered in this experiment:

$$\frac{d}{dt}\mathbf{x}_1 = \mathbf{A}_1\mathbf{x}_1 + \mathbf{B}_1\mathbf{u} \quad (20)$$

3.4 Configuration of Simulation 3 – Applying the Online SVR as a Predictor

In this study, an online SVR is used as a predictor (see Fig. 3). In addition, in this experiment, a radial basis function (RBF) kernel is applied to this predictor. The RBF kernel of samples is notated by $\mathbf{x}$, $\mathbf{x}'$, which represents the feature vectors in any input space and are defined as follows:

$$k(\mathbf{x}, \mathbf{x}') = \exp\left(-\beta\left|\left|\mathbf{x} - \mathbf{x}'\right|\right|^2\right) \quad (21)$$

Moreover, the predictor's parameters are listed in Table 2. In the table, $i \in \{1, 2, 3, 4\}$.

3.5 Configuration of Simulation 4 – Applying the LQR as a Predictor

In this study, a linear quadratic regulator (LQR) is applied as shown in Fig. 3. The feedback gain $\mathbf{k}_f$ is applied so as to minimize the quadratic cost function J; this is calculated by the LQR as in eq. (22).

$$J = \int_0^\infty \left[\mathbf{x}^\top(t)\mathbf{Q}\mathbf{x}(t) + \mathbf{u}^\top(t)\mathbf{R}\mathbf{u}(t)\right]dt \quad (22)$$

In this verification experiment, matrices $\mathbf{Q}$ and $\mathbf{R}$ are defined as follows:

$$\mathbf{Q} = \begin{bmatrix}1 & 0 & 0 & 0 & 0\\ 0 & 6\times10^5 & 0 & 0 & 0\\ 0 & 0 & 1 & 0 & 0\\ 0 & 0 & 0 & 1 & 0\\ 0 & 0 & 0 & 0 & 4\times10^2\end{bmatrix} \quad (23)$$

$$\mathbf{R} = 1\times10^3 \cdot \begin{bmatrix}1 & 0\\ 0 & 1\end{bmatrix} \quad (24)$$

In the above equation, $\mathbf{k}_f$ is the gain for optimal feedback obtained by minimizing J. From these results, $\mathbf{k}_f$ is applied as the action predictor [1]. Furthermore, the feedback gain $\mathbf{K}_f$ of the state-feedback stabilizer is applied. However, in the verification experiment, planar movement of the two-wheeled inverted pendulum is not considered. Hence, $\phi = 0, \theta_{ml} = \theta_{mr}$, and $\mathbf{u} = u, \mathbf{d}(t) = d(t)$ are considered.

3.6 Simulation Conditions – Training Set Acquisition

In this experiment, the command input is sent to the inverted pendulum, and the environment is changed based on the command input. The shape of the floor is changed from flat to undulating. Figure 6 shows the command input, and Fig. 7 shows the shape of the floor. The shape of the floor is given by the following equation:

$$z = 0.01U(x_f)\sin\left[2\times2\cdot\pi\cdot(x_f)\right] \quad \text{[cm]} \quad (25)$$

$$x_f = x - 2 \quad \text{[cm]}$$

In this equation, $U(x)$ indicates the unit step function, x [cm] indicates the length of the floor, and z [cm] indicates the undulating

Table 1. Physical parameters of NXTway-GS

Symbol	Value	Unit	Physical property
g	9.81	[m/s^2]	Gravity acceleration
m	0.03	[kg]	Wheel weight [12]
R	0.04	[m]	Wheel radius
J_w	$\frac{mR^2}{2}$	[kgm^2]	Wheel inertia moment
M	0.635	[kg]	Body weight [12]
W	0.14	[m]	Body width
D	0.04	[m]	Body depth
H	0.144	[m]	Body height
L	$\frac{H}{2}$	[m]	Distance of Center of mass from wheel axle
J_ψ	$\frac{ML^2}{3}$	[kgm^2]	Body pitch inertia moment
J_ϕ	$\frac{M(W^2+D^2)}{12}$	[kgm^2]	Body yaw inertia moment
J_m	1×10^{-5}	[kgm^2]	DC motor inertia moment [12]
R_m	6.69	[Ω]	DC motor resistance [12]
K_b	0.468	[V·s/rad.]	DC motor back EMF constant [12]
K_t	0.317	[N·m/A]	DC motor torque constant [12]
n	1	[1]	Gear ratio [12]
f_m	0.0022	[1]	Friction coefficient between body and DC motor [12]
f_W	0	[1]	Friction coefficient between wheel and floor [12]

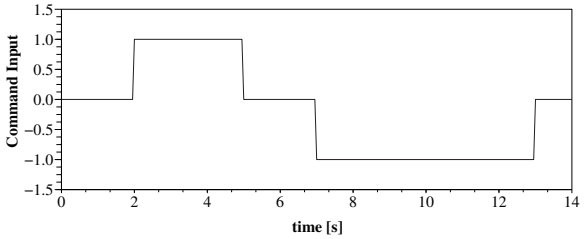


Figure 6. Command input signals (+1 indicates running forward; 0 indicates stationary balancing).

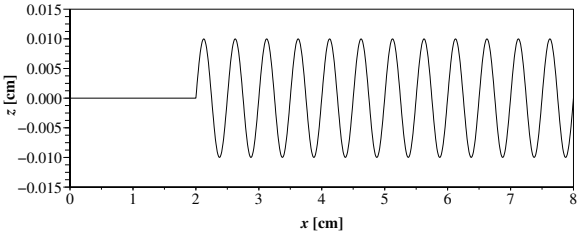


Figure 7. Simulation environment (the shape of the floor).

height of the floor. Using the settings mentioned above, we attempted to drive the inverted pendulum model on the floor by using the command input (Fig. 8). Thus, the train-

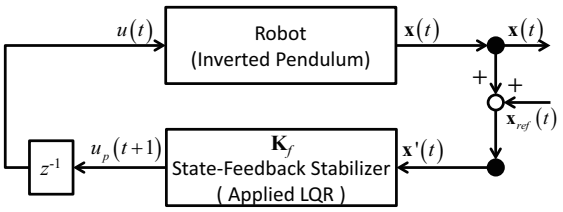


Figure 8. Control input obtained by mixing the action and disturbance inputs.

ing sets for the two-wheeled inverted pendulum can be acquired. Here, the movement distance for displaying the position is given by the following equation:

$$x = 100 \cdot R \cdot \int \dot{\theta}(t)dt \quad \text{[cm]} \qquad (26)$$

The position of the mass of the inverted pendulum z_m is given in the following equation:

$$z_m = 100 \cdot \left[R + R\sin\gamma \cdot \int \dot{\theta}(t)dt + L\cos\left\{ \int \dot{\psi}(t)dt \right\} \right] \quad \text{[cm]} \qquad (27)$$

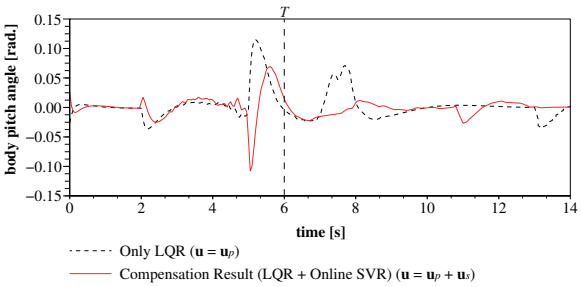


Figure 9. Control response for body pitch angle ψ using LQR as the training set vs. control response of the proposed method.

In this verification experiment, the "NXTway-GS" model achieves self-balancing. In addition, the properties of the disturbance signal that we provide as input and other conditions of the simulation are listed in Table 3.

Table 2. Learning parameters of the online SVR.

Symbol	Value	Property
C_i	300	Regularization parameter or predictor of x_i
ϵ_i	0.02	Error tolerance for predictor of x_i
β_i	30	Kernel parameter for predictor of x_i

3.7 Simulation Results

In this experiment, the training sets provided to the NXTway-GS are based on predicted results generated by the proposed method. In addition, the NXTway-GS model performs stationary balancing on the basis of these sets, and thereby moves backward or forward. Figures 9 and 10 show the compensation results and ordinary results for comparing state $\mathbf{x}_1$, and Fig. 11 shows the compensation results for control input $\mathbf{u}$. In this section, the part of the graph obtained from the actual training sets is not considered. Thus, only those parts of the graph pertaining to the state prediction part shown in T (at $t = 6.00$ [s]) of Figs. 9 through 11 are analyzed.

Moreover, Figs. 12 and 13 show the position of the mass of the inverted pendulum. The graph in Fig. 14 shows enlargement around the z position of the mass of the inverted pendulum in Fig. 13. Here, the movement distance used for displaying the trajectory was obtained by eqs. (26) and (27).

Table 3. Parameters for the simulation using the proposed method.

Symbol	Value	Unit	Physical property
ψ_0	0.0262	[rad.]	Initial value of body pitch angle
γ_0	0.0	[rad.]	Slope angle of movement direction
t_s	0.05	[s]	Sampling rate
N_s	120	—	Initial dataset length
$N_{\max}$	270	—	Maximum dataset length for the prediction
N	20	—	Step size of outputs for N-ahead state-action pair predictor's outputs
N_σ	10	—	Calculation range of the standard deviation of the predicted values

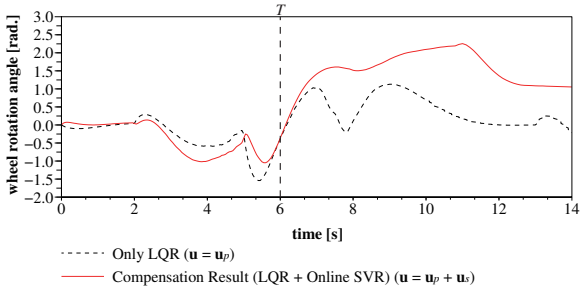


Figure 10. Control response of wheel rotation angle θ using LQR as the training set vs. control response of the proposed method.

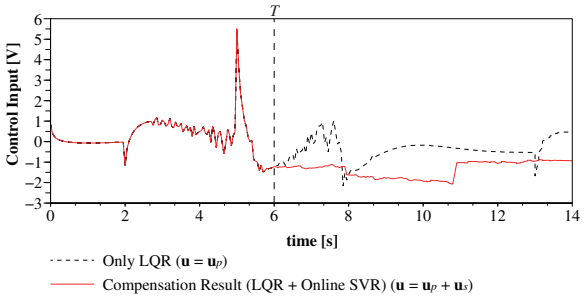


Figure 11. Control response of the control input u using LQR as the training set vs. control response of the proposed method.

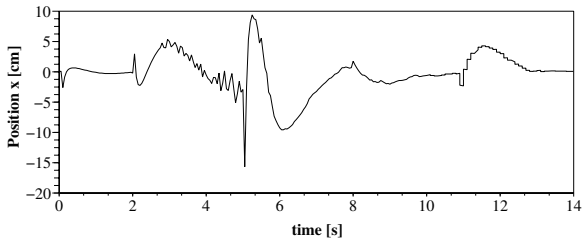


Figure 12. Position of the inverted pendulum on the x-axis using the proposed method.

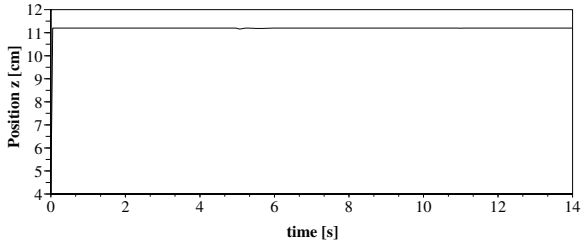


Figure 13. Position of the center of mass of the inverted pendulum on the z-axis using the proposed method.

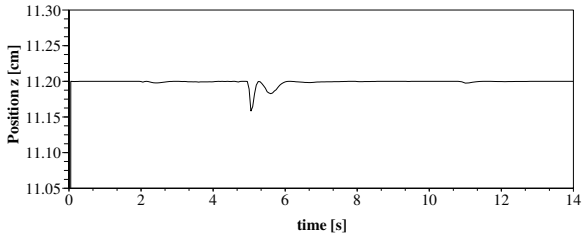


Figure 14. Position of the center of mass of the inverted pendulum on the z-axis using the proposed method focused around $z = R + L$ [cm] in Figure 13.

3.8 Discussion of Simulated Results using the Proposed Method

Here, the state prediction point is shown after $t = 6.00$ [s]. Therefore, we focus only on the part of the graph pertaining to the state prediction part shown in T. According to Fig. 9, the compensation result obtained using the proposed method (shown as the solid line) approaches and oscillates around zero with time. From Fig. 10, $\theta(t)$ is rotated backward continuously and can be confirmed. Therefore, the wheel is moving while trying to decrease the body pitch angle $\psi(t)$. However, the wheel rotation angle rotates in the backward direction continuously from 12 [s]. On the other hand, it is confirmed that the slope of the wheel rotation angle is not steep.

Additionally, with the control input $u(t)$ shown in Fig. 11, compensation results obtained using the proposed method generate control input to drive the wheel backward continuously. Furthermore, this result is taken for more advanced states than the conventional method that uses only LQR. This is because the compensated control input combines consideration of the current action and future anticipated action. In this case, as the future action, compensation control input uses the result of state-action pair prediction directly. In other words, $\theta(t)$ and $u(t)$ are influenced by the sum of the compensation input and disturbance input $d(t)$. Therefore, the compensated control input generates an action considering a future disturbance. As a result, the effectiveness of the disturbance signal is reduced. From this system characteristic, the control response for body pitch angle converges to the desired state.

