A Novel Method of Automatic Training Data Selection for Estimating Missing Well Log Zone Using Neural Networks[+]*

Yingwei Yu[1], Douglas Seyler[2], and Michael D. McCormack[3]

Search and Discovery Article #41055 (2012)
Posted October 29, 2012

*Adapted from extended abstract prepared for poster presentation at AAPG International Conference and Exhibition, Singapore, 16-19 September, 2012; modified from publication of Seismic Micro-Technology, Inc. (http://www.ihs.com/ar/images/Estimating_Missing_Well_Log_Zone_Using_neural_network.pdf).

[+]U.S. Patent No. 8,065,086 (issued November, 2011)

[1]IHS Global, Inc., Houston, TX (yingweiy@gmail.com)
[2]Blueback Reservoir Americas, Houston, TX
[3]Olympic Geoscience Consulting, Seattle, WA

Abstract

Well-log interpolation is of great importance to understand the lithology and history of a sedimentary basin. A neural network-based algorithm is presented that estimates missing intervals of log curves using log curves from the same borehole and other nearby wells. In general, for approaches using neural networks, the accuracy of the prediction is highly dependent on the quality of the training data. The user has to select the training data manually, but this method is time-consuming, and sometimes can be inaccurate. A unique feature of this algorithm is the extensive analysis and preprocessing of all candidate wells and their log curves to determine the set of wells, curves, as well as log data samples within each curve that will yield the best estimate of the missing log interval. This refined set of curves and data samples are used to train a neural network that then estimates the missing log interval. An additional output of this analysis is a confidence value for each estimated log sample that provides a qualitative measure of the accuracy of the neural network prediction.

Introduction

There are occasions during the acquisition or processing of borehole logs that portions of log data are lost, corrupted, or missing. As borehole logs provide a critical source of subsurface information for interpretation, it is important that a method be available to accurately reconstruct the lost data. The concept of recovering information in gaps in well logs has been discussed extensively in the literature (Hampson et al., 2001, Russell et al., 2002, Herrera et al., 2006, Bhatt and Helle, 2002, Holmes et al., 2003), and has been implemented in at least two commercial products. The technology in these applications relies heavily on the expertise of a trained log analyst to manually select training data to estimate missing log sections. The selection of training data is generally observed at the following three levels:

- **Well selection**: select the set of related wells from a neighborhood that can be used as training wells.
- **Curve selection**: select a set of curves in a well that can be used as training curves.
- **Sample selection**: in one well, a sample is defined as a vector with multiple data values from a given set of curves at a given depth. The sample selection is to select a set of samples which is related to the available data in the missing zone and used as the training samples.

In this article, we describe a new method that uses multiple log types and boreholes to reconstruct missing sections of log curves by automatic selection at the above three levels (see "Intra-well Interpolation" below for curve selection and sample selection, and "Inter-well Interpolation" below for well selection). The approach described here automates this data selection through the application of novel data analysis techniques which:

- increase the speed with which these calculations can be performed.
- obviate the need for users to have domain knowledge in advanced computing technologies, such as neural networks.
- provide a confidence measure that informs the log analyst of the reliability of the reconstructed well-log zone.

In the next section we describe the new methodology of training data selection. In the following section, we will present examples of the application of the technique to recover missing sections of a log curve.

Methods

There are two sub-problems of well-log interpolation: intra-well and inter-well interpolation. Intra-well interpolation estimates a depth interval of missing data (called a gap) in a log curve using other curve types in the same well. We discuss the methods of curve selection and sample selection in the intra-well subsection. Inter-well interpolation estimates a gap using all available curve types from the well with the missing data plus corresponding curves from neighboring wells, and it is built upon the intra-well interpolation algorithm.

Intra-well Interpolation

A well has N curves, and the m^{th} curve is called *the target curve* if it contains a gap at zone **Z**, while all other curves are called *the input curves*, which are denoted by a vector ***a***. **Z** is called the *testing zone*. $\overline{Z}$ refers to a zone outside **Z** and is called to the *training zone*. Let $D_{x,y}$ denote a matrix defined by log curves **x** in an arbitrary zone **Y**. Therefore, the task of intra-well interpolation is to learn the correlation between the training data (i.e., the matrix $D_{a,\overline{Z}}$) and the target values (i.e $D_{m,\overline{Z}}$), and predict data in the testing zone $\hat{D}_{m,Z}$ (i.e., the prediction).

