ROOM GEOMETRY ESTIMATION FROM ROOM IMPULSE RESPONSES USING CONVOLUTATIONAL NEURAL NETWORKS

Wangyang Yu* W. Bastiaan Kleijn†

* EEMCS, Delft University of Technology, Mekelweg 4, 2628 CD, Delft, The Netherlands
† ECS, Victoria University of Wellington, Kelburn, Wellington 6012, New Zealand

ABSTRACT

We describe a new method to estimate the geometry of a room given room impulse responses. The method utilises convolutional neural networks to estimate the room geometry and uses the mean square error as the loss function. In contrast to existing methods, we do not require the position or distance of sources or receivers in the room. The method can be used with only a single room impulse response between one source and one receiver for room geometry estimation. The proposed estimation method can achieve ten centimetre accuracy and is shown to be computationally efficient comparing to state-of-the-art methods.

Index Terms— Room geometry estimation, room impulse response, convolutional neural network.

1. INTRODUCTION

Augmented reality (AR) is a specific immersive audio-visual environment where the objects in an artificial scenario are augmented in real environment by computer-generated perceptual information and give the user an interactive experience [1]. Augmented reality in real environment by computer-generated perceptual information from room impulse responses as that facilitates a quick and practical measurement.

The room impulse response, the transfer function between the sound source and the listener, characterises the acoustics environment of a room. It is composed of direct-direction sound, early reflections, and late reverberation. The image-source method is commonly used to model reflections in a room [2-3]. A room impulse response is affected by the position of the sound source and the receiver, the room geometry and the reverberation time. Consequently, a room impulse response contains information about the room geometry.

Existing algorithms to estimate room geometry from room impulse responses all require a priori information about the configuration of the sources and the microphones [4-7], [8] uses room impulse responses and a set of time of arrival (TOA) measurements to estimate 2D room geometry. It does not require the positions of sources and receivers. However, it assumes that the TOA measurements are labelled and room impulse responses consist of the first and the second order reflections. [4] proposed a method to estimate the 3D room shape from room impulse responses by exploiting the properties of Euclidean distance matrices and the first order reflections. Although it requires only a single source, it requires at least four receivers and their pairwise distances. In addition, it may misclassify higher order reflections as the first order reflections [5]. In [5], the room geometry is estimated from one sound source by a two-step intuitive geometrical method. The proposed method still requires five receivers and their pairwise distances. [6] infers the room geometry efficiently by correctly labelled echoes and estimated image source positions, which requires at least two sources and five receivers.

In contrast to the above mentioned state-of-art methods, we would like to estimate the room geometry without multiple sources or receivers. A nature solution is the method based on machine learning, which requires just a single room impulse response between one source and one receiver.

Convolutional neural networks (CNNs) were first proposed by [8] for visual pattern recognition. CNNs can be composed of convolutional layers, pooling layers, fully connected layers and so on. As a result of the increased computational power and the availability of large databases, CNNs have seen a rapid increase in usage in recent years. Many variations of CNN architectures have been developed, such as AlexNet [9] and VGG-16 [10]. In addition, CNNs have been used for various applications such as image classification [11-13], speech recognition [14-16]. Recent applications, such as reverberation time estimation [17], prove that CNNs are able to show a good modelling ability for acoustic problems and outperform state-of-art algorithms, which motivates us to use CNNs for our problem. Moreover, CNNs contain filters to extract information from the input signal, which is a natural fit to our problem to estimate room geometry from room impulse responses. Among numerous applications of CNNs, we are not aware of existing work on room geometry estimation using CNNs.

The main contribution of our paper is that we use convolutional neural networks to estimate room geometry. In contrast to state-of-art methods for room geometry estimation, our method does not require the position or distance of receivers and sources. In its basic form, the method requires only one room impulse response between a single sound source and a single receiver. The proposed method is computationally efficient compared to state-of-art algorithms.

This paper is organised as follows. In section 2, we formulate our room geometry estimation problem, describe the network architecture and list the objectives of experiments. In section 3, we describe how we generate database, discuss our experimental setup and analyse the results. Finally, we conclude our paper in section 4.

2. CNN BASED ROOM GEOMETRY ESTIMATION

We use CNNs to estimate room geometry from room impulse responses. The room impulse response depends on reflection coefficients and room geometry. Room geometry is defined as a three-dimensional vector, which contains the length, width, and height
of a room. We consider reflection coefficients as a nuisance factor in our problem. The relationship between room impulse responses and room geometry is difficult to write as a mathematical equation. We seek a model to map the relationship between the room impulse responses and room geometry. In this section, we first describe the architecture and metrics of our CNN model. Then we propose a method to improve accuracy. Finally, we discuss the effect of reflection coefficients and reverberation time.

