Backward Explicit Congestion Control in Image Transmission on the Internet

Jeong Ha Kim[*] , Hyoung Bae Kim[**], Hak No Lee, and Boo Hee Nam[***]

Dept of Electrical and Computer Engineering, Kangwon National University
Chunchon, Kangwondo, 200-701, Korea
Tel : +82-33-250-6341 Fax : +82-33-242-2059
E-mail: [*]jkim@is.kangwon.ac.kr, [**] hkim@is.kangwon.ac.kr, [***]boonam@kangwon.ac.kr

Abstract-**In this paper we discuss an algorithm for a real time transmission of moving color images on the TCP/IP network using wavelet transform and neural network. The image frames received from the camera are two-level wavelet-trans formed in the server, and are transmitted to the client on the network. Then, the client performs the inverse wavelet-transform using only the received pieces of each image frame within the prescribed time limit to display the moving images. When the TCP/IP network is busy, only a fraction of each image frame will be delivered. When the line is free, the whole frame of each image will be transferred to the client. The receiver warns the sender of the condition of traffic congestion in the network by sending a special short frame for this specific purpose. The sender can respond to this information of warning by simply reducing the data rate which is adjusted by a back-propagation neural network. In this way we can send a stream of moving images adaptively adjusting to the network traffic condition.**

I. Introduction

The transmission control protocol/internetworking protocol (TCP/IP) is a protocol suite that defines how all transmissions are exchanged across the Internet and has been in active use and has demonstrated its effectiveness on a worldwide scale [1]-[3]. The TCP is a reliable and connection-oriented stream transport port-to-port protocol. Since the TCP/IP uses a CSMA/CD (carrier-sense multiple access with collision detection), the network can be slowed down according to the condition of the network traffic. When we send a stream of moving color images, the heavy load of moving images can be a big burden to the network.

Wavelet transforms are useful for image processing [4]-6]. Performing a one-step wavelet decomposition of an image gives four pieces of image: approximation portion, horizontal detail, vertical detail, and diagonal detail, where all four of approximation portion and details are of the same size. To perform a level 2 decomposition of the image gives seven pieces of data: an approximation portion of one-sixteenth size, three details of one-sixteenth size, and three details of one-fourth size. The sequence to be sent on the network from the sender is in the order of an approximation portion of one sixteenth size, horizontal detail of one sixteenth size, vertical detail of one sixteenth size, diagonal detail of one sixteenth size, horizontal detail of one fourth size, vertical detail of one fourth size, and diagonal detail of one fourth size. The reason for this sending sequence is that we can get a better reconstructed image using the received pieces of image data within a prescribed time limit. In the worst case of the traffic, we send only the approximation portion of one sixteenth size. When the line is free, the whole frame of each image will be transferred to the client.

The neural network (NN) is particularly useful for solving problems that use imprecise or fuzzy input patterns and have abundant example data [7]-[8]. When a large amount of input/output data is available, but you're not sure how to relate it to the output, with a NN, you simply show it: "this is the correct output, given this input". With an adequate amount of training, the network will mimic the function that you are demonstrating. In this paper, the neural network will be used to determine the amount of the image data to be sent on the communication network from the sender.

The receiver warns the sender of the condition of traffic congestion in the network by sending a special frame for this specific purpose. The sender can respond to this information of warning by simply reducing the data rate which is adjusted by a back-propagation neural network. In this way we can send a stream of moving images adaptively adjusting to the network traffic condition.

This paper describes and implements an algorithm of real-time transmission of moving image using wavelet transform and the NN. In section II, the overall scheme is presented. In section III, the wavelet transform for images and the neural network are dealt. In section IV, we show the results, followed by the conclusion.

II. Configuration Of Moving Image Transmission

Fig. 1 describes the overall diagram of moving image transmission system. First, we obtain a moving image data by CCD (Charge-Coupled Device) camera at a rate of 15 frames per second. Each image frame is of the size 160 by 120 with RGB colors. And we perform the two-step 2D (two dimensional) wavelet transform for them. Then we obtain a set of wavelet coefficients for each frame of moving image data. Next, we construct a proposed packet stream from approximation data to details data. And then, we transmit in real-time the moving image data to client on the TCP/IP network. When the network traffic is heavy, the client can’t receive all the image data. In this case, we decrease data rate of sending packet using the NN to cope with the network traffic.