The intra-well interpolation algorithm includes the following three steps: pre-processing, prediction with General Regression Neural Network (GRNN) (Specht, 1991), and post-processing. The input of the algorithm contains three parts: the well curves ***D*** to interpolate, the target curve m, and the testing zone Z. The algorithm outputs the predicted curve, $\hat{D}_{m,Z}$ with a confidence measurement coded in a color bar (in Figure 3, a color bar is below the predicted curve).

Pre-processing

In the pre-processing step, we first normalize the curves, and then we apply curve selection and sample selection. The curve selection is aimed at selecting the best subset of available curves to predict the target curve. The purpose of sample selection is to select the set of samples that maximally correlates to the data in the testing zone.

Curve Selection

In curve selection, we select the curve subset that can best correlate to the target curve. First, we cluster these curves into groups based on the absolute value of the correlation coefficient. We choose one curve in each group that best correlates to the target curve. Hence for groups, there are candidate curves. We can apply a sequential search and random test with a subset of these candidate curves. In each random test, we eliminate the curve with the lowest correlation coefficient value to the target curve in the subset, and randomly use a portion of the data in the well as the training dataset and the testing dataset. We feed both of these random samples and the candidate curve subset into a general regression neural network. The performance of each candidate curve subset is evaluated by a score *S*:

$$S = w_0 + w_1\alpha = w_2\phi, \qquad (1)$$

where α is the normalized accuracy of the random test, ϕ the correlation coefficient of the predicted values to the actual values, and *w* the weight of each factor. Values for *w* are empirically determined. The searching algorithm selects the subset of curves with the highest score *S*.

The basic idea of this curve selection algorithm is that if an input curve is useful to the prediction of the target curve; then it either has a strong linear relationship or nonlinear relationship with the target curve. To address the linear relationship, we use the correlation coefficient. For the nonlinear relationship, we apply random tests with the neural network. The proposed method is more efficient than an exhaustive search of input curves as proposed by Russell et al. (2002). For example, for *N* curves in a well, the exhaustive search method has to test 2Ncombinations of the curve subsets. With our method, the number of random tests is only *N* times at most. The grouping of similar curves also efficiently reduces the redundant curves even before the random tests. Our sequential searching method preserves the curves with high correlation coefficients in the candidate subsets. The resulting curves with the highest heuristic score (Equation 1) may not be the optimized curve subset, but experimentally we observed that they are an efficient approximation to the optimum curve subset.

Sample Selection

The sample selection process is to identify samples in the training zone that reside near a testing sample point in space (Figure 1). This figure is an *n*-dimensional plot, where *n* is the number of curves in the selected curve set. Each red or blue point in this figure corresponds to a vector, constructed from the selected log curves at a particular depth in the borehole. This step has two benefits. First, it uses only the subset of all available samples that are most likely to yield the best estimate of missing data in the target curve. Second, by reducing the number of samples, computing performance is enhanced.

Now the problem is to check if a sample is close to a given testing sample. We do this selection by calculating their Euclidean distance. Mathematically, a sample x = $D_{b,z}$ is selected for a testing sample $y = D_{b\bar{z}}$ when:

$$d(\mathbf{x},\mathbf{y}) < \theta, \qquad (2)$$

where d (x,y) denotes the distance between training sample **x** and testing sample ***y*** and θ is a threshold. Note that each testing sample has a set of selected training samples, which are enclosed by a hypersphere with diameter θ.

Russell et al. (2002) firstly employed a general regression neural network (GRNN) to interpolate the missing zone in a well log. The use of GRNN to predict data with GRNN can be summarized as follows:

Let b be the subset of selected curves from the curve selection step, and P the set of selected samples for testing sample y from the sample selection step. The training dataset for the GRNN is $D_{b,P}$, while the target for training data is $D_{m,P}$, and the testing data is a sample y in $D_{b,Z}$. The GRNN calculates the underlying relationship between the training data and their target values, then utilizes this relationship to make predictions for the testing data (see Specht, 1991) for details of the GRNN algorithm):

$$\text{GRNN}\ (y;\ D_{b,P},\ D_{m,P}) \rightarrow \hat{v}, \qquad (3)$$

where $\hat{v}$ is the prediction for the testing samples in the missing zone.

