https://doi.org/10.7840/kics.2024.49.6.883

A Convolutional Neural Network Combined with Local Binary Pattern and Self-Attention Mechanism based on MC4L Device for Indoor Positioning

Nan Yin◆, Zhengyang Zou*, Yuxiang Sun**, Jaesoo Kim°

ABSTRACT

In earlier research, we proposed a vision-based ranging algorithm based on a Monocular Camera and four Lasers(MC4L) device for indoor positioning in dark environment call as Logarithmic Regression Algorithm(LRA). The linear relationship between the irradiation area and the real distance is established based on a LRA to control the positioning error within 2.4 cm. However, limited by the ranging mode of MC4L device, the indoor positioning algorithm cannot distinguish whether the measured object is a wall or an obstacle. Hence, its application in environments with obstacles is limited. In order to address this issue, we proposed a Convolutional Neural Networks(CNNs) combined with a Local Binary Pattern(LBP) and self-attention mechanism called as LBP-CNNs model. This LBP-CNNs model can achieve distance measurement and obstacle recognition by modifying activation function and loss function of output layer. Experimental results show that the LBP-CNNs model can reduce the indoor positioning error to 1.27 cm, and the obstacle recognition accuracy reaches 92.3%.

Key Words : Local binary pattern, Convolutional neural networks, Self-attention mechanism

Ⅰ. Introduction

In order to improve the accuracy of indoor positioning in dark environment, we proposed a vision-based ranging algorithm based on a MC4L device and logarithmic regression algorithm in previous research called as LRA[1]. Due to the angle between laser emitter and base, the center point and laser irradiation area will change with distance. Therefore, the corresponding relationship between irradiation area and real distance can be obtained by a logarithmic regression algorithm. However, since we only consider local feature changes of images, we cannot distinguish whether the current object is a wall or an obstacle. Because the illumination angle and smoothness are different on different materials, the reflection of laser light is also different. Fig. 1 shows reflections from wall and obstacle. We can find that the reflection of the obstacle is more obvious than wall. Based on this feature, we proposed an LBP-CNNs model combined with LBP and CNNs based on MC4L device. The LBP-CNNs model can achieve ranging error reduction and obstacle recognition by modifying activation function and loss function of output layer. LBP is often utilized to extract local texture features of images, which have

※ This research was supported by the Korean MSIT (Ministry of Science and ICT), under the National Program for Excellence in SW)(2021-0-01082) supervised by the IITP(Institute of Information & communications Technology Planning & Evaluation)"(2021-0-01082)

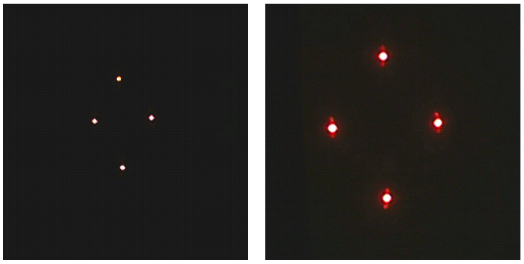
◆ First Author : Kyungpook National University, Computer Science and Engineering, yinnan1990@gmail.com, 정회원
° Corresponding Author : Kyungpook National University, Computer Science and Engineering, kjs@knu.ac.kr, 종신회원
* Kyungpook National University, Department of French Language and Literature, zzy@knu.ac.kr, 학생회원
** Kyungpook National University, Software Technology Research Institute, yxsun@knu.ac.kr, 정회원

논문번호 : 202306-123-B-RN, Received December 29, 2023; Revised February 2, 2024; Accepted February 22, 2024

the characteristics of rotation invariance and gray value invariance. Therefore, we can convert the original image into grayscale without losing image features. This operation can greatly reduce the computational intensity of convolutional layers. The experimental results show that the proposed LBP-CNNs model has a minimum error of 0.8 cm and an average error of 1.27 cm within 3 meters and the obstacle recognition accuracy reaches 92.3%. Our contributions can be summarized as follows:

1. We solve the problem presented in the previous paper without changing the hardware design and save half cost.
2. The utilization of LBP greatly reduces the computational pressure and improves the accuracy of obstacle recognition.
3. Compared with other models, the model we proposed has the characteristics of simple structure, high efficiency and strong robustness.

