공학박사학위논문

# 품질 조절이 가능한 센서 데이터의 스케일러블 부호화 분석

# ANALYSIS FOR SCALABLE CODING OF QUALITY-ADJUSTABLE SENSOR DATA

2014 년 2 월

서울대학교 대학원

전기·컴퓨터공학부

이 동 은

# Abstract

## Analysis for Scalable Coding of Quality-Adjustable Sensor Data

Dongeun Lee

Department of Electrical Engineering and Computer Science

College of Engineering

Graduate School

Seoul National University

Machine-generated data such as sensor data now comprise major portion of available information. This thesis addresses two important problems: storing of massive sensor data collection and efficient sensing. We first propose a quality-adjustable sensor data archiving, which compresses entire collection of sensor data efficiently without compromising key features.

Considering the data aging aspect of sensor data, we make our archiving scheme capable of controlling data fidelity to exploit less frequent data access of user. This flexibility on quality adjustability leads to more efficient usage of storage space. In order to store data from various sensor types in cost-effective way, we study the optimal storage configuration strategy using analytical models that capture characteristics of our scheme. This strategy helps storing sensor data blocks

with the optimal configurations that maximizes data fidelity of various sensor data under given storage space.

Next, we consider efficient sensing schemes and propose a quality-adjustable sensing scheme. We adopt compressive sensing (CS) that is well suited for resource-limited sensors because of its low computational complexity. We enhance quality adjustability intrinsic to CS with quantization and especially temporal downsampling. Our sensing architecture provides more rate-distortion operating points than previous schemes, which enables sensors to adapt data quality in more efficient way considering overall performance. Moreover, the proposed temporal downsampling improves coding efficiency that is a drawback of CS. At the same time, the downsampling further reduces computational complexity of sensing devices, along with sparse random matrix. As a result, our quality-adjustable sensing can deliver gains to a wide variety of resource-constrained sensing techniques.

**keywords :** quality-adjustable sensor data, data archiving, data aging, optimal storage management, compressive sensing, downsampling

**student number :** 2006-21240

# Contents

# List of Figures

# List of Tables

# Chapter 1

# Introduction

Rapid advances of hardware technology have created massive information flow generated by various sensors. In order to handle this, we have to consider how to capture and store sensor data efficiently. In this chapter we look at our research motivation and characteristics of sensor data such as spatio-temporal correlation and quality adjustability. We also summarize major contributions of our research and outline the contents of the thesis.

## 1.1 Motivation

We are now witnessing the ubiquity of computers and computing devices in our everyday life. To build better information-based environments, great research efforts have been concentrated in pervasive computing. In particular, as mobile computing became prevalent in the form of hand-held devices such as smart phones, their rich sensing capabilities are enabling new applications and spawning new research topics [1]. Crowd sourcing and opportunistic sensing are derivatives of this richer sensing capability compared to traditional sensing in terms of both quality and quantity [2].

In addition, the sensing boundary has now broadened to urban areas [3]. In modern society, most population tends to be concentrated in urban areas. The vision for smart cities originated as a natural evolution of research in smart homes and other smaller scale smart spaces [4-8]. In smart city, 'things' and people are intimately connected through diverse technologies. However the key technology behind the smart city is various sensors that gauge physical infrastructure such as power grids and oil pipelines, and even mobile objects such as humans and vehicles. People can also act as active sensors using their hand-held devices to gather intelligence on city operations.

Meanwhile, the progress of hardware technology with respect to storage, computation, and communication capabilities has enabled continuous and rapid flow of data items. This progress allows us to create and replicate more information, which has promoted the tendency of generating any data that were once neglected or merely provided in aggregate form [9].

Among this generated information, less than half can be accounted for by user activities, while the rest represents machine-generated information such as sensor data. It is evident that data generated from countless sensors will keep increasing. As various types of sensors are being deployed at more places, information generated by these sensors is also rapidly increasing. This tremendous amount of information we are faced with give rise to so-called 'information explosion' crisis [10-12].

While the disk storage cost keeps decreasing and data storage capacity keeps increasing, this faster information generation rate now leads to a paradox that increasing storage capacity cannot catch up with the rate of information explosion. The amount of information created, captured, or replicated has already exceeded available storage for the first time in 2007. Moreover, it is reported that almost half

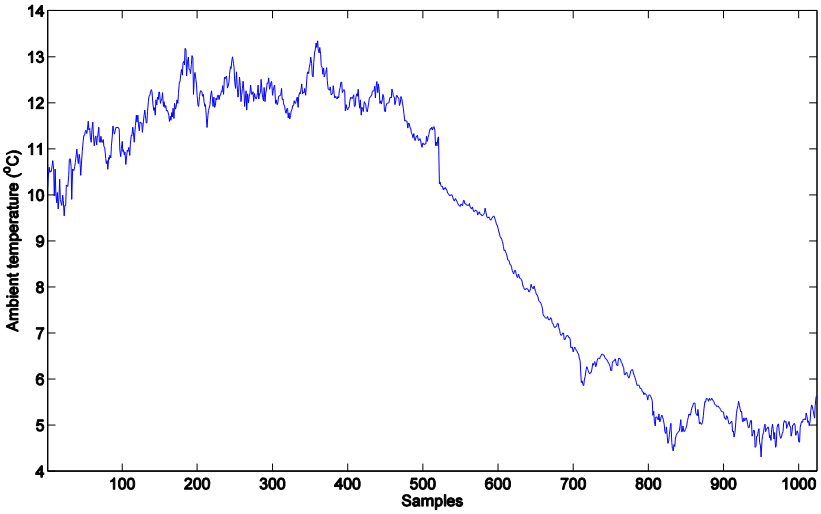
of information created and transmitted cannot be stored in 2011 [13, 14]. It is apparent that this gap between available storage and information creation will keep widened.

In order to resolve this issue, we have to reconsider how to store data generated by 'things' such as sensors. Sensor data have several characteristics that differentiate them from other data. First, sensor data are highly correlated in nature within both spatial and temporal domain [15]. Second, accuracy of sensor data need not be strictly precise [16, 17]. Finally, retrieval of sensor data is gradually decreased as time goes by [18-20].

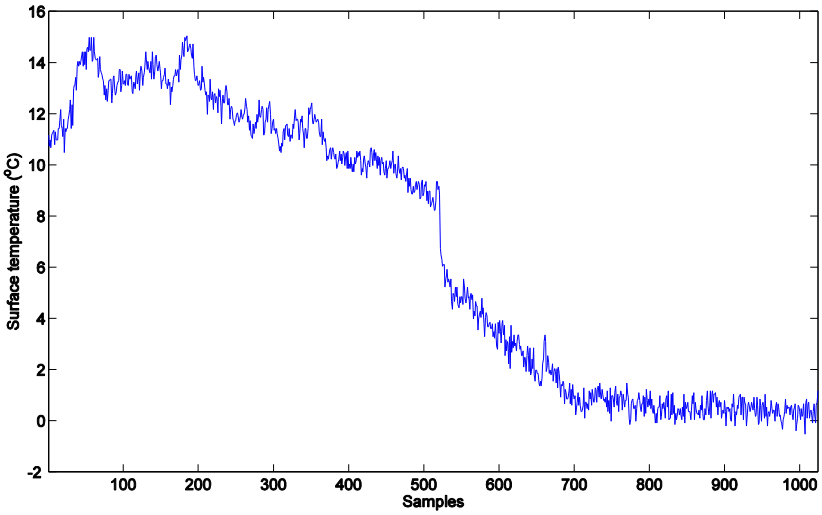
Using these characteristics of sensor data, we address the information explosion from the perspective of both storage and sensing environment: we develop (i) quality-adjustable data archiving scheme for storage efficiency, and (ii) quality-adjustable sensing scheme for individual sensing devices.