Post-processing

The predicted values of the missing section of log by GRNN must first be de-normalized to the original log scale. The accuracy of GRNN prediction depends on the presence of enough samples in the neighborhood. If a prediction is made at the location in the high dimensional space where there are few observed samples, the confidence of such a prediction is low. Similarly, if a testing point has a high density of observed samples in its neighborhood, the confidence is high. The confidence $\mathbf{c}_\mathbf{y}$ for each testing sample ***y*** is defined as

$$\mathbf{c}_\mathbf{y} = \sum_{x \in P} (1 - d(x, y)), \qquad (4)$$

where ***P*** is the set of selected samples for testing sample ***y***.

Inter-well Interpolation

Inter-well interpolation predicts the missing section with all available curves in surrounding wells together with the target well (i.e., the one with missing section to be interpolated). We call the area in which wells are to be used in the analysis the “neighborhood". As a general notation in this section, assume the neighborhood **W** has *n* wells. Let one of the wells with index *k* in **W** be the target well. The desired output is an interpolated log curve *m* in the testing zone **Z**. To interpolate with multiple wells, a “pair and merge" method was invented and is described as follows.

The “Pair and Merge" Method

The interpolation for multiple wells uses a new method called “pair and merge". The algorithm iteratively selects one of the wells, say well *i*, and pairs it with the target well *k* utilizing a curve name alias table. After the pairing process, the curves in wells *i* and *k* can be treated as if they come from one well. Therefore, the intra-well interpolation can be applied to get one prediction $\mathbf{P}_i$. For *n* wells in the neighborhood **W**, we can get *n* predictions. Finally, a curve merging algorithm merges those predicted curves into one final prediction.

There are three advantages of this “pair and merge" method:

1. Only two wells are loaded into memory at any given time, resulting in significant decreases in the amount of memory required even when numerous wells are in the neighborhood.
2. Each pair of wells is processed individually, producing data that is more coherent than using multiple wells at one time.
3. Processing for each pair of wells is independent, thus making the algorithm easy to implement in parallel for increased speed.

The pairing process is applied for wells when $i \neq k$. For the target well itself ($i = k$), this procedure can be skipped. This process includes two steps: (1) find log curves common to wells *i* and *k*, and (2) shift curves to common means. After the pairing process, apply the intra-well interpolation algorithm, and record the curve selection score S_i for well *i*, which is to be used in the merge process.

In the merge process, we first apply *well selection*. Only wells with a curve selection score above a certain threshold (e.g., 0.75) are selected. The well selection process can effectively eliminate unrelated wells, and improve the final accuracy. For the selected wells, the confidence for each prediction can be used to generate the final prediction. One can sum up the predictions by using their confidence as weights (Equation 5).

$$\boldsymbol{q}_i = \sum_{j=1}^{n} (C_{i,j} P_{i,j}) \,, \qquad (5)$$

where *n* is the number of automatically selected wells, *j* the index of selected wells, *i* the index of testing samples, *q* the merged prediction, ***C*** the confidence matrix, and ***P*** the prediction matrix. Finally, the overall confidence measurement ***m*** for each prediction can be the sum of the individual confidence values (Equation 6).

$$\boldsymbol{m}_i = \sum_{j=1}^{n} C_{i,j} \qquad (6)$$

Results

In this section we demonstrate two applications of the new algorithms using a portion of a field with 12 wells (see Figure 2). In the first example, we predict a “missing" section of the ρ_b log curve in the interval from 808-892*m* in target well 2.

Log curves from both the target and surrounding 11 wells were analyzed to determine the wells that were best candidates for predicting the missing zone in the target well. Wells 1, 2 (in regions outside of the missing zone) and 5 were selected in this curve analysis and used to train the neural network to estimate the ρ_b log in the missing interval. Since we actually had ρ_b data in our missing zone, we can compare the estimated ρ_b log curve (blue) with the real ρ_b log curve (red) in the missing zone. The results, shown below in Figure 3, show a good qualitative correlation between the two curves.

The confidence of the prediction is coded in the color bar at the bottom of Figure 3. Maroon represents high confidence, green is intermediate and blue indicates low confidence. Quantitatively, the average error difference between the two curves is 2% and the correlation coefficient is 84.61%.

It is interesting to note that the algorithm selected wells 1 and 5 for training the neural network that were 6.7 km and 4.1 km away from the target well and excluded several closer wells. Intuition would lead one to assume that wells nearer the target well are more likely to have similar petrophysical properties, and hence be chosen for training. This is not the case in this example. The curve and sample selection processing rigorously culls through all the wells and selects only the best matching data to use for neural network training.