The remainder of the article is organized as follows. In the related works section, we analyze existing monocular depth estimation technologies and small object recognition technologies. In LBP-CNNs model section, we introduce the LBP-CNNs model and detail processing. Section 4 is experimental environment and indoor positioning algorithm based on LBP-CNNs mode. Next section is performance evaluation including determination coefficient for LBP-CNNs regression model and confusion matrix for LBP-CNNs classification model. Finally, a conclusion and future research are drawn.



a. wall　　b. obstacles

Fig. 1. Reflections on different materials

Ⅱ. Related Works

2.1 Monocular Depth Estimation Technologies

Monocular depth estimation methods can be divided into traditional machine learning and deep learning. The former generally uses Markov Random Fields(MRF) or Conditional Random Fields(CRF) to model depth relationships. Under the maximum posterior probability framework, through energy Function minimization solution depth. According to whether the model contains parameters, this method can be further divided into parametric learning methods and non-parametric learning methods. The former assumes that the model contains unknown parameters, and the training process is to solve the unknown parameters; the latter uses existing data sets to conduct similarity Retrieve inferred depth without learning parameters. However, in recent years, with the improvement of computing power, methods based on deep learning have gained attention. Eigen et al.[2] first utilized deep neural networks for monocular depth estimation tasks and proposed two-scale neural networks to estimate depth for a single image. The coarse-scale network predicts the global depth of images, and the fine-scale network optimizes the local depth. feature. Eigen et al.[3] improved on the above work and proposed a unified multi-scale network framework, which was used for three tasks: depth prediction, surface normal vector estimation and semantic segmentation. Applying the same framework independently to different tasks is not a unified learning of multiple tasks. Hence, it is classified as a single-task method. Different tasks adopt different loss functions and different datasets for training. The network model is end-to-end and requires no post-processing. Liu et al.[4] combined a deep convolutional neural network with a continuous CRF and proposed a deep convolutional neural field to estimate depth from a single image. By analytically solving the integral of the partition function, the likelihood probability optimization problem can be solved exactly. Liu et al.[5] encoded super pixel information into a neural network based on the research results of Trigueiros et al.[6] to improve computational efficiency. In addition, Li et al.[7] proposed a multi-scale depth estimation method. They first used a deep neural network to regress the depth at the super pixel scale, and then used multi-layer conditional random field post-processing to optimize the depth at the super pixel scale and pixel scale. Laina et al.[8] proposed a fully convolutional network architecture based on residual learning for monocular depth estimation. The network structure is deeper and no post-processing is required. In order to improve the output resolution while optimizing efficiency, this paper proposes a new up sampling method; taking into account the numerical distribution characteristics of depth, inverse Huber Loss is also introduced as an optimization function. Garg et al.[9] proposed using stereo image pairs to achieve unsupervised monocular depth estimation without the need for depth labels, and its working principle is similar to an automatic encoding machine. During training, a stereo image pair composed of the original image and the target image is used. The encoder is first used to predict the depth map of the original image, and then the decoder is used to reconstruct the original image by combining the target image and the predicted depth map. The reconstructed image is compared with the original image. Compare and calculate the loss. Godard et al.[10] further improved the above method and used the consistency of the left and right views to achieve unsupervised depth prediction. Epipolar geometric constraints are used to generate disparity maps and left, and right disparity consistency is used to optimize performance and improve robustness. In [11], The limitations of the deep learning-based monocular depth estimation model in terms of accuracy, computation time requirements, real-time inference, transferability, input image shape and domain adaptation, and generalization are discussed.

2.2 Small Object Recognition Technologies

Image object classification and detection are two important basic issues in computer vision research and are also the basis for other high-level vision tasks such as image segmentation, object tracking, and behavior analysis. In current research, feature fusion is basically used, that is, shallow feature maps and deep feature maps are combined into a feature fusion. The main reason is that shallow feature maps have better location information for objects, while deep feature maps have stronger semantic information for objects. Lim et al.[12] believes that the key to solving this problem depends on how to use context as additional information to help detect small targets. For example, by just looking at the objects in the image below, it would be difficult for humans to even identify these objects. However, by considering its location in the sky, the object can be identified as a bird. To improve the performance of small target detection, Mate et al.[13] proposed to oversample small target samples, and then copy and paste small targets in the sample to provide enough small targets to match anchor. The author proposed three types of copying to paste a small target. First select a small target in images, then copy and paste it multiple times at random locations. Second, select many small targets in the image and copy-paste them once anywhere. Finally, copy and paste all small targets in the image multiple times at any location. Since the points reflected by laser are very small in entire background, just binarizing them is not enough to extract their features. Therefore, we added LBP operation in the image preprocessing stage.