## 1.2 Spatio-Temporal Correlation in Sensor Data

Since the sensors usually capture physical phenomenon such as environmental data [21], we use data sets downloaded from the SensorScope website throughout this thesis [22]. The SensorScope website has various wireless sensor network (WSN) deployment scenarios that are mostly environmental data samples. In particular, we employ three different sensor types for our data archiving: (i) ambient temperature, (ii) surface temperature, and (iii) relative humidity. Figure 1.1 illustrates these three sample data sets captured from a certain sensor node deployed with other sensor nodes at École Polytechnique Fédérale de Lausanne (EPFL) campus in Lausanne, Switzerland.

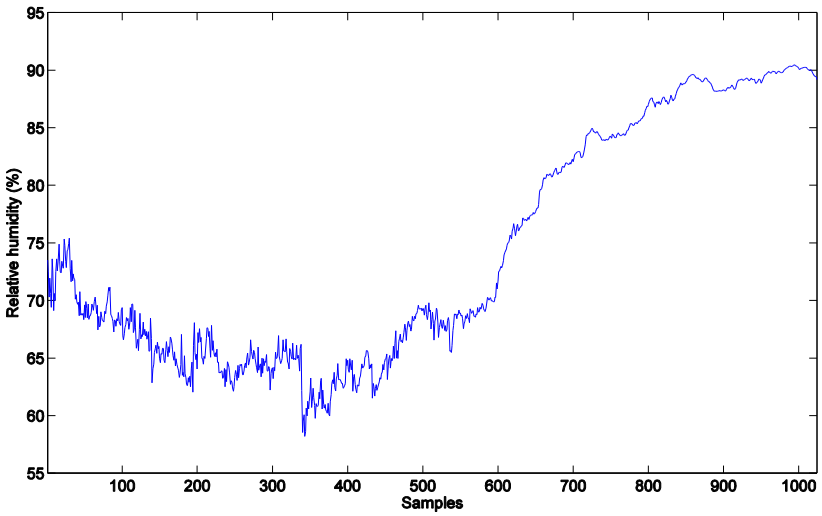



(a)




(b)

Figure 1.1 Excerpts of sensor data samples from: (a) ambient temperature; (b) surface temperature; (c) relative humidity.

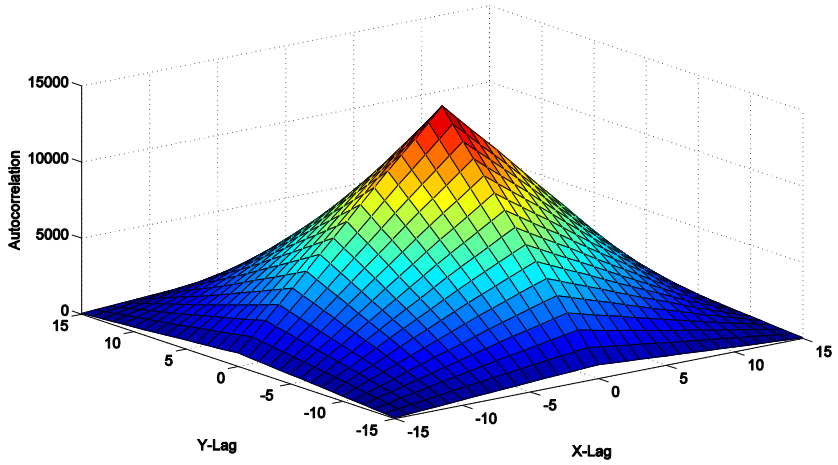



(c)

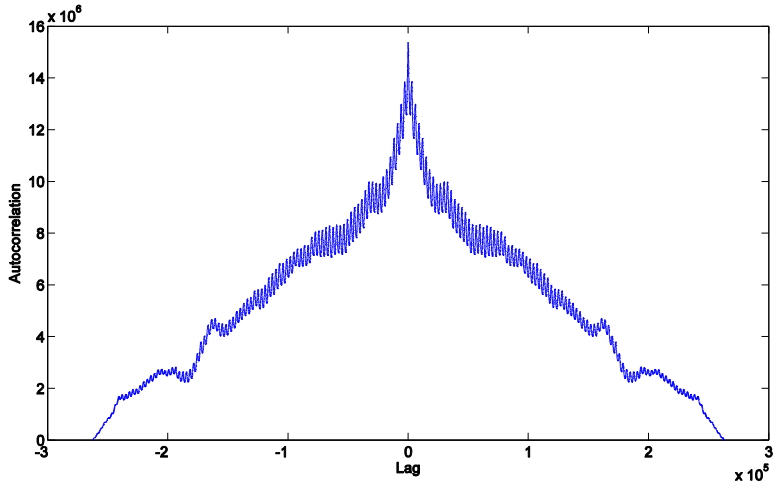
Figure 1.1 (Continued).

Data shown in Fig. 1.1 is highly correlated in nature within spatial and temporal domain [15, 23]. These correlations are presented in Fig. 1.2 where the autocorrelation of the ambient temperature data are plotted for demonstrating the spatial correlation and the temporal correlation that exist in general environmental data. In particular, Fig. 1.2a shows the two-dimensional autocorrelation of the ambient temperature data samples collected at certain time instance, by sensor nodes deployed within a certain area, whereas Fig. 1.2b describes the autocorrelation of consecutive ambient temperature data samples collected by a specific sensor node. In Fig. 1.2a, we can identify that close nodes observe more correlated data, and conversely less correlated data is observed as the distance between nodes increases. Similarly, the correlation depends on the time difference between signal samples as shown in Fig. 1.2b: the ambient temperature data is more

correlated within short time interval. It should be noted that between the spatial and temporal correlation, the temporal correlation exhibits stronger correlation than the spatial correlation.




(a)




(b)

Figure 1.2 (a) 2-D autocorrelation of ambient temperature data samples (b) Autocorrelation of ambient temperature data samples collected by a sensor node.

This spatio-temporal correlation can be exploited to remove the redundancy in both spatial and temporal dimensions, which results in compressing the entire data set into smaller form [24]. This process is analogous to the intra prediction and inter prediction of video compression standards where the intra prediction removes redundancy among texture information of a certain video frame, while the inter prediction removes redundancy among consecutive frames in a video sequence [25].

## 1.3 Quality Adjustability of Sensor Data

Individual sensor data does not require either bit-level accuracy or intactness due to several reasons: (i) each sensor node is equipped with inexpensive and imprecise sensors that only guarantee moderate level of sensing accuracy, (ii) sensor nodes are densely deployed and they periodically capture environmental data that are highly correlated in spatio-temporal domain, which makes storing all of data unnecessary, (iii) we are usually interested in overall trend of sensor data, thus we can tolerate a certain amount of distortion and approximate results are sufficient most of the time [16]. This characteristic of impreciseness, together with strong spatio-temporal correlation, allows us to cope with high information generation rates of sensors via lossy source coding that greatly reduces the amount of required storage space.

In addition, the frequency of access to sensor data is gradually decreased as time goes by. Although fresh data could be frequently accessed and therefore they should retain high fidelity, aged data could be seldom retrieved and only find their use in offering a digest of historical trend in sensor readings. We can exploit this property by controlling the fidelity of sensor data, that is, gradually lowering data quality so that the accuracy of data is decreased over time. In other words, it is

sufficient to store key features of sensor data in most sensor applications especially for long-term storage [15, 18-20].