Next, the movement distance is considered. In Fig. 12, it can be confirmed that the inverted pendulum is moving forward because it received a forward-running command input. In parallel, the inverted pendulum is also moving backward. From this command input, the inverted pendulum approaches the undulating floor. From this result it is clear that the inverted pendulum moved autonomously based on predicted results and was able to balance itself. Therefore, the resulting body pitch an-

gle was near zero. Subsequently, the trajectory of the mass of the inverted pendulum is analyzed. In Fig. 13, it cannot be confirmed that the body is pitching around the balance point. Accordingly, Fig. 14 is considered next. In Fig. 14, it can be confirmed that the mass of the inverted pendulum is only slightly pitched around the 0 [deg.] balance point. These results also show that the inverted pendulum moved autonomously based on predicted results, and was able to balance itself. Based on these results, the body pitch angle was confirmed to be near zero.

In this experiment, we only focused on the body pitch angle. From this viewpoint, it can be concluded that the simulation results attain a stable state despite the continuously changing shape of the road. Moreover, this response of the wheel rotation angle can be further improved if the distribution function is reconsidered. Therefore, it is concluded that the results of the verification experiment are reasonable.

4 CONCLUSION

In this paper, the relationship between the states and actions of a robot with stochastic properties was examined. To achieve this, on the basis of our previous work, we proposed a method that decides a compensation action at each sampling time based on predictions obtained from recent stochastic tendencies. Moreover, in the proposed method, LQR was applied to derive the action predictor's gain, and the normal probabilistic function was applied to define the weight coefficients. Applying this proposed method, compensated actions for rapid convergence were obtained. In other words, it was determined that the body pitch angle of the NXTway-GS model converged to zero with time.

Using the proposed method, the command input was sent to the inverted pendulum. From this command input, the shape of the floor was changed from flat to undulating. As a result, we confirmed that the inverted pendulum moved autonomously based on the predictions and could balance itself. From the results of the verification experiment, the proposed method could converge to the desired state. In particular, the slope of the body pitch angle of the NXTway-GS model converged to zero based on state and action predictions and decisions. Accordingly, the proposed method which considers a stochastic technique can be adapted to situations of stochastic transitions in external environments.

REFERENCES

[1] Masashi Sugimoto, Kentarou Kurashige : A Study of Effective Prediction Methods of the State-action Pair for Robot Control using Online SVR, *Journal of Robotics and Mechatronics*, Vol. 27 No. 5, pp. 469-479 (2015)

[2] Mark Cannon, Basil Kouvaritakis, Saša V. Raković, and Qifeng Cheng : Stochastic Tubes in Model Predictive Control with Probabilistic Constraints, Proc. 2010 American Control Conference, pp. 6274-6279 (2010)

[3] Tomoaki Hashimoto : Probabilistic Constrained Model Predictive Control for Linear Discrete-time Systems with Additive Stochastic Disturbances, Proc. IEEE 52nd Annual Conference on Decision and Control, pp. 6434-6439 (2013)

[4] Lars Blackmore, Masahiro Ono, Askar Bektassov, and Brian C. Williams : A Probabilistic Particle-Control Approximation of Chance-Constrained Stochastic Predictive Control, *IEEE Trans. on Robotics*, Vol.26 No. 3, pp. 502-517 (2010)

[5] Andrew Gray, Yiqi Gao, Theresa Lin, J. Karl Hedrick, and Francesco Borrelli : Stochastic Predictive Control for Semi-Autonomous Vehicles with an Uncertain Driver Model, Proc. IEEE 16th Annual Conference on Intelligent Transportation Systems (ITSC 2013), pp. 2329-2334 (2013)

[6] Shridhar K. Shah, Chetan D. Pahlajani, Nicholaus A. Lacock, and Herbert G. Tanner : Stochastic receding horizon control for robots with probabilistic state constraints, Proc. 2012 IEEE International Conference on Robotics and Automation (ICRA), pp. 2893-2898 (2012)

[7] Florian Weissel, Thomas Schreiter, Marco F. Huber, and Uwe D. Hanebeck : Stochastic model predictive control of time-variant nonlinear systems with imperfect state information, Proc. 2008 IEEE International Conference on Multisensor Fusion and Integration for Intelligent Systems (MFI 2008), pp. 40-46 (2008)

[8] Masashi Sugimoto, Kentarou Kurashige : A Study on the Deciding an Action Based on the Future Probabilistic Distribution, Intelligent Robotics and Applications, 9th International Conference, ICIRA 2016, Tokyo, Japan, August 22-24, 2016, Proceedings, Part II, Lecture Notes in Computer Science, vol. 9835, pp. 383-394, Springer (2016)

[9] Masashi Sugimoto, Kentarou Kurashge : The Proposal for Real-time Sequential-decision for Optimal Action using Flexible-weight Coefficient based on the State-Action Pair, Proc. 2015 IEEE Congress on Evolutionary Computation, pp. 544-551 (2015)

[10] Masashi Sugimoto, et al. : A Study of Effectiveness of Dynamically Varying Sampling Rate for State-action Pair Prediction, Proc. the Second International Conference on Electronics and Software Science, pp.79-87 (2016)

[11] John Devcic : "Weighted Moving Averages: The Basics," 2006. [Online]. Available: http://www.investopedia.com/articles/technical/060401.asp

[12] Masashi Sugimoto, Hitoshi Yoshimura, Tsukasa Abe, and Isao Ohmura : A Study on Model-Based Development of Embedded System using Scilab/Scicos, Proc. the Japan Society for Precision Engineering 2010 Spring Meeting, Saitama, D82, pp. 343-344 (2010)

International Journal of New Computer Architectures and their Applications (IJNCAA) 6(4): 132-139
The Society of Digital Information and Wireless Communications, 2016 ISSN 2220-9085 (Online); ISSN 2412-3587 (Print)

Development of 3D Video Communication System in The Synchronized AR Space

Yuki Fujibayashi and Hiroki Imamura
Department of Information Systems Science, Graduate School of Engineering, Soka University
Mailing Address: 1-236, Tangi-machi, Hachioji-Shi, Tokyo, Japan 192-8577
E-mail: e15m5223@soka-u.jp, imamura@soka.ac.jp

ABSTRACT

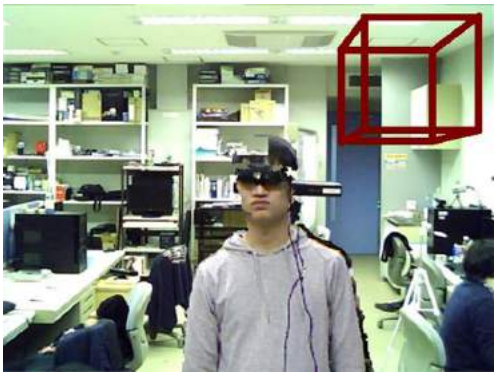
In recent years, with the evolution of communication technology and video technology, development of three-dimensional video communication system attracts attention. However, existing systems have various issues as communication tools. To solve these issues, our laboratory has proposed three-dimensional system using Kinect and Head Mounted display. Moreover, as a prototype system, we already developed a three-dimensional communication system targeting two users. In this paper, we propose improvement accuracy of the prototype system and support for multiple users. As a proposed method, we implemented UDP, hidden surface removal and parallax to improve the prototype system. Furthermore, we developed a server system for multiple users.

KEYWORDS

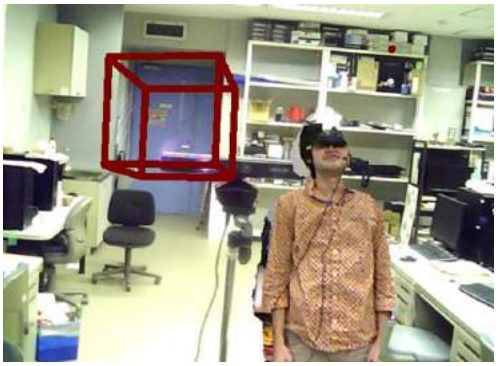
3D video communication system, Kinect, HMD (Head Mounted Display), Server, AR (Augmented Reality)

1 INTRODUCTION

In recent years, by the development of network technology and the spread of PCs, smart phones and tablets, communication system using a network become an indispensable part in our lives. For example, there are E-mail, SNS, and the Internet call service. These systems have transmitted information through 2D intermediaries such as characters, images, voice, and video. Moreover with the development of video technology, recently we have seen many techniques using 3D information.
If we combine these technologies and show the body of the other user and object in 3D, we consider that the communication can obtain smoother and realistic.



(a) The view of User A.



(b) The view of User B.

Figure 1. The view of users in the prototype system.

There are various systems as existing 3D video communication systems. For example, there are the hologram system using the 10 Kinects [1], the 3D reconstruction system by using the marker [2] and the system which reflects the movement of the user in an avatar [3]. However, they cannot share space, and realistic sensation is insufficient.
To solve the issues, we have developed a 3D video communication system using Kinect and Head Mounted Display (HMD) [4], [5]. In this system, multiple users at remote locations are reconstructed each other in the AR space[6]. Furthermore, they can share a CG object and move it intuitively. Moreover, we implemented a prototype system. Figure 1 shows the view of users in the prototype

system. In the prototype system, two users can see the 3D reconstructed human body within the sight of HMD as the 3D video communication. The prototype system indicates a CG object that users can move by hand operation.

To implement a more practical 3D video communication system, we improve the prototype system in this research. The improvements are the following three points.

- Improvement of Frame Rate
  High frame rate video is required in a video communication system. However, the frame rate is about 3.09 frames per second (FPS) in the prototype system. We aim to improve the frame late to 10 FPS because the human eye can recognize a video more than about 10 FPS.

- Providing Sense of Reality
  In a 3D video communication system, sense of distance is required in order to improve the realism. However, the prototype system shows 3D reconstructed models flat because the system uses only one camera. Moreover, the hand of the user is shown behind of the 3D reconstructed models. To provide a sense of distance, we implement hidden surface removal and parallax. Parallax means a difference of distance between both eyes.

- Support for Multiple Users
  When the system intended for use in the business meeting, it is necessary to use of the multiple users. To improve the system for multiple users, we implement server and client system.

2 PROPOSED SYSTEM

2.1 Equipment in the Proposed System

2.1.1 Kinect

We describe an outline of the Kinect using in this research. Kinect has the IR Depth Sensor, the Color Sensor which is the RGB camera and the Microphone Array consisting of four single directional microphones. The vertical viewing angle of these cameras is 43 degrees, the horizontal viewing angle is 57 degrees, and the resolution is 640 × 480 pixels.

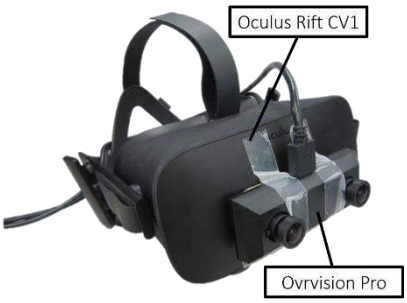


Figure 2. The structure of the HMD.

2.1.2 Head Mounted Display

We describe an outline of the HMD that is combination of Oculus Rift CV1 and Ovrvision Pro. Figure 2 shows the structure of the HMD. Oculus Rift CV1 is a HMD for virtual reality (VR). Moreover, we can get the orientation of the HMD on the three-dimensional space by the head tracker. To convert the HMD from VR to AR, we combine Oculus Rift CV1 and Ovrvision Pro. Ovrvision Pro is a stereo camera that can obtain two RGB images. Oculus Rift CV1 displays at 1080 × 1200 resolution per eye. Ovrvision Pro captures video at up to 60 frames per second.

2.1 System Overview

Figure 3 shows the configuration of this system. The system uses a server PC and multiple client PCs. A Kinect and HMD are connected to the each client PC. Each client PC sends the obtained data to

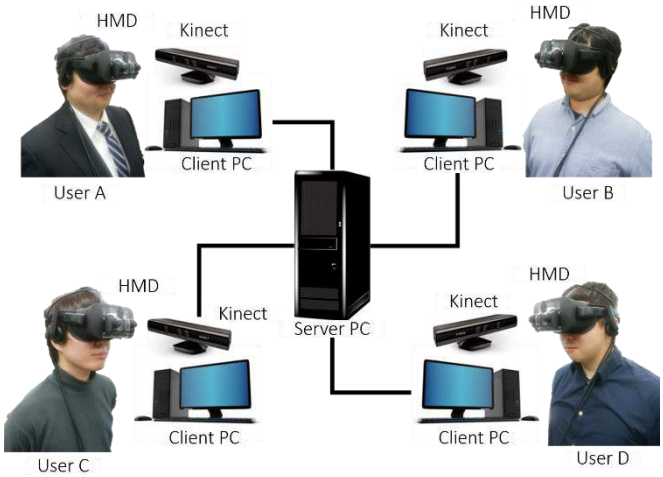


Figure 3. The component of the proposed system.

the server PC by using User Diagram Protocol (UDP). The server PC creates a server space and integrates the received data in the server space. Moreover, the server PC sends integrated information to each client PC by using UDP. Each client PC indicates the view in the server space based on received information.
The proposed method consists "Client Process" and "Server Process". We describe each process from the following section.

2.3 Client Process

Figure 4 shows the flow of the client process. We explain about each process.

2.3.1 Obtainment of the data from Kinect

In this process, each client PC obtains RGB image and depth image from each Kinect.

2.3.2 Extracting the human body

In this process, each client PC extracts each user body by basic function of Kinect. Kinect can detect skeletal data of a human body and its 20 joint positions. We use this basic function to extract the user body. When each Kinect detects skeletal data of each user body, each client PC keeps depth data of the user body, and deletes other pixels depth data.

