J. Chosun Natural Sci.
Vol. 12, No. 4 (2019) pp. 105 – 115
https://doi.org/10.13160/ricns.2019.12.4.105

Smart Control System Using Fuzzy and Neural Network Prediction System

Tae Yeun Kim[1] and Sang Hyun Bae[2†]

Abstract

In this paper, a prediction system is proposed to control the brightness of smart street lamps by predicting the moving path through the reduction of consumption power and information of pedestrian's past moving direction while meeting the function of existing smart street lamps. The brightness of smart street lamps is adjusted by utilizing the walk tracking vector and soft hand-off characteristics obtained through the motion sensing sensor of smart street lamps. In addition, the motion vector is used to analyze and predict the pedestrian path, and the GPU is used for high-speed computation. Pedestrians were detected using adaptive Gaussian mixing, weighted difference imaging, and motion vectors, and motions of pedestrians were analyzed using the extracted motion vectors. The preprocessing process using linear interpolation is performed to improve the performance of the proposed prediction system. Fuzzy prediction system and neural network prediction system are designed in parallel to improve efficiency and rough set is used for error correction.

Keywords : Smart Control System, Fuzzy Prediction System, Neural Network Prediction System, Motion Vector, Rough Set

1. Introduction

The purpose of street lamps is to maintain the visual abilities of pedestrians and drivers the same as the daytime so as to relieve traffic safety, prevent crime, relieve anxiety of road users, reduce fatigue, and improve efficiency of road use. Street lamps are also important public utilities and consume 40 percent of the city's energy budget[1]. Therefore, smart LED street lamps are an important area in the smart city field. Smart street lamps are systems that reduce energy and operating costs by replacing conventional street lamps with LED-based lamps and install sensors and network devices to enable administrators to access and control information about street lamps over the Internet. In other words, it is a system that can grasp the state and surrounding situation of all street lamps in real time and control it according to the situation[2].

As such, smart street lamps can adjust the brightness of street lamps according to the traffic volume of people, vehicles, etc. and can identify the entire street lamp situation in real time, reduce the manpower and cost required to maintain the lamp, eliminate lamp lights that are turned on at unnecessary times, and operate it in time and place. In addition, it is expected that more than just lighting can be generated through its role in enhancing efficiency of operation, management, etc. as well as providing infrastructure for a number of smart city services, including smart parking meters, traffic lights and traffic management systems.

However, the existing smart street lamps do not reveal the path of pedestrians in advance, unlike the existing street lamp environment, where all streetlights are always on, because the street lamps operate only when they are detected passing by a pedestrian through an attached. Therefore, it is necessary to predict the moving path through the reduction of power consumption and information on the past moving direction of pedestrians while satisfying the functions of the existing street lamps.

In this paper, a system to locate pedestrians through motion sensing sensor of smart street lamp, obtain motion vector based on past movement direction information, and predict pedestrian path through route prediction system to adjust brightness of smart street lamp.

[1]SW Convergence Education Institute, Chosun University, Gwangju
[2]Department of Computer Science & Statistics, Chosun University, Gwangju

†Corresponding author : shbae@chosun.ac.kr
(Received : November 29, 2019, Revised : December 4, 2019, Accepted : December 17, 2019)

2. System Configuration and Design

2.1. Smart Street Lamp System

Smart street lamp-based route prediction system is composed of IoT sensor lighting controller, gateway, monitoring control server, and path prediction system[3]. IoT sensor lighting controller is designed to work with lighting controller and switched mode power supply for dimming LED (DALI, 1-10V, etc.) street lamps, and it is designed so that two-way communication can be made using gateway and mesh ZIGBEE communication. There are many communication obstacles on the road, and the shape of the obstacles also varies. These obstacles make communication difficult and unstable. In order to overcome this problem, this paper constructs a mesh network[4-6].

In the case of the gateway, brightness is controlled and controlled by the surveillance control server by configuring about 200 lighting controllers as a set. The street lamp with the motion detection sensor transmits the measurement information to the gateway, and the measured information is transmitted to the monitoring and control server through CDMA communication. The monitoring and control server identifies pedestrian locations based on measurement information, and commands the gateway to increase the brightness of the light if there is a pedestrian, otherwise lower the brightness. Upon receiving the command, the gateway adjusts the brightness of the street lamp within its control range according to the command of the monitoring control server.

2.2. Pedestrian Tracking Vector

The existing smart street lamp system was operated by raising the brightness of the street lamps when pedestrians are detected through motion sensors attached to each street lamp. However, in this paper, it predicts the pedestrian’s path in advance based on the motion sensor attached to the street lamp and operates the method to raise the brightness of the next street lamp in advance. Fig. 1 shows the motion sensor detection range and pedestrian tracking vector.

