Machine-Learning-Based Throughput Estimation Using Images for mmWave Communications

Conference Paper · June 2017

DOI: 10.1109/VTCSpring.2017.8108570

6 authors, including:

Takayuki Nishio
Tokyo Institute of Technology
205 PUBLICATIONS **4,809** CITATIONS

SEE PROFILE

Koji Yamamoto
Kyoto Institute of Technology
268 PUBLICATIONS **2,450** CITATIONS

SEE PROFILE

Daisuke Murayama
NTT
29 PUBLICATIONS **128** CITATIONS

SEE PROFILE

Machine-Learning-Based Throughput Estimation Using Images for mmWave Communications

Hironao Okamoto*, Takayuki Nishio*, Masahiro Morikura*, Koji Yamamoto*,
Daisuke Murayama†, and Katsuya Nakahira†
*Graduate School of Informatics, Kyoto University, Yoshida-honmachi, Sakyo-ku, Kyoto, 606-8501 Japan
†NTT Network Innovation Laboratories, NTT Corporation, Yokosuka, Kanagawa 239-0847, Japan
E-mail: info16@imc.cce.i.kyoto-u.ac.jp

***Abstract*—The human blockage problem is an open issue in next-generation wireless access networks using millimeter-wave (mmWave) communications. A proactive base station (BS) handover system leveraging RGB and depth (RGB-D) cameras is proposed for solving the human blockage problem. RGB-D cameras observe mmWave communication ranges, and BS handover is conducted proactively before a human blockage causes serious performance degradation. However, this system must rely on a scheme that provides a guideline for selecting a BS to which the transfer can be done. In this paper, we propose a mmWave throughput estimation scheme using an online machine learning algorithm and depth images obtained by the RGB-D camera. The algorithm learns the relationship between depth images and measured throughputs, and estimates throughput from depth images. The scheme enables the handover system to estimate throughput quickly and adaptively to the wireless environment without transmitting any control frames. We conducted a proof-of-concept experiment by using a testbed consisting of IEEE 802.11ad mmWave wireless local area network devices and an RGB-D camera. The experiment confirmed that the proposed scheme estimates throughput from a depth image with an RMS error of 114–178 Mbit/s in real time.**

I. Introduction

The millimeter wave (mmWave) band is seen as a key enabler of high-speed and high-capacity wireless access in next generation mobile networks [1]–[3]. At least 5 GHz of continuous bandwidth is available in many countries worldwide, and that enables multi-Gbit/s transmission rates using the IEEE 802.11ad standard [4], which is a standard of wireless local area networks (WLANs) using 60 GHz band. However, mmWaves are strongly absorbed by moisture and oxygen, thus when pedestrians block line-of-sight (LOS) paths, received signal strength seriously decreases [5].

To solve the human blockage problem, base station (BS) handover and directivity control have been suggested [6]–[8]. In the handover method, a mobile station (STA) communicating with a BS is transferred to another BS with better link quality (LQ) when the LOS path blockage occurs. In the directivity control method, the mmWave beam is directed in another direction so that an STA can receive signals using reflected or diffracted waves. Here, the LQ is determined by performance metrics such as the throughput, received signal strength indicator (RSSI), and frame loss rate. Since blockage decreases LQ sharply, the handover or directivity control need to be performed quickly in order to keep LQ high.

We proposed a proactive handover scheme and experimentally confirmed that the scheme significantly reduced the amount of time during which human blockage degraded throughput performance [9]. The system predicts the timing of human blockages using an RGB-D camera and performs handover before blockages occur. Using RGB-D cameras, the local wireless environment, including the locations of obstacles, can be detected, which enables accurate prediction of current or future wireless environments. However, the previous work assumed that the LQ between every pairs of STAs and BSs are given.

This paper focuses on how estimate LQ between each STA and BS, including pairs not transmitting frames to each other. For a better LQ, a handover controller must decide which BS to be transferred to or which directivity to be used. Furthermore, since directly measuring LQ requires transmitting several frames between each STA and each BS, which leads to consumption of bandwidth and delay in proactive handover, the controller must estimate LQ from historical link information.

LQ estimation (LQE) has been studied intensively in sensor networks, and two approaches have been employed: hardware-based and software-based [10]. Hardware-based LQE uses metrics obtained directly from radio modules such as RSSI and signal-to-noise ratio. These metrics can be measured by sending overhead frames. However, since mmWave communication employs directional antennas and beam forming, the metrics of overhead frames and frames after beam forming could be very different, and large differences could cause degradation of estimation accuracy. On the other hand, software-based LQE uses metrics obtained from the statistics of packet reception or transmission, such as packet reception ratio and the required number of packets. In this approach, there is a trade-off between overhead and accuracy. In order to obtain an accurate estimation at every moment of operation during handover, BSs must establish a connection with each STA and exchange packets regularly, which consumes bandwidth.