Ⅲ. LBP-CNNs Model

Fig. 2 shows the architecture of LBP-CNNs model. The entire LBP-CNNs model is divided into three parts, data preprocessing, CNNs with self-attention mechanism and post-processing. There are two post-processing tasks: one is regression for distance estimation and the other is binary classification for obstacle recognition. Our CNNs has a total of eight 1D Convolutional(Conv1D) layers, and each two Conv1D is a group. Conv1D is a 1D convolution operation that extracts features at different levels by scanning filters on the input data using a sliding window. Each group is followed by a Max Pooling layer(MaxPooling1D). MaxPooling1D is a down sampling operation used to reduce the dimensionality and computational complexity of feature maps. It selects the maximum value on each sliding window

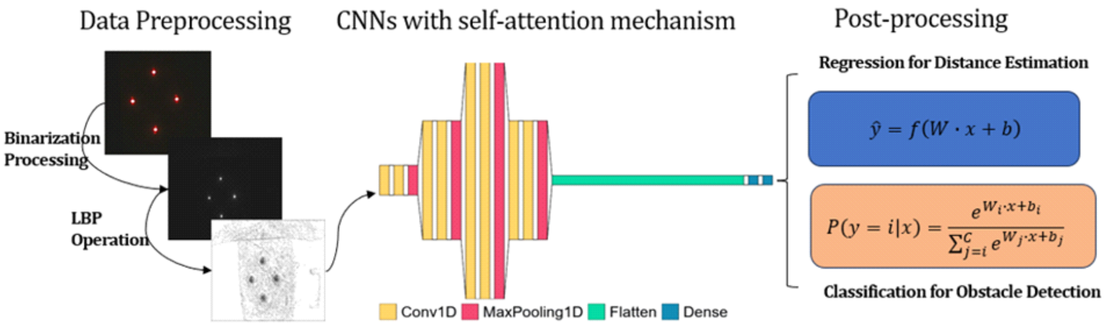

Data Preprocessing
CNNs with self-attention mechanism
Post-processing
Binarization Processing
LBP Operation
Regression for Distance Estimation
$\hat{y} = f(W \cdot x + b)$
$P(y = i|x) = \frac{e^{W_i \cdot x + b_i}}{\sum_{j=i}^{C} e^{W_j \cdot x + b_j}}$
Classification for Obstacle Detection
Conv1D
MaxPooling1D
Flatten
Dense

Fig. 2. Architecture of LBP-CNNs model.

of 1D data and takes these maximum values as output. The activation function uses *ReLU*. Then Flatten function is used to convert the 2D data into 1D data. It converts multi-dimensional input data into a 1D form and maintains the order of all elements. The number of neurons in the fully connected layer is 10 by Dense function provided by Keras library[14]. The Dense function is used to learn relationships and patterns between data. The final output layer is divided into two types: one is a linear function for distance estimation. The other is a SoftMax function for object detection shown in Fig. 2.

3.1 Data Preprocessing

The entire data preprocessing is divided into binarization process and LBP operation.

3.1.1 Binarization Processing

The first step is the binarization process shown in equation (1), The purpose of the binarization process is to reduce the computational pressure of CNNs model. The x and y in the formula represent the coordinates of pixel values. The T is threshold. If the pixel value exceeds the threshold T, it will be set to white, otherwise it will be black. where g represents the input image and G represents the output image.

$$G(x,y) = \begin{cases} 255, & if \;\; g(x,y) \geq T \\ 0, & otherwise \end{cases} \tag{1}$$

3.1.2 LBP Operation

A formal description of the LBP operator can be shown as equation (2).