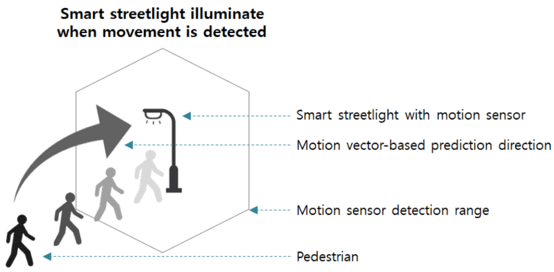


Fig. 1. Motion sensor detection range and pedestrian tracking vector.

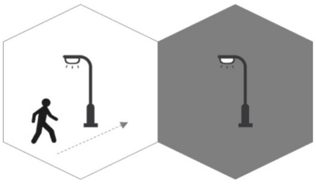
A street lamp motion sensor is attached to detect the pedestrian's movement and its direction. The street lamps are positioned such that the detection range of the motion sensors overlaps with each other. When the pedestrian is detected by the motion detection sensor, a walking tracking vector is obtained based on the position of the pedestrian's movement. By dividing this by time unit, a motion vector is obtained and the most recent motion vector is predicted by the next prediction path[7]. The unit of time is a human’s average walking speed is 4 km per hour, so a motion vector is obtained at 1 second intervals assuming about 1 m per second[8].

This calculation allows the measurement of pedestrian movement coordinates at intervals of about 1m. Based on this, if the predicted movement path is within the range of motion detection of a street lamp, only the brightness value of the street lamp to which the sensor is attached is increased. If the predicted movement path reaches the motion-sensing sensor detection overlay area, a brightness transition between motion-sensing sensors is required. For this reason, soft hand-off between street lamps is used.

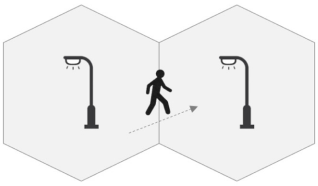
Soft hand-off is a continuous call, even though the device moves between the base station and the base station during the call, to perform a hand-off, which is characterized by keeping the traffic channel of the previous base station connected until the terminal is fully transferred to the new base station[9]. It is called Connect before Break. Soft hand-offs are intended for hand-offs with the help of base stations, unlike using hand-offs in standby mode.

In this paper, the soft hand-off characteristic is applied to the street lamp. In other words, it predicts the next base station location based on the past base station information, so that the next street lamp can be revealed in advance based on the past street lamp information as if the soft hand-off is performed.

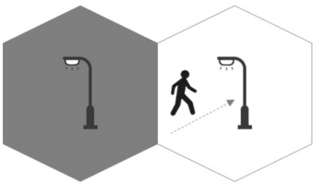
If the predicted moving path is outside the detection range of one motion sensor and reaches the detection range of the new sensor, the brightness value of the



(a) The brightness change of smart street lamp when the pedestrian tracking vector does not reach the overlapped sensing range of sensor.



(b) The brightness change of the smart street lamp when the pedestrian tracking vector reaches the overlapped sensing range of sensor.



(c) The brightness change of smart street lamp when the pedestrian tracking vector does not reach the overlapped sensing range of sensor.

Fig. 2. The brightness change of smart street lamp according to the pedestrian tracking vector.

street lamp to which the new motion sensor is attached is increased and the brightness value of the street lamp to which the previous motion sensor is attached is lowered.

Fig. 2 shows the variation in the brightness of the street lamps according to the walking tracking vector. If the walking tracking vector does not fall within the sensor's nested detection range, as shown in (a), increase the brightness value of the lamp in the sensing range only. If the distance between the new motion detection sensor and the walking tracking vector is closer than the distance between the previous motion sensing sensor and the walking tracking vector in the overlapping area as shown in (b), the operation is performed in the following order. First, lower the brightness value of the street lamp with the previous motion sensor, and then increase the brightness value of the street lamp with the new motion sensor. Finally, if the connection with a street lamp with a previous motion-sensing sensor is severed, the old street lamp becomes dark and the new street lamp becomes brighter, as shown in (c).

2.3. Motion Vector

The motion vector is used for the path analysis and prediction of the pedestrian, and the GPU is used for the high-speed computation. First of all, using adaptive Gaussian mixing technique from the foreground, weighted differential imaging technique for prominent motion, and motion vector to detect pedestrians and analyze pedestrian movements using extracted motion vectors[10].