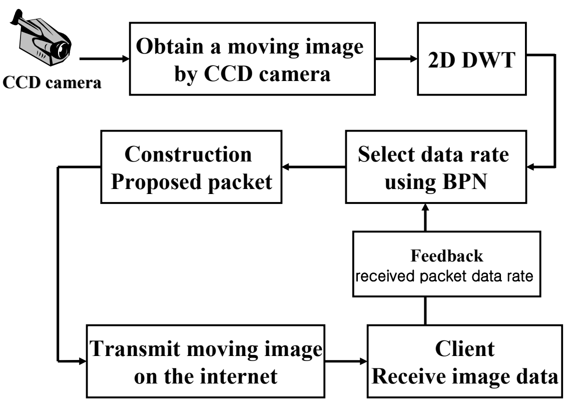


Fig. 1. System diagram.

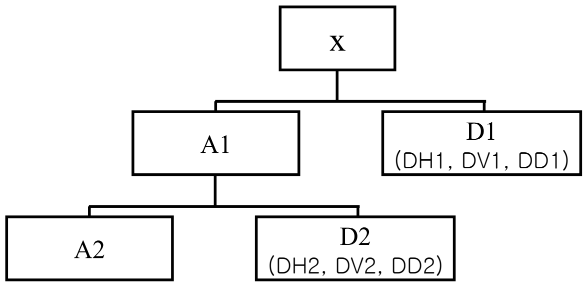
III. WAVELET TRANSFORM OF MOVING IMAGES

In this section, we perform a two-level 2D wavelet transform for an image data achieved from CCD camera, and we propose an algorithm to transmit the moving images.

A. Wavelet Transform

We will use the DWT to get the approximation and details from the original image. DWT analyzes the signal at different frequency bands with different resolutions by decomposing the signal into a coarse approximation and detail information. DWT employs two sets of functions, called scaling functions and wavelet functions, which are associated with low pass and highpass filters, respectively. The decomposition of the signal into different frequency bands is simply obtained by successive lowpass and highpass filtering of the time domain signal.

Fig. 2 shows a scheme to decompose the image into an approximation coefficient and detail coefficients using the wavelet transform. In Fig. 2, A1 corresponds to one-fourth of the original image data. If we continually perform the 2D wavelet transform like this, the image quality will be lowered due to the loss of partial data in an image. In this paper, we perform the 2D wavelet transform for an original image twice. Fig. 4 shows a transformed image performed through 2D DWT for the image shown in Fig. 3.



$$X = A1+D1$$
$$= A2+D2+D1 ,$$
where $D1 = DH1+DV1+DD1$
$D2 = DH2+DV2+DD2$

Fig. 2. 2D wavelet transform of image data.