$$LBP(x_c, y_c\ ) = \sum_{p=0}^{p-1} 2^p s\left(i_p, i_c\right) \tag{2}$$

Where (x_c, y_c) as central pixel with intensity i_c; and p is the p-th pixel of neighborhoods, i_p is the gray value of neighborhood pixels, i_c is the gray value of center pixel, and s(x) is the sign function defined as:

$$s(x) = \begin{cases} 1, & if \;\; x \geq 0 \\ 0, & else \end{cases} \tag{3}$$

3.2 Convolution Operation with Self-Attention Mechanism

Assume that there is an input feature map as I, a convolution kernel as K, and the corresponding output feature map as O. In the two-dimensional convolution operation, each element in output feature map can be obtained by the weighted sum of input feature map and convolution kernel at the corresponding position. This weighted summation process can be expressed by equation (4).

$$O(i,j) = \sum_{m=0}^{M-1} \sum_{n=0}^{N-1} I(i+m, j+n) \times K(m,n) \tag{4}$$

Where $O(i, j)$ represents the position in output feature map of (i, j). $I(i+m, j+n)$ represents the position in the input feature map of ($i+m$, $j+n$). $K(m,n)$ represents the position of (m,n) in convolution kernels. M and N are the height and width of convolution kernels respectively.

In CNNs, the attention mechanism can be used to enhance the model's attention to specific part of images. Among the attention mechanisms, a common implementation is the self-attention mechanism, which is often used in transformer model. In self-attention mechanism, assuming there is an input image *X*, its attention weight *A* can be calculated by following equation:

$$score(X_i, X_j) = X_i^T W_q \times (X_j^T W_k) \tag{5}$$

Where X_i and X are different position pixel value in input image *X* W_q and W_k are weight matrix. Next we calculate the attention weight by equation (6).

$$attention(X_i, X_j) = softmax\left(score(X_i, X_j)\right) \tag{6}$$

This step normalizes the attention score to a probability distribution so that the sum of attention weights is 1. Finally, we can obtain weight sum by equation (7).

$$weight(X_i) = \sum_{j=1}^{N} attention(X_i, X_j) \times (X_j^T W_v) \tag{7}$$

W_v is another weight matrix used to project the input sequence *X* into the value space. *N* is the length of input sequence. We use the mapping relationship between query, key and value to calculate attention weights, and generate output by weighted summation.

3.3 post-processing

In this section our post-processing is mainly divided into regression and classification. Regression is used to predict the measured distance and classification is used to determine whether the front is an obstacle or a wall.

3.3.1 Regression for Distance Estimation

For regression problems, the output layer is usually a single node or multiple nodes that predict continuous values. Since we want to predict a single continuous value, the formula of output layer can be described as:

$$\hat{y} = f(W \cdot x + b) \tag{8}$$

Where $\hat{y}$ is output of model. *W* is weight and *x* is input feature vectors. *b* is bias, *f* is identity function.

3.3.2 Classification for Obstacle Detection

We use the SoftMax function as activation function because it converts the raw output of neural network into a probability distribution for class predictions shown in equation (9).

$$P(y = i|x) = \frac{e^{W_i \cdot x + b_i}}{\sum_{j=i}^{C} e^{W_j \cdot x + b_j}} \tag{9}$$

Where P($y = i|x$) is the predicted probability of class *i* given input *x*. *C* is the total number of class and set to 2. W_i *and* b_i is the weight and bias of *i-th* class. During training we adopt the cross-entropy loss function to measure difference between predicted values and true labels.

Ⅳ. Indoor positioning algorithm

4.1 Experimental Environment

Fig. 3 shows a 2D diagram of the experimental environment. We randomly placed a target object cube ABCD with a length, width, and height of 15 cm in a dark experimental environment of 3 × 3 m. Since our algorithm can automatically identify whether the