The trade-off between data fidelity and storage consumption can be explained by rate-distortion theory, where rate and distortion are inversely proportional to each other. When coding a source, one can allow some amount of distortion in original source to reduce rate that are expressed by *mutual information* between original and reconstructed source. Furthermore, the *successive refinement* concept [26, 27] enables sending a description with a particular amount of distortion and later deciding that the description needs to be specified more accurately. Then when an addendum to the original description is sent, this refinement should be as efficient as if the more strict requirements had been known at the start. Figure 1.3 illustrates this concept, where a refinement from $\hat{X}1$ to $\hat{X}2$ achieves the rate-distortion limit at each of the two stages.

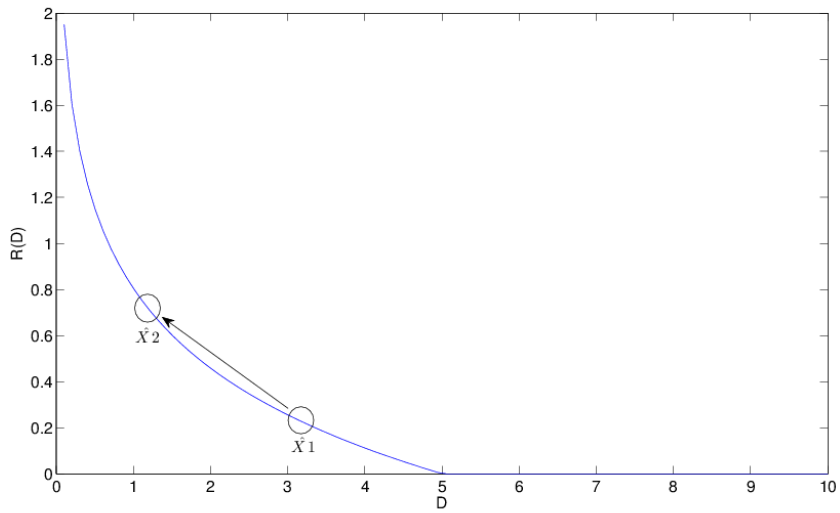



Figure 1.3 Successive refinement from $\hat{X}1$ to $\hat{X}2$.

The successive refinement concept has been realized to many applications, especially in multimedia field in the name of scalable coding, which has been also successful in practical applications [28-31]. However, a well defined successive refinement theory does not hold completely in reality. Even the most advanced scalable encoder to date is not able to achieve the rate-distortion limit shown in Fig. 1.3 due to several reasons: (i) source distribution mismatch between actual source and theoretical source which has simple distributions such as Gaussian or Laplacian, (ii) the impossibility of assuming infinite block length for the codebook generation as in the case of rate-distortion theory, and (iii) side information and protocol overhead.

In spite of a little inefficiency intrinsic to the scalable coding, utilizing it to control the fidelity of sensor data would deliver notable gain in handling the fast information generation rate that is triggered by vast amount of sensors. Specifically, we can combine multiple layers to constitute a whole data block and later easily discard the highest layer one by one, which should result in efficient usage of storage space.

## 1.4 Research Contributions

The contributions of this thesis are summarized as follows:

- We propose a *quality-adjustable sensor data archiving* that exploits both spatio-temporal correlations inherent in sensor data collection, which can be employed as a quality management module in conventional distributed file system. This archiving scheme provides digested set of sensor data without compromising much fidelity. The performance of our scheme can

be demonstrated as outstanding coding efficiency with data fidelity corresponding to the order of sensor accuracy.

- We focus on the gradually decreasing access pattern of sensor data, which can be translated into decreasing data fidelity as time elapses: it is sufficient to store only key features of sensor data in most sensor applications especially for long-term storage. Thus we offer multiple fidelity levels in our archiving scheme, which facilitates efficient storage management.
- We delve into the relationship between quality parameters, distortion, and size of sensor data, from which we derive models that closely reflect the characteristics of our quality management scheme. Using these analytical models, we further find the optimal rate allocation strategy which minimizes distortion under a certain allowable rate. Furthermore, we study the *optimal storage configuration strategy* with which huge data from various sensor types have to be efficiently stored.
- We propose a *quality-adjustable sensing* for an individual sensing device. To this end, *compressive sensing* (CS) is adopted that shifts the complexity burden of conventional source coding from the sensing device to data collection points that have more computational power. In particular, we extend general CS framework with *downsampling* in order to enhance the quality adjustability of the sensing device. We show that not only sensing data quality can be adapted in more efficient manner depending on various contexts sensors are subject to, but coding efficiency is improved using the downsampling approach.

## 1.5 Thesis Organization

The rest of this thesis is organized as follows. Chapter 2 presents the quality-adjustable archiving scheme of sensor data utilizing their properties, and compares its performance with other coding methods. We also discuss the optimal storage configuration strategy that is derived from analytical models. In Chapter 3, we enhance the number of fidelity control options through the addition of quality enhancement layer to the archiving scheme, whose effect on the optimal storage configuration is discussed as well. Chapter 4 presents the quality-adjustable sensing in sensing environment. We introduce a low-complexity compressive sensing that is suitable for resource-limited sensors. Finally, we draw conclusions and address future research directions in Chapter 5.

# Chapter 2

# Archiving of Sensor Data

We have seen several properties of sensor data, especially the spatio-temporal correlation and the quality adjustability. In this chapter we begin our study of archiving the collection of sensor data. We show our quality-adjustable archiving scheme is competitive by demonstrating its coding efficiency. We derive analytical models that reflect the operation of our scheme. These models in turn lead to the optimal storage configuration strategy for handling massive data generated from various sensors.

## 2.1 Encoding Sensor Data Collection

Figure 2.1 illustrates the scenario of various sensors collecting and transmitting data to storages, where most of sensors are static and densely deployed. Storage optimization is essential in these circumstances, which calls for more efficient compression algorithms that can enable us to handle more information with the same amount of hardware.

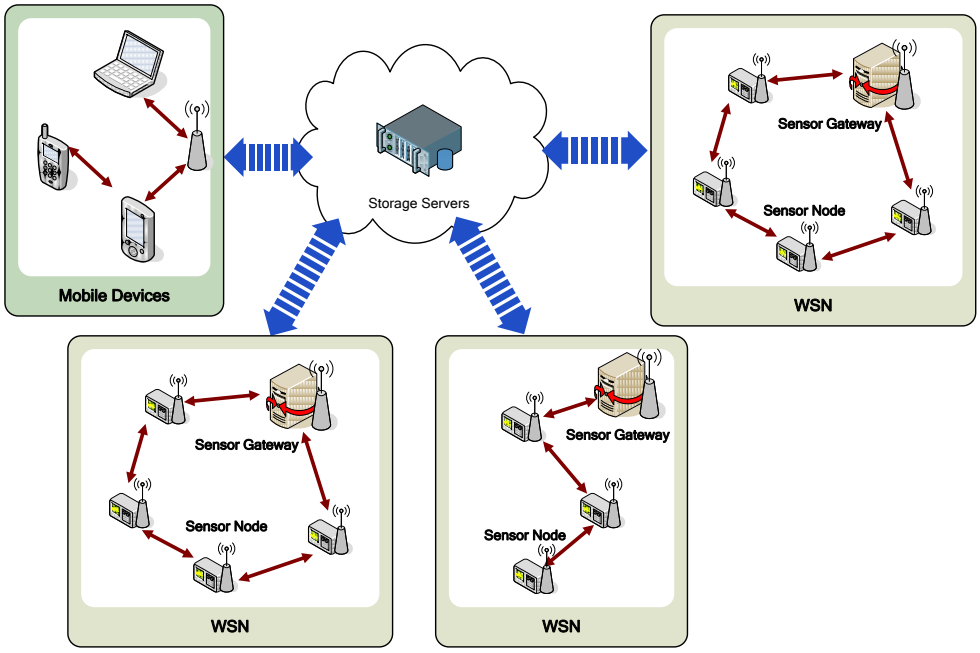



Figure 2.1 Data collection scenario from various sensors.