In this paper, we propose a machine-learning-based throughput estimation scheme using images obtained by an RGB-D camera for a proactive mmWave BS handover system. The scheme enables us to obtain the estimated throughput as LQ, which provide a guideline to select a BS with highest throughput without overhead signals. The scheme uses an online machine learning algorithm [11] which enables fast and adaptive estimation. In addition, we discuss an accuracy improvement method to manage accuracy degradation caused by human blockages in estimation using the proposed scheme. To demonstrate the effectiveness of the proposed system, we built a testbed using commercially available IEEE 802.11ad mmWave WLAN devices and an RGB-D camera.

The focus of this paper is a proof of concept of image-based LQE for mmWave communications. Therefore, we use depth images which include the three-dimensional geometry of the communication area. In terms of implementation cost, a depth camera is more expensive than an RGB camera. However, this LQE approach is expected to be applicable for RGB images. Additionally, we note that the implementation cost of cameras can be offset by sharing images with other applications such as surveillance and user analysis applications; moreover, mmWave communications would generally be deployed in crowded places such as stations and commercial facilities, which would be primary targets of surveillance and user analysis applications.

The contributions of this paper are threefold. 1) This is the first attempt to quantitatively estimate mmWave throughput using depth images and machine learning. 2) We experimentally reveal estimation accuracy degradation induced by PHY rate control and propose a filtering method to solve this problem. 3) Proof of concept experiments demonstrate the feasibility and estimation accuracy of the proposed scheme.

II. System Model

Fig. 1 shows the system model. The system consists of the following components: mmWave BSs, RGB-D cameras, machine learning units (ML units), and a controller. Each ML unit is connected with a mmWave BS, RGB-D cameras, and a controller via wired local area networks. The RGB-D cameras, typified by Microsoft Kinect [12], can capture not only color but also depth images. The camera-to-object distance data in depth images accurately represent the locations of objects which affect mmWave LQ. The RGB-D image is sent to the connected ML unit. The unit estimates current LQ based on past LQ and images using a machine learning algorithm. Specifically, when the unit receives the measured LQ from the BS, the unit learns the relationship between the LQ and the RGB-D camera image and sends the measured LQ to the controller. In contrast, when the unit does not receive the measured LQ, it estimates the current LQ and sends the estimation result to the controller. Note that the ML units can estimate the LQ by using multiple cameras and can share cameras with other ML units. The controller controls the handover between BSs based on the measured or estimated LQ received from the units.

In this paper, we use the throughput as the LQ since other metrics such as the RSSI and MCS index cannot be obtained on IEEE 802.11ad devices used in our experiments. Additionally, we assume saturated traffic conditions in order to estimate the time series of the low-layer LQ. We use only depth images from RGB-D images.

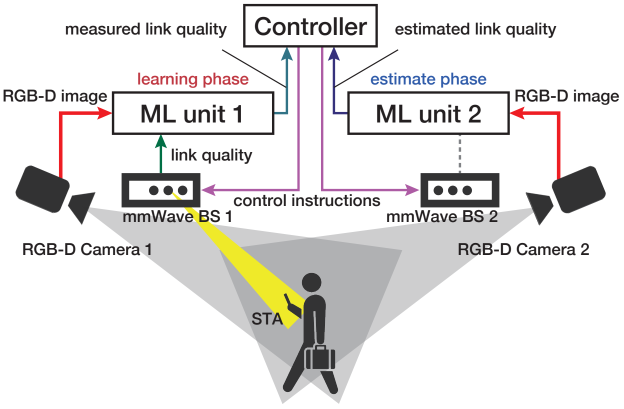


Fig. 1. System model.

RGB-D cameras have been studied in image recognition, and existing image recognition techniques provide highly accurate human or object recognition [13], [14]. The combination of human or object location data obtained by this technique and mmWave propagation characteristics enables LQ estimation. In addition, by using machine learning techniques for LQ estimation, we can construct a system that adapts to the wireless environment, particularly the positions of the BS, the camera, and the objects, as changes in the environment.