Among the techniques for detecting moving objects in an image in real time, optical flow and background modeling are widely used. The optical flow can detect objects that move independently without information about the background. However, it requires a lot of computations and needs the help of hardware. Therefore, weight difference imaging, adaptive Gaussian mix models, and motion vectors were used to use motion detection techniques that can be implemented in software without the help of hardware.

The use of these extracted objects has been largely limited to three cases: {Active, Inactive}, {Position moving, Fixed moving}, and {Walking, and Running}. As such, feature parameters are used to distinguish each of these three steps[11].

2.3.1. Motion Vector Extraction Method

As a method to separate the objects with prominent movement from the background, the object is detected through the difference image adaptive to the background model generated by the Gaussian mixture model. The proposed system retained the value of motion vectors from the Enhanced Predictive Zonal Search (EPZS) function as shown in Table 1[12].

The functions used in this function are mx and my, which are the values of the motion vectors for x and y.

Table. 1. EPZD Function

```
simpleme_epzs_motion_search(s, 0, &mx, &my, P,
pred_x, pred_y, rel_xmin, rel_ ymin, rel_xmax,
rel_ymax, s->p_mv_table, (1<<16)>>shift, mv_penalty);
```

In the case of high resolution, a large amount of computation is required to extract the motion vector. In this case, the GPU is used to reduce the CPU load.

If the variable where the motion vector is stored is $psMV_Z$, it can be represented as a double array of $psMV_Z[X][Y]$. where is shown in equation (1).

$$psMV_Z = \sqrt{MV_X^2 + MV_Y^2} \quad (1)$$

When the motion vector is extracted at various resolutions for the image of the pedestrian walking, as the resolution increases, the number of motion vectors increases. The extracted vector values can be used for various analysis.

2.3.2. Leverage GPU

GPU is used to speed up the large amount of computations required by computer graphics. Because GPU has more ALU than CPU, it processes data in parallel. All threads of the GPU execute the same sequential code but operate in the SIMD method that performs operations with different data. The proposed algorithm was designed to fit the GPU parallel processing structure and implemented using CUDA C.

In CUDA, the program execution unit is threads and provides the functions of blocks and grids to efficiently manage and execute multiple threads. In CUDA, threads are gathered into blocks and blocks are gathered into a grid. This is called a grid block model. Parallel processing requires allocating the appropriate number of threads and blocks per block according to the kernel program.

To generate a single image, a two-dimensional index is used. Therefore, each thread will have a specific (x, y) index and can be easily accessed to a location corresponding to one pixel in the output image.

In order to increase the efficiency of the program, the number of blocks and threads is changed accordingly according to the input resolution. The number of threads cannot exceed 512. In addition, the number of threads a block has was designated 128, 256, and 512, because the Streaming Multiprocessor is supposed to operate in 32 multiple units[13]. Fig. 3 shows how the algorithms proposed in this paper can be utilized in GPU and CPU.

As shown in Fig. 3, the GPU is used to process a large amount of motion vector, adaptive Gaussian mixed background modeling, and a difference image using significant motion. The part that analyzes the motion using the extracted motion vector is processed by the CPU due to the small amount of computation.

2.3.3. Analysis of Pedestrian Movement

The pedestrian movement analysis was limited to three cases: {Active, Inactive}, {Position moving, Fixed moving}, and {Walking, Running}.

Fig. 4 shows the motion analysis method of the pedestrian using the motion vector, which is processed by the CPU using the values of the motion vector extracted from the GPU. Since the experiments were performed at various resolutions, the reference values had to be changed slightly for each resolution.

After separating the object from the low background, determine whether the pedestrian is moving or not. If there is no pedestrian movement, it is not necessary to recognize the movement.

In the first step, the frequency of occurrence of the entire motion vector is examined to determine {Active, InActive}. If the foreground object has moved above a certain threshold, it can indicate that the pedestrian is moving and that it is active. That is, if $psMV_Z$ detects

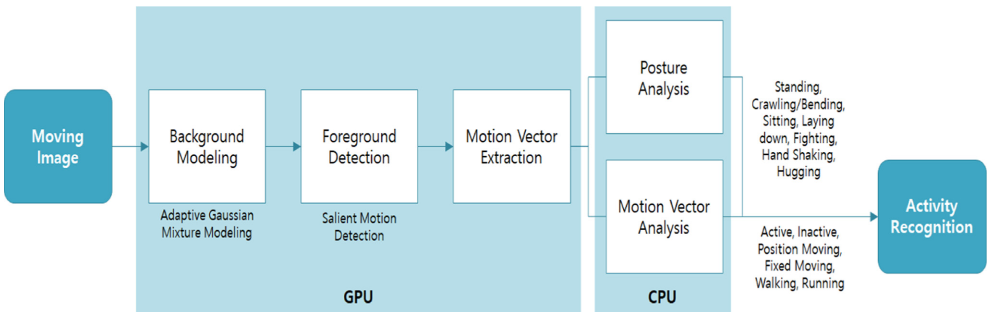


Fig. 3. The method GPU utilization.