### 2.1.1 Archiving Architecture

Figure 2.2 illustrates the block diagram of our quality management module working with conventional distributed file system that stores collected data from various sensors which are mostly static and densely deployed. Incoming sensor input is first filtered through the spatio-temporal decorrelation module where most of redundancy inherent in input data is removed in both spatial and temporal direction.

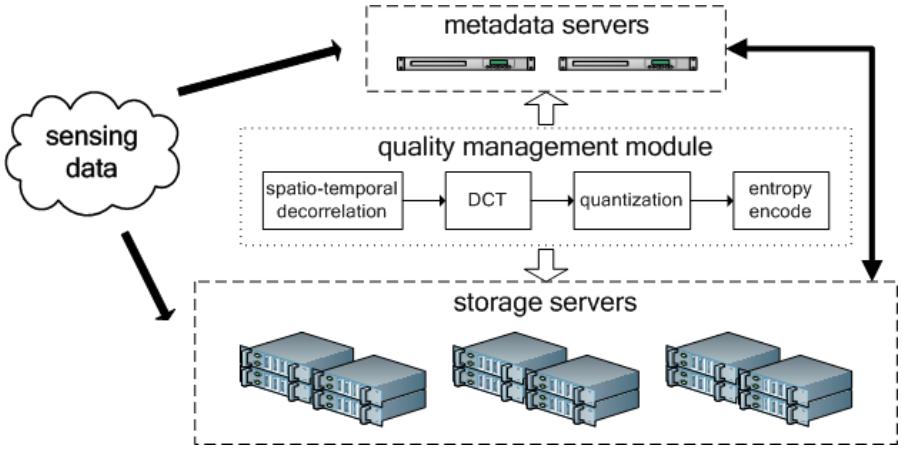



Figure 2.2 Quality management module working with conventional distributed file system.

Incoming sensor input is first filtered through the spatio-temporal decorrelation module where most of redundancy inherent in input data is removed in both spatial and temporal direction. In particular, spatial correlation shown in Fig. 1.2a is removed by predicting a particular sensor value with its neighboring sensor values; whereas temporal correlation shown in Fig 1.2b is removed by predicting collection of sensor values at a certain time instant with collections at previous time instants. Since the temporal correlation generally exhibits stronger correlation than the spatial correlation, the signal decorrelation effect is stronger over temporal direction.

The output from spatio-temporal decorrelation module in turn undergoes discrete cosine transform (DCT) for signal compaction. DCT has important characteristics such as energy compaction and signal decorrelation [32, 33], which is an approximation of Karhunen-Loève transform (KLT) that is optimal in reducing the dimensionality of feature space.

Similar to DCT, the wavelet transform also has desirable properties for the

compression of data such as energy compaction and signal decorrelation [32]. However it is well known that the performance of wavelet-based and DCT-based coding is almost same [34].

After DCT, the transformed data is subject to quantization and entropy encode processes. The quantization process is related to the rate-distortion theory explained in Section 1.3, which is concerned with the task of representing a source with the fewest number of bits possible for given reproduction quality. In other words, quality of data is compromised in the quantization process to yield compact representation of data, that is, lossy compression. Finally the entropy encode process further compresses quantized output losslessly by representing frequently occurring quantized labels with fewer bits and infrequently occurring quantized labels with more bits [32]. Meanwhile, the decoding process is straightforward: the entire process can be reversed to reconstruct data that approximate the original data.

Note that this process is analogous to modern image and video encoding scheme [25, 32], whose performance overhead is insignificant with respect to today's standard [16, 28, 31, 35, 36]. In fact, regarding sensor and environmental data as two-dimensional array of pixels has been embraced in literatures. Utilizing the inter prediction concept of a video coding standard, the watching of a 'sensor movie' idea was realized in monitoring data sensed from large WSN to increase sensor lifetimes [37, 38]. In addition, handling environmental data directly in floating-point format and making them compressible by an image compression standard was studied [39].

Compressing of sensor data in distributed environment using lossless or lossy approach has also been proposed in literatures [18-20, 40-42]. Since they focused on distributed storage of WSN, the spatial and temporal correlation inherent in sensor data were not fully exploited, thereby underutilizing latent correlation in

contrast to our archiving architecture.

### 2.1.2 Data Conversion

Most sensors capture physical phenomenon that can be represented using IEEE 754 single precision floating-point format which is 32 bits long. Previous studies have addressed lossless compression of floating-point data [39, 43-45]. However, 4-byte length to represent the physical phenomenon such as environmental data is more than necessary most of the time. Due to the structure of the floating-point format that is made up of exponent and fraction bits, a number around zero enjoys excessively fine granularity. In addition, each sensor embedded in sensor nodes has only limited accuracy as discussed in Section 1.3.

Taking these into consideration, we can represent sensor data using just one byte without much penalty. We could use fixed-point number instead of floating-point, and divide normal operating range with 256 steps, while reserving both the first and last steps for handling anomalous data that are out of the normal operating range. This one byte representation leads to an immediate effect of reducing entire data size by three-fourths at the cost of little distortion that is $\Delta^2/12$ assuming a mean squared error (MSE) distortion and a uniform distribution of the quantization error, where $\Delta$ is the quantization step size.

The conversion of 4-byte to 1-byte data results in smooth adaptation to the quality management module shown in Fig. 2.2 whose performance is optimized with the input of 1-byte unsigned integers. In Fig. 2.2, sensor data fed into the quality management module undergoes another quantization process to control the data fidelity, which can be adjusted through the quantization parameter (QP). The combination of one quantization from data conversion and the other quantization

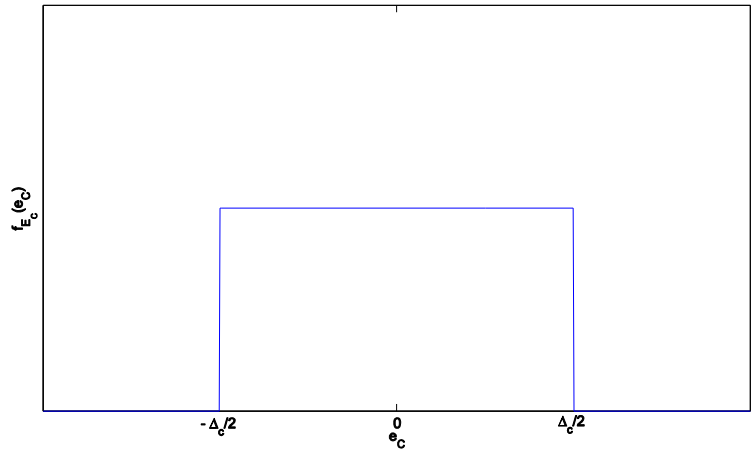
from lossy coding itself seems quite complicated to analyze at first glance. However, they can be treated separately as the following lemma.

*Lemma 2.1*: The joint distortion $D_{quant}$ caused by the quantization from data conversion and the quantization from lossy coding is separable and can be expressed by sum of both distortions.