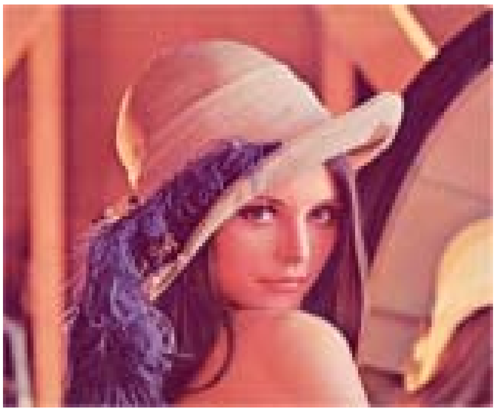


Fig. 3. Original image data

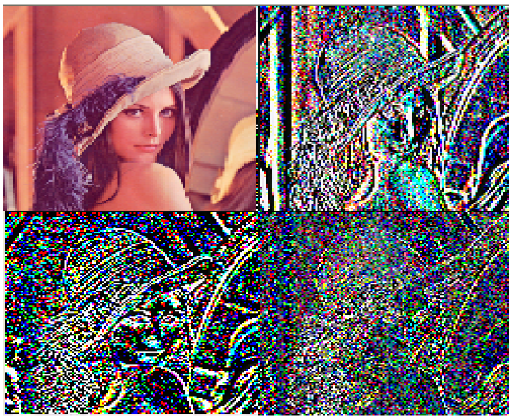


Fig. 4. Image data after one-level 2D wavelet transform

B. Construction of sending sequences of image packets

Fig. 5 shows a set of image pieces through 2D wavelet transform. To perform a level 2 decomposition of the image gives seven pieces of data: an approximation portion of one-sixteenth size, three details of one-sixteenth size, and three details of one-fourth size. The sequence to be sent on the network from the sender is in the order of ①: an approximation portion of one sixteenth size, ②: horizontal detail of one sixteenth size, ③: vertical detail of one sixteenth size, ④: diagonal detail of one sixteenth size, ⑤: horizontal detail of one fourth size, ⑥: vertical detail of one fourth size, and ⑦: diagonal detail of one fourth size. The reason for this sending sequence is that we can get a better reconstructed image using the received pieces of image data within a prescribed time limit and easily change the sending data rate with good image quality. These seven pieces of image data are variable-sized packets.

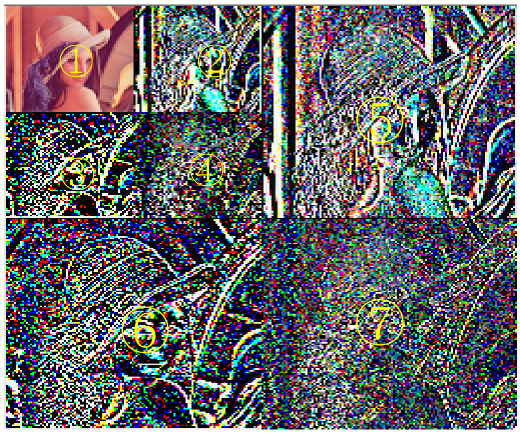


Fig. 5. Image data after perform 2D wavelet transform twice

In Fig. 6, after we perform the level-2 DWT, we construct a sequence of packets as shown by the arrow direction from approximation data to detail data ① through ⑦. We send as many packets as the network traffic is allowed within the sampling time. With all the seven packets, we can reconstruct the original image as shown in Fig. 6. Fig. 7 shows the reconstructed image with packets ① through ⑥. Fig. 8 shows the reconstructed image with packets ① through ⑤. Fig. 9 shows the reconstructed image with packets ① through ④. Fig. 10 shows the reconstructed image with packets ① through ③. And Fig. 11 shows the reconstructed image with packets ① through ②. Fig. 12 shows the reconstructed image with packet ① only. Each image of Fig. 7, 8, 9, 10, 11 and 12 loses one-fourth, two-fourths, three-fourths, 13 sixteenths, 14 sixteenths and 15 sixteenths of the original image, respectively. Even if the network traffic congestion happens, we can transmit the packet in real-time by changing the data rate using the variable sized packets.

C. Neural network to determine the sending data rate

The neural network is adjusted, or trained, so that a particular input leads to a specific target output. Fig. 13 shows backpropagation network. Generally, multilayered neural network which has error backpropagation algorithm is used in the pattern recognition and system recognition. In this paper, backpropagation neural network will be used to infer the network speed. To calculate the network weights, first we compute the output.
The output y can be expressed as:

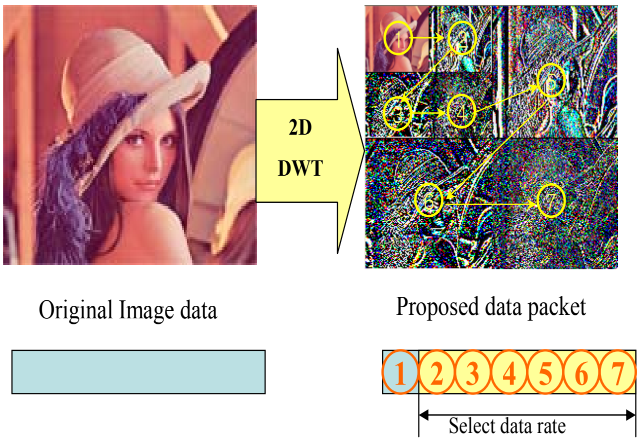


Fig 6. Proposed packet

Fig 7. Using packet ①−⑥

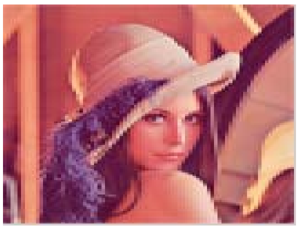


Fig 8. Using packet ①−⑤

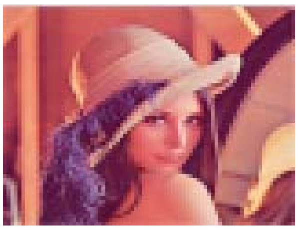


Fig 9. Using packet ①−④

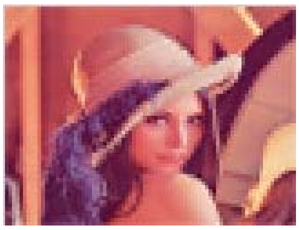


Fig 10. Using packet ①−③

Fig 11. Using packet ①− ②

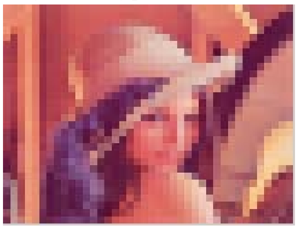


Fig 12. Using packet ① only

$$y = f(\psi); \quad \psi = \sum_{j=1}^{M} w_j h_j \qquad (1)$$

where

$$h_j = g(\phi_j); \quad \phi_j = \sum_{k=1}^{N} c_{jk} x_k; \quad j = 1,2,\cdots,M \qquad (2)$$

And then we compute the error by comparing the output and the target. Finally, we adjust the weights of the output layer and hidden layer using the error back-propagation.

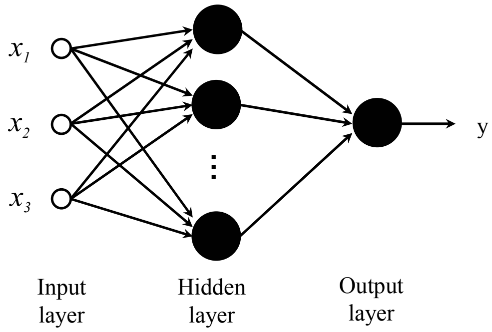


Fig. 13. backpropagation network with N=3

Equation (3) and (4) show the expressions to adjust the weights of the output layer and hidden layer.

$$w_j(n+1) = w_j(n) + \eta\delta_y(n)h_j(n) \qquad (3)$$

where δ_y : error of output layer's neuron

$$c_{jk}(n+1) = c_{jk}(n) + \eta\delta_{hj}(n)x_k(n) \qquad (4)$$

where δ_{hj} : error of hidden layer's jth neuron.

Selection of data rate is affected by the network speed. We assume that we send the 15 image frames a second. Within 50 ms, one image frame is captured from the camera, is 2-level wavelet-transformed, is transmitted on the internet, is inverse wavelet-transformed and displayed on the receiver. Also, we assume the maximum speed of the network is 10 Mbps, when we can send all 7 pieces of the image data which is 2-level wavelet transformed. When the network speed is low (1 Mbps), we send only the approximation portion whose size is one-sixteenth of the image frame. Depending on the network speed is increasing, decreasing or maintaining the same speed, the sender adjusts adaptively the number of data pieces of the image frame.

Table 1 shows the data rate which can transmit without delay to be coped with network traffic. If network speed is 2M bps, we can transmit the image packets consisting of data numbering 1, 2 and 3 to the client without any delay within the sampling time 50ms.

We need a training data to train a neural network. In this paper, we use a training data which has three inputs x_1, x_2, x_3 and one output y, to train a neural network. Input x_1 is the data rate which the server sends the data packets, input x_2 is the data rate which the receiver receives the data packets within a prescribed time, and input x_3 is a flag for the existence of delay. If there is a delay, the flag is 0, and 1 otherwise. The output y is the data rate which the server will send in the next sampling time.

TABLE 1

DATA RATE BEFFITTING NETWORK SPEED

Internet speed (Mbps)	Data rate which is able to translate for 50ms (bit)	Number of packet data	Data rate of packet(bit)
1	50000	1	28800
1.5	75000	1,2	57600
2	100000	1,2,3	86400
3	150000	1,2,3,4	115200
5	250000	1,2,3,4,5	230400
7	350000	1,2,3,4,5,6	345600
10	500000	1,2,3,4,5,6,7	460800

Table II shows the training data. For example, for a set of dada {4, 4, 1 and 5}, The first number '4' is x_1. The second number '4' is x_2 , and the third number '1' is x_3 , which represents no time delay in this case. It means that if the server transmits the data packets ① through ④ to the client, the client can receive the whole pieces of image data within a prescribed time without any delay and the server might send the data packets ① through ⑤ of the image frame.