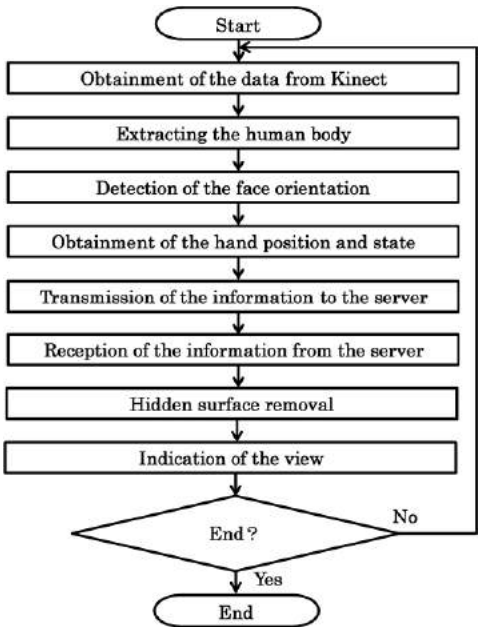


Figure 4. The flow of the client process.

2.3.3 Detection of the face orientation

In this process, each client PC detects face orientation of each user. HMD obtains orientation of the HMD as quaternion. The server PC needs rotation matrix to move the viewpoint in server space. Therefore, each client PC converts the quaternion to rotation matrix

$$\boldsymbol{R} = \begin{pmatrix} 1-2y^2-2z^2 & 2xy+2wz & 2xz-2wy \\ 2xy-2wz & 1-2x^2-2z^2 & 2yz+2wx \\ 2xy+2wy & 2yz-2wx & 1-2x^2-2y^2 \end{pmatrix}, (1)$$

where x, y, z and w denote elements of a quaternion.

2.3.4 Obtainment of the hand position and state

In this process, each client PC obtains the hand position and status that is grab or release. Firstly, each Kinect obtains 3-dimensional hand coordinate (H_x, H_y, H_z) based on skeletal data.
Secondly, each client PC converts 3-dimensional hand coordinate to 2-dimensional hand coordinate (H'_x, H'_y) of RGB image. Thirdly, each client PC creates a square area centered on the coordinate (H'_x, H'_y). The side length of the square

$$L[\text{pixels}] = \frac{110}{H_z} \qquad (2)$$

changes according to the distance between the hand and the Kinect as shown in Figure 5. Fourthly, each client PC detects skin color within the square area.

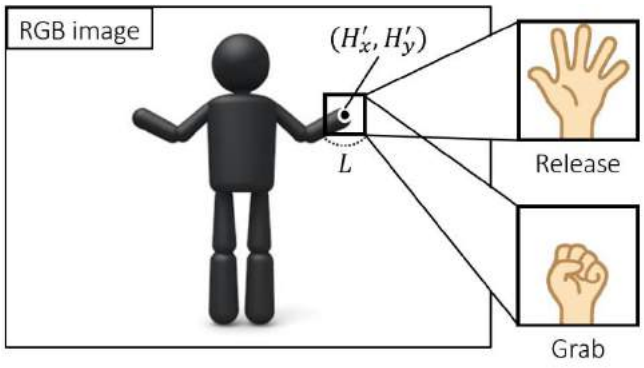


Figure 5. Judgement of hand status.

Fifthly, each client PC calculates skin color rate in the square area

$$S = \frac{px_s}{px_a}, \quad (3)$$

where px_s denotes the total number of pixels in the square area and px_a denotes the number of skin color pixels in the square area. If skin color rate is less than the threshold, then each client PC judges that hand status of the user is grab.

2.3.5 Transmission of the information to the server

In this process, each client PC sends the RGB image, the depth image, the rotation matrix, coordinate of joint, and hand status through socket programs [7]. The proposed system employs UDP as a communication protocol. Moreover, each client PC reduces the RGB image and the depth image before sending to reduce send data.

2.3.6 Reception of the information from the server

In this process, each client PC receives the view in the server space. The view images of left eye and right eye are sent from the server PC after server process.

2.3.7 Hidden surface removal

In this process, each client PC performs hidden surface removal to provide sense of distance. Firstly, each client PC sets the image obtained from left-eye camera of HMD as a background of left eye view. The same process is performed for the right eye view. Secondly, each client PC compares the position of user hand and CG object and other users. If user hand is closer to user head than CG object and other users, then the client PC performs skin color detection. Finally, the client PC overwrites the detected skin color area on the image of view.

2.3.8 Indication of the view

In this process, each client PC displays the view on the HMD. At that time, the left eye image is displayed on the left display of HMD. The same process is performed on the right side.

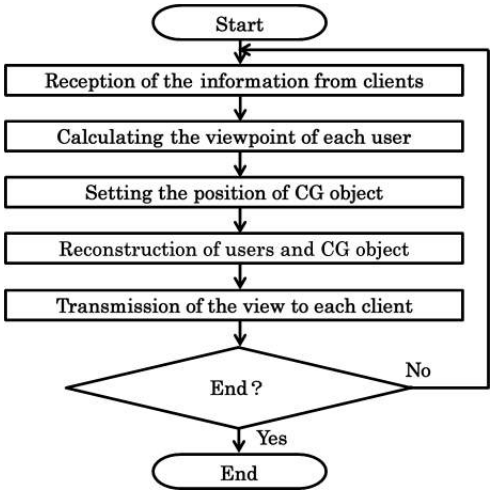


Figure 6. Flow of the server process.

2.4 Server Process

Figure 6 shows the flow of the client process. We explain about each process.

2.4.1 Reception of the information from clients

In this process, the server PC receives the RGB image, the depth image, the rotation matrix, coordinate of joint, and hand status from each client PC.

2.4.2 Calculating the viewpoint of each user

In this process, the server PC calculates viewpoint and fixation point of each client user in the server space. Firstly, the server PC calculates a viewpoint of Kinect coordinate system. A viewpoint of left eye $\left(E_{l_x}, E_{l_y}, E_{l_z}\right)$ is calculated by using

$$\boldsymbol{E_l} = \boldsymbol{U} + \boldsymbol{R} \cdot \begin{pmatrix} -\frac{D_1}{2} \\ 0 \\ D_2 \end{pmatrix} - \begin{pmatrix} 0 \\ 0 \\ D_3 \end{pmatrix}, \quad (4)$$

where $\boldsymbol{U}$ and $\boldsymbol{R}$ denote the position of head and the rotation matrix received from the user. As shown in Figure 7, D_1 denotes distance between both eyes, D_2 denotes distance between U_z and R_z and D_3 denotes distance between Kinect and origin. The

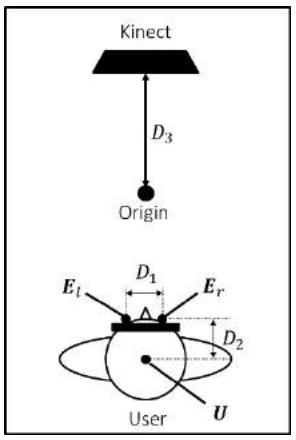


Figure 7. Space of a user.

values of D_1 , D_2 and D_3 can be changed. Furthermore, server PC calculates a viewpoint of right eye $\left(E_{r_x}, E_{r_y}, E_{r_z}\right)$ by using

$$\boldsymbol{E_r} = \boldsymbol{U} + \boldsymbol{R} \cdot \begin{pmatrix} \frac{D_1}{2} \\ 0 \\ D_2 \end{pmatrix} - \begin{pmatrix} 0 \\ 0 \\ D_3 \end{pmatrix}. \qquad (5)$$

Moreover, the server PC converts the calculated viewpoint of each user into the coordinate system of the server space. In case of two users, the server PC assigns number to each user as shown in Table 1. In case of three or more users, the server PC assigns number to each user as shown in Table 2. The server PC converts each viewpoint $\boldsymbol{E}$ by using user number n and

$$\boldsymbol{E'_n} = \begin{cases} \begin{pmatrix} E_{n_x} \\ E_{n_y} \\ E_{n_z} \end{pmatrix} + C \begin{pmatrix} 0 \\ 0 \\ n/4 \end{pmatrix} & (n \bmod 4 = 0) \\ \begin{pmatrix} E_{n_z} \\ E_{n_y} \\ -E_{n_x} \end{pmatrix} + C \begin{pmatrix} n/4 \\ 0 \\ 0 \end{pmatrix} & (n \bmod 4 = 1) \\ \begin{pmatrix} -E_{n_x} \\ E_{n_y} \\ -E_{n_z} \end{pmatrix} + C \begin{pmatrix} 0 \\ 0 \\ -n/4 \end{pmatrix} & (n \bmod 4 = 2) \\ \begin{pmatrix} -E_{n_z} \\ E_{n_y} \\ E_{n_x} \end{pmatrix} + C \begin{pmatrix} -n/4 \\ 0 \\ 0 \end{pmatrix} & (n \bmod 4 = 3) \end{cases}. \qquad (6)$$

where C denotes the distance between the front and rear users.

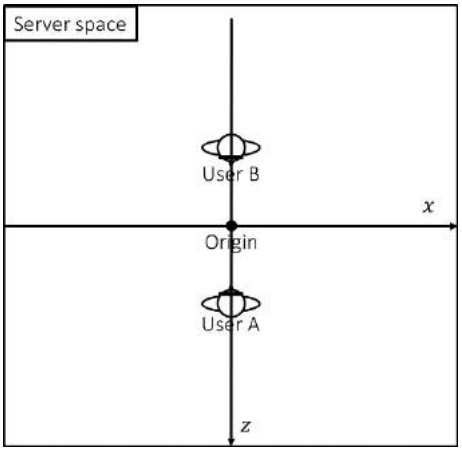
Table 1. User number of two users.

User	A	B
Number	0	2

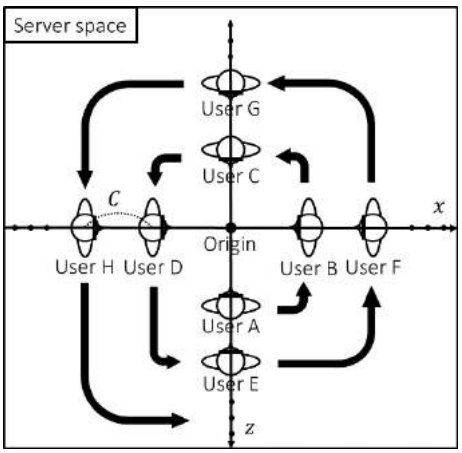
Table 2. User number of three or more users.

User	A	B	C	D	E	F	⋯
Number	0	1	2	3	4	5	⋯

Figure 8 shows the location of users in the server



(a) In case of two users.



(b) In case of three or more users.

Figure 8. The initial location of users in the server space after conversion.

space after conversion. The origin in Figure 7 and the origin in Figure 8 overlap by conversion. Finally, the server PC calculates a fixation point of each eye

$$\boldsymbol{F_n} = -\boldsymbol{E'_n} - 20\boldsymbol{R_n} \cdot \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix}. \qquad (7)$$

2.4.3 Setting the position of CG object

In this process, the server PC locates the position of CG object based on hand status and coordinate of users. If a received hand status of a user is grab, then the server PC judges that the hand coordinate of the user (H_x, H_y, H_z) is within the CG object (C_x, C_y, C_z) using

$$(|H_x - C_x| < S) \wedge (|H_y - C_y| < S) \wedge (|H_z - C_z| < S), \quad (8)$$

where S denotes the size of the CG object. If conditional expression (8) is satisfied, then the server PC changes color of the CG object and conforms the coordinate of the object to the coordinate of the hand.

2.4.4 Reconstruction of users and CG object

In this process, the server PC performs three-dimensional reconstruction of the users and the CG object in the server space. The users are reconstructed to the position calculated using initial position and received head coordinate of the users. The CG object is reconstructed to the position set in section 2.4.3.

2.4.4 Transmission of the view to each client

In this process, the server PC sends left eye view and right eye view in the server space to each client PC. The two views are obtained from the viewpoint $\boldsymbol{E'_n}$ and the fixation point $\boldsymbol{F_n}$ set in section 2.4.2.

3 EVALUATION EXPERIMENT

3.1 Outline of the experiment

In evaluation experiment, 10 subjects use the proposed system. A subject wear HMD, move, grasp the CG object and move it. Subjects perform these actions with the prototype system and the proposed system. After that, they evaluated following items.

item1: The communication speed and frame rate were good.
item2: There was no discomfort when you moved the fixation point.
item3: It was easy to feel sense of distance.
item4: The three-dimensional reconstructed person was realistic.
item5: It was easy to move the CG object.

They evaluated with evaluate value in 5 scales. Following is the evaluation value and evaluation word.
5: Strongly agree, 4: Agree, 3: Neutral, 2: Disagree, 1: Strongly disagree
From Figure 9 to Figure 14 show example images of experiments.

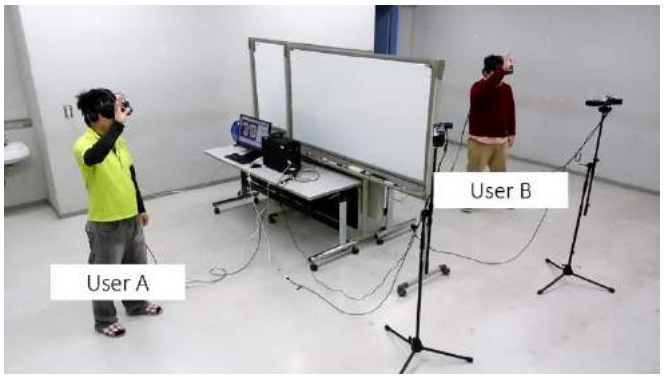


Figure 9. The experimental environment of user A and user B.

Figure 10. The experimental environment of user C.