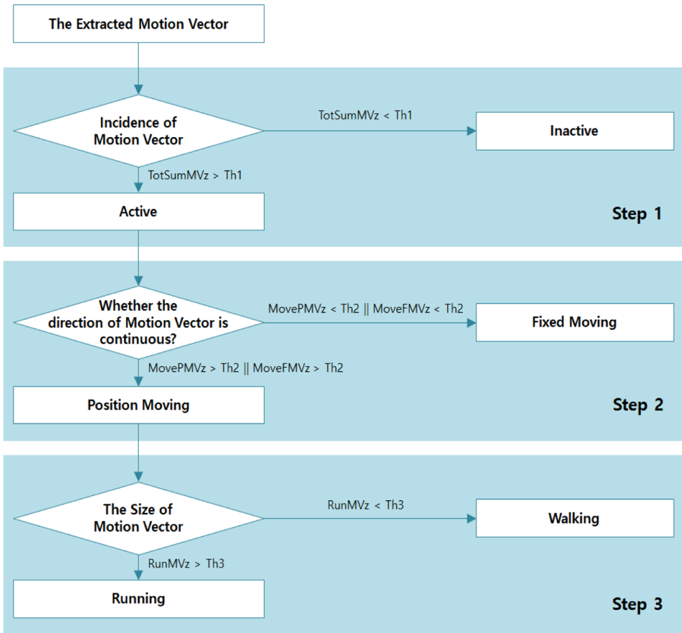


Fig. 4. The pedestrian movement analysis method.

a value greater than 0 and detects a motion vector above a certain threshold, it is determined to be Active and proceeds to the next step. If not, the object is regarded as a stationary object and belongs to the background.

Where $TotSumMV_Z$ represents the sum of cases where $psMV_Z$ is greater than 0. The threshold value Th1 for this step may be determined fluidly, but at least three values must be given. In this step, it is only to judge activity or not.

In the second step, {Position Moving, Fixed Moving} is classified for the foreground object determined to be active in the first step classification. In this step, the directionality of the motion vector is checked.

Check whether the distribution of MV_Z values appears continuously from top / bottom / right / left. For example, if the distribution of values is more than three consecutive from right to left, from two to three, this is determined by Position Moving that a pedestrian is walking or running. Otherwise, it is determined by Fixed Moving.

In the third step, {Walking, Running} is classified for the object determined as Position Moving in the above two-stage classification. The magnitude of the motion vector is examined and used. Examine the case where the value of MV_Z has a large number of 20 or more.

2.4. Smart Street lamp Based Path Prediction System

To enhance the performance of the proposed smart street lamp-based route prediction system using data obtained through motion vectors, a rough set for preprocessing processes, fuzzy systems, neural network systems and error correction models were used[14].

Data preprocessing methods include Moving Average (MA), Integrated Moving Average (IMA), and Linear Interpolation method. In this paper, preprocessed through linear interpolation method. Generate an appropriate number of differential data with nonlinear time series data characteristics and minimize incomplete elements that can occur when designing a system model if the characteristics of the generated differential data can well represent the characteristics of the circular data. Therefore, the linear interpolation method is proposed to expect the learning effect of insufficient data, and this is the first step to enable accurate prediction by processing the data[15,16].

The first order fuzzy prediction system is used as a predictor to generate interpolation input data. The data value to be predicted is unknown. Therefore, this means that the value of the data to be inserted in the preprocessing cannot be generated and is a step to solve this.

The secondary prediction system has been designed in parallel with fuzzy prediction systems and neural network prediction systems. Fuzzy prediction systems tend to have good predictive performance for typical load patterns in full load forecasting, but they do not predict sudden increases or decreases in data. In addition, the neural network prediction system has a lower performance than the fuzzy prediction system in comparison with the fuzzy prediction system, but shows better performance than the fuzzy prediction system in case of sudden increase or decrease of data. Therefore, it was designed in parallel to use the advantages of both prediction systems.

The rough set is used to select the error correction model based on the results of the two secondary prediction systems. The processing of the system was simplified by reducing attributes and eliminating unnecessary rules through data analysis of the two secondary prediction system result values to be used according to the resulting values.

2.4.1. Data Preprocessing

In general, the performance of the prediction system shows a close relationship to the learning data used for the design of the system. Too much variability can result in too much learning data being inconsistent and unnecessary. Also, less training data may not contain the information required by the system. Therefore, this paper preprocesses the data by applying linear interpolation which can contain more information method in a relatively small amount of data.