### 2.1. Architecture and metrics

Convolutional neural networks are considered a powerful modelling technique in various applications. Furthermore, CNNs contain a set of filters of different levels to extract the various features from the signals. Each filter slides along the entire signal to extract a certain kind of information from the signal by the convolution operation. The parameters of each filter are learned through the training process. CNNs can thus learn the features of a signal. Room geometry is an underlying feature of room impulse responses. Consequently, applying CNNs on room impulse responses is expected to extract room geometry information.

Since the room geometry is described by continuous variables, we formulate the room geometry estimation problem as a regression problem rather than a classification problem, where the model can output the continuous room geometry estimates directly. To solve the problem, our neural network has three output nodes for the length, width and height of a room. Our model takes one room impulse response in the time domain as the input without any preprocessing. The network estimates the geometry of a room for each room impulse response of the given room.

We adopt a commonly used CNN architecture as a basis. In this architecture each convolutional layer is followed by a batch normalisation layer and an activation function. Since our input signal is the raw time-domain signal, we use one-dimensional convolutional layers and one-dimensional batch normalisation layers. To keep a balance between the number of parameters and the modelling ability of neural networks, our proposed neural network consists of five one-dimensional convolutional layers and three fully connected layers. The number of the filters in convolutional layers increases with depth because the output dimensionality of the convolutional layers decreases. We use a rectified linear unit (ReLU) activation function as the activation function which introduces non-linearity to the model without adding parameters. To prevent overfitting and add regularization of the coefficients, a dropout layer is added after the second, fourth and fifth convolutional layer. We set the keep probability to be 0.1 since we do not have many units per layer. Our network architecture and corresponding parameters are shown in Table 1 where \( n \) denotes the batch size. Our network contains 178093 trainable parameters in total.

The mean square error is used as our loss function to minimise the squared distance between the estimated room geometry and the true room geometry. The room geometry is denoted as \( L \). The loss function is defined as

\[
MSE(L, \hat{L}) = \frac{1}{3} \sum_{i=1}^{3} (L(i) - \hat{L}(i))^2, \tag{1}
\]

where \( L \) denotes the true room geometry, and \( \hat{L} \) denotes the estimated room geometry. We chose the mean square error loss since it is relative sensitive to outliers, which we would like to suppress in room geometry estimation problem.

### 2.2. An improved algorithm

With the estimates that are not significantly biased, we proposed an improved algorithm for room geometry estimation. For each estimated room, we selected \( N \) random room impulse responses. We then have \( N \) estimates for each room. The method is to average over the \( N \) estimates to calculate the final estimate for the room. Since the accuracy is limited by the bias, we use experiments to investigate the accuracy we can reach and the effect of the number of estimates.

### 2.3. The effect of reflection coefficients and reverberation time

Besides room geometry, room impulse responses are also affected by reflection coefficients. We aim to investigate if fixed or varied reflection coefficients have an effect on the accuracy of estimation. Our hypothesis is that fixed reflection coefficients result in a more accurate estimate.

Sabine’s formula commonly quantifies reverberation time,

\[
RT_{60} = \frac{24ln10}{c_{20}} \frac{V}{S} \approx 0.1611 \frac{V}{S} a \tag{2}
\]

where \( c_{20} \) is the speed of the sound in the room for 20 degrees Celsius, \( V \) is the room volume, \( S \) is the total surface area of the room and \( a \) is the average absorption coefficient of room surfaces. From (2), we can conclude that reverberation time is related to room geometry and reflection coefficients. As a result, varying reflection coefficients is a sub-case of varying reverberation time.

| Output | Kernel Size | Stride | Filter | Output Size |
|--------|-------------|--------|--------|-------------|
| Input  |             |        |        | (n, 1024)  |
| Kernel |             |        |        | (60, 1, 1024) |
| BatchNorm1D |             |        |        | (60, 1, 1024) |
| ReLU   |             |        |        | (60, 1, 1024) |
| Conv1D | 4, 4, 10    | 1      |        | (60, 1024) |
| BatchNorm1D |             |        |        | (60, 1024) |
| ReLU   |             |        |        | (60, 1024) |
| Conv1D | 4, 4, 20    | 1      |        | (60, 1024) |
| BatchNorm1D |             |        |        | (60, 1024) |
| ReLU   |             |        |        | (60, 1024) |
| Conv1D | 4, 4, 100   | 1      |        | (60, 100) |
| BatchNorm1D |             |        |        | (60, 100) |
| ReLU   |             |        |        | (60, 100) |
| DropOut|             | 0.5    |        | (60, 100) |
| RelU   |             |        |        | (60, 100) |
| Reshape|             |        |        | (60, 1000) |
| Fully connected |             |        |        | (60, 1024) |
| Fully connected |             |        |        | (60, 1024) |
3. EXPERIMENTS

In this section, we present our experiments. In the first subsection, we describe how we generate our database for training and testing. We describe the setup our experiments in the second subsection. Finally, we show and analyse our experimental results.