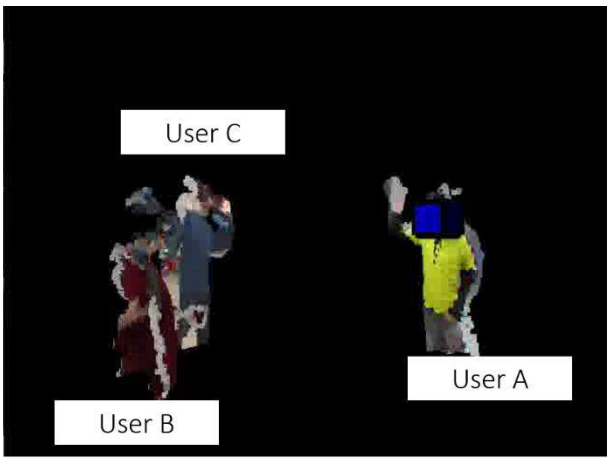


Figure 11. The image of server space.

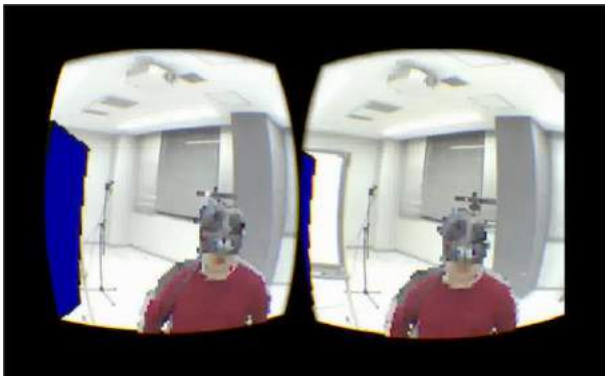


Figure 12. The view of user A.

Figure 13. The view of user B.

Figure 14. The view of user C.

3.2 Result

Table 3 and Table 4 show the result of experiment. In the proposed system, the average score from item1 to item3 and item5 are higher than the prototype system. Moreover, all standard deviations are lower than 1.0.

Table 3. The result of prototype system.

No.	item1	item2	item3	item4	item5
Average of evaluation	2.10	2.40	2.00	3.50	1.40
Standard deviation	0.54	0.66	0.89	0.50	0.49

Table 4. The result of proposed system.

No.	item1	item2	item3	item4	item5
Average of evaluation	4.30	4.30	3.60	2.60	3.50
Standard deviation	0.46	0.64	0.66	0.66	0.81

3.3 Discussion

Firstly, item1 is a question about frame rate. From the result of item1, many of the subjects felt that the frame rate was good. Actually, the frame rate of proposed system is about 7.2 FPS when we measured it. Although we did not reach 10 FPS, we succeeded in improving the frame rate enough.
Secondly, from item2 to item5 are questions about sense of reality. Many of the subjects felt that sense of distance was better than the prototype system, because the result of item2, 3 and 5 is higher than the prototype system.
From the above, the proposed system was evaluated as improving practicality.
However, the result of item4 was low. We consider that it was caused by reduction of transmission data for improvement of frame rate. We consider that one of the improvement measures is to change the method of send and receive body data of users.

4 CONCLUSION

In this paper, we proposed the 3D video communication system in synchronized AR space and improve the prototype system of the proposed system along the following three items.

- Improvement of frame rate
- Providing sense of reality
- Support for multiple users

We implemented UDP, hidden surface removal and parallax to improve the prototype system. Furthermore, we developed a server system for multiple users. From the results of evaluation experiment, it can be said that the prototype system has been improved in the above three items.

As future work, we will add voice communication function and 3D reconstruction of whole body through multiple Kinects. Since this system uses one Kinect for each, this system can obtain only front part of human body data, and users can see it only. Therefore, we need to put more Kinect for obtaining whole body data to make full 3D reconstruction.

5 REFERENCES

1. Kibum Kim, John Bolton, Audrey Girouard, Jeremy Cooperstock and Roel Vertegaal, "TeleHuman: Effects of 3D Perspective on Gaze and Pose Estimation with a Life-size Cylindrical Telepresence Pod", Proceedings of Chi'12 Conference on Human Factors in Computing Systems, 2012.
2. Ryo Jozaki, "AR Chat: remote communication support system that uses augmented reality", The national convention of IPSJ, 2012.
3. Avatar Kinect, http://marketplace.xbox.com/en-US/Product/Avatar-Kinect/66acd000-77fe-1000-9115-d8025848081a
4. Hideyuki Hashimoto, Yuki Fujibayashi and Hiroki Imamura, "Development for 3D Video Communication System by Using Kinect and Head Mount Display in the AR Space", The International Conference on Computer Graphics, Multimedia and Image Processing (CGMIP2014), 2014.
5. Yuki Fujibayashi and Hiroki Imamura, "Development of File Transfer Application by Handover for 3D Video Communication System in Synchronized AR Space", International Journal of New Computer Architectures and their Applications (IJNCAA) Vol.5, No4, pp141-147, 2015.
6. Van Krevelen, D. W. F., R. Poelman, A survey of augmented reality technologies, applications and limitations., International Journal of Virtual Reality 9.2 (2010): 1.
7. Lewis Napper, Winsock 2.0: Windows Socket Power Guide, John Wiley & Sons; Pap/Cdr edition (1997).

International Journal of New Computer Architectures and their Applications (IJNCAA) 6(4): 140-149
The Society of Digital Information and Wireless Communications, 2016 ISSN 2220-9085 (Online); ISSN 2412-3587 (Print)

GPU-based Object Identification in Large-scale Images for Real-time Radar Signal Analysis

Young-Min Kang
Tongmyong University
Busan, 48520, Korea
ymkang@tu.ac.kr

Sung-Soo Kim
ETRI
Daejeon, 34129, Korea
sungsoo@etri.re.kr

Gyung-Tae Nam
GCSC Inc.
Busan, 47607, Korea
gtnam@gcsc.co.kr

ABSTRACT

As the computing power of processors is being drastically improved, the sizes of image data for various applications are also increasing. However, CCL cannot be easily implemented in a parallel fashion because the connected pixels can be found basically only by graph traversal. In this paper, we propose a GPU-based efficient algorithm for object identification in large-scale images and the performance of the proposed method is compared with that of the most commonly used method implemented with OpenCV libraries. The experimental results show that the proposed method outperforms the reference method when the pixel density is below 0.7. Object identification in image data is the fundamental operation and rapid computation is highly requested as the sizes of the currently available image data rapidly increase. The method proposed in this paper is applied to the analysis of radar signals. Since the radar signals contains complex and large data, the efficiency of the method is crucial when realtime analysis is required. The experimental results show the proposed method can be a good solution to the object identification in large-scale image data such as radar signals.

KEYWORDS

GPGPU, Object Identification, CCL, Radar Signals, Realtime Analysis.

1 INTRODUCTION

As the computing power of processors is being drastically improved, the sizes of image data for various applications are also increasing. Therefore, efficient algorithms for manipulating the large-scale image data are required. One of the most basic operations on image data is to identify objects within the image, and the connected component labeling (CCL) is the most frequently used strategy for this problem.

Various image processing techniques can be easily implemented in parallel fashion, and GPU parallelism has been successfully exploited in this field. However, CCL cannot be easily implemented with parallel tasks because the connected pixels are represented as adjacent nodes in a graph and the adjacency among all the nodes can be investigated basically only by graph traversal.

In this paper, we propose a GPU-based efficient algorithm for object identification in large-scale images and the performance of the proposed method is compared with that of an OpenCV-based CCL algorithm.

2 RELATED WORK

Object identification is a fundamental problem in image processing. In many cases, the object identification is performed based on CCL. The most CCL algorithms are reduced to the traversal of adjacent nodes (pixels) along the edges in the graph that represents the input image. The traversal approaches are naturally sequential, and parallel implementations of typical CCL algorithms have not been very successful [10].

Even in the early stage of computer vision research, it was found that the connectivity cannot be determined by completely parallel tasks [7]. However, the rapid development of general purpose graphics processing unit (GPU)

technologies made it possible for GPU-based parallel approaches to achieve better performance than CPU-based traditional CCL methods [1, 9].

The basic approach to CCL is to use *union-find* algorithm which can determine whether two nodes in an undirected graph are connected or not [8]. However, such methods based on this approach use sequential computations. Several methods have been proposed to exploit parallel computing architectures [5, 9]. However, these methods were applied to relatively small images. Some methods utilized cluster architectures [4]. These methods split the data volume and the data segments are assigned to different computing units. Parallelism within a single GPU, therefore, cannot be exploited in these methods.

Label equivalence method is implemented on GPU [6], and this method iterates to resolve the label equivalence by finding the roots of equivalence trees. However, this approach relies on decision tables which cannot be efficiently handled on GPU [10].

Block-based labeling and efficient block processing with decision table were proposed in [2, 3]. Block equivalence method based on "scan mask" was proposed [10]. However, this method also has to iterate the equivalence resolving until it converges to the state where no label update is found.

3 PROPOSED METHOD

In this section, a GPU-based parallel approach to CCL is proposed. The method is composed of four major tasks: 1) data initialization, 2) computation of column-wise label runs, 3) label merge of connected components. Each task is explained in details in the following subsections. The proposed method is to apply to radar signals. Therefore, the radar signals must be efficiently read and reconstructed in the format of normal images.

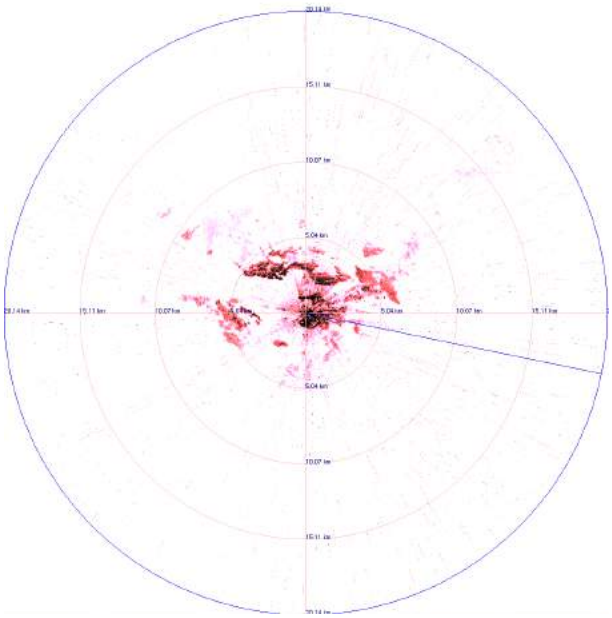


Figure 1. Radar signal visualization example

3.1 Real-time Radar Signal Visualization

The method proposed in this paper is to be applied to realtime analysis of large images such as radar signals shown in Fig. 1. ASTERIX is a well-known standard for the exchange of air traffic services (ATS) information developed by the European ATS organization Eurocontrol. In order to analyze the ASTERIX radar signal with image processing techniques, the signals must be parsed and reconstructed as rectangular image. The image reconstruction process is as follows:

1. Data Parsing: In this step, we read the radar signals in accordance with the data format. The reflectance intensity within the radar range is streamed into the system in azimuthal angle order.

2. Image Construction: The raw data in the radar signal is not in the form of rectangular images. In this step, we map the intensity signal into 4096×4096 image space.

3. Realtime Visualization: In order to visualize the radar signal mapped into 4096×4096 image in an efficient fashion, we allocate a buffer in the texture memory

to hold the image. The realtime visualization can then be easily implemented with GPU technologies.

3.2 Label Initialization

The pixels in an image can be classified into either 'on' pixels and 'off' pixels. The 'on' pixels are regarded nodes in graph representation, and it is assumed that an edge exists between two nodes of which positions neighbor each other in the image space.

The goal of CCL algorithms is to assign an identical label for linked nodes. In order to achieve this goal, each pixel is assigned unique label in the initialization stage. The simplest method is to assign sequential numbers to the pixels. Suppose we have an image with $w \times h$ pixels and the pixel at (x,y) is denoted as $p(x, y)$ where $x = [1, w], y = [1, h]$. The 'on' pixel at (x, y) is then labeled with the number $x + w(y - 1)$. Therefore, the labeling numbers range from 1 to wh. All the 'off' pixels are labeled to be -1.

The label map I_λ is an image composed of label of each pixel, and each label at (x, y) in the label map are denoted by $I_\lambda^{x,y}$. In other words, the image with initial labels can be described as follows:

$$\begin{aligned} I_\lambda &\in \mathbb{Z}^{w \times h} \\ id^{x,y} = x + w(y-1) &\quad \in [1, wh] \\ p(x,y) = 1 \Rightarrow l_\lambda^{x,y} &\quad = id^{x,y} \\ p(x,y) = 0 \Rightarrow l_\lambda^{x,y} &\quad = -1 \end{aligned} \tag{1}$$

After the initialization is done, the rest of the algorithm is to merge the positive labels in each connected component into a single label.

3.3 Computing Column-wise Label Runs

In order to merge labels, adjacent positive labels are merged. The block of contiguous object pixels in a column is a 'run.' The first stage of label merge is to find runs. In other words, each run is identified and labeled with a unique number. Each column is assigned to a CUDA thread and processed in parallel fashion. Therefore, we have w threads running separately.

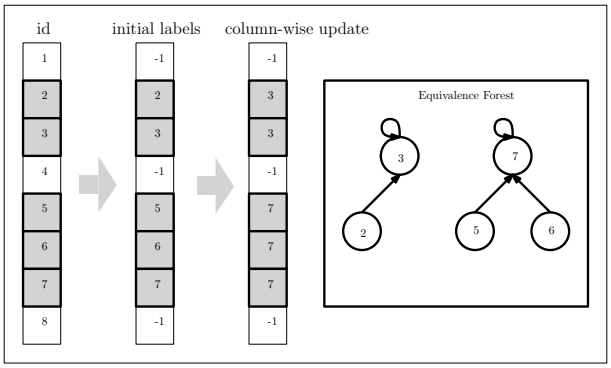


Figure 2. Label runs and equivalence forest

The computation within a thread is to simply scan pixels and change the label of the current pixel to be that of the previously scanned pixel if the both pixels are 'on.'