If the data are given as shown in $x(1)$, $x(2)$,..., $x(t)$,..., (N) $(t = 1...n)$, the interpolation interval between data is defined as if the number of data to be used for interpolation is . Therefore, in this paper, the interpolation data can be defined as follows because the average value of each data is defined as interpolation value.

$$m(t)1 = \frac{1}{\Delta}(x(t)+x(t+1)) \tag{2}$$

Here is $t = 1...N$ and $1 = 1{:}L$.

The TSK Fuzzy Prediction System and Neural Network Forecasting System used in this paper are generally based on the regression model (AR) and can produce good results for relatively three inputs. Therefore, in this paper, since the regression model using three input data is used, Equation (2) can be expressed as the following form.

$$m(t)1 = \frac{1}{4}(x(t)+x(t+1)) \tag{3}$$

Here, it will be given as $1 = 1,2,3$.

Thus, re-representation of observed data into interpolated data structures can be expressed as shown in equation (3).

2.4.2 Fuzzy Prediction System

In this paper, two types of prediction systems are used. First, a fuzzy prediction system is used to predict data in a circular form using the portion of the forecast system's input data and the input data generated. In this paper, the predicted input value found by the previous input prediction is used as input, and the input/output data structure is transformed as shown in equation (4).

$$\hat{D} = \begin{bmatrix} \widehat{m}_{(1)1} & \widehat{m}_{(1)2} & \widehat{m}_{(1)3} \\ \vdots & \vdots & \vdots \\ \widehat{m}_{(t-1)1} & \widehat{m}_{(t-1)2} & \widehat{m}_{(t-1)3} \\ \vdots & \vdots & \vdots \\ \widehat{m}_{(N-1)1} & \widehat{m}_{(N-1)2} & \widehat{m}_{(N-3)1} \end{bmatrix}, \; Y = \begin{bmatrix} x_{(2)} \\ \vdots \\ x_{(t)} \\ \vdots \\ x_{(N)} \end{bmatrix} \tag{4}$$

Where $\hat{D}$ is the interpolated data value predicted by the input data prediction system and is the actual given data values corresponding to these inputs. Therefore, the structural change of the input/output data can be expressed as in Equation (5).

$$R^j\text{: IF } \widehat{m}_{(t-1)1} \text{is } F_1 \text{and } \widehat{m}_{(t-1)2} \text{is } F_2 \text{ and } \widehat{m}_{(t-1)3} \text{is } F_3$$
$$THEN\ x_{(t)=}\ a_0 + a_1\widehat{m}_{(t-1)1} + a_1\widehat{m}_{(t-1)2} + a_1\widehat{m}_{(t-1)3} \tag{5}$$

In addition, the output equation of the fuzzy prediction system based on the prediction input is shown in Equation (6).

$$\hat{x}_{(t)} = \frac{\sum_{j=1}^{M} f^{j\hat{x}^j_{(t)}}}{\sum_{j=1}^{M} f^j} = \frac{\sum_{j=1}^{M}(\hat{a}_0^j + \hat{a}_1^j\widehat{m}_{(t-1)1} + \hat{a}_2^j\widehat{m}_{(t-1)2} + \hat{a}_3^j\widehat{m}_{(t-1)3}}{\sum_{j=1}^{M} f^j} \tag{6}$$

The rule-base and parameters for obtaining fuzzy power allow the previously designed values to be applied to avoid the complexity of the computation. However, only the ignition strength is renewed according to the input data pair.

Consequently, the proposed prediction method can be viewed as an architecture in which the three interpolation values before t-point are predicted first, and then the values at t-point are used to predict the values. The input/output data pairs of Equation (6) can also be used in the neural network prediction system performed later.

2.4.3. Neural Network Prediction System

In order to utilize the neural network circuit, a part of the second prediction was executed using the time series analysis app of the neural network tool box provided by MATLAB. The neural network tool box provided by MATLAB passes a total of seven steps to extract the output.

Fig. 5 shows the sequence diagram of Step 7.

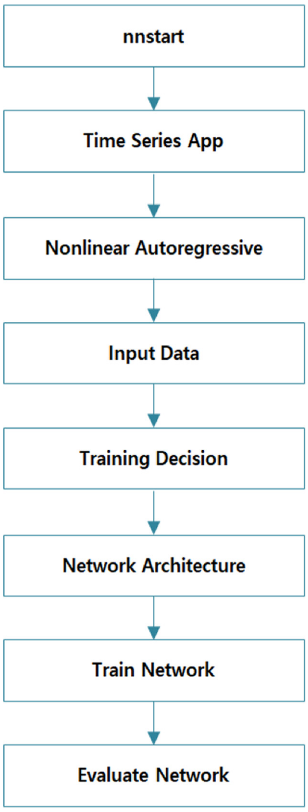


Fig. 5. MATLAB tool box architecture of neural network.