3.1. Database Generation

We need a large-scale dataset of good quality to train our deep neural networks. To build a large-scale dataset, we used the image-source method to simulate RIRs. We assume the room is a shoe box shaped. The speed of sound was set to $c = 340$ m/s. The sample frequency was 8000 Hz. In addition, the length of each RIR was 4096. The room impulse response of length 4096 corresponds to approximate 0.5 seconds, which contains at least direct path signal and early reflections in an indoor environment. Each dimension of room geometry, i.e. length $\times$ width $\times$ height, was assumed to be uniformly distributed between 6 $\times$ 5 $\times$ 4 m and 10 $\times$ 8 $\times$ 6 m. We randomly placed one source and three receivers in each room and computed the room impulse responses between them. We generated three RIRs in each room since it outperforms other cases. In addition, we recorded the corresponding room geometry. The test dataset is generated in the same way as the training dataset.

To achieve a good training performance and prevent overfitting at the same time, it is crucial to separate training and test dataset properly. We set the ratio between the size of training dataset and test dataset to be 4 : 1. In our experiment, the size of training dataset was 24000 RIRs and the size of test dataset was 6000 RIRs.

3.2. Experimental Setup

In this subsection, we describe how we set up our experiments. We first discuss the experiments for reverberation time and error analysis. Then we discuss the setup of the experiments for improved methods. Finally, we describe the general experimental setup.

Our first experiment was to determine the effect of reflection coefficients and reverberation time. We divided our experiments into two cases, fixed and varying reflection coefficients. We first fixed the reflection coefficients and generated the database on this randomly generated set of reflection coefficients. In this setup, the reverberation time is only related to the change of room geometry. We then remove this restriction. The varying reflection coefficients database were generated to guarantee the reverberation time uniformly distributed between 0.4 s and 1 s, which can be representative of real-world environments. After that, we compared the performance between these two cases. We recorded the training error and test error. In addition, we computed the bias for varying reflection coefficients. Our further experiments will base on varying reflection coefficients.

In order to improve our room geometry estimation, we need to do an error analysis. We refer a sample as large error data when there exist at least one element of the three dimensions of estimated geometry vector whose squared distance is larger than 1 m. To begin with, we plotted the log error distribution in the test set. Then we compared the mean square error distribution between normal data and large error data with respect to room volume, reverberation time and direct path distance. We would like to determine if the large error data show some specific pattern with regard to a certain label. To further analyse the reason for the existence of large error data, we randomly generated ten rooms with a random reverberation time each. In each room, we randomly placed 100 sources and 100 receivers and calculated the room impulse responses. After that, we plotted the log error distribution in each room and analyse the result. In addition, we computed the bias for each room to figure out if the bias is constant in different room configuration.

We proposed an improved method to increase the estimation accuracy. The training progress is not changed. During the test step, we generated four different databases independently. In each database, there are 6000 randomly generated rooms. In each room, there are one, three, five, and nine random room impulse responses respectively. We perform our proposed improved method on these four datasets. That is, we average over the selected number of estimates in each room as our final estimates. Then we computed the mean square error in each database.

We used PyTorch to implement our neural network and perform training. We used the default initialisation method in PyTorch. We used a GPU node to train our neural network. The batch size was set to be 50. The learning rate of Adam optimiser was 0.001 and the coefficients used for computing running averages of gradient and its square were set to be (0.9, 0.999). We iterated for 2000 epochs and recorded the mean square error loss for each epoch. After training, we set the model on the evaluation mode and computed the test error. In addition, we recorded the running time for the geometry estimation of each room.

3.3. Experimental Results

In this subsection, we show and analyse our experimental results. We first compare the results of fixed and varying reflection coefficients. Then we show the results of error analysis. After that, we plot the error distribution for improved estimation methods and list the mean square errors. Finally, we compare our estimation error and running time with one of the traditional methods.