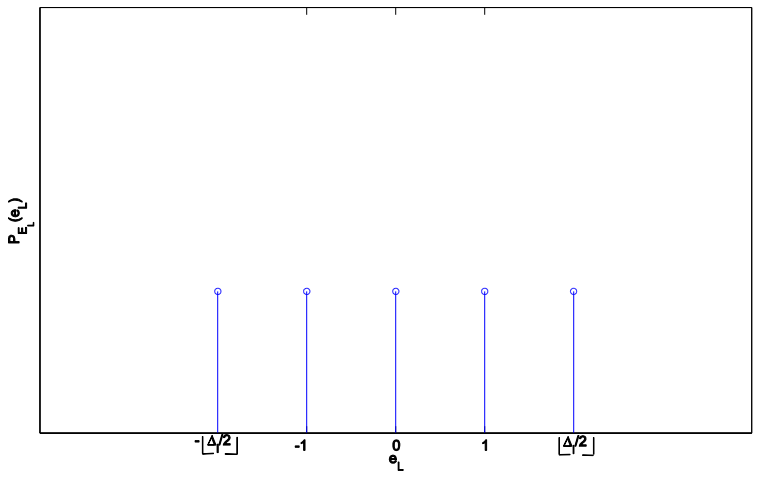
*Proof*: Assuming the original data of single precision floating-point type is nearly continuous, the quantization step size $\Delta_c$ is a division of normal data range by 256 steps. Then the probability density function (pdf) of quantization error from data conversion can be shown as in Fig. 2.3a. On the other hand, the quantization step size $\Delta_l$ is controlled by QP of a lossy encoder that usually performs quantization operation in DCT domain. However it is well known that in an ideal encoder-decoder system, spatial-domain distortion and DCT-domain distortion are equal [32, 46], which enables us to render the probability mass function (pmf) of quantization error from lossy coding as in Fig. 2.3b. Assuming $\Delta_l$ is an odd number without loss of generality, we can express the pmf of the quantization error from lossy coding as follows:

$$P_{E_L}(e_L) = \begin{cases} \dfrac{1}{\Delta_l} & e_L = -\left\lfloor \dfrac{\Delta_l}{2} \right\rfloor, -\left\lfloor \dfrac{\Delta_l}{2} \right\rfloor + 1, ..., \left\lfloor \dfrac{\Delta_l}{2} \right\rfloor, \\ 0 & \textit{otherwise} \end{cases} \tag{2.1}$$

where $E_L$ is a discrete random variable that denotes an amount of quantization error in integer domain.




(a)




(b)

Figure 2.3 (a) Pdf of quantization error from data conversion (b) Pmf of quantization error from lossy coding.

When a specific $E_L$ is given by $e_L$, we can express the conditional pdf of quantization error from data conversion as follows:

$$f_{E_C|E_L}(e_C \mid e_L) = \begin{cases} \dfrac{1}{\Delta_c} & e_L \cdot \Delta_c - \dfrac{\Delta_c}{2} < e_C < e_L \cdot \Delta_c + \dfrac{\Delta_c}{2} \\ 0 & otherwise \end{cases}, \tag{2.2}$$

where $E_C$ is a continuous random variable that denotes an amount of quantization error. We can identify from (2.2) that the pdf shown in Fig. 2.3a can be shifted to left or right according to given $E_L$.

Assuming MSE distortion measure, $D_{quant}$ can thus be formulated using joint distribution:

$$D_{quant} = \sum_{-\left\lfloor \Delta_l/2 \right\rfloor}^{\left\lfloor \Delta_l/2 \right\rfloor} \int_{e_C} f_{E_L E_C}(e_L, e_C) \cdot e_C^2 de_C. \tag{2.3}$$

Then (2.3) yields the following:

$$D_{quant} = \sum_{-\left\lfloor \Delta_l/2 \right\rfloor}^{\left\lfloor \Delta_l/2 \right\rfloor} \int_{e_C} P_{E_L}(e_L) \cdot f_{E_C|E_L}(e_C \mid e_L) \cdot e_C^2 de_C = \frac{\Delta_c^2}{12} + \frac{\Delta_c^2 \left\lfloor \frac{\Delta_l}{2} \right\rfloor \left( \left\lfloor \frac{\Delta_l}{2} \right\rfloor + 1 \right)}{3}, \tag{2.4}$$

which continues in

$$D_{quant} = \frac{\Delta_c^2 + \Delta_c^2 (\Delta_l - 1)(\Delta_l + 1)}{12} \approx \frac{\Delta_c^2}{12} + \frac{\Delta_c^2 (\Delta_l^2 - 1)}{\beta}, \tag{2.5}$$

where $\beta$ is a denominator which is 12 for small quantization step size and larger than 12 in case of a larger quantization step size compared to the signal variance.

This is because when the quantization step size becomes large, quantization errors can no longer be treated as uniformly distributed [47].

In the right-hand side of (2.5), the first term is the data conversion distortion and the second term is the lossy coding distortion normalized by $\Delta_c$. In fact, this result is owing to the independence of two different distortion sources. ■

The above lemma helps us analyze and model the distortion of lossy coding by separating two different sources of quantization errors. Since $\Delta_c$ is typically very small, we can confine distortion due to the data conversion to ignorable amount and rather focus on lossy coding itself.

## 2.2 Compression Ratio Comparison

In order to suggest the coding efficiency of our scheme, we compared the compression ratios of popular lossless coding methods with our quality-adjustable archiving scheme. We first focus on lossless coding methods and compare their performance of encoding raw environmental data. We employed several methods whose brief descriptions are as follows [32]. First, *gzip* is widely used file compression tool in Unix-like operating systems that is based on DEFLATE algorithm, which is a combination of LZ77 and Huffman coding [48]. *bzip2*, which generally yields more coding efficiency than *gzip*, is based on a combination of Burrows Wheeler Transform (BWT), move-to-front transform, and Huffman coding [49]. *PPMd* is an optimized implementation of prediction by partial matching (PPM) algorithm [50]. Lastly, *7-Zip* is a relatively recent compressor that is based on the Lempel-Ziv-Markov chain algorithm (LZMA) [51].

Figure 2.4 shows compression ratios of lossless coding methods mentioned

above, where the compression ratios are expressed by the original raw data size divided by the compressed size. Unconverted, environmental data of single precision floating-point format were compressed using four lossless coding methods. In Fig. 2.4, we can identify that coding efficiency depends on the characteristics of each environmental data, where the relative humidity fluctuates vibrantly compared to ambient and surface temperature, hence yielding low compression ratios. In addition, we can observe that apart from *gzip*, three compression methods show similar compression performance throughout three data sets.

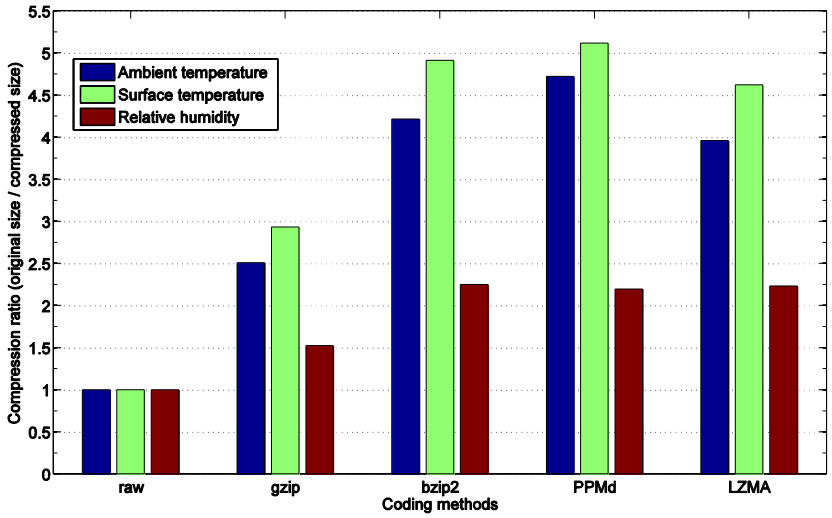



Figure 2.4 Compression ratios of various lossless coding methods.