Fig. 2 shows how the label assigned to each pixel is updated through column-wise label run computation described in Algorithm 1. After the update, the each label represents the root node in the equivalence tree it belongs to as shown in Fig. 2

Algorithm 1: Column-wise label run

kernel **vertLabel**

Data: $I_\lambda \in \mathbb{Z}^{w \times h}$: In, Out
begin
  col = thread:$[1, w]$
  for *row:* $h - 1$ *downto* 1 **do**
    if $I_\lambda^{row,col} > 0$ *and* $I_\lambda^{row+1,col} > 0$ **then**
      $I_\lambda^{row,col} = I_\lambda^{row+1,col}$

3.4 Label Merge

Once the column-wise label update is finished, the horizontal connectivity must be investi-

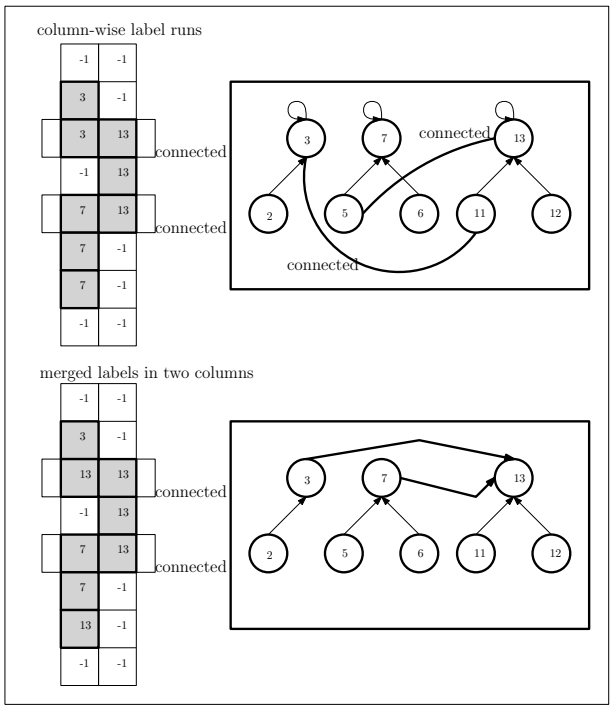


Figure 3. Label merge with two columns

gated. Let us suppose, for simplicity, that we have only two columns. In the previous column-wise update, the labels are merged to the largest value in the equivalence tree. If two pixels in a row are connected, the equivalence trees those pixels belong to should be merged into a single tree. In order to achieve this, the root node of each tree must be found and the root node with a smaller label is relabeled to point the other root with a larger label as shown in Fig. 3.

Note that the labels of the connected pixels are not directly updated. Fig. 4 shows the merge process. As shown in Fig. 4 (a), two pixels a and b in different equivalent trees are found to be connected. The direct update of connected pixels does not successfully merge the equivalence trees as shown in Fig. 4 (b). The correct label merge can be done by comparing and update of the roots of the equivalence trees as shown in Fig. 4 (c).

In order to perform the two-column label merge for an images with w columns, $w/2$ column pairs are separately merged with $w/2$ threads. After every two adjacent column pairs are merged, we iterate the label merge. In

Algorithm 2: Label merge algorithm

host **callMergeLabel**

Data: $I_\lambda \in \mathbb{Z}^{w \times h}$: In, Out
begin
  div = 2
  for *i:* 0 *upto* $\lg w - 1$ **do**
    merge <<$h \cdot w$/div>> (div,I_λ)
    div = 2 $\times$ div

kernel **merge**

Data: div $\in \mathbb{Z}$: in, $I_\lambda \in \mathbb{Z}^{w \times h}$: In, Out
begin
  thread:[$1, wh$/div]
  nBoundary = w/div
  col = div/2 + (thread % nBoundary) $\times$ div
  row = thread / nBoundary
  if $I_\lambda^{row,col} > 0$ *and* $I_\lambda^{row,col+1} > 0$ **then**
    $root_L$ = **findRoot**(row,col)
    $root_R$ = **findRoot**(row,col+1)
    $I_\lambda^{\min(root_L,root_R)} = I_\lambda^{\max(root_L,root_R)}$

device **findRoot**

Data: row, col $\in \mathbb{Z}$: In, label $\in \mathbb{Z}$: Out
begin
  if $I_\lambda^{row,col} < 0$ **then**
    return -1
  label = w row + col
  while $I_\lambda^{label} \neq$ label **do**
    label := I_λ^{label}
  return label

Algorithm 3: Relabeling

kernel **relabel**

Data: $I_\lambda \in \mathbb{Z}^{w \times h}$: In, Out
begin
  thread:[$1, w \cdot h$]
  row = thread/w, col = thread%w
  $I_\lambda^{row,col} = \mathbf{findRoot}(row, col)$

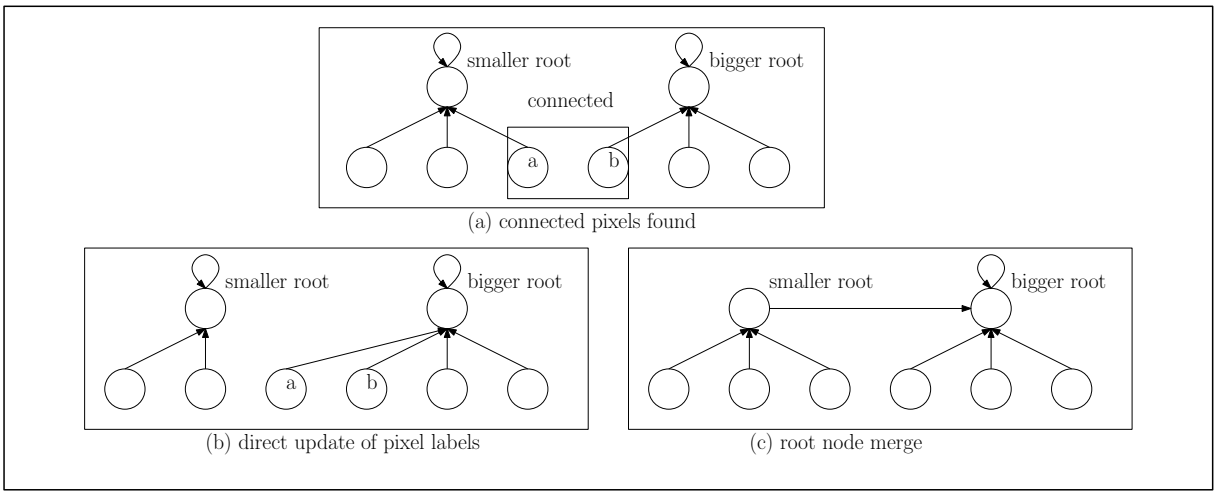


Figure 4. Label merge for equivalence tree merge

the second merge phase, as shown in Fig. 5, we have only to consider the boundaries between the previously merged column pairs, and the computation is reduced to half the previous one.

It is easily noticed that the $\lg w$ iterations are sufficient for finding the final label equivalence. Let us denote the computational cost for the first iteration to be $C(1)$. The total computational cost for the label merge is $\sum_{i=0}^{\lg w-1} \frac{1}{2^i} C(1) = \mathbf{O}(C(1))$.

Algorithm 2 shows the implementation details of our method. The label merge is implemented with one host function, one kernel function, and one device function. In the host function callMergeLabel, we determine the number of boundaries where column-pairs are merged and call kernel function merge with the necessary number of threads. The host function iterates this call $\lg w$ times, and the number of threads to be called decreases as the iteration is repeated. In the i=th call, $w \cdot h/2^i$ threads are required.

Every thread executes the kernel function merge. In the kernel function, every two pixels across the merge boundaries are investigated in parallel fashion. If the pixels are both 'on', the root nodes of equivalence trees the pixels belong to are found and compared. If they have identical root node, they are already in a single equivalence tree. Therefore, no more operations are required. Otherwise, they are in different equivalence trees. However, the horizontal adjacency of the pixels informs that they must be actually in an identical equivalence group. In this case, we must reassign the root labels of the pixels.

The reassign process can be understood as a tree merge as shown in Fig. 4. The label equivalence trees are merged by relabeling the root with smaller label to have the same label with the other root. The device function findRoot is called in this process to find the root of the pixel currently being investigated.

After the execution of Algorithm 2, the equivalence tree will be obtained. However, the final goal of CCL is to make all the pixels in a connected component have an identical label. This can be achieved by applying the device function findRoot to each pixel and updating its label to be the returned value. This process can be easily performed in a parallel fashion because the label updates for any nodes in the tree do not destroy the equivalence of the nodes in the tree.

Algorithm 3 describes the operations of relabeling thread. Total $w \cdot h$ threads separately call findRoot for corresponding pixels and update the label in order to make the connected pixels have an identical label.

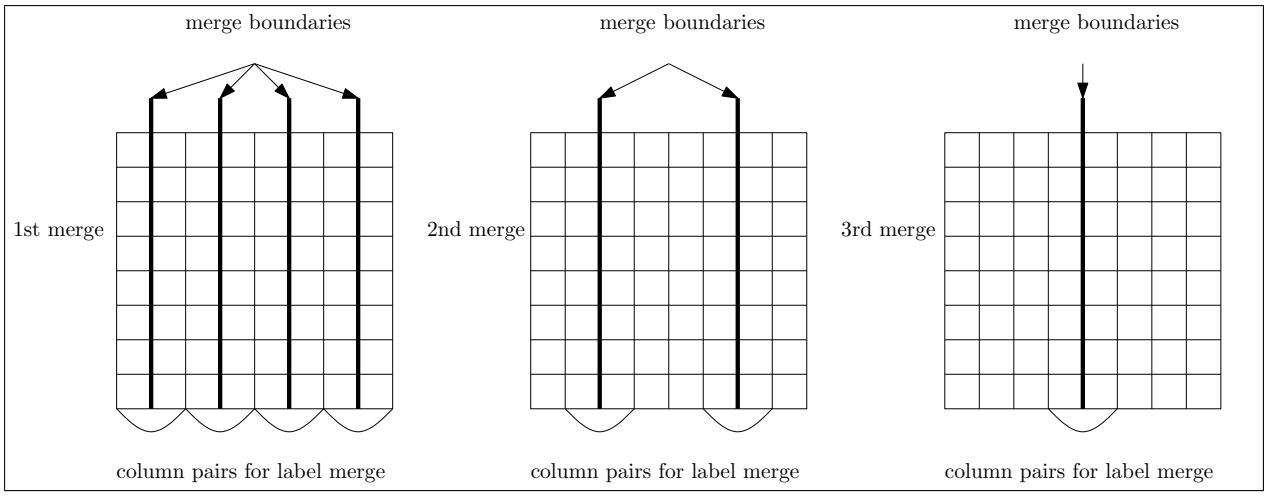


Figure 5. Label merge iteration

4 EXPERIMENTAL RESULTS

Table 1. CCL performance comparison with random density noise patterns (2048×2048 pixels).

2048×2048		
density	**Grana** (ms)	**Proposed method** (ms)
0.1	22.8	6.3
0.2	30.7	7.0
0.3	46.0	7.7
0.4	51.3	8.6
0.5	47.6	10.2
0.6	43.6	15.2
0.7	35.5	26.1
0.8	28,0	40.8
0.9	20.8	64.6

The method proposed in this paper was implemented on computing environments with commodity CPUs and GPUs. The experimental results were collected from the tests on a system with i7-3630QM 2.4 GHz CPU and Geforce GTX 670MX GPU.

In order to verify the efficiency of our method, we compared the performance of our method with the most commonly used method. The reference method was proposed in [3], and implemented with OpenCV libraries.

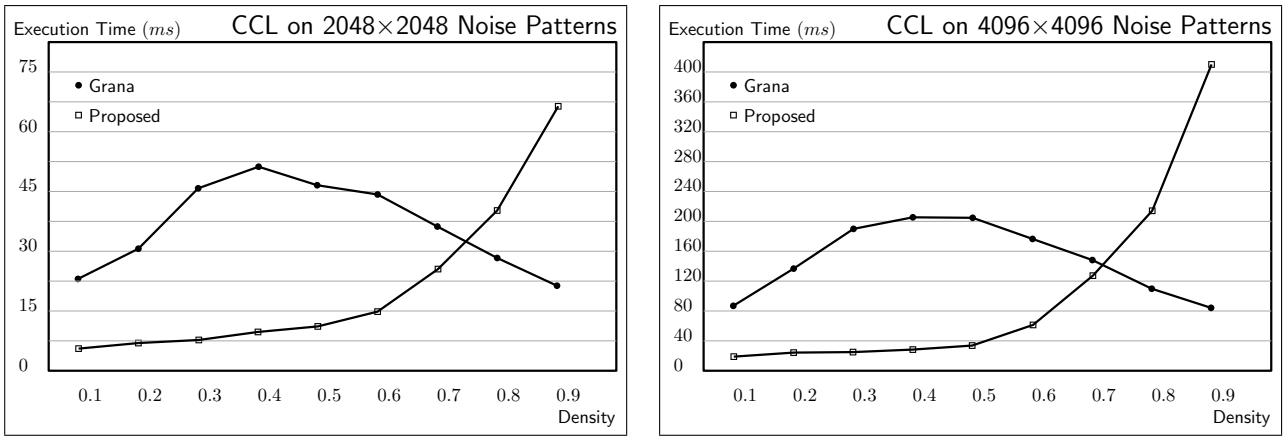
In the first experiment, the performance of each method was measured by applying images with random noise. The random noise was automatically generated and the density of the noise ranges from 0.1 to 0.9. Table. 1 shows the experimental results when 2048×2048 images with random noise are applied. The first column represents the noise density, and the second column shows the measured execution time of the reference method in milliseconds. The third column shows the execution time of the proposed method. As shown in the table, the reference method (denoted by Grana) requires more execution time when the noise density is around 0.5 while the proposed method requires more time as the density increases.