III. Machine Learning Algorithm

A. Online Machine Learning

We use an online learning algorithm called adaptive regularization of weight vectors (AROW) [15] as the estimation method in the ML units. Machine learning [11] is broadly divided into batch learning and online learning. In batch learning, a complete training input-output data set is required to learn models because models are constructed by learning the entire set of training data at once. Batch learning provides high-accuracy estimation; however, it requires long calculation time, consumes much memory for storing whole dataset, and has difficulty reflecting new input-output data in learned models. In contrast, online learning algorithms operate in rounds. In online learning, training data becomes available one by one and models are updated based on stochastic optimization when each new data point becomes available. Online learning takes little time for each update, consumes little memory, and easily reflects new input-output data in learned models.

Examples of online learning algorithms include linear classifiers and some neural networks based on stochastic gradient descent methods. Linear classifiers can be calculated quickly, but cannot learn non-linear problems. Neural networks can learn non-linear problems, but incur greater calculation costs.

In the proposed system, it is desirable for the RGB-D camera to take high-resolution and high-frame-rate video to improve estimation accuracy. As a consequence, the ML units receive large-size image data sequentially. Thus, the online learning algorithm is appropriate for this system because the algorithm updates estimation models sequentially and needs not to store each data point. Although AROW is algorithms to assign discrete values to input data, it can assign continuous values to input data by applying to regression problems.

B. Adaptive Regularization of Weight Vectors

A linear classifier receives a training input $\boldsymbol{x}_t \in \mathbb{R}^d$ and output $y_t \in \{-1, 1\}$ as a pair in round t and learns the relationship parameterized by a weight vector $\boldsymbol{w}_t : y_t = \mathrm{sign}(\boldsymbol{x}_t^{\mathrm{T}} \boldsymbol{w}_t)$. In AROW, $\boldsymbol{w}_t$ is assumed to follow a normal distribution $\mathcal{N}(\boldsymbol{\mu}_t, \Sigma_t)$. $\boldsymbol{\mu}_t \in \mathbb{R}^d$ is the mean of the weight vector, and $\Sigma_t \in \mathbb{R}^{d \times d}$ is the covariance matrix. The AROW algorithm minimizes the following unconstrained objective function in each round:

$$C(\boldsymbol{\mu}_t, \Sigma_t) = \mathrm{D_{KL}}(\mathcal{N}(\boldsymbol{\mu}_t, \Sigma_t) \parallel \mathcal{N}(\boldsymbol{\mu}_{t-1}, \Sigma_{t-1})) + \lambda_1 \ell_{\mathrm{h}^2}(y_t, \boldsymbol{x}_t^{\mathrm{T}} \boldsymbol{\mu}_t) + \lambda_2 \boldsymbol{x}_t^{\mathrm{T}} \Sigma \boldsymbol{x}_t, \quad (1)$$

where $\mathrm{D_{KL}}(\mathcal{N}(\boldsymbol{\mu}, \Sigma) \parallel \mathcal{N}(\boldsymbol{\mu}_{t-1}, \Sigma_{t-1}))$ is the Kullback-Leibler (KL) divergence, also called relative entropy, between the distribution before update $\mathcal{N}(\boldsymbol{\mu}_{t-1}, \Sigma_{t-1})$ and one after update $\mathcal{N}(\boldsymbol{\mu}_t, \Sigma_t)$. $\ell_{\mathrm{h}^2}(y_t, \boldsymbol{x}_t^{\mathrm{T}} \boldsymbol{\mu}) = (\max\{0, 1 - y_t \boldsymbol{x}_t^{\mathrm{T}} \boldsymbol{\mu}\})^2$ is the squared hinge loss function. $\lambda_1, \lambda_2 \geq 0$ are hyperparameters. For simplicity, it is assumed that $\lambda_1 = \lambda_2 = 1/(2r)$ for some $r > 0$.

The first term of (1) shows that the algorithm minimizes the KL divergence between the pre-training model and the post-training model. That is, AROW updates $\boldsymbol{\mu}$ and Σ minimally, which improves noise immunity. The second term minimizes the squared hinge loss function, which shows that the algorithm updates models so that correct values are assigned to input data. The third term shows that Σ becomes smaller as the rounds proceed, which suppresses the fluctuation of well-trained weight values.

The Σ is usually defined as a diagonal matrix. Although this definition decreases estimation accuracy, it reduces calculation cost. AROW can be applied to regression by modifying y as $y \in [-1, 1]$ and changing squared hinge loss function $\ell_{\mathrm{h}^2}(y_t, \boldsymbol{x}_t^{\mathrm{T}} \boldsymbol{\mu})$ to the squared loss function $\ell(y_t, \boldsymbol{x}_t^{\mathrm{T}} \boldsymbol{\mu}) = (y_t - \boldsymbol{x}_t^{\mathrm{T}} \boldsymbol{\mu})^2$.