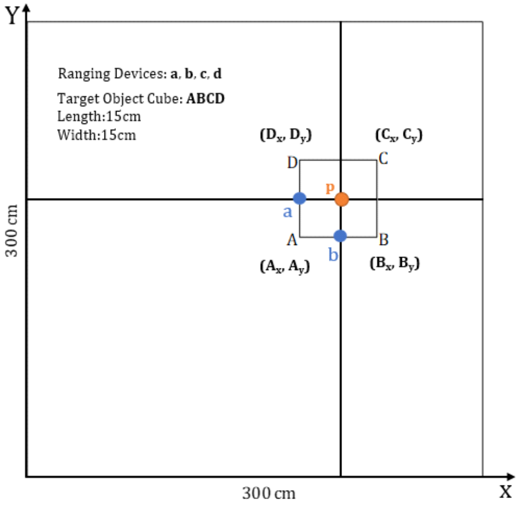
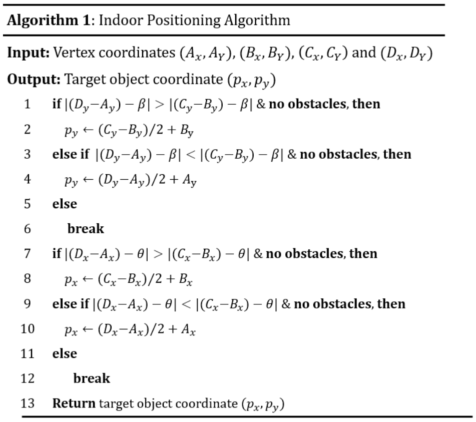


Fig. 3. 2D diagram of experimental environment.

front is an obstacle or a wall, we only need to place two devices *a* and *b* on the surface of ABCD object. This can directly reduce equipment costs by 50% compared to the previously proposed LRA method. We measured each position 5 times at 130 consecutive positions with intervals of 1 cm. Calculate the average error by comparing the five prediction results for each location with the real distance. During five image collections for each location, we randomly placed obstacles in front of the wall once. Therefore, our entire obstacle collection is 130 times and non-obstacle collection is 520 times.

Fig. 4 shows the proposed indoor positioning algorithm based on MC4L (Algorithm 1). β and θ are the width and height of the target object ABCD, respectively. First, we obtain the distances of DA(Dy - Ay) and CB(Cy - By) and compare the error between them and object width β (15cm) shown in line 1. If the error of DA is greater than that of CB and there are no obstacles in front, then the center point of CB will be the ordinate of the target object ABCD shown in line 2. Otherwise, the center point a of DA will be used as the ordinate of the target object ABCD shown in line 4. If there is an obstacle ahead, the entire ranging algorithm will break. The abscissa of ABCD can be obtained by repeating the same operations from lines 7 to 12. Finally, the center point coordinates (Px , Py) of ABCD can be obtained. The biggest difference from the ranging algorithm mentioned in LRA is the addition of obstacle recognition. Although the computational complexity is slightly increased, the current position can be calculated more accurately.



```
Algorithm 1: Indoor Positioning Algorithm
Input: Vertex coordinates $(A_x, A_Y)$, $(B_x, B_Y)$, $(C_x, C_Y)$ and $(D_x, D_Y)$
Output: Target object coordinate $(p_x, p_y)$
1   if $|(D_y - A_y) - \beta| > |(C_y - B_y) - \beta|$ & no obstacles, then
2     $p_y \leftarrow (C_y - B_y)/2 + B_y$
3   else if $|(D_y - A_y) - \beta| < |(C_y - B_y) - \beta|$ & no obstacles, then
4     $p_y \leftarrow (D_y - A_y)/2 + A_y$
5   else
6     break
7   if $|(D_x - A_x) - \theta| > |(C_x - B_x) - \theta|$ & no obstacles, then
8     $p_x \leftarrow (C_x - B_x)/2 + B_x$
9   else if $|(D_x - A_x) - \theta| < |(C_x - B_x) - \theta|$ & no obstacles, then
10    $p_x \leftarrow (D_x - A_x)/2 + A_x$
11  else
12    break
13  Return target object coordinate $(p_x, p_y)$
```

Fig. 4. Indoor positioning algorithm based on LBP-CNNs

V. Performance evaluation

5.1 Determination Coefficient for LBP-CNNs Regression Model

Determination coefficient is usually utilized to measure the fitting degree of a regression model and expressed as R^2. It can be calculated by following equation:

$$R^2 = 1 - \frac{RSS}{TSS} \tag{1}$$

Where RSS(Residual Sum of Squares) represents the sum of squares of the difference between model predictions and actual observations. TSS(Total Sum of Squares) represents the sum of squares differences between predicted variables and its mean. The value of R^2 ranges from 0 to 1. The closer it is to 1, the better the model fits data, while the closer it is to 0, the worse the model fits data. Fig. 5 shows the determination coefficient of proposed regression model under 7 random distances and reaches 0.9934.