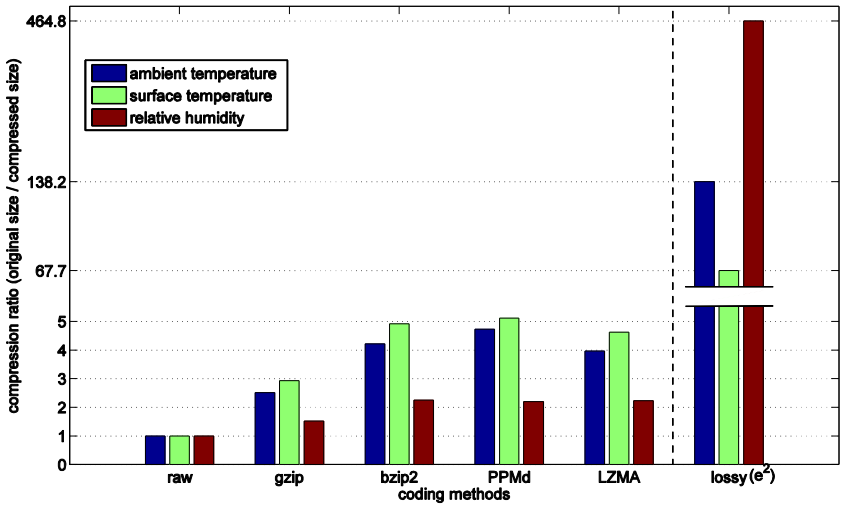
We are also interested in the results of our quality-adjustable archiving scheme that handle the data with converted 1-byte unsigned integer format. As mentioned in Section 2.1.2, the conversion leads to reduction of data size by three-fourths, incurring distortion $\Delta_c^2/12$. Lossless coding methods in this case, compress the

converted data without loss that already carry the error due to the data conversion. In contrast, our quality-adjustable archiving scheme can compress more than lossless encoders through data fidelity adjustment at the cost of extra distortion.

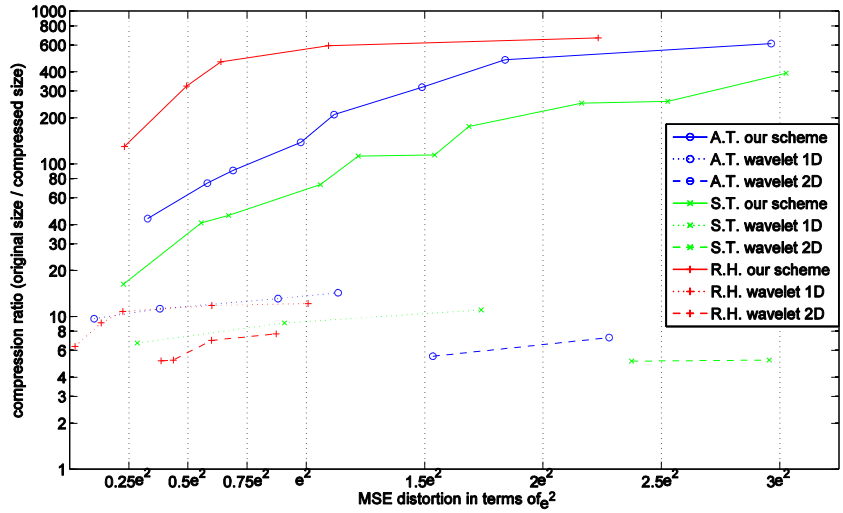
We compare our scheme with lossless coding methods in Fig. 2.5a. Although compressed size can be as small as how much we allow distortion, it might be unfair to directly compare lossy coding with lossless coding in terms of coding efficiency. Hence we set out a reference point for distortion, which is the sensor accuracy explained in Section 1.3: "each sensor node is equipped with inexpensive and imprecise sensors that only guarantee moderate level of sensing accuracy." Vendors manufacturing sensors usually provide sensor accuracy information [52]. Table 2.1 shows the sensor type and its accuracy which corresponds to the sensor error margin *e*. In Fig. 2.5a, we allow total distortion up to $e^2$ in terms of MSE for our archiving scheme where data conversion distortion $\Delta_c^2/12$ is within boundary of $e^2$.

Table 2.1 Sensor accuracy and type for three data types used in experiments

| **Data Type** | **Accuracy** | **Sensor Type** |
|---|---|---|
| Ambient Temperature (A.T.) | ±0.3℃ | Sensirion SHT75 |
| Surface Temperature (S.T.) | ±0.3℃ | |
| Relative Humidity (R.H.) | ±2% | |




(a)




(b)

Figure 2.5 (a) Compression ratios of our archiving scheme compared with various lossless coding methods (b) Log-scale compression ratios of our archiving scheme compared with wavelet-based methods with limited correlations at various data fidelities.

In Fig. 2.5a, LZMA performs best among lossless coders, while other lossless coding methods show moderate results with compression ratios under 5.0. However the most notable result comes with our archiving scheme that is up to 464.8 depending on data types, allowing distortion comparable to the order of sensor error margin.

The utilization of both spatio-temporal correlations culminates in outstanding coding efficiency as shown in Fig. 2.5b, where our archiving scheme contrasts with wavelet coding methods with limited correlation. Wavelet is another popular lossy coding method apart from DCT-based coding. Although the performance of wavelet-based and DCT-based coding is almost same as mentioned in Section 2.1.1, compression ratio shown in Fig. 2.5b juxtaposes a consequence of restricting the use of correlation to either spatial dimension or temporal dimension: wavelet 1D only exploits *temporal correlation* for signal compression, whereas wavelet 2D only exploits *spatial correlation* for signal compression. After signal compaction, both methods apply threshold, quantization and entropy encode processes for lossy compression of signal. Between both wavelet-based methods, wavelet 1D shows better results than wavelet 2D, thanks to the stronger correlation in the temporal domain than the spatial domain as shown in Fig. 1.2.

It should be again noted that similar approaches to our scheme in distributed environment such as WSN have been proposed to reduce traffic and storage usage inside the network itself [18-20, 40-42]. Apart from these efforts, an efficient data compression technique that fully exploits spatio-temporal correlation of huge sensor data set, in contrast to the limited correlation of distributed environment case, is demanded for better management of storage space.

The results in Fig. 2.5 convince us that our scheme is a viable solution for archiving huge amount of sensor data. In the following section, we will show more

comprehensive results of quality-adjustable archiving scheme with varying data fidelities, where we will study the effects of data fidelity control on both rate and distortion aspects.

## 2.3 Quality-Adjustable Archiving Model

We have seen the importance of utilizing both spatio-temporal correlations in our sensor data archiving scheme by comparing coding efficiency. We further focus on the quality adjustability of our archiving scheme; therefore, we derive analytical models that reflect the effect of adjusting data fidelity on both rate and distortion aspects. We show our model is close to actual results, which subsequently enables us to develop the optimal storage configuration strategy.

### 2.3.1 Data Fidelity Model: Rate

While the size of data can be controlled by adjusting QP at the quantization process in Fig. 2.2 in the traditional rate-distortion theoretical sense, it can also be controlled by adjusting the granularity in temporal domain, which is equivalent to the temporal scalability. Figure 2.6 shows the temporal coding structure of our spatio-temporal decorrelation module. There are total five temporal levels shown in Fig. 2.6, where each increasing temporal level corresponds to a double of frequency at which collections of sensor data at certain time instance are included in coded data set. Thus, the highest temporal level shall contain all of data sampled in line with temporal dimension. Figure 2.6 also displays the temporal prediction structure shown by arrows, which exploits strong temporal correlation we have seen in Fig. 1.2b. It should be noted that this type of the temporal coding and prediction structure has been adopted in various video coding standards, where its efficiency

has been verified as well.