IV. Proposed Throughput Estimation

A. Overview of Proposed Scheme

The proposed scheme consists of two operation phases: learning and estimation. When either the BS or the STA is transmitting frames to the other device, the ML unit connected with the BS is in the learning phase. Conversely, when both the BS and the STA are not transmitting frames to each other, the ML unit connected with the BS is in the estimation phase.

In the learning phase, the ML unit learns the relationship between depth images from the RGB-D camera and measured throughput from the mmWave BS by using a machine learning algorithm with a training dataset filtering, details of which are explained in Sections IV-B and IV-C, respectively. The unit then sends the measured throughput to the controller.

In the estimation phase, the ML unit estimates throughput from a depth image from the RGB-D camera by using the learned model and estimation output filtering, details of which are explained in Section IV-C. The unit then sends the estimated throughput to the controller.

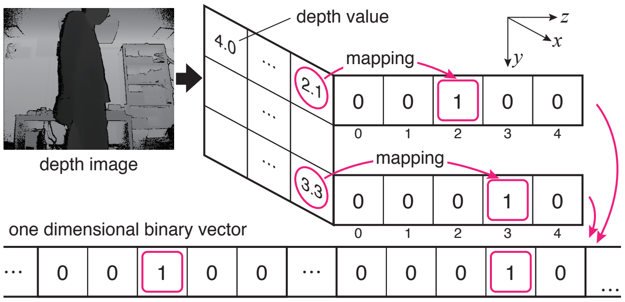


Fig. 2. An example of the transformation of a depth image to an input vector for AROW. Resolution of a depth image is reduced to $(x, y, z) = (3, 3, 5)$ in this figure.

The controller receives the measured throughput from units in the learning phase and estimated throughput from those in the estimation phase. The controller performs the handover based on measured and estimated throughputs. Note that this paper focuses on throughput estimation, so the handover algorithm is outside of the present scope.

B. Throughput Estimation with AROW

The proposed scheme uses AROW for regression as the machine learning algorithm. Let s_t be the measured throughput obtained from a mmWave BS and $\boldsymbol{i}_t$ be the transformed depth image at time t. In the learning phase the AROW learns the relationship between the normalized throughput $r_t = s_t/\eta$ and the transformed depth-image $\boldsymbol{i}_t$. η is a throughput normalization factor, so that r_t ranges from 0 to 1. In the estimation phase the algorithm estimates the normalized throughput r'_t from $\boldsymbol{i}_t$ and outputs the estimated throughput $s'_t = \eta r'_t$.

Fig. 2 shows the transformation of an original depth image into a transformed depth image, which is an one-dimensional binary vector. First, the resolution of the depth image is reduced to H × W pixels in order to reduce the calculation complexity. Second, we normalize each depth value such that its maximum value becomes D. Let $d_{x,y}$ denote a depth value contained by pixel (x, y). We generate D-sized binary vectors for each pixel, the $(\lfloor d_{x,y} \rfloor)$-th value of which is 1 with other values being 0. Finally, we combine each vector to obtain the transformed depth image i_t for the input vector of AROW.

C. Accuracy Improvement Method Using a Filtering of Training Dataset and Estimation Output

In addition to the throughput estimation from depth images using AROW, a training dataset and estimation output filtering is proposed for preventing estimation accuracy degradation. Fig. 3 shows the results of a pre-experiment and illustrates the low-throughput period following a blockage. The throughput remains low for several seconds after the blockage ends in spite of the LOS communication. A depth image in the low-throughput period, for example, B in Fig. 3, is similar to one in LOS communications, for example, C in Fig. 3, which means that the LOS path is not blocked at B.