Step 1 is the selection of which applications MATLAB will use. The application required in this paper was used as a predictor in Time series app.

Step 2 provides a total of three prediction system models as part of the selection of models for the forecasting system. Among them, a nonlinear autoregressive (NAR) model was selected that fits the current time series prediction data.

Step 3 should be set up for that part because it inserts a portion of the processed data as part of inserting inputs into the prediction system and is recognized as a matrix.

Step 4 is to set the number of training, validation, and testing of input data, which can be set to percent rather than specific number of settings.

Step 5 is to set the number of hidden layers and the number of Time Delays used to predict the structure of neural network circuits.

Step 6 is the part set up for the learning method. Learning methods include Levenberg-Marquardt, Baysian Regularization, and Scaled conjugate Gradient. Among them, Levenberg-Marquardt, the same concept of back propagation, was used for learning.

Step 7 is a step to evaluate the deployed neural network, which is to verify the performance of the deployed neural network by inserting the rest of the data to validate the learned system and can extract the predicted result values[17,18].

2.4.4. Selecting a Prediction Model Using a Rough Set

Model selection for prediction systems is a necessary structure to take advantage of both fuzzy prediction systems and neural network prediction systems. Therefore, the performance of the proposed system is proved by presenting the final error correction result by selecting the model of the better prediction system when correcting the error of arbitrary value.

The system proposed in this paper has two result values predicted through fuzzy and neural network prediction systems in internal output results. In general, the final output of multiple prediction systems is the simplest way to use the output of the best prediction systems in the training process, the way to use the output averages of multiple prediction systems, and the method to take weighted averages according to performance.

The rough set theory is used to select and drive the prediction system to perform the prediction according to the properties of the input data. In order to obtain the effective rule base of the reduced form using the rough set theory, it is necessary to first define the attributes and attribute values of the conditional and decision units[19].

Accordingly, conditional attributes of the decision table are given three attributes corresponding to the input data, and the attributes of the decision section use the results of fuzzy and neural network predictors as attributes. The scope for the conversion of conditional attribute values given as data is grouped into four clusters by the K-mean clustering technique, and the data contained in each cluster is represented by the number of the cluster. For this purpose, the center values of clusters to be searched can be defined as three, and if the data are classified by the boundary of the cluster, four kinds of classification groups can be obtained.

After all data classification, the data in the range smaller than the first center value is converted to the attribute value as 1, and the data in the range between the first and second center is converted to 2. Input data located in the range between the second center and the third center is converted to the attribute value 3, and input data located in the range larger than the third center value is converted to the attribute value 4. In addition, if a rule is expressed by searching for and removing unnecessary attributes and using attributes necessary for system selection among the remaining attributes, it is possible to select an effective prediction system with fewer rules.

As shown in Table 2, when the first attribute value of the conditional attribute of the input data set is 2 and the second attribute value is 3, the neural network prediction system performs prediction according to rule 7 regardless of whether the third attribute value is present. As such, when the conditional attribute values are determined by all the input data, the corresponding prediction system performs the prediction.

Table 2. Final rule for the system selection

	Conditional attribute			Decision-making attribute
				Prediction system
1	2	2	2	FP
2		(3)	2	
3	3		(4)	
4	3	3		
5	4	2	3	
6	(1)		2	NP
7	(2)	3		
8	3	(2)		
9		4	(3)	

3. Performance Evaluation of System

The motion vector is used to analyze and predict the pedestrian path, and the experiment was performed using Visual Studio 2017, OpenCV 4.0.1 version to evaluate the GPU performance for high speed computation. The CPU uses Intel i7-7700K 4.2GHz and GPU uses NVIDIA GTX 1060. The same images were tested after changing them to various resolutions in the Windows OS.

Table 3 compares the processing using the CPU alone and the processing using the GPU together. It can be seen that speed is about 5 to 10 times faster than CPU only method. When the CPU-based operation was performed, the total execution time increased more than four times for every 2 times the resolution of the image. However, the speed difference was not significant even when the image was enlarged by the GPU-based technique.