Table 2. CCL performance comparison with random density noise patterns (4096×4096 pixels).

4096×4096		
density	**Grana** (ms)	**Proposed method** (ms)
0.1	85.7	21.4
0.2	135.3	24.3
0.3	190.4	27.1
0.4	206.1	30.8
0.5	205.4	36.9
0.6	177.6	59.0
0.7	153.1	126.7
0.8	112.1	216.7
0.9	85.6	411.5

Table. 2 shows the similar experimental results except that the size of the input images is 4096×4096. As shown in the table, the computational cost is similarly changing in accordance with the density.

Fig. 6 (a) visualizes the experimental results



(a)2048×2048-sized noise patterns (b) 4096×4096-sized noise patterns

Figure 6. CCL execution time on noise patterns with different noise densities

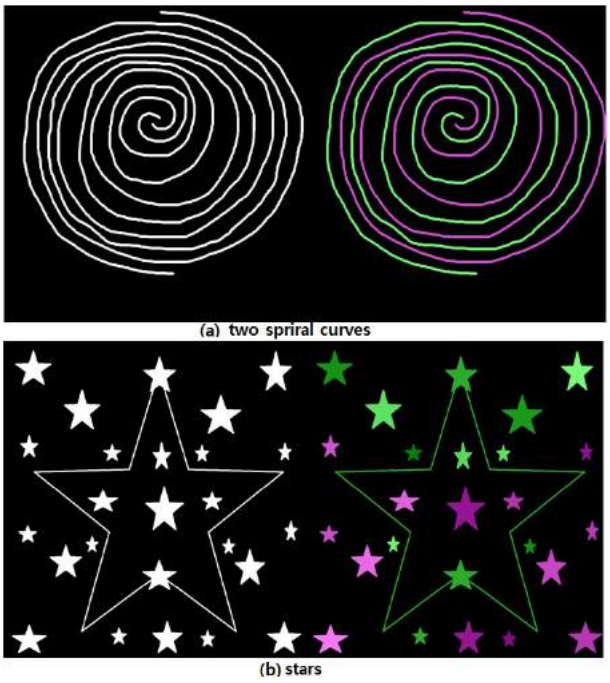


Figure 7. Test images for practical labeling: (a) two spiral curves (b) scattered stars

shown in Table. 1. As shown in the figure, the computational cost of the proposed method increases as the density increases. However, the proposed method is far better until the density is below 0.7.

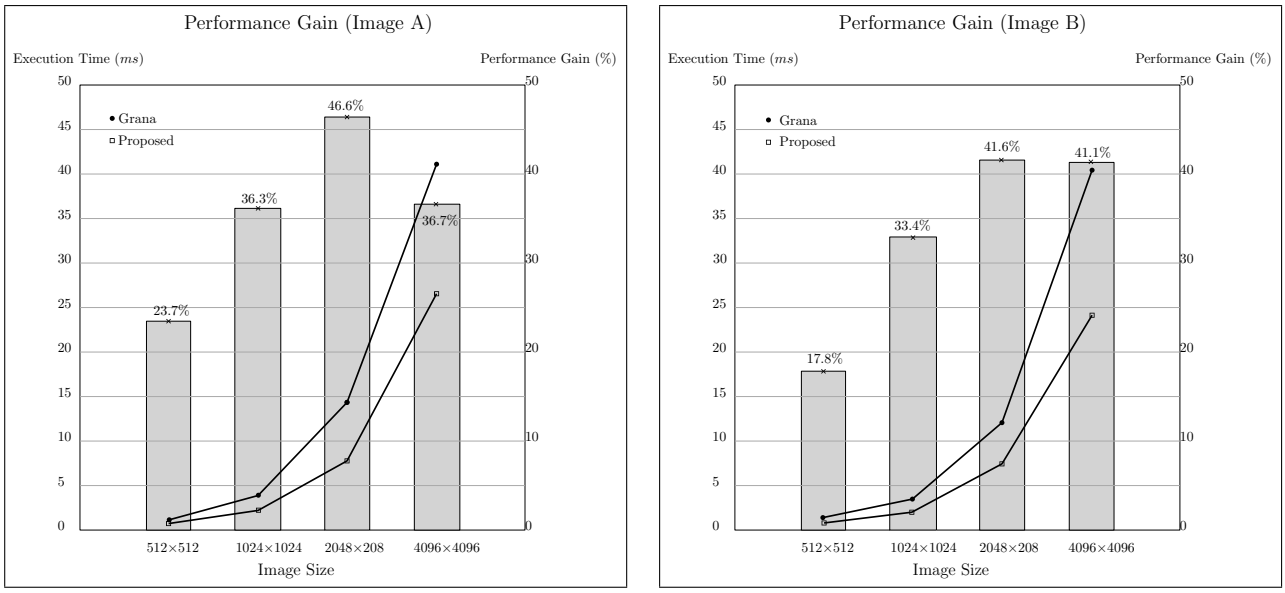
Fig. 6 (b) shows the similar results. This figure visualizes the result shown in Table. 2. As shown in the figure, the computational cost of the proposed method increases as the density increases. However, the proposed method is far better until the density is below 0.7.

Table 3. CCL performance comparison with test images A and B with different image sizes.

methods	Images sizes 512^2	1024^2	2048^2	4096^2
Test Image A				
Grana (ms)	1.14	3.86	14.39	41.1
Proposed (ms)	0.87	2.46	7.68	26.0
Gain (%)	23.7	36.3	46.6	36.7
Test Image B				
Grana (ms)	1.01	3.53	12.35	40.78
Proposed (ms)	0.83	2.35	7.21	24.03
Gain (%)	17.8	33.4	41.6	41.1

The CCL algorithms are not actually applied to noise data. In order to measure the performance of the proposed method in more feasible environments, we prepared two test images shown in Fig. 7. There are two different test images and the sizes of the images can be either 512^2, 1024^2, 2048^2 or 4096^2. Test image A has two components, and each of them is a long spiral curve without touching the other component. The other test image B has many scattered stars, and two of them are connected with a star-shaped thin line.

Table. 3 shows the execution time required for reference method and the proposed method applied to the test images with different sizes. In the last row in each data set obtained by using each test image, the performance gain is computed and shown. The performance gain is obtained by computing the ratio of the reduced computational cost by applying the proposed



(a)test image A (b)test image B

Figure 8. CCL execution time for test images and measured performance gains

Table 4. Performance analysis of GPU-based connected components labeling.

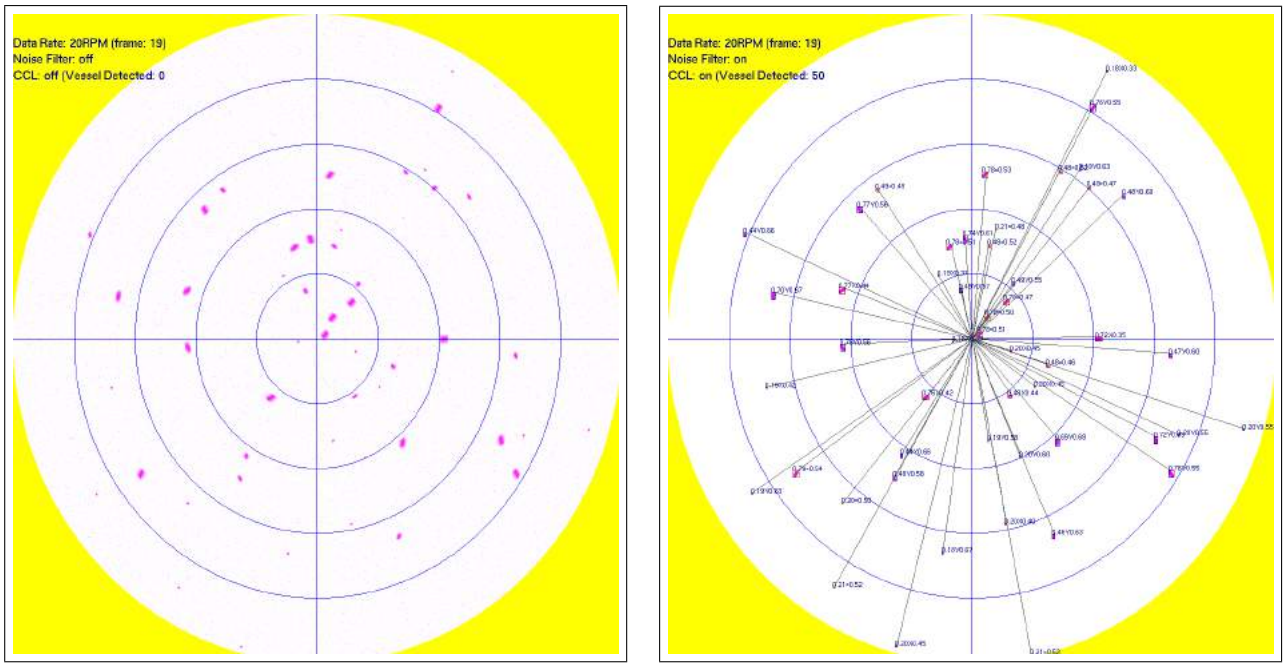
	Number of Connected Components	
Operations	50 connected components	1869 connected components
denoising	16.05 $msec$	16.05 $msec$
Preprocessing	16.05 $msec$	16.05 $msec$
label initialization	2.30$msec$	2.29$msec$
column-wise merge	3.59$msec$	3.63$msec$
hierarchical merge	7.97$msec$	8.35$msec$
relabel	3.07$msec$	3.17$msec$
CCL	16.93$msec$	17.44$msec$
counting components	2.12$msec$	2.16$msec$
bounding box computation	2.55$msec$	32.71$msec$
Bounding Box	4.67$msec$	34.87$msec$
Total Computation Time	21.60$msec$	52.31$msec$

method to the cost of the reference method. As shown in the table, the performance gain is more noticeable as the size of the input image increases.

Fig. 8 visually compares the computational costs of the reference method and our method (lines), and the performance gain is also visualized with bars. Fig. 8 (a) shows the result when the test image A was used as input image. The formance gain was the largest when the size of the input image is 2048×2048.

Fig. 8 (b) shows the similar results when the other test image is used. As shown in the figure, the performance gain is again the most noticeable when the size of the image is 2048×2048.

The method was applied to radar signal. In order to simulate the situation where a lot of moving vessels are detected by radar, we generated a radar signal as shown in Fig. 9 (a). The final images to be displayed has the size of 4096×4096. The input images were denoised and the proposed method was applied to those images. The connected components were successfully extracted as shown in Fig. 9 (b) in realtime. Fig. 10 (a) and (c) are zoom-in images of the result shown in Fig. 9 (a) and (b) respec-



(a) received radar signal (b) extracted connected commponents

Figure 9. The vessel tracking with proposed CCL techniques

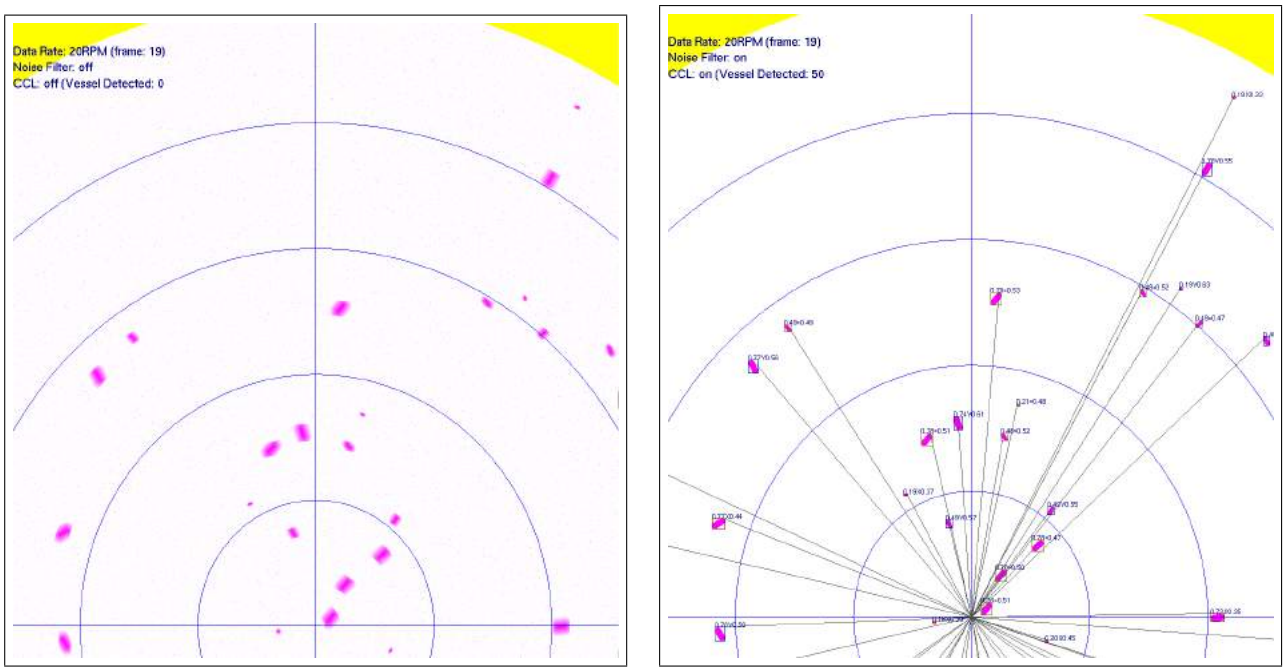
tively. As shown in the figure, the objects in the image were successfully labeled and tight bounding boxes wraping the components were obtained.

The performance analysis of GPU-based connected components labeling with radar signal is shown in Table 4. Each row shows the time required for each operations named in the first column in milliseconds. The operations can be categorized into three operation groups, and gray-colored rows shows the time consumed for those categories. We used two different radar signal data which have 50 and 1869 objects respectively.