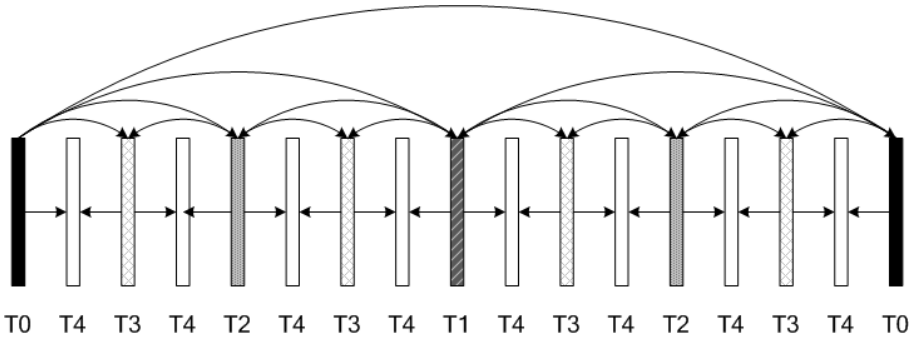



Figure 2.6 Temporal coding and prediction structure of our spatio-temporal decorrelation module.

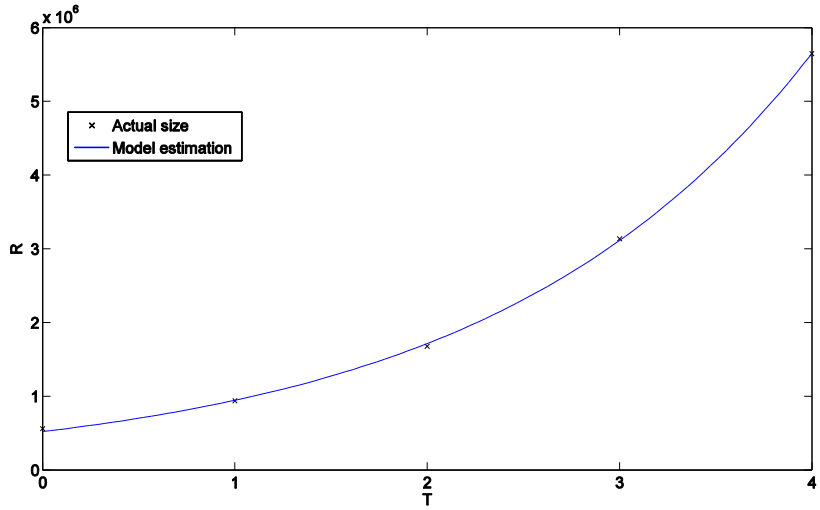
It is quite intuitive to reckon that the size of compressed data *R* is reduced by half as the temporal level decreases by one step. However, due to the temporal prediction structure shown in Fig. 2.6, the amount of reduction becomes less than half per one temporal level decrease. We can model this relation as:

$$R = \alpha(\Delta_l) \cdot \exp(\beta(\Delta_l) \cdot T), \tag{2.6}$$

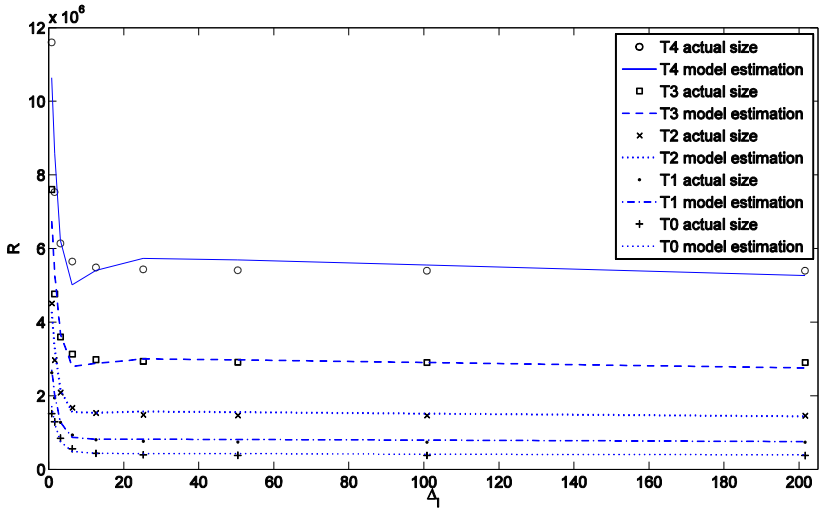
where $\alpha(\Delta_l)$ and $\beta(\Delta_l)$ are model parameters dependent on the quantization step size $\Delta_l$, and $T \in \{0,1,2,3,4\}$ denotes the temporal level. In fact, the range of temporal levels can be extended or reduced depending on applications, which entails modification of the temporal coding and prediction structure shown in Fig. 2.6. Without loss of generality, we use the structure shown in Fig. 2.6 throughout this thesis.

Comparison between actual data size and the model in (2.6) with a fixed $\Delta_l$ is shown in Fig. 2.7a, where we can confirm the model effectively follows the

varying size of actual sensor data with respect to the temporal level.




(a)




(b)

Figure 2.7 (a) Rate curve as a function of temporal level estimated by (2.6) with *QP*=20 (b) Rate curves as functions of quantization step sizes for different temporal levels estimated by (2.6).

In (2.6), two model parameters $\alpha(\Delta_l)$ and $\beta(\Delta_l)$ have to be estimated from the real data based on the quantization step size, which are represented by

$$\alpha(\Delta_l) = a_\alpha \exp(b_\alpha \Delta_l) + c_\alpha \exp(d_\alpha \Delta_l), \tag{2.7}$$

$$\beta(\Delta_l) = a_\beta \exp(b_\beta \Delta_l) + c_\beta, \tag{2.8}$$

where $a_\alpha$, $b_\alpha$, $c_\alpha$, and $d_\alpha$ are data-dependent constants supplementary to $\alpha(\Delta_l)$ in (2.6), and similarly, $a_\beta$, $b_\beta$, and $c_\beta$ are constants for $\beta(\Delta_l)$ in (2.6). It should be noted that $b_\alpha$ and $d_\alpha$ in (2.7) and $b_\beta$ in (2.8) are all negative valued parameters that reflect decay of $\alpha(\Delta_l)$ and $\beta(\Delta_l)$ with an increasing $\Delta_l$. Combining (2.7) and (2.8) with (2.6), we can represent the total rate as a function of both the quantization step and the temporal level. The resulting model function is plotted in Fig. 2.7b, where five lines represent each temporal level and actual data points are also plotted for comparison. We can confirm the model effectively follows the varying size of actual sensor data.

### 2.3.2 Data Fidelity Model: Distortion

In addition to the rate modeling discussed above, we can estimate the distortion of data due to the quantization as well, which is given by

$$D_{quant} = a_{quant} \cdot \exp(b_{quant} \cdot QP) + c_{quant}, \tag{2.9}$$

where $a_{quant}$, $b_{quant}$, and $c_{quant}$ are data-dependent constants. It should be noted that (2.9) is a function of *QP*, whose relationship with the quantization step size $\Delta_l$

can be expressed by $\Delta_l = 0.625 \cdot 2^{QP/6}$ [53]. Figure 2.8 shows actual distortion points and its approximation using (2.9).