The second example illustrates the use of the new algorithm to create a synthetic log in a borehole, i.e., set the missing zone to be whole depth of the well. We will use the same well 2 as in example 1 and apply the inter-well interpolation algorithm to the whole sonic transit time log (DT). Well 2 has a recorded sonic log that we will use for comparison with the predicted sonic log. The analysis follows the identical steps as in example 1, except that no sonic log information in well 2 is incorporated into the computations. Thus the estimate of the sonic log in well 2 is derived solely from logs in the other 11 wells in this field.

Figure 4 compares the estimated sonic log (blue) with the original sonic log in this well (red). Qualitatively, there are differences between the two log curves - the most noticeable being the lower amplitude excursions of the predicted curve as compared to the actual log curve. The average error difference over the entire sonic log interval is less than 6.5% and the correlation coefficient is 74.07%.

Conclusion

In this article, we demonstrated a new algorithm which can automatically select the training data for well-log missing zone interpolation. The data selection is carried out in three levels--well selection, curve selection, and sample selection. All of these pre-processing techniques can significantly improve the performance and the accuracy. The inter-well interpolation algorithm can be simply extended to synthesize a well log curve as well.

Acknowledgments

The authors wish to thank Hank Chambers for helpful discussions and the management of IHS for their encouragement and support of this work.

References

Bhatt, A., and H.B. Helle, 2002, Committee neural networks for porosity and permeability prediction from well logs: Geophysical Prospecting, v.50, p. 645–660.

Hampson, D.P., J.S. Schuelke, and J.A. Quirein, 2001, Use of multiattribute transforms to predict log properties from seismic data: Geophysics, v. 66, p.220–236.

Herrera, V.M., B.H. Russell, and A. Flores, 2006, Neural networks in reservoir characterization: The Leading Edge, v. 25, p. 402–411.

Holmes, M., D. Holmes, and A. Holmes, 2003, Generating missing logs - techniques and pitfalls (extended abstract): AAPG Annual Meeting, Salt Lake City, Utah. Search and Discovery Article #90013 (2003), Web accessed 15 October 2012.
http://www.searchanddiscovery.com/abstracts/pdf/2003/annual/extend/ndx_79985.PDF

Russell, B.H., L.R. Lines, and D.P. Hampson, 2002, Application of the radial basis function neural network to the prediction of log properties from seismic attributes: CREWES Research Report 2002, v. 14, 1-21 p.

Specht, D.F., 1991, A general regression neural network: IEEE Transactions on Neural Networks, v. 2:6, p. 568–576.