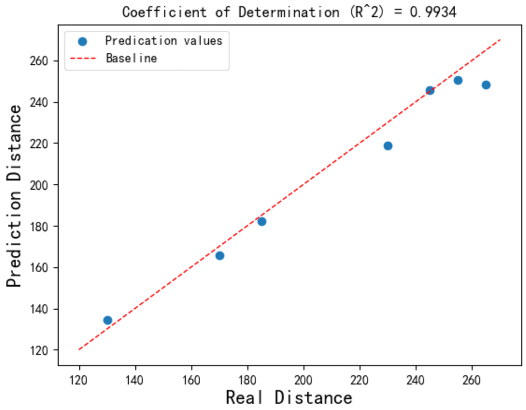


Fig. 5. Determination coefficient based on LBP-CNNs for regression under 7 random distances

5.2 Confusion Matrix for LBP-CNNs Classification Model

Fig. 6 shows the confusion matrix for proposed classification model. The horizontal line represents the prediction results of the model, and the vertical line represents the actual category. The four regions represent TP(True Positive), FN(False Negative), FP(False Positive) and TN(True Negative). The *F*1 *Score* is a value between 0 and 1. When the precision and recall are both high, the *F*1 *score* will also be high. *F*1 *score* is an indicator that comprehensively considers precision and recall and is used to evaluate the overall performance of a binary classification model. The *F1 score* of our proposed classification model reaches 0.89, accuracy is 0.923, precision is 1 and *Recall* is 0.8.

$$Precision = \frac{TP}{TP + FP} \tag{11}$$

$$Recall = \frac{TP}{TP + FN} \tag{12}$$

$$F1\ Score = \frac{Precision \times Recall}{Precision + Recall} \tag{13}$$

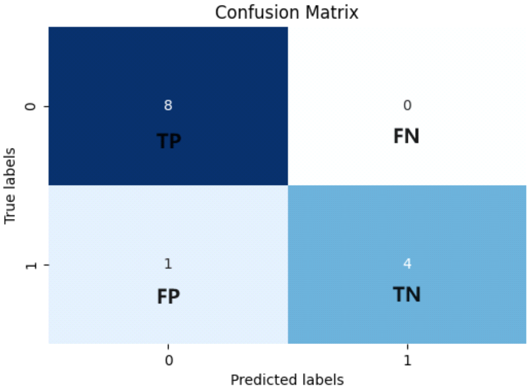


Fig. 6. Confusion matrix based on LBP-CNNs for classification under 13 random obstacles.

5.3 Indoor Positioning Error Comparison

Fig. 7 shows indoor positioning error comparison based on different depth estimation model. Compared with the model proposed by Eigen et al in 2014 and 2015, the performance is improved by 53.7% and 51.6% respectively. Mainly because the multi-scale deep network model they proposed can only improve the accuracy by increasing the amount of training data, but its feature extraction ability is weak. However, the LBP module in our model can better extract the features of images. Even if the model proposed by Liu in 2016 adds encoded super pixel information, it still has a 51.1% performance improvement. Garg et al used stereo image pairs to achieve excellent performance, but there was still a 54.2% gap with the LBP-CNNs model. Stereoscopic image pairs may suffer from viewpoint inconsistencies because the camera's position and orientation may not be precisely aligned, or due to dynamic elements in the scene. This can lead to errors in tasks such as depth estimation. Laina inverts the parameters of the Huber loss function, making it more sensitive to smaller errors but less robust to outliers. Compared with the LRA method, which is also based on MC4L device, the performance is improved by 47.1%. Godard replaced the use of explicit depth data during training with easier-to-obtain binocular stereo footage. This method saves a lot of manpower without explicit depth labels and enables real-time depth perception. However, binocular stereo lenses usually rely on texture and lighting information. Therefore, the accuracy of depth estimation may decrease for scenes that lack texture or have large lighting changes.