As shown in the table, the computational costs of each operations do not rely on the number of connected components. Since the denoising process can be perfectly parallelized, the computational costs for both cases were exactly the same. Any preprocessing techniques can be applied in the phase, and the majority of image processing techniques can be easily parallelized. Even in the CCL process, any significant difference was not found in the aspect of computational costs. The CCL operation is composed of four subtasks. Each task required similar execution time for both cases. However, the bounding box computation required more time when we have more objects in the image because the generation of bounding boxes and determination of its properties requires computations of which complexity is proportional to the number of objects.

5 CONCLUSION

In this paper, an efficient GPGPU implementation of connected component labeling (CCL) was proposed. The method exploits the data parallelism of GPUs to improve the performance of CCL. Object identification in image data is the fundamental operation and rapid computation is highly requested as the sizes of the currently available image data rapidly increase. The experimental results show the proposed method can be a good solution to the object identification in large-scale image data. The CCL method proposed in this paper was successfully applied to realtime object tracking for radar signals. The computational cost of the proposed method does not hinder realtime visualization of radar signals.



(a) zoom-in image of input (b) zoom-in after CCL

Figure 10. The vessel tracking with proposed CCL techniques (Zoomed)

ACKNOWLEDGMENT

The research was supported by 'Software Convergence Technology Development Program', through the Ministry of Science, ICT and Future Planning (Development of Big Data Processing and Decision Support System for Offshore Maritime Safety, S0142-15-1014).

REFERENCES

[1] P. Chen, H. Zhao, C. Tao, and H. Sang. Block-run-based connected component labelling algorithm for gpgpu using shared memory. *Electronics letters*, 47(24):1309–1311, 2011.

[2] C. Grana, D. Borghesani, and R. Cucchiara. Connected component labeling techniques on modern architectures. In *International Conference on Image Analysis and Processing*, pages 816–824. Springer, 2009.

[3] C. Grana, D. Borghesani, and R. Cucchiara. Optimized block-based connected components labeling with decision trees. *IEEE Transactions on Image Processing*, 19(6):1596–1609, 2010.

[4] C. Harrison, H. Childs, and K. P. Gaither. Data-parallel mesh connected components labeling and analysis. In *Eurographics Parallel Graphics and Visualization Symposium, Llandudno, Wales*, 2012.

[5] K. A. Hawick, A. Leist, and D. P. Playne. Parallel graph component labelling with gpus and cuda. *Parallel Computing*, 36(12):655–678, 2010.

[6] O. Kalentev, A. Rai, S. Kemnitz, and R. Schneider. Connected component labeling on a 2d grid using cuda. *Journal of Parallel and Distributed Computing*, 71(4):615–620, 2011.

[7] M. Minsky and S. Papert. *Perceptrons*. MIT press, 1988.

[8] B. Preto, F. Birra, A. Lopes, and P. Medeiros. Object identification in binary tomographic images using gpgpus. *International Journal of Creative Interfaces and Computer Graphics (IJCICG)*, 4(2):40–56, 2013.

[9] O. Šťava and B. Beneš. Connected component labeling in cuda. *Hwu., WW (Ed.), GPU Computing Gems*, 2010.

[10] S. Zavalishin, I. Safonov, Y. Bekhtin, and I. Kurilin. Block equivalence algorithm for labeling 2d and 3d images on gpu. *Electronic Imaging*, 2016(2):1–7, 2016.

International Journal of New Computer Architectures and their Applications (IJNCAA) 6(4): 150 -155
The Society of Digital Information and Wireless Communications, 2016 ISSN 2220-9085 (Online); ISSN 2412-3587 (Print)

Development of a City Planning Simulation System Using Leap Motion in the AR Space

Toshiharu Katahira and Hiroki Imamura
Department of Information Systems Science, Graduate School of Engineering, Soka University
Mailing Address: 1-236, Tangi-machi, Hachioji-Shi, Tokyo, Japan 192-8577
E-mail: e15m5207@soka-u.jp, imamura@soka.ac.jp

ABSTRACT

Some simulation systems of the city planning in VR space have been proposed. For example, there are landscape simulation system and sunshine simulation systems. However, the conventional simulation system in VR space are limited in the 3DCG of buildings modeled in advance and don't assume that discussion of city planning with an architect in remote locations. In order to solve these issues, we have proposed methods that superimposition virtual models on actual models using AR and object recognition and using COMSAS. In this paper, we report the development and evaluation of a prototype of this city planning simulation system using Leap Motion in AR space which becomes a cornerstone of the proposed system. In this prototype system provides users intuitive manipulation of AR buildings.

KEYWORDS

Virtual Reality, Augmented Reality, City Planning, Simulation, Natural User Interface, Leap Motion, Unity.

1 INTRODUCTION

Recently, city renewal projects are carried out in various area in Japan because of decrepit buildings, anti-disaster and change of the inhabitant’s needs. A city planning, such as the city renewal project, advances through consultations between an architect and government officials. They develop an idea of the city project with either making an accurate model of the city or drawing the illustrations of it. Moreover, recently they use a 3DCG model of the city for landscape simulation in virtual reality (VR) space [1][2]. In addition, research on sunshine simulation [3] and wind simulation with visualizing wind [4] have been proceeded. However, the conventional city simulation systems in VR space are limited in the 3DCG of buildings prepared in advance. Furthermore, the conventional system does not assume that discussion of city planning in remote locations. As a solution to these issues, we use two methods. The first method is superimposition virtual models on actual models using Augmented Realty (AR) and object recognition. The second method is using a 3D video communication system in synchronized AR space (COMSAS) [5]. In COMSAS, multiple users at remote locations are reconstructed both each other in the AR space. Owing to these two methods, users can add information of texture and shadow to actual models. Furthermore, it is possible to share virtual models in a remote location. To implement the above system, in this study, we develop a city planning simulation system using Leap Motion [6] in AR space which becomes a cornerstone of the proposed system.

2 PROPOSED SYSTEM

2.1 Equipments in the Proposed System

2.1.1 Leap Motion

We describe about Leap Motion. Fig. 1 shows the appearance of the Leap Motion. The Leap Motion has optical sensors and infrared light. The sensors are directed along the y-axis upward when the controller is in its standard operating position and have a field of view of about 150 degrees. The effective range of the Leap Motion Controller extends from approximately 25 to 600 millimeters above the device. The Leap Motion can obtain information of user's hands (such as three-dimensional coordinates, postures, and gestures) and track user's hands. To realize an intuitive user interface, we use the Leap Motion Controller extends from approximately 25 to 600 millimeters above the device. The Leap Motion can obtain information of user's hands (such as three-dimensional coordinates, postures, and gestures)

Fig. 1: Leap Motion

Fig. 2: The structure of HMD

and track user's hands. To realize an intuitive user interface, we use the Leap Motion.

2.1.2 HMD and Stereo Camera

We describe an outline of the HMD and the stereo camera. We use an Oculus Rift CV1 [7] that is a headset as the HMD. The Oculus Rift CV1 has a stereoscopic OLED display, 1080* 1200 resolution per eye, a 90 Hz refresh rate, and 110 degrees' field of view. The Rift has full 6 degree of freedom rotational and positional tracking. This tracking is performed by a USB stationary infrared sensor that is picking up light that is emitted by IR LEDs. To convert the HMD from VR to AR, we combine Oculus Rift CV1 and Ovrvision Pro [8]. Fig. 2 shows the structure of combined these devises. Ovrvision Pro is a stereo camera that can obtain two RGB images. Ovrvision Pro captures video at up to 60 frames per second.

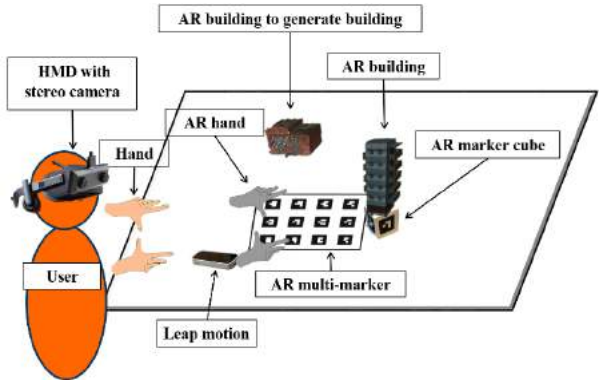


Fig. 3: The configuration of the prototype system

2.2 System Overview

Fig. 3 shows the configuration of the prototype system. The prototype system uses an Oculus Rift CV1 with Ovrvision, an AR multi-marker, an AR marker cube and a Leap Motion[7]. While the Leap Motion is detecting user's hands, the prototype system displays AR hands (means virtual objects represent user's hands). In addition, the prototype system uses Unity [9] to implement the physics simulation. By using collision decision program of Unity, the user can manipulate AR buildings (means virtual objects of buildings) with only the AR hands.

2.3 Processing of the Proposed System

Fig. 4 shows the flow of processing in this prototype system. The descriptions of each step are in following sections.

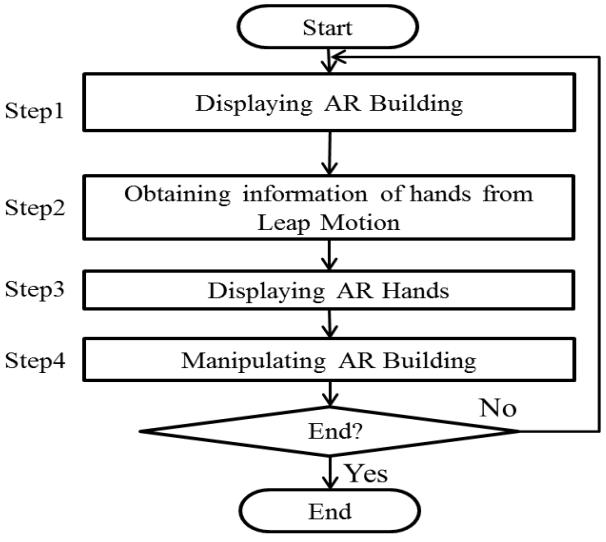


Fig. 4: The flow chart of this system

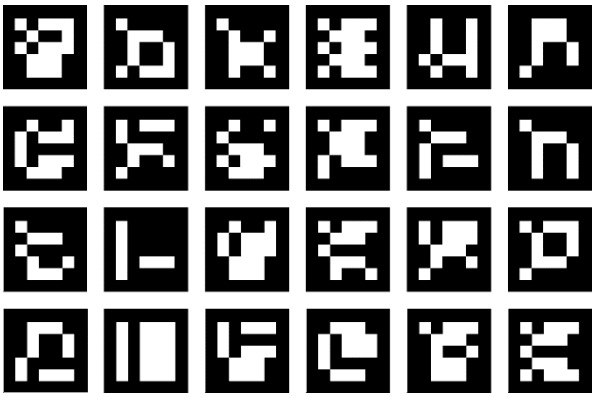


Fig. 5: AR multi-marker.

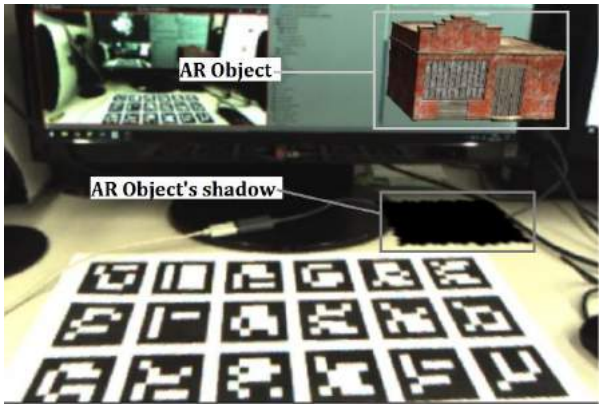


Fig. 6: The displayed AR object

2.3.1 Displaying of AR Buildings

The prototype system displays virtual models based on the position and the inclination of AR multi-maker. Fig. 5 shows the AR multi-marker. The center of AR multi-marker is origin of the virtual models, so that each virtual model is displayed based on the center. Thereby the user can observe the virtual models from various angles by moving own viewing position. In addition, AR multi-marker is robust to occlusion. The prototype system can recognize the AR multi-marker if only one or more markers of it are detected. Fig. 6 shows the displayed AR object. A shadow is drawn under the AR object by the directional light in function of Unity.

2.3.2 Obtaining Information of Hands from Leap Motion

The prototype system obtains information of user's hands from the Leap Motion. Out of those, positions of Fingertips, palms and pinch are needed to display AR hands, translate AR buildings, and change the scale AR buildings.

2.3.3 Displaying AR Hands

While the Leap Motion is detecting hands, the prototype system displays AR hands based on obtained user's hands information. Fig. 7 shows AR hands.

2.3.4 Manipulating AR Buildings

A collision detecting function between virtual objects is implemented as basic function of Unity. Based on that function, the users can touch and manipulate virtual objects with the AR hands. In following section, we describe 5 manipulations implemented in the prototype system.

2.3.4.1 Generating of AR Buildings

When the user touches a virtual model to generate building with the tip of the forefinger of the AR hand (Fig. 8-a), a new virtual model is generated (Fig. 8-b).