To begin with, we show the mean square error on fixed and varying reflection coefficients in Table 2. We find the error is smaller when the reflection coefficients are fixed. This proves that the varying reflection coefficients is a nuisance factor in our estimation problem and varying reflection coefficients has an effect on the accuracy of estimation. For the varying reflection coefficients, there are around 6.6% large error data. The bias in the test set is $-0.0084$ m, where the negative sign indicates our prediction is smaller than the true geometry. The estimation bias of length, width, and height is $-0.0679$ m, 0.0752 m, and $-0.0326$ m respectively. This confirms that our CNN model is not significantly biased after training.

| Reflection coefficients | Fixed | Varying |
|-------------------------|-------|---------|
| Training error (m)      | 0.0607| 0.1605  |
| Test error (m)          | 0.0968| 0.1781  |

Next we show the results of the error analysis. The log error distribution in the test set and ten different rooms is shown in Figure 1. Observing the error distribution in the test set, the error follows a long-tailed distribution, which confirms that the test error is mainly due to the existence of large error data and most estimation errors are relatively small. The error distribution of the randomly generated ten rooms with 10000 room impulse responses in each room all follow a long-tailed distribution. But the proportion of small errors is different for different rooms. We listed the bias of each room in Table 3. Comparing the bias of each room, we found there will...
Table 3: The bias of each room under varying reflection coefficients.

| Room number | Bias (m) | Room number | Bias (m) |
|-------------|----------|-------------|----------|
| Room 1      | −0.2301  | Room 2      | −0.0559  |
| Room 3      | −0.0234  | Room 4      | −0.1165  |
| Room 5      | −0.3294  | Room 6      | 0.2497   |
| Room 7      | 0.0657   | Room 8      | −0.1442  |
| Room 9      | −0.1261  | Room 10     | 0.0475   |

Table 4: Mean squared error of improved method under varying reflection coefficients.

| Number of RIRs | MSE (m) |
|----------------|---------|
| 1              | 0.1854  |
| 3              | 0.1272  |
| 5              | 0.1159  |
| 9              | 0.1101  |

be a bias for each room and the sign is different. In Figure 2, we show the error distribution of normal data and large error data with respect to room volume, reverberation time and direct path distance respectively. Observing Figure 2, the distributions between normal data and large error data do not show obvious patterns with respect to a certain label. Consequently, we can conclude that the outliers do not result from a certain label pattern. As a result, some room configurations may outperform others.

Figure 1: Mean squared error (in log scale) distribution under varying reflection coefficients.

(a) Test set

(b) 10 different rooms

Figure 2: Error distribution comparison between normal data and large error data under varying reflection coefficients.

Table 5: Comparison of proposed method and state-of-art method.

|                    | Proposed method | Method in [6] |
|--------------------|-----------------|---------------|
| Average error (m)  | 0.1101          | 0.0235        |
| Average run time (s)| 2.47 × 10⁻³     | 2.43          |

result is shown in Table 5. The method in [6] requires at least two sources and five receivers while our proposed method only requires nine random RIRs. From the experimental results, on the one hand, the traditional method performs approximately five times better in terms of average estimation error. On the other hand, in terms of average run time, our proposed CNN based method performs 10³ better than the method proposed in [6]. To conclude, our proposed CNN based room geometry estimation method is computationally efficient with acceptable estimation error and does not require priori required knowledge or setup compared to traditional methods.

4. CONCLUSIONS

In this paper, we use convolutional neural networks to estimate room geometry. We formulate our problem as a regression problem with the mean square error as a loss function. The proposed method only requires one random room impulse response between a single source and a single receiver. Knowledge of positions or relative distance is not required. With our proposed method, we can arrive at ten centimetre estimation accuracy. Moreover, our method is computationally efficient. Our further work will focus on more advanced error analysis to figure out the reason for the bias and approaches that minimize bias.
5. REFERENCES

[1] Wikipedia, “Augmented reality,” 2019. [Online]. Available: https://en.wikipedia.org/wiki/Augmented_reality

[2] H. Kuttruff, "Room acoustics." New York: CRC Press, 2014.

[3] J. B. Allen and D. A. Berkley, “Image method for efficiently simulating small-room acoustics,” J. Acoust. Soc. Am., vol. 65, no. 4, pp. 943–950, 1979.

[4] I. Dokmanić, R. Parhizkar, A. Walther, Y. M. Lu, and M. Vetterli, “Acoustic echoes reveal room shape,” Proceedings of the National Academy of Sciences, vol. 110, no. 30, pp. 12 186–12 191, 2013. [Online]. Available: https://www.pnas.org/content/110/30/12186

[5] T. Rajapaksha, X. Qiu, E. Cheng, and I. Burnett, “Geometrical room geometry estimation from room impulse responses,” in 2016 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), March 2016, pp. 331–335.