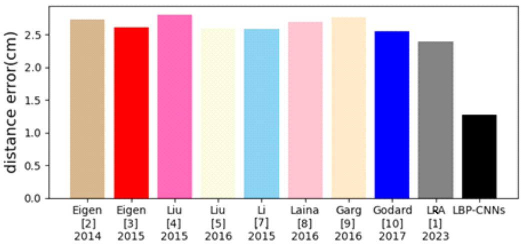


Fig. 7. Indoor positioning error comparison based on different depth estimation model

5.4 Errors from different locations

Fig. 8 shows errors at different distances from target object. The LRA method is a positioning algorithm based on MC4L devices that we proposed previously. it obtains the corresponding relationship between the irradiated area and real distance by a logarithmic regression algorithm. Since laser has different scattering effects on different target surfaces, deviations will occur in the process of positioning the laser irradiation point. Therefore, the LRA method will have uneven distribution of measurement errors.

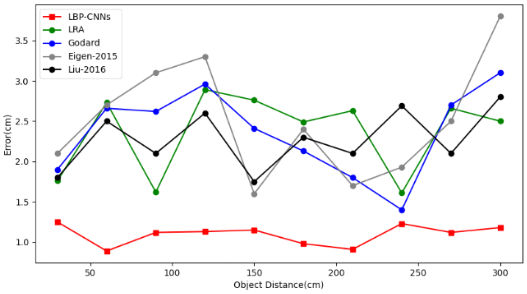


Fig. 8. Error at different distances from target object

Although the two algorithms shown in the figure are based on MC4L, the LBP-CNNs method we proposed not only leads in error but also has more stable error control. Godard adopts a stereo camera to reduce the error, but it also makes the error vary greatly in different situations. The difference between the maximum error and the minimum error is 41.9%. Compared with Eigen, Liu adds encoded super pixel information based on CNNs. Hence, its error fluctuations are relatively small.

Ⅵ. Conclusions

Our article is a supplement and improvement to the previously proposed indoor positioning algorithm called as LRA method. The LRA is also based on a MC4L device, which uses four lasers to mark the target object and record the irradiation spot with a high-definition monocular camera. The corresponding relationship between the irradiation area and the real distance is obtained by logarithmic regression algorithm.

Compared with the MC4L-based LRA method, the indoor positioning accuracy is improved by 47.1% and obstacle recognition accuracy reach 92.3% without changing the model structure. LBP is mainly utilized to extract texture information. Hence, it performs well in tasks such as texture recognition, and object recognition. Combined with the self-attention mechanism, it can more accurately locate the position of lasers. It overcomes the difficulty of finding laser irradiation point location due to laser irradiation divergence under different environments. In addition, the introduction of CNNs enables it to identify whether there are obstacles ahead. The implementation of this function can reduce the number of indoor positioning devices by half, effectively reducing costs.

References

[1] N. Yin, Y. Sun, Z. Zou, and J. Kim, "A vision-based ranging algorithm combining monocular camera and laser for indoor positioning," *J. KICS*, vol. 48, no. 11, pp. 1378-1386, 2023. (https://doi.org/10.7840/kics.2023.48.11.1)

[2] D. Eigen, C. Puhrsch, and R. Fergus, "Depth map prediction from a single image using a multi-scale deep network," in *Proc. Int. Conf. NIPS*, pp. 2366-2374, 2014. (https://doi.org/10.48550/arXiv.1406.2283)

[3] D. Eigen and R. Fergus, "Predicting depth, surface normals and semantic labels with a common multi-scale convolutional architecture," in *Proc. IEEE ICCV*, pp. 2650-2658, 2015. (https://doi.org/10.48550/arXiv.1411.4734)

[4] F. Y. Liu, C. H. Shen, and G. S. Lin, "Deep convolutional neural fields for depth estimation from a single image," in *Proc. 2015 IEEE CVPR*, pp. 5162-5170, 2015. (https://doi.org/10.48550/arXiv.1411.6387)

[5] F. Y. Liu, C. H. Shen, G. S. Lin, and I. Reid, "Learning depth from single monocular images using deep convolutional neural fields," *IEEE Trans. Pattern Anal. and Mach. Intell.*, vol. 38, no. 10, pp. 2024-2039, 2016. (https://doi.org/10.1109/TPAMI.2015.2505283)

[6] P. Trigueiros, F. Ribeiro, and L. P. Reis "A comparison of machine learning algorithms applied to hand gesture recognition," in *Proc 7th Iberian Conf. Inform. Syst. and Technol.*, pp. 1-6, 2012.