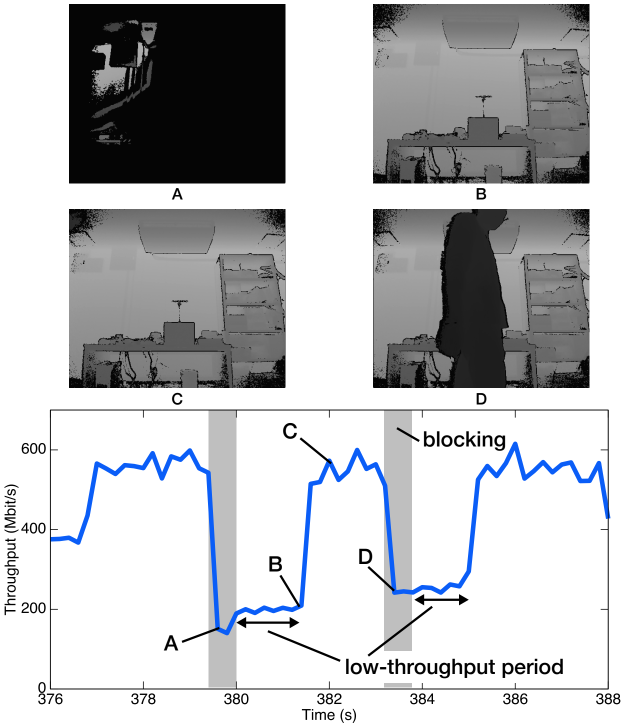
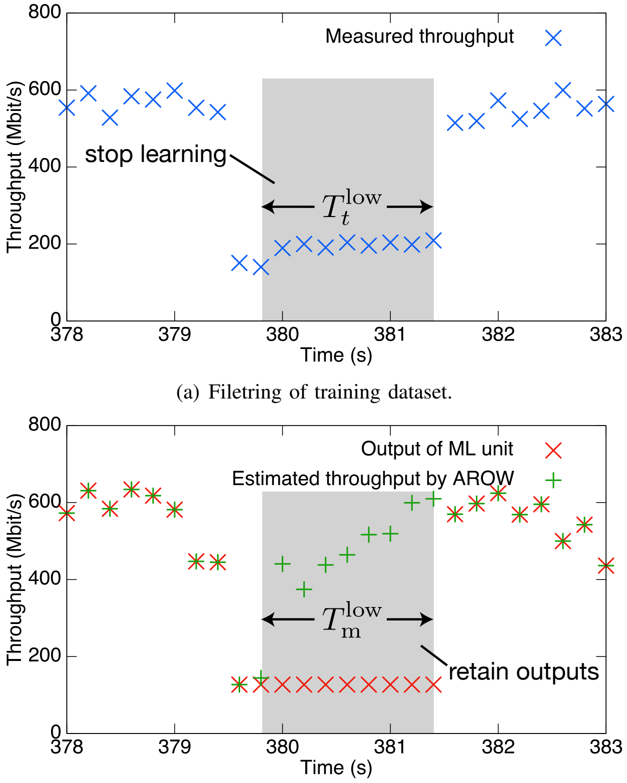


Fig. 3. Time series of depth images and measured throughputs. The shaded areas represent periods when the LOS path is blocked.



(a) Filetring of training dataset.

(b) Filtering of estimation output.

Fig. 4. Filtering of training dataset and estimation output.

Such a low-throughput period after a period of low channel quality was observed in conventional WLANs wherein IEEE 802.11g devices using a rate adaptation PID algorithm [16]. In that case, PID decreases the PHY rate when a long period of interference or signal level reduction occurs, and keeps the PHY rate low for a few seconds after the recovery of radio channels [17]. The low-throughput period for mmWave communications seems to be caused via a similar mechanism to that of conventional WLANs.

In practical use, there are no transmissions between the BS and the STA in the estimation phase. Hence, blockages do not affect rate adaptation and do not cause the low-throughput period. However, there are transmissions in the learning phase, therefore blockages cause the periods. If the AROW learns the relationship during this period, the estimation accuracy is degraded because AROW cannot distinguish between images B and C in Fig. 3. In addition, when we estimate throughput between a BS and an STA pair with frame transmissions, we must take account of the low-throughput period.

To solve the problem caused by the low-throughput period, we perform a filtering of training dataset and estimation output. This filtering prevents the unit from learning an undesirable model in learning phase and lets the unit simulate the low-throughput period in estimation phase. Modifying the rate-adaptive algorithm could be another solution for this problem, but this approach is outside the scope of this paper. Fig. 4 illustrates the filtering. In the training phase, first the ML unit judges the presence of the low-throughput period based on temporal variation of measured throughputs. Next, the unit stops the algorithm from learning during this period. This prevents the unit from learning an undesirable model and improves estimation accuracy. In addition, the unit stores the length of the low-throughput period T_t^{low}. The details of this procedure are given in Algorithm 1.

When we estimate throughput between a BS and an STA pair with frame transmissions, first the ML unit judges the beginning of the period based on temporal variation of estimated throughputs. Next, the unit retains outputs over the $T_{\mathrm{m}}^{\mathrm{low}}$ seconds at the estimated throughput from the algorithm at the beginning of the period. $T_{\mathrm{m}}^{\mathrm{low}}$ is the median of T_t^{low}. From this filtering, the unit simulates the low-throughput period. The details of this procedure are given in Algorithm 2.