For the experiment, 200 different situations were produced, and because of the overlapping situations, active

Table 3. Compared of the execution time

Resolution	CPU(sec)	GPU(sec) + CPU(sec)
320×240	0.11	0.03
720×480	0.43	0.06
1280×720	1.43	0.12
1920×1080	3.23	0.23

Table. 4. The result of simulation

Situation	Simulation	Success	Fail	Success rate (%)
Active	100	95	5	95
InActive	20	17	3	85
Position Moving	75	69	6	92
Fixed Moving	30	26	4	86.7
Walking	45	41	4	91.1
Running	45	38	7	84.4

100 times, inactive 20 times, position moving 75 times, fixed moving 30 times, walking 43 times, and running 45 times occurred. Two or more of them have created different situations 15 times.

Table 4 shows the simulation results and the average success rate was 89.03%.

In addition, two largely predictive simulation processes were performed to assess the performance of the prediction system. Both predictive simulation processes sampled the data into 64 data lengths, 54 of which were used for system design and the remaining 10 for system performance evaluation. Finally, both root mean square error and mean relative error were used as indicators for system performance evaluation. The following equations (7) and (8) can be used.

$$\mathrm{RMSE} = \sqrt{\frac{1}{N}\sum_{i-1}^{N}(y_i - \hat{y}_i)^2} \quad (7)$$

$$\mathrm{MRE} = \frac{1}{N}\sum_{i=1}^{N}\frac{|y_i-\hat{y}_i|}{|y_i|} \times 100\% \quad (8)$$

In the case of using 54 data for system design, when the regression model is used, the initial three data are used only as input of the system design, so the 51 training data cannot sufficiently reflect the characteristics of the data. Thus, in this paper, a linear interpolation between data and data was used to preprocess the given data. The preprocessed data consists of 253 pieces of data and the training data is 213 pieces long enough to contain enough information and length. These generated data are used to predict the first pair of input data that will be used as input to fuzzy prediction systems and neural network prediction systems to predict circular data. As such, the data are collected through four iterations by the k-mean clustering method, and the central change is also made clear by the four iterations.

Table 5 and 6 show the performance evaluation results of the path prediction system through two prediction simulation processes.

As shown in Table 5, the performance of the neural network prediction system is superior to that of the fuzzy prediction system in all index values. However, the final prediction results carried out through the prediction system selection process showed that some characteristics of the fuzzy prediction system were involved in the final prediction and that the performance of the proposed prediction system was high.

In addition, Table 6 shows that the RMSE metrics at the verification interval performed better than the performance of the fuzzy prediction system, whereas the performance of the fuzzy prediction system at the MRE metrics showed better performance than that of the neural network prediction system. It was also confirmed that the final predictive performance of the proposed system, which reflects these indicator values, was significantly improved in both indicators.

Therefore, the validity of the proposed prediction system design process can be confirmed through the performance evaluation results of Table 5 and Table 6, and it can be confirmed that the noise prediction error can be compensated through the system selection process. In addition, it can be seen that the linear interpolation can overcome the problem of lack of information or lack of data in the design of the prediction system.

Table. 5. Performance evaluation of prediction system

	Learning Section		Evaluation Section		
	Fuzzy System	Neural Network System	Fuzzy System	Neural Network System	Proposed Prediction System
RMSE	36.671	32.383	26.461	18.395	17.936
MRE(%)			0.312	0.214	0.202

Table 6. Performance evaluation of the system

	Learning Section		Evaluation Section		
	Fuzzy System	Neural Network System	Fuzzy System	Neural Network System	Proposed Prediction System
RMSE	47.713	48.771	36.331	39.316	34.154
MRE(%)			0.459	0.427	0.405

4. Conclusions

The existing smart street light does not reveal the pedestrian's path in advance unlike the existing street lamp environment where all street lamps are always turned on because the street lamp operates only when the pedestrian is detected by the attached sensor.

In this paper, a prediction system is proposed to control the brightness of smart street lamps by predicting the moving path through the reduction of consumption power and information of pedestrian’s past moving direction while meeting the function of existing smart street lamps. Using the walk tracking vector and soft hand-off characteristics obtained through the motion sensing sensor of smart street lamps, the brightness of smart street lamps is adjusted and the GPU is utilized for high-speed operation using motion vectors to analyze and predict the path of pedestrians. In addition, pedestrian movements were analyzed using adaptive Gaussian mixing techniques, weighting differential imaging techniques, and motion vectors. Lastly, to improve the performance of the prediction system, we designed the preprocessing process using linear interpolation, fuzzy prediction system and neural network prediction system in parallel structure to improve the efficiency. Rough set was used for error correction.

The prediction system proposed in this paper predicts the path of pedestrians and is expected to be effective in securing pedestrian safety, light pollution and energy saving by linking with smart street light system.

Acknowledgments

This study was supported by research funds from Chosun University, 2019.