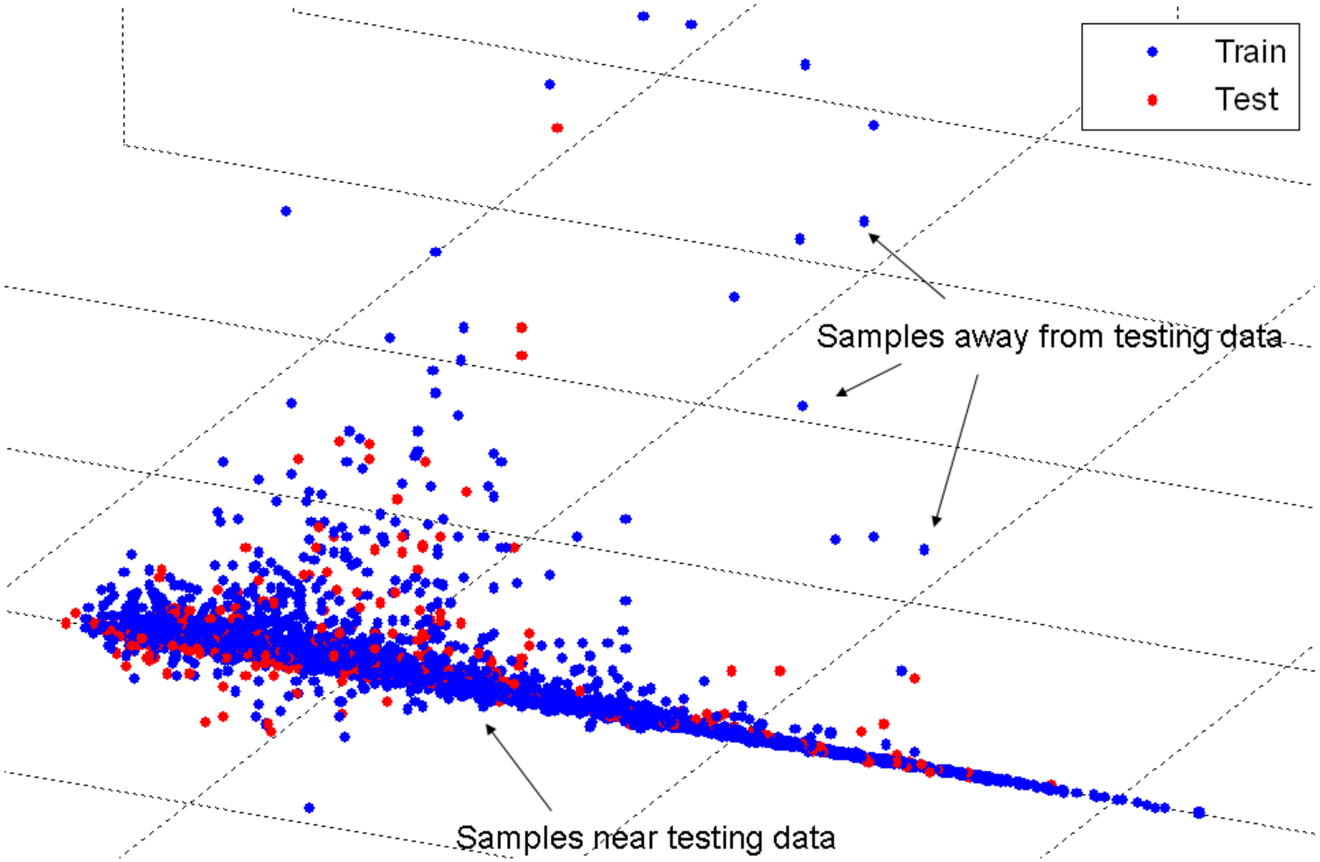

Figure 1. Sample selection of the training data. Samples of both training data (blue) and testing data (red) are projected as points in high-dimensional space (as shown in 3-D here). The sample selection process of the training data preferentially utilizes samples near (or even mixed with) the testing data and avoids using samples far away.

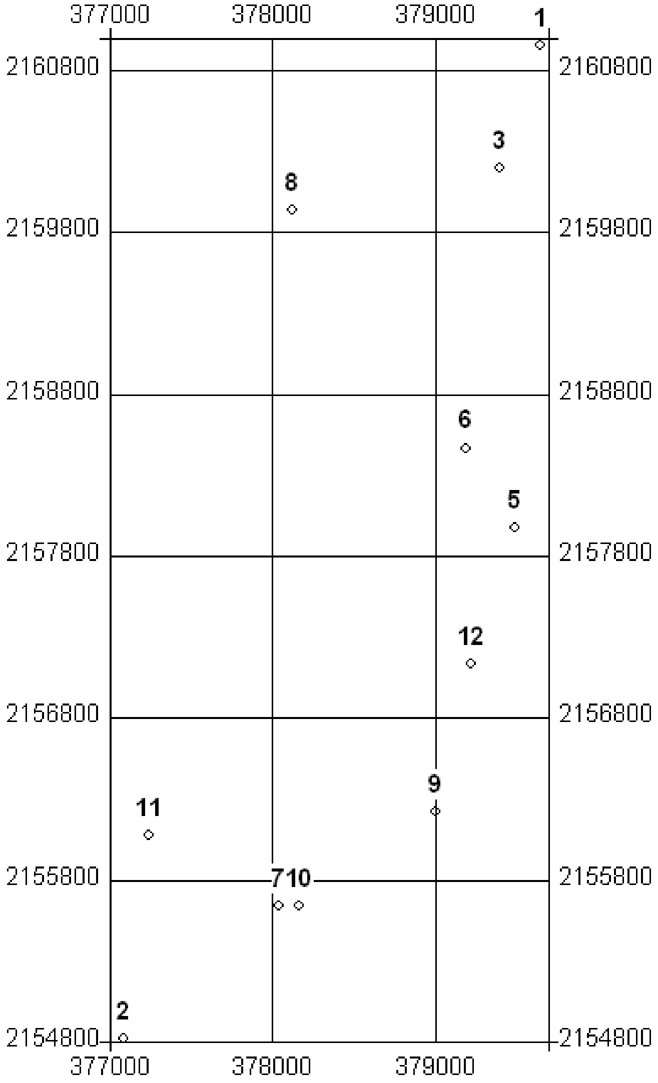

Figure 2. Base map with well locations. All units are in meters.