V. Experiment

A. Experimental Scenario

Fig. 5 shows the experimental setup. The location of the mmWave BS and STA were fixed during the experiment. The controller generated TCP traffic with iPerf3 [18] for throughput measurement and obtained throughput and depth images at 0.2-second intervals.

A pedestrian passed in front of the BS from backward and forward alternately at a velocity of 1 m/s, and these blockages occurred about every 4 s. We performed several experimental trials. In each trial of the experiment, the camera was placed at one of the following positions: (A) on the BS and (B) 1.8 m forward and 1 m in front of the BS. In addition, a pedestrian

Algorithm 1 Learning phase with training dataset filtering

```
input: The tolerance of throughput variability $\alpha > 0$.
initialize: set filter_state ← IDLE.
for time $t = 1, \dots, T$ do
    Get the current throughput $s_t$ and depth image $\boldsymbol{i_t}$;
    if $s_{t-1} < s_{t-2} - \alpha$ and $s_t > s_{t-1} - \alpha$ then
        set filter_state ← ACTIVE and $T_{\mathrm{s}} \leftarrow t$;
    else if filter_state = ACTIVE and $s_t > s_{t-1} + \alpha$ then
        set filter_state ← IDLE;
        Store the low-throughput period $T_t^{\mathrm{low}} = T_{\mathrm{s}} - t$;
    end if
    if filter_state = IDLE then
        Learn the relationship between $s_t$ and $\boldsymbol{i_t}$;
    end if
end for
```

walked along one of the following paths: (p) 0.2 m in front of the BS, (q) 0.8 m in front of the BS and (r) along both paths p and q. The BS and the camera were placed at a height of 1 m above the floor, and the STA was placed at a height of 0.9 m. The detection allowance of blockage α was set to 100 Mbit/s. The parameter of AROW r was set to 200, and the input for AROW was a one-dimensional binary vector of depth images with a resolution of $(x, y, z) = (20, 20, 20)$. The throughput normalization factor η was set to 1 Gbit/s because 1000BASE-T Gigabit Ethernet limits the maximum throughput to 1 Gbit/s.

Our experiments consisted of a learning period and an estimation period. In the learning period, the ML unit learned the relationship between depth images and throughputs for 10 minutes. In the estimation period, the ML unit stopped learning and estimated throughputs from depth images for 10 minutes, where the unit continued to measure throughput for comparison. We also estimated throughput with a scheme which does not filter training datasets and estimation outputs. After conducting the experiments, we compared the root mean square (RMS) error of the estimated throughput of both the proposed scheme and the scheme without filtering.

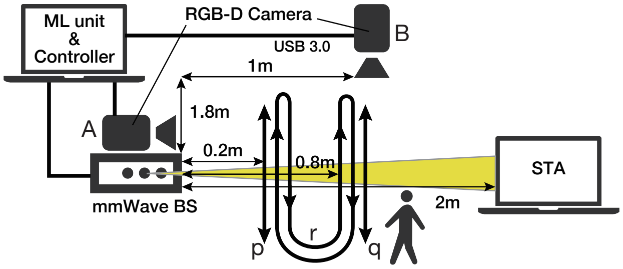
B. Experimental Results

Fig. 6 shows a part of the measured and estimated throughput time series when the camera was placed at position A and a pedestrian walked along path (r). The simultaneous decrease of measured throughput and each estimated throughput indicates that both schemes are able to detect LOS path blockages.

In the low-throughput periods following blockages, as shown in Fig. 6, after a pedestrian blocks the LOS path, the scheme without filtering raised its estimated throughput to a high value while the measured throughput was low. On the other hand, the proposed scheme retained its low throughput estimate. The measured throughput sometimes increased immediately after the human blockage occurred, as shown at 341 s in Fig. 6, but such cases are less common than those where a low-throughput period occurs. Therefore, the proposed scheme averagely provided more accurate estimation than the scheme without filtering in the experiment.

Algorithm 2 Estimation phase with estimation output filtering

```
input: The tolerance of throughput variability $\alpha > 0$.
initialize: set the estimated low-throughput period
    $T_m^{\mathrm{low}}$ ← the median of $T_t^{\mathrm{low}}$ and filter_state ← IDLE.
for time $t = 1, \dots, T$ do
    Get the current depth image $\boldsymbol{i_t}$;
    Estimate the throughput $s'_t$ from $\boldsymbol{i_t}$ using AROW;
    if $s'_{t-1} < s'_{t-2} - \alpha$ and $s'_t > s'_{t-1} - \alpha$ then
        set filter_state ← ACTIVE and $T_{\mathrm{s}} \leftarrow t$, and
    $s'_{\mathrm{r}} \leftarrow s'_{t-1}$;
    else if filter_state = ACTIVE and $t > T_{\mathrm{s}} + T_m^{\mathrm{low}}$ then
        set filter_state ← IDLE;
    end if
    if filter_state = ACTIVE then
        Output $s'_{\mathrm{r}}$ as the estimated throughput;
    else
        Output $s'_t$ as the estimated throughput;
    end if
end for
```