References

[1] S.-K. Jeon, “ Research on Smart-LED Street Lighting System with Micro ESS for Demand side Energy Management,” Master Thesis, Chung-Ang Universtiy, 2018.

[2] J.-S. Kim, and K.-S. Kim, “An Analysis on the Policy Trends of LED-Lighting Policy in Major Countries,” Electronics and Telecommunications Trends, Vol. 28, No. 6, pp. 192-205, 2013.

[3] T.-Y. Kim, and N.-H. Park, “Design of Path Prediction Smart Street Lighting System on the Internet of Things,” Journal of the Chosun Natural Science, Vol. 12, No. 1, pp. 14-19, 2019.

[4] Z. Kaleem, T.-M. Yoon, and C. Lee, “Energy efficient outdoor light monitoring and control architecture using embedded system,” IEEE Embedded Systems Letters, Vol. 8, No. 1, pp. 18-21, 2016.

[5] F. Leccese, “Remote-control system of high efficiency and intelligent street lighting using a zigbee network of devices and sensors,” IEEE Transaction on Power Delivery, Vol. 28, No. 1, pp. 21-28, 2013.

[6] J.-S. Jeon, “A study on the street security light management system using Zigbee network,” Journal of the Korean Society of Marine Engineering, Vol. 38, No. 4, pp. 430-436, 2014.

[7] S.-H. Park, J.-Y. Hong, M.-S. Seok, J.-Y. Um, and J.-S. Ahn, “An Autonomous Street Light Switch Based on Motion Vector,” in Proc. of Korea Information Processing Society, Vol. 23, No. 2, pp. 810-813. 2016.

[8] C.-S. Chung, I.-S. Shin, J.-S. Seo, S.-D. Eun, and K. In, “The Analysis of a Gait Pattern and the Mechanical Efficiency on Ages and Speed Conditions,” Journal of Korean Society of Sport Biomechanics, Vol. 10, No. 2, pp. 205-219, 2001.

[9] J. Kim, and S.-W. Park, “Performance Analysis of Uplink Transmit Power Control during Soft Handoff,” The Journal of Korean Institute of Communications and Information Sciences, Vol. 37A, No. 8, pp. 632-638, 2012.

[10] S.-W. Kim, and Y.-S. Choi, “Analysis of Activity Using Motion Vector and GPU,” The Journal of the Korea Institute of Electronic Communication Sciences, Vol. 9, No. 10, pp. 1095-1102, 2014.

[11] H.-T. Kim, G.-H. Lee, J.-S. Park, and Y.-S. Yu, “Vehicle Detection in Tunnel using Gaussian Mixture Model and Mathematical Morphological Processing,” The Journal of Korean Institute of Communications and Information Sciences, Vol. 7, No. 5, pp. 967-974, 2012.

[12] Y. Ismail, J. B. McNeely, M. Shaaban, H. Mahmoud, and M. A. Bayoumi, “Fast Motion Estimation System Using Dynamic Models for H.264/AVC Video Coding,” IEEE Transactions on Circuits and Systems for Video Technology, Vol. 22, No, 1, pp. 28-42, 2012.

[13] J. Snaders, and E. Kandrot, CUDA by Example : An Introdution to General Purpose GPU Programing, Boston : Addison Wesley, 2010.

[14] J.-H. Kim, “The design of the system forecasting short-term peak load using neural network and fuzzy predictor,” Master Thesis, Kangwon National

University, 2016.

[15] Y.-K. Bang, and C.-H. Lee, “Design of Multiple Model Fuzzy Predictors using Data Preprocessing and its Application,” The Transactions of The Korean Institute of Electrical Engineers, Vol. 58, No. 1, pp. 173-180, 2009.

[16] R. Zhang, B. Ashuri, and Y. Deng, “A novel method for forecasting time series based on fuzzy logic and visibility graph,” Advances in Data Analysis and Classification, Vol. 11, No. 4, pp. 759-783, 2017.

[17] Z. Xiao, S. J. Ye, B. Zhong, and C. X. Sun, “BP neural network with rough set for short term load forecasting,” Expert Systems with Applications, Vol. 36, No. 1, pp. 273-279, 2009.

[18] H. Quan, D. Srinivasan, and A. Khosravi, “Short-Term Load and Wind Power Forecasting Using Neural Network-Based Prediction Intervals,” IEEE Transactions on Neural Networks and Learning Systems, Vol. 25, No. 2, pp. 303-315, 2014.

[19] W. Zhang, L. Ren, and L. Wang, “Method of Deep Belief Network Image Classification Based on Probability Measure Rough Set Theory,” International Journal of Pattern Recognition and Artificial Intelligence, Vol. 32, No. 11, 2018.