We train the neural network using the data of Table II until the error becomes 10^{-5} or below. The training time is about 10 seconds. Fig. 14 shows a diagram of the neural network system.

The TCP is responsible for the reliable transmission of data from one node to another. It is a connection-based protocol and establishes a connection between two machines before any data is transferred. We send the packets of image data determined by the neural network according to the network traffic and the amount of the packets to be sent.

TABLE II

TARAINING DATA

Input	x_1	1 1 1 2 2 2 2 3 3 3 3 4 4 4 4 4 5 5 5 5 5 6 6 6 6 6 6 7 7 7 7 7 7 7
	x_2	1 1 0 2 2 1 1 3 3 2 1 4 4 3 2 1 5 5 4 3 2 6 6 5 4 3 2 7 7 6 5 4 3 2
	x_3	1 0 0 1 0 1 0 1 0 1 1 1 0 0 0 0 1 0 0 0 0 1 0 0 0 0 0 1 0 0 0 0 0 0
Output	y	2 1 1 3 2 1 1 4 3 2 1 5 4 3 2 1 6 5 4 3 2 7 6 5 4 3 2 7 7 6 5 4 3 2

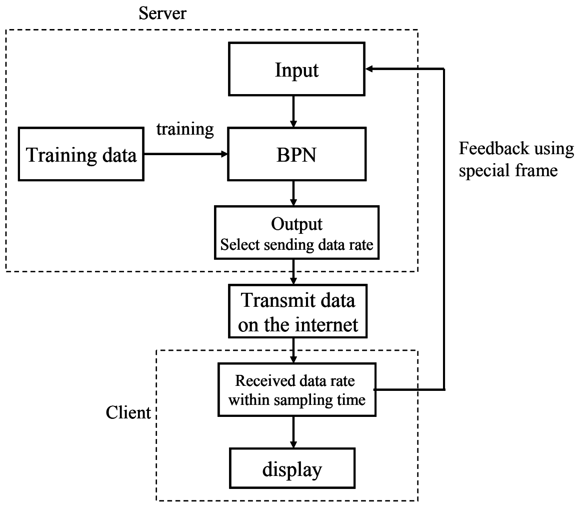


Fig 14. A diagram of neural network system

IV. EXPERIMENT RESULT

We use the result for the sending rate of image packets determined by the neural network. The upper part in Fig. 15 shows that when the network speed is increasing, the data rate of the sender is accordingly increasing, corresponding to the network speed. The lower part in Fig. 15 shows that when the network speed is increasing, the data rate of the receiver is accordingly increasing, corresponding to the network speed. Fig. 15 also shows the case where the network speed is changing abruptly up and down and decreasing. Using the neural network, the probability of selected data rate which is subject to the network speed is 99% or more. In the system, however, when we use the if-sentence, the probability was below 98%. So, if we use our proposed method, we can transmit a stream of moving images without disconnection on real-time. Experiments on the network sending the moving images show the validity of the proposed method.

V. CONCLUSION

In this paper, we propose a method to transmit the moving image in real time using the discrete wavelet transform and the neural network on the internet, and implement the algorithm to obtain a satisfactory result. The image frames received from the camera are two-level wavelet tr ansformed in the server, and are transmitted to the clien t on the network. Then, the client performs the inverse