mmWave BS	Dell Wireless Dock D5000
ML unit/Controller	Apple MacBook Air Mid 2013 (CPU: Intel Core i7, RAM: 8 GB)
mmWave STA	Dell Latitude E5540 (CPU: Intel Core i5, RAM: 4 GB)
RGB-D camera	Microsoft Kinect for Windows (Model: 1656)
Wired Network	1000BASE-T Gigabit Ethernet
WLAN	IEEE 802.11ad
Channel	62.64 GHz

Fig. 5. Experimental setup.

In addition, in LOS communications, except for low-throughput periods, the proposed scheme provided more accurate estimation than the comparison scheme. This is because the training dataset filtering in the low-throughput periods prevents the AROW algorithm from learning undesirable relationships between depth images and throughputs, as they are removed from desirable ones. From the above discussion, the proposed scheme provides more accurate throughput estimation than the scheme without filtering in both LOS communication and the low-throughput period.

The RMS errors in each experiment are shown in Table I. In the situation where the camera was placed at position A, the proposed scheme improved accuracy by about 9% to 25% compared with the scheme without filtering. The average execution time of the algorithm for learning and estimation from an image is 1.40ms and 1.36 ms, respectively, which is much shorter than a duration of a human blockage.

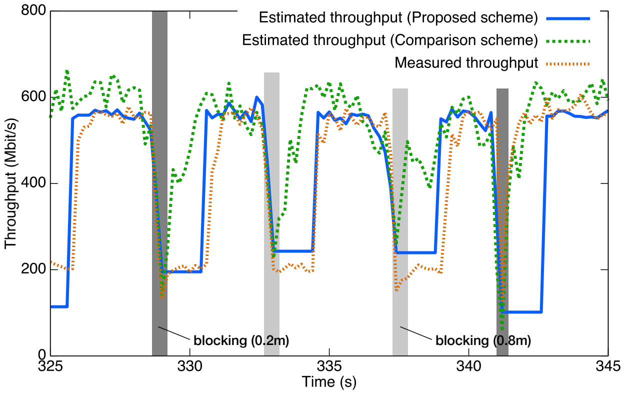


Fig. 6. Part of the measured and estimated throughput time series when the camera was placed at position A and a pedestrian walked along path r (the pedestrian walked alternately 0.2 m and 0.8 m in front of the BS).

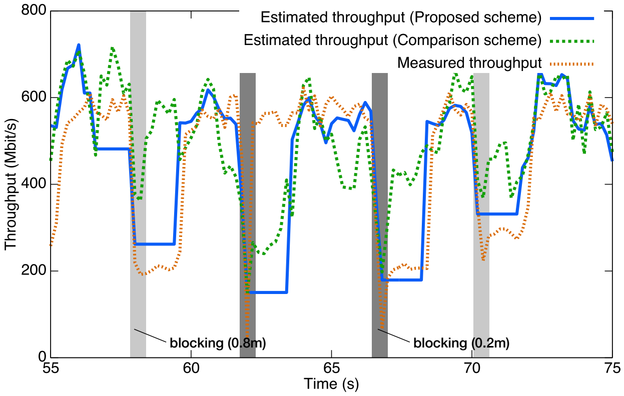


Fig. 7. Part of the measured and estimated throughput time series when the camera was placed at position B and a pedestrian walked along path r (the pedestrian walked alternately 0.2 m and 0.8 m in front of the BS).

Fig. 7 shows a part of the measured and estimated throughput time series when the camera was placed at position B and a pedestrian walked along path (r). As with the previous situation, estimated throughputs with the proposed scheme were more accurate than those using the comparison scheme. Table I shows that the proposed scheme improved accuracy by about 10% compared to the comparison scheme when the camera was placed at position B. These results suggest that the proposed scheme expected to be work well for other camera placements.