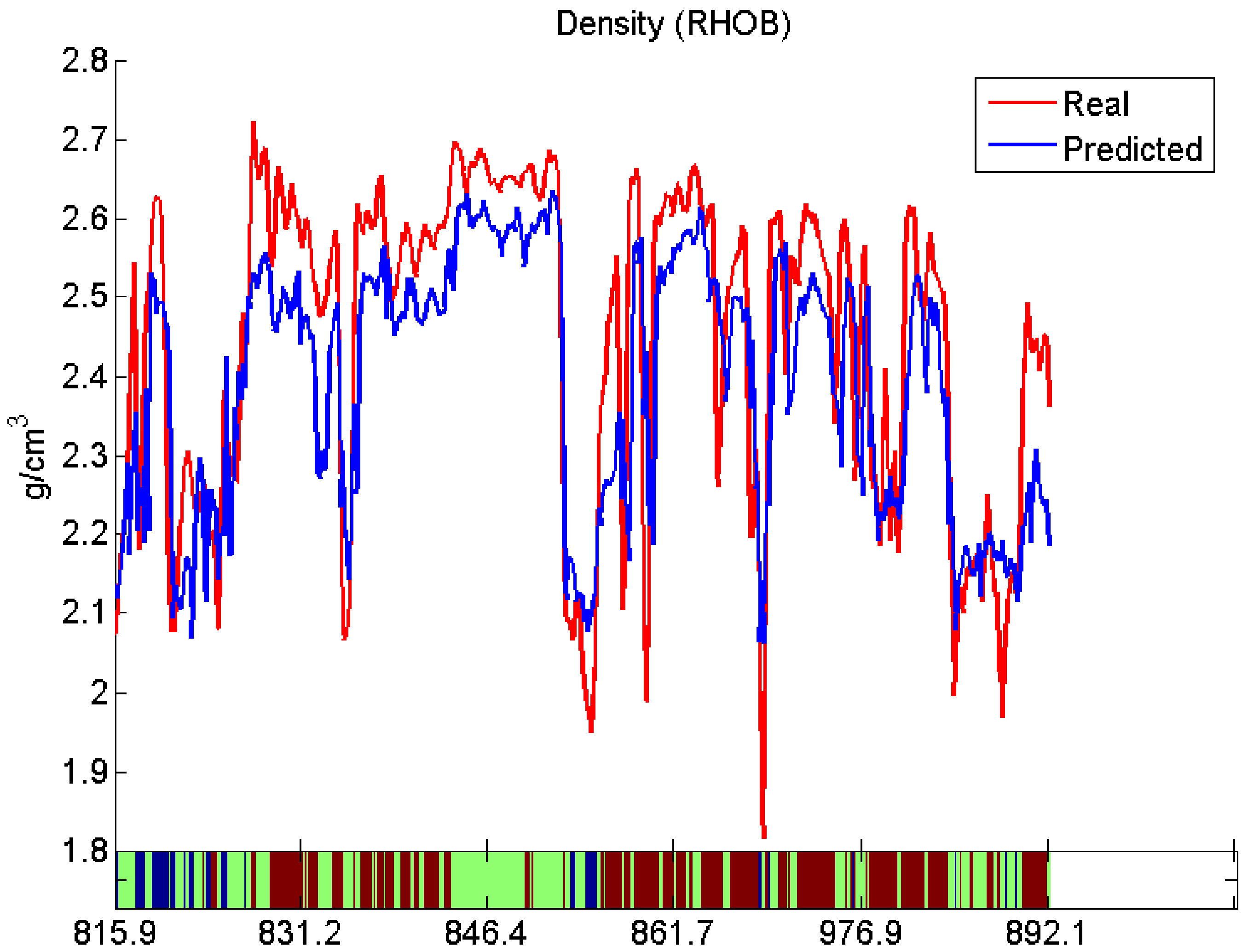

Figure 3. Predicted curve with confidence measurement**.** The blue curve is the predicted ρ_b log, and the red curve is the recorded ρ_b log. Units are in *g/cm*^{3}. The confidence of the prediction is coded in the color bar at the bottom. Maroon represents high confidence; green is intermediate; and blue indicates low confidence.