[7] B. Li, C. H. Shen, Y. C. Da, H. A. Vanden, and M. Y. He, "Depth and surfacenormal estimation from monocular images usingregression on deep features and hierarchical," in *Proc. IEEE Conf. CVPR*, pp. 1119-1127, 2015.

(https://doi.org/10.1109/CVPR.2015.7298715)

[8] I. Laina, C. Rupprecht, V. Belagiannis, F. Tombari, and N. Navab, “Deeper depth prediction with fully convolutional residual networks,” in *Proc. 4th Int. Conf. 3D Vis.*, pp. 239-248, 2016.
(https://doi.org/10.48550/arXiv.1606.00373)

[9] R. Garg, B. G. VijayKumar, G. Carneiro, and R. Ia, “Unsupervised CNN for single view depth estimation: Geometry to the rescue,” in *Proc. 14th ECCV,* pp. 740-756, 2016.
(https://doi.org/10.48550/arXiv.1603.04992)
(https://doi.org/10.1007/978-3-319-46484-8_45)

[10] C. Godard,O. M. Aodha, and G. J Brostow, “Unsupervised monocular depth estimation with left-right consistency,” in *Proc. IEEE Conf. CVPR,* pp. 270-279, 2017.
(https://doi.org/10.48550/arXiv.1609.03677)

[11] A. Masoumian, H. A. Rashwan, J. Cristiano, M. S. Asif, and D. Puig, “Monocular depth estimation using deep learning: A review,” *Sensors*, vol. 22, no. 14, pp. 1-24, 2022.
(https://doi.org/10.3390/s22145353)

[12] J. S. Lim, M. Astrid, H. J. Yoon, and S. I. Lee, “Small object detection using context and attention,” in *Proc. 3rd Int. Conf. Artificial Intell. in Inform. and Commun.*, pp. 181-186, 2021.
(https://doi.org/10.48550/arXiv.1912.06319)

[13] M. Kisantal, Z. Wojna, J. Murawski, J. Naruniec, and K. Cho, “Augmentation for small object detection,” in *Proc. 9th Int. Conf. Advances in Comput. and Inform. Technol.*, pp. 119-133, 2019.
(https://doi.org/10.5121/csit.2019.91713)

[14] F. J. J. Joseph, S. Nonsiri, and A. Monsakul, “Keras and TensorFlow: A hands-on experience,” *Advanced Deep Learn. for Eng. and Sci.*, pp. 85-111, 2021
(https://doi.org/10.1007/978-3-030-66519-7_12)

인 난 (Yan Yin)

2013 : Bachelor’s, Process Equipment and Control Engineering, Beijing University of Chemical Technology

2015 : M.Eng Mechanical Engineering, University of Ottawa

2022~Current : Ph. D. School of Computer of Kyungpook National University

<Research Interests> Machine Learning, Artificial Intelligence, Internet of Things

[ORCID:0009-0007-3506-9592]

순 위 샹 (Yuxiang Sun)

2019 : M.Sc, Computer Science and Engineering, Kyungpook National University

2022 : Ph. D. Computer Science and Engineering, Kyungpook National University.

2023~Current : Researcher, Software Technology Research Institute of Kyungpook National University

<Research Interests> Semantic Web, Entity Alignment, Machine Learning

[ORCID:0000-0003-0165-7664]

쩌우쩡양 (Zhengyang Zou)

2022 : Bachelor’s, Department of French Language and Literature, Kyungpook National University

<Research Interests> Signal Processing, Machine Learning, Internet of Things

[ORCID:0009-0009-6048-9614]

김 재 수 (Jaesoo Kim)

1985 : Bachelor's, Electronic Engineering, Kyungpook National University

1987 : M.Sc Computer Science, Joong-Ang University

1999 : Ph. D. Computer Engineering, Kyungnam University

1987~1996 : Senior Researcher, Korea Electrical Research Institute

2003~2004 : Visiting Professor, The University of Cincinnati, OH, USA

1996~Current : Professor, School of computer of Kyungpook National University

<Research Interests> Mobile Computing, Sensor Network, Internet of Things, UAV Network

[ORCID:0000-0003-2541-1669]