VI. Conclusion

This paper proposed a mmWave throughput estimation scheme using depth images based on a machine learning algorithm. Specifically, the relationship between depth images and measured throughputs were learned and the throughput was estimated only from depth images. In addition, we proposed a filtering of training dataset and estimation output to prevent estimation accuracy degradation caused by unexpected throughput degradation. We conducted proof-of-concept experiments using IEEE 802.11ad mmWave WLAN devices and an RGB-D camera, and confirmed the feasibility of the proposed scheme. Our experiments show that in a situation where the throughput is limited to 1 Gbit/s, the proposed scheme estimates throughput from depth images with RMS errors of 114–178 Mbit/s, which is up to 25% less than the scheme without filtering.

TABLE I
RMS error.

Position of Camera	Scheme & Reduction	Pedestrian Path		
		path p	path q	path r
A	proposed comparison	144 Mbit/s 188 Mbit/s	178 Mbit/s 196 Mbit/s	122 Mbit/s 162 Mbit/s
	reduction	23.4%	9.1%	24.7%
B	proposed comparison	144 Mbit/s 161 Mbit/s	114 Mbit/s 126 Mbit/s	140 Mbit/s 156 Mbit/s
	reduction	10.6%	9.5%	10.3%

References

[1] C. Dehos *et al.*, “Millimeter-wave access and backhauling: The solution to the exponential data traffic increase in 5G mobile communications systems?” *IEEE Commun. Mag.*, vol. 52, no. 9, pp. 88–95, Sep. 2014.

[2] P. Wang *et al.*, “Multi-gigabit millimeter wave wireless communications for 5G: From fixed access to cellular networks,” *IEEE Commun. Mag.*, vol. 53, no. 1, pp. 168–178, Jan. 2015.

[3] IEEE P802.11 Task Group ay, https://mentor.ieee.org/802.11/dcn/14/11-14-1151-08-ng60-ng60-proposed-par.docx.

[4] “Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications Amendment 3: Enhancements for Very High Throughput in the 60 GHz Band,” IEEE Std. 802.11ad, IEEE 802.11 Working Group, 2012.

[5] S. Collonge *et al.*, “Influence of the human activity on wide-band characteristics of the 60 GHz indoor radio channel,” *IEEE Trans. Wireless Commun.*, vol. 3, no. 6, pp. 2396–2406, Nov. 2004.

[6] B. Gao *et al.*, “Double-link beam tracking against human blockage and device mobility for 60-GHz WLAN,” in *Proc. IEEE WCNC*, Istanbul, Turkey, Apr. 2014, pp. 323–328.

[7] Y. M. Tsang and A. S. Y. Poon, “Detecting human blockage and device movement in mmWave communication system,” in *Proc. IEEE GLOBECOM*, Houston, Texas, USA, Dec. 2011, pp. 1–6.

[8] S. Singh *et al.*, “Blockage and directivity in 60 GHz wireless personal area networks: from cross-layer model to multihop MAC design,” *IEEE J. Sel. Areas Commun.*, vol. 27, no. 8, pp. 1400–1413, Oct. 2009.

[9] Y. Oguma *et al.*, “Proactive base station selection based on human blockage prediction using RGB-D cameras for mmWave communications,” in *Proc. IEEE GLOBECOM*, San Diego, CA, USA, Dec. 2015, pp. 1–6.

[10] N. Baccour *et al.*, “Radio link quality estimation in wireless sensor networks: A survey,” *ACM Trans. Sen. Netw.*, vol. 8, no. 4, pp. 34:1–34:33, Sep. 2012.

[11] C. M. Bishop, *Pattern Recognition and Machine Learning*. Secaucus, NJ, USA: Springer-Verlag New York, Inc., 2006.

[12] Kinect, https://dev.windows.com/en-us/kinect, Jan. 2016.

[13] J. Shotton *et al.*, “Real-time human pose recognition in parts from single depth images,” in *Proc. IEEE CVPR*, Colorado Springs, USA, Jun. 2011, pp. 1297–1304.

[14] L. Bo *et al.*, “Depth kernel descriptors for object recognition,” in *Proc. IEEE/RSJ IROS*, San Francisco, USA, Sep. 2011, pp. 821–826.

[15] K. Crammer *et al.*, “Adaptive regularization of weight vectors,” *J. Mach. Learn.*, vol. 91, no. 2, pp. 155–187, May 2013.

[16] W. Yin *et al.*, “Performance of mac80211 rate control mechanisms,” in *Proc. MSWiM*, Miami, Florida, USA, Oct. 2011, pp. 427–436.

[17] S. Yamashita *et al.*, “Rate adaptation based on exposure assessment using rectenna output for wlan station powered with microwave power transmission,” *IEICE Trans. Commun.*, vol. 98, no. 9, pp. 1785–1794, 2015.

[18] Iperf, http://iperf.fr, Jan. 2016.