[6] I. Jager, R. Heusdens, and N. D. Gaubitch, “Room geometry estimation from acoustic echoes using graph-based echo labeling,” in 2016 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), March 2016, pp. 1–5.

[7] A. H. Moore, M. Brookes, and P. A. Naylor, “Room geometry estimation from a single channel acoustic impulse response,” in 21st European Signal Processing Conference (EUSIPCO 2013), Sep. 2013, pp. 1–5.

[8] K. Fukushima, “Neocognitron: A hierarchical neural network capable of visual pattern recognition,” Neural Networks, vol. 1, no. 2, pp. 119 – 130, 1988. [Online]. Available: http://www.sciencedirect.com/science/article/pii/0893608088900147

[9] A. Krizhevsky, I. Sutskever, and G. E. Hinton, “Imagenet classification with deep convolutional neural networks,” in Proceedings of the 25th International Conference on Neural Information Processing Systems - Volume 1, ser. NIPS’12. USA: Curran Associates Inc., 2012, pp. 1097–1105. [Online]. Available: http://dl.acm.org/citation.cfm?id=2999134.2999257

[10] K. Simonyan and A. Zisserman, “Very deep convolutional networks for large-scale image recognition,” CoRR, vol. abs/1409.1556, 2014. [Online]. Available: http://arxiv.org/abs/1409.1556

[11] A. Krizhevsky, I. Sutskever, and G. E. Hinton, “Imagenet classification with deep convolutional neural networks,” in Advances in neural information processing systems, 2012, pp. 1097–1105.

[12] Y. Wei, W. Xia, M. Lin, J. Huang, B. Ni, J. Dong, Y. Zhao, and S. Yan, “Hcp: A flexible cnn framework for multi-label image classification,” IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 38, no. 9, pp. 1901–1907, Sep. 2016.

[13] H. Lee and H. Kwon, “Going deeper with contextual cnn for hyperspectral image classification,” IEEE Transactions on Image Processing, vol. 26, no. 10, pp. 4843–4855, Oct 2017.

[14] O. Abdel-Hamid, A. Mohamed, H. Jiang, L. Deng, G. Penn, and D. Yu, “Convolutional neural networks for speech recognition,” IEEE/ACM Transactions on Audio, Speech, and Language Processing, vol. 22, no. 10, pp. 1533–1545, Oct 2014.

[15] S. Park, Y. Jeong, and H. S. Kim, “Multiresolution cnn for reverberant speech recognition,” in 2017 20th Conference of the Oriental Chapter of the International Coordinating Committee on Speech Databases and Speech I/O Systems and Assessment (O-COCOSDA), Nov 2017, pp. 1–4.

[16] Y. Qian, M. Bi, T. Tan, and K. Yu, “Very deep convolutional neural networks for noise robust speech recognition,” IEEE/ACM Transactions on Audio, Speech, and Language Processing, vol. 24, no. 12, pp. 2263–2276, Dec 2016.

[17] M. Lee and J. Chang, “Blind estimation of reverberation time using deep neural network,” in 2016 IEEE International Conference on Network Infrastructure and Digital Content (IC-NIDC), Sep. 2016, pp. 308–311.

[18] S. Ioffe and C. Szegedy, “Batch normalization: Accelerating deep network training by reducing internal covariate shift,” CoRR, vol. abs/1502.03167, 2015. [Online]. Available: http://arxiv.org/abs/1502.03167

[19] V. Nair and G. E. Hinton, “Rectified linear units improve restricted boltzmann machines,” in Proceedings of the 27th International Conference on International Conference on Machine Learning, ser. ICML’10. USA: Omnipress, 2010, pp. 807–814. [Online]. Available: http://dl.acm.org/citation.cfm?id=3104322.3104425

[20] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, “Dropout: A simple way to prevent neural networks from overfitting,” Journal of Machine Learning Research, vol. 15, pp. 1929–1958, 2014. [Online]. Available: http://jmlr.org/papers/v15/srivastava14a.html

[21] D. P. Kingma and J. Ba, “Adam: A method for stochastic optimization,” CoRR, vol. abs/1412.6980, 2014. [Online]. Available: http://arxiv.org/abs/1412.6980

[22] I. A. L. Erlangen, “RIR generator,” 2014. [Online]. Available: https://www.audiolabs-erlangen.de/fau/professor/habets/software/rir-generator