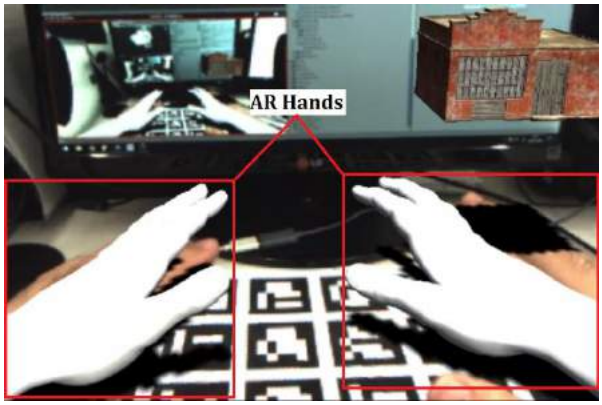
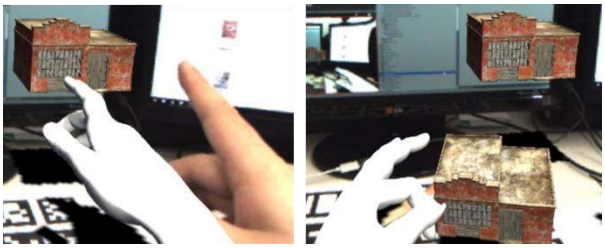
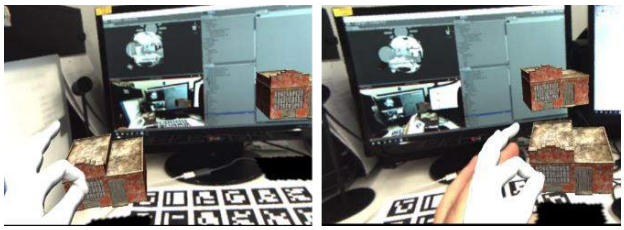


Fig. 7: Displaying the AR hands



(a)Before generate (b) After generate

Fig. 8: Generating AR building



(a) Before translation (b) After translation

Fig. 9: Translating AR building

2.3.4.2 Translating of AR Buildings

The user can pick up an AR building and translate it by pinching with thumb and forefinger of the AR hand. When the user releases the AR building, it falls to ground plane following the gravity. Fig. 9 shows this manipulation. *T*, a moving amount of the AR object, is determined by using

$$T = \begin{pmatrix} x'_a - (x_a - x_0) \\ y'_a - (y_a - y_0) \\ z'_a - (z_a - z_0) \end{pmatrix}, \qquad (1)$$

where (x_a, y_a, z_a) is a picking position, (x'_a, y'_a, z'_a) is a position after movement and (x_0, y_0, z_0) is a position before movement of the AR hands.

2.3.4.3 Rotating of AR Buildings

When the user picks up an AR building, the user can rotate the AR building with user' s palm of pinching hand. Fig. 10 shows outline in calculation of the rotation angle of the palm. Firstly, as shown in Fig. 10-a, the proposed system calculates the normal vector $\vec{P}_n$ to the palm from the function of Leap motion when the time is t-1. Secondly, as shown in Fig. 10-b, the proposed system calculates the normal vector $\vec{P}'_n$ to the palm from the function of Leap motion when the time is t. Finally, to calculate the rotation angle of the hand, the proposed system converts 3-dimensional vectors $\vec{P}_n$ and $\vec{P}'_n$ into 2-dimensional vectors P of a X-Z plane and P' respectively and calculates the rotation angle θ of the hand by using

$$\theta = \sin^{-1} \frac{P \cdot P'}{|P| \cdot |P'|}. \qquad (2)$$

Fig. 11 shows this manipulation.

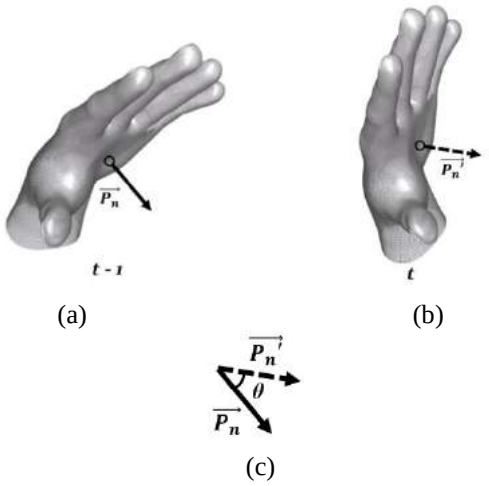


Fig. 10: Calculation the angle of rotation of the palm

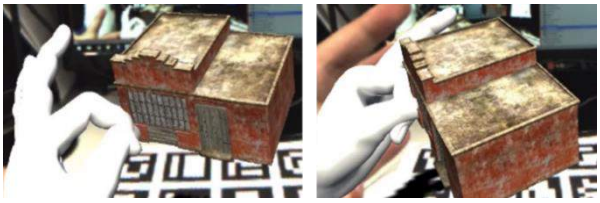


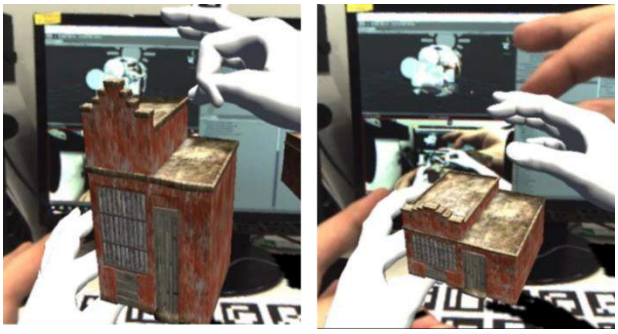
Fig. 11: Rotate AR building

2.3.4.4 Changing Scale of AR Buildings

In a state of picking AR building, the user can change the scale of AR building such as expanding (Fig. 12-a, 13-a, 14-a) and shrinking (Fig. 12-b, 13-b, 14-b) a rubber band with both hands. The scale of AR buildings can be changed in each xyz-axis direction.

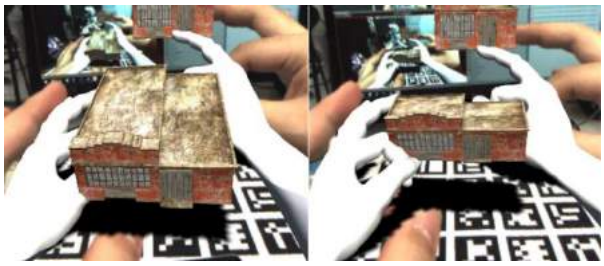
(a) Scale up (b) Scale down

Fig. 12: Changing scale AR building in x axis

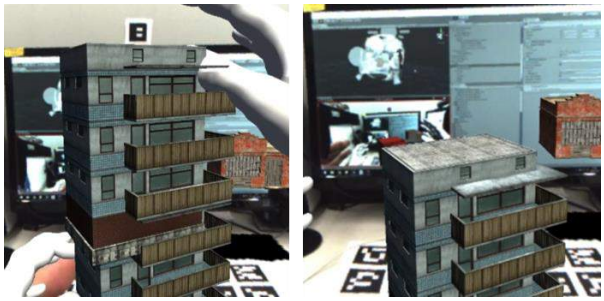


(a) Scale up (b) Scale down

Fig. 13: Changing scale AR building in y axis



(a) Scale up (b) Scale down
Fig. 14: Changing scale AR building in z axis



(a) Increasing story (b) Decreasing story
Fig. 15: Changing Story of AR Buildings

The positional relationship between both AR hands determines which scale of axis will be changed. The change amount of the scale of AR buildings is determined by varying of the distance of AR hands. When the distance of AR hands at the picked moment is larger than the distance of AR hands at the current moment, the AR building is expanding. When the distance of AR hands at the picked moment is smaller than the distance of AR hands at the current moment, the AR building is shrinking. *S*, a change amount of the scale of AR buildings, is determined by using

$$S = \begin{pmatrix} |(x'_{l2} - x'_{r1}) - (x_{l2} - x_{r1})| \\ |(y'_{l2} - y'_{r1}) - (y_{l2} - y_{r1})| \\ |(z'_{l2} - z'_{r1}) - (z_{l2} - z_{r1})| \end{pmatrix}, \qquad (3)$$

where (x_{r1}, y_{r1}, z_{r1}) is position of right AR hand picking a AR building,(x_{l2}, y_{l2}, z_{l2}) is position of left AR hand picking the same AR building, $(x'_{r1}, y'_{r1}, z'_{r1})$ and $(x'_{l2}, y'_{l2}, z'_{l2})$ are position of after movement of each right and left AR hand.

2.3.4.5 Changing Story of AR Buildings

A displayed AR building on the AR marker cube can be changed the story of it, such as in Fig. 15-a and Fig. 15-b. When user performs expanding manipulation in height direction, the same AR building is added on the top of the AR building. Conversely, when the user performs shrinking manipulation in height direction, the added AR building on the top is deleted.

3 EVALUATION EXPERIMENTS

3.1 Outline of the Experiment

We had an evaluation experiment on the proto type system. We evaluated whether the implemented function behave correctly and this system is intuitive or not. A user wears a HMD with a Ovrvision on and manipulates AR hands with his/her hands above the Leap Motion, and then, the user performs manipulation of translating and changing story of AR buildings. 10 subjects performed the experiment. After that, they evaluated following items. 10 people performed the experiment. After that, they evaluated following items.

① Could you pick up AR buildings?
② Could you translate AR buildings?
③ Could you rotate AR buildings?
④ Could you expand AR buildings?
⑤ Could you shrink AR buildings?
⑥ Could you increase story of AR building on the AR marker cube?
⑦ Could you decrease story of AR building on the AR marker cube?
⑧ Did you feel this system is intuitive?

Each evaluation is 5 levels as the highest score is 5.

3.2 Results of Experiments

Table 1 shows the result of the evaluation experiment. Average scores from ① to ③ are higher than 4.0. Among all the items, items ① and ② has the highest average score and the lowest standard deviation, conversely, the item ⑦ has the lowest average score and the highest standard deviation.

3.3 Discussion of Experiments

Average scores of items ⑥ and ⑦ are lower than 3.5 and the standard deviation is higher than 1.0.

Table 1. Evaluation experiments

No.	①	②	③	④	⑤	⑥	⑦	⑧
Average Score	4.60	4.60	4.10	3.70	3.70	3.30	3.10	4.00
Standard Deviation	0.66	0.66	0.70	0.46	0.46	1.10	1.22	0.77

It is considered that the cause is unstable displaying the AR building on the AR marker cube in manipulation of the hanging the story. An area of AR marker cube visible with a Ovrvision is smaller than the area of AR multi-marker, therefore, the AR marker cube is easy to be invisible with the Ovrvision by hands of the subject in manipulation of changing the story. Accordingly, it is seeming to be difficult for the subject to perform manipulation of changing the story. In order to solve this problem, we will implement a method of object recognition with 3D sensor. The average score of the item ⑧ is higher than 4.0 and the standard deviation of that is lower than 0.8, therefore, it can be said that most subjects feel that the prototype system is highly intuitive.

4 CONCLUSION

In this paper, we showed the outline of the city planning simulation system using a Leap motion, Ovrvision and Oculus Rift CV1. Moreover, we implemented and evaluated the intuitive manipulations of AR buildings. As the result, we found the prototype system is intuitive. However, we found a problem of the changing story of AR building displayed on AR marker cube. In the future, we will solve this problem by implementing shape recognition with 3D sensor. As future work, we will try to implement shape recognition with 3D sensor. In addition, we will try to implement sunshine simulation and wind simulation. Moreover, we will consult an expert on the city planning and determine the finally function of this system.

REFERENCES

[1] Construction of the creation and design consultation system of vision using landscape simulation in around Kashiwa Station, http://www.mlit.go.jp/common/000112023.pdf [Accessed 12 January 2017]

[2] KAWAI Yasuo, SANO Masami, IKEDA Takeshi, and MASUOKA Ryo, “Landscape simulation system using a game engine for landscape planning” Summaries of technical papers of annual meeting 2013, 93-94, 2013-08-30

[3] Noriyoshi Fujisawa, Takamasa Miyazaki, and Kiminori Nakazawa “A Study on Landscape Architecture Design using BIM and GIS” Architectural Institute of Japan, Vol. 21, No. 47, pp.355-360, 2015

[4] Visualization by stereoscopic vision based on VR technology, http://www.civil.chuou.ac.jp/lab/keisan/vr1.html [Accessed 12 January 2017]

[5] Hideyuki Hashimoto, Yuki Fujibayashi and Hiroki Imamura, "3D Video Communication System by Using Kinect and Head Mounted Displays", International Journal of Image and Graphics, Vol.15, No.2, pp.1540003-1-1540003-12, 2015

[6] Leap Motion. Inc. https://developer.leapmotion.com/ [Accessed 12 January 2017]

[7] Oculus Rift CV1. https://www3.oculus.com/en-us/rift/ [Accessed 12 January 2017].

[8] Ovrvision Pro. http://ovrvision.com/ [Accessed 12 January 2017].

[9] Unity Technologies. http://unity3d.com/jp/unity [Accessed 12 January 2017]

International Journal of
NEW COMPUTER ARCHITECTURES AND THEIR APPLICATIONS

The *International Journal of New Computer Architectures and Their Applications* aims to provide a forum for scientists, engineers, and practitioners to present their latest research results, ideas, developments and applications in the field of computer architectures, information technology, and mobile technologies. The IJNCAA is published four times a year and accepts three types of papers as follows:

1. **Research papers**: that are presenting and discussing the latest, and the most profound research results in the scope of IJNCAA. Papers should describe new contributions in the scope of IJNCAA and support claims of novelty with citations to the relevant literature.
2. **Technical papers**: that are establishing meaningful forum between practitioners and researchers with useful solutions in various fields of digital security and forensics. It includes all kinds of practical applications, which covers principles, projects, missions, techniques, tools, methods, processes etc.
3. **Review papers**: that are critically analyzing past and current research trends in the field.

Manuscripts submitted to IJNCAA **should not be previously published or be under review** by any other publication. Plagiarism is a serious academic offense and will not be tolerated in any sort! Any case of plagiarism would lead to life-time abundance of all authors for publishing in any of our journals or conferences.

Original unpublished manuscripts are solicited in the following areas including but not limited to:

- Computer Architectures
- Parallel and Distributed Systems
- Storage Management
- Microprocessors and Microsystems
- Communications Management
- Reliability
- VLSI