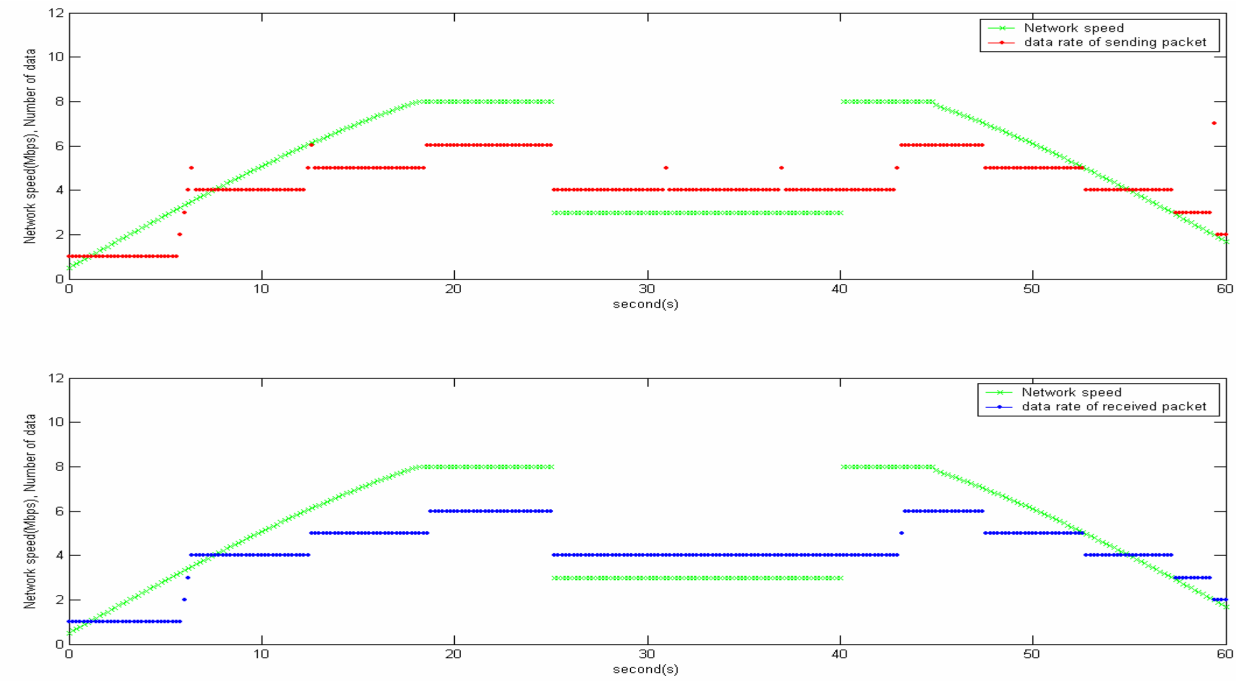


Fig 15. A Result of Experiment

wavelet-transform using only the received pieces of each image frame within the prescribed time to display the moving images. When the TCP/IP network is busy, only a fraction of each image frame will be delivered. When the line is free, the whole frame of each image will be trans ferred to the client. The receiver warns the sender of the condition of traffic congestion in the network by sending the special frames for this specific purpose. The sender can respond to this information of warning by controlling the number of pieces which is determined by a back-propagation neural network. In this way we can send a stream of moving images adaptively adjusting to the network traffic condition.

ACKNOWLEDGMENT

This paper was supported by BK21(Brain Korea 21) Project of Kangwon National University in 2003.

REFERENCES

[1] A. Ben-Artzi, A. Chandna, and U. Warrier, "Network management of TCP/IP networks: present and future," IEEE Network , Volume 4, Issue 4, pp. 35 –43, July 1990,

[2] P. P. Dang, and P. M. Chau, "Robust image transmission over CDMA channels," IEEE Transactions on Consumer Electronics, Volume 46, Issue 3, pp.664-672, Aug 2000.

[3] P. M. Corcoran, "Mapping home-network appliances to TCP/IP sockets using a three-tiered home gateway architecture," IEEE Transactions on Consumer Electronics, Volume 44, Issue 3, pp. 729 –736, August 1998.

[4] R. A. Devore, B. Jawerth, and B. J. Lucier, "Image compression through wavelet transform coding," IEEE Tr. On information theory, pp. 719-746, March 1992.

[5] Juan Liu and P. Moulin,"Information-theoretic analysis of interscale and intrascale dependencies between image wavelet coefficients," IEEE Transactions on Image Processing, Volume 10, Issue 11, pp.1647-1658, Nov 2001.

[6] A. N. Sarlashkar, M. Bodruzzaman, and M. J. Malkani, "Feature extraction using wavelet transform for neural network based image classification," Proceedings of the Thirtieth Southeastern Symposium on System Theory, pp. 412 –416, 8-10 March 1998.

[7] R. Hecht-Nielsen, "Theory of the backpropagation neural network," IJCNN., International Joint Conference on Neural Networks, Vol.1, pp. 593-605, 18-22 Jun 1989.

[8] W. Chang, B. Bosworth, G.. C. Carter, "On using back-propagation neural networks to separate single echoes from multiple echoes," ICASSP-93 IEEE International Conference on Acoustics, Speech, and Signal Processing , Volume 1, pp.265-268, 27-30 Apr 1993.