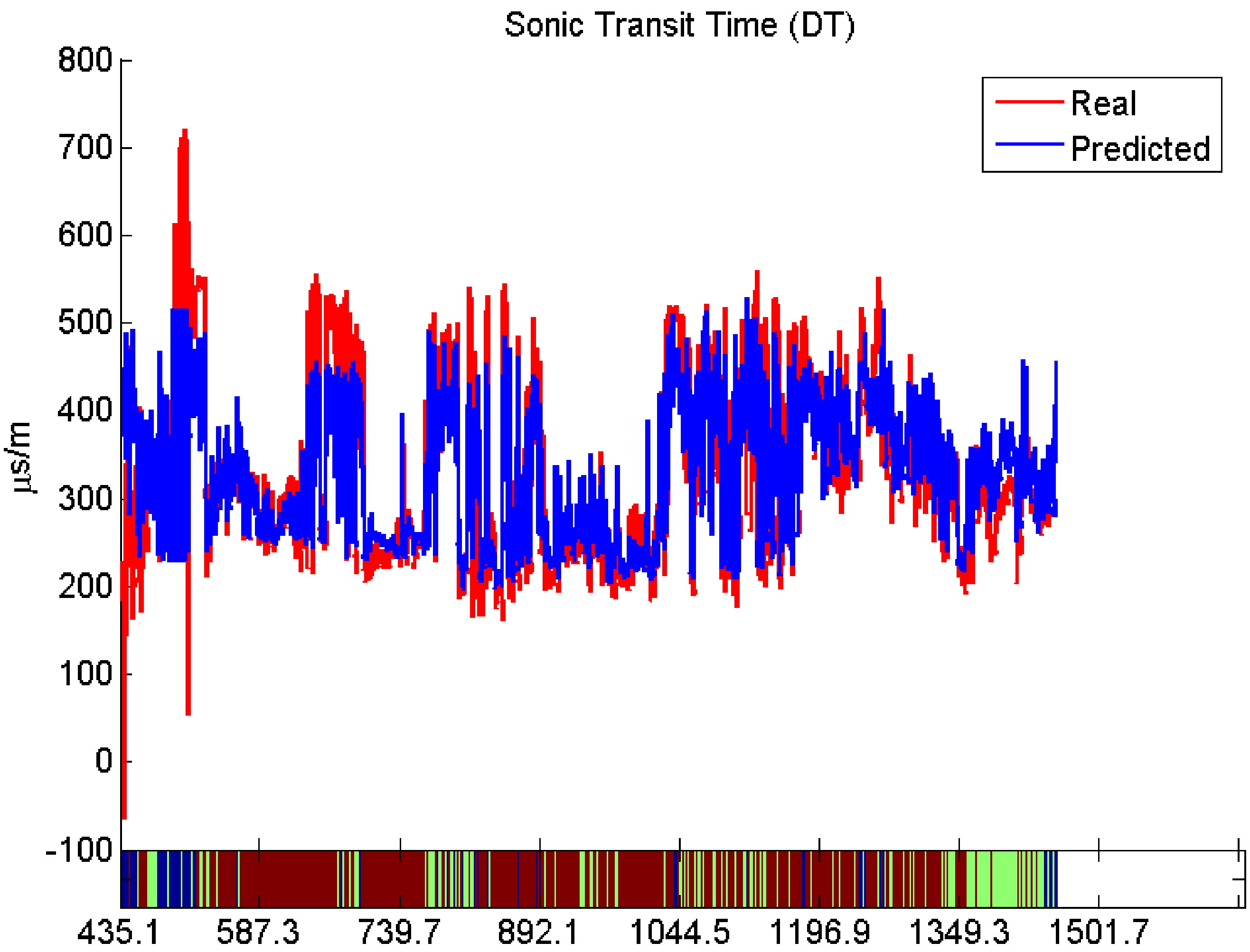

Figure 4. Predicted synthetic curve with confidence measurement. The blue curve is the predicted synthetic DT curve, and the red curve is the actual DT curve. Units are in *µs/m*. The confidence of the prediction is coded in the color bar at the bottom. Maroon represents high confidence; green is intermediate; and blue indicates low confidence.