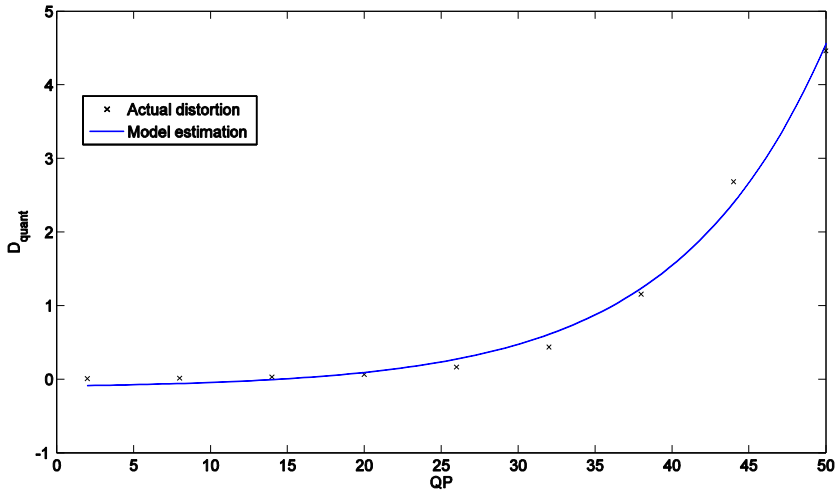



Figure 2.8 Distortion curve as a function of *QP* estimated by (2.9).

Although (2.9) effectively models the distortion caused by quantization, the source of distortion is not limited to the quantization. As the temporal level *T* varies, the amount of sampled data along temporal dimension varies as well, which causes another distortion. Recalling the temporal coding structure shown in Fig. 2.6, as *T* decreases by one step, half of data are excluded from data set, which leads to the condition that omitted data should be estimated using previous data samples. As a result, the total distortion increases as *T* decreases.

In order to incorporate the temporal distortion into total distortion along with the quantization distortion, we assume that the temporal distortion is measured by mismatch between actual data samples and omitted data samples that are replaced

by previous data samples. Although the combination of these two different types of distortion seems tightly coupled, they can be separated as in the case of the lemma 2.1. The following lemma proves their separability.

*Lemma 2.2*: The joint distortion $D_{total}$ caused by the quantization from lossy coding and the omission of data samples along temporal dimension is separable and can be expressed by sum of both distortions.

*Proof*: First we assume an arbitrary pdf of distance between actual data samples and reconstructed data samples, in which missing samples are covered by previous existing data samples. This pdf is denoted by $f_{E_T}(e_T)$, where random variable $E_T$ reflects the near continuity of distance between data samples.

When a specific $E_T$ is given by $e_T$, the conditional pmf of the quantization error from lossy coding is given by

$$P_{E_L|E_T}(e_L \mid e_T) = \begin{cases} \dfrac{1}{\Delta_l} & e_L = e_T - \left\lfloor \dfrac{\Delta_l}{2} \right\rfloor, e_T - \left\lfloor \dfrac{\Delta_l}{2} \right\rfloor + 1, ..., e_T + \left\lfloor \dfrac{\Delta_l}{2} \right\rfloor, \\ 0 & \textit{otherwise} \end{cases} \quad (2.10)$$

which indicates that the pmf shown in Fig. 2.3b can be shifted to left or right according to given $E_T$.

We can express $D_{total}$ using joint distribution:

$$D_{total} = \int_{e_T} \sum_{e_L} f_{E_T E_L}(e_T, e_L) \cdot e_L^2 de_T. \quad (2.11)$$

Then we have

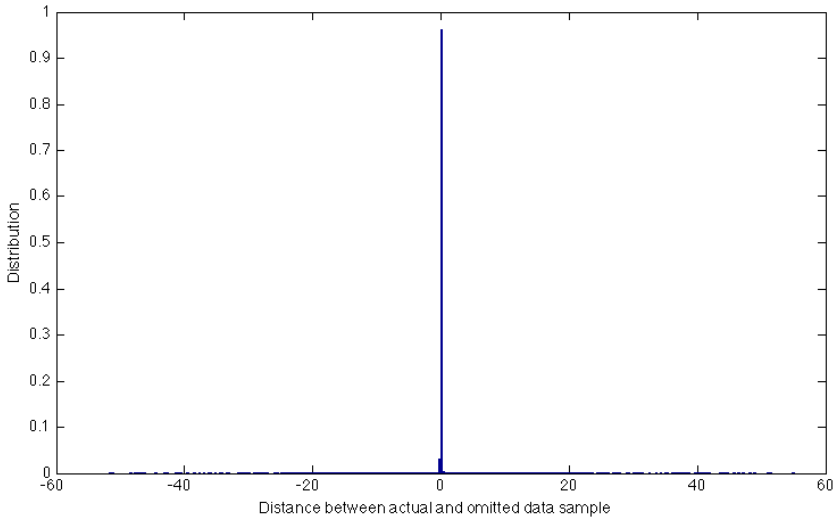
$$
\begin{aligned}
D_{total} &= \int_{e_T} \sum_{e_L=-\lfloor \Delta_l/2 \rfloor}^{\lfloor \Delta_l/2 \rfloor} f_{E_T}(e_T) P_{E_L|E_T}(e_L \mid e_T)\left(e_L + e_T\right)^2 de_T \\
&= \int_{-\infty}^{\infty} f_{E_T}(e_T) \frac{1}{\Delta_l} \sum_{e_L=-\lfloor \Delta_l/2 \rfloor}^{\lfloor \Delta_l/2 \rfloor} \left(e_L + e_T\right)^2 de_T,
\end{aligned}
\tag{2.12}
$$

which continues in

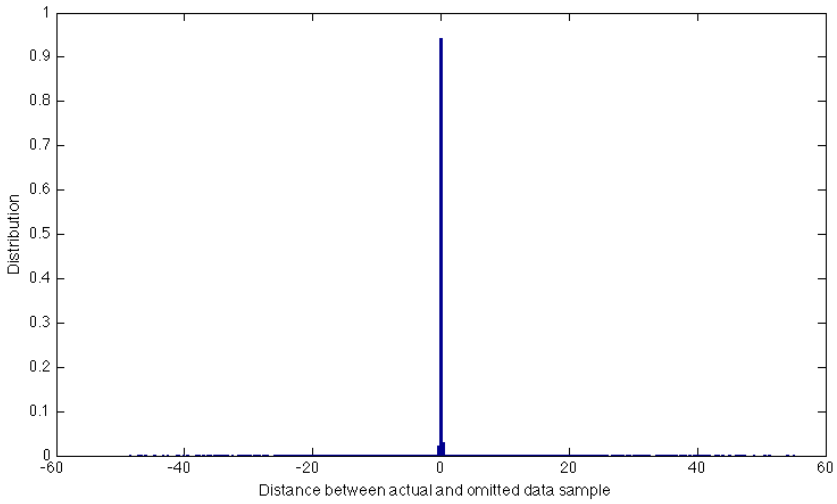
$$
D_{total} = \int_{-\infty}^{\infty} f_{E_T}(e_T)\left(e_T^2 + \frac{\Delta_l^2 - 1}{12}\right) de_T \approx \int_{-\infty}^{\infty} e_T^2 f_{E_T}(e_T) de_T + \frac{\Delta_l^2 - 1}{\beta}. \tag{2.13}
$$

In the right-hand side of (2.13), the first term is the distortion in temporal dimension and the second term is the lossy coding distortion. Again, this result explains the independence of two different distortion sources. ■

Using the above lemma, the total distortion can be simply expressed by summing distortions from two different sources, which later will be proved as a useful property for modeling distortion. Meanwhile, throughout the lemma, we assumed an arbitrary pdf $f_{E_T}(e_T)$ that illustrates the distribution of distance between data samples. Intuitively we can conjecture that the range of distances between actual and omitted data samples which are replaced by previous data samples is widened as more data samples are dropped along the temporal dimension. Indeed, this conjecture can be confirmed as shown in Fig. 2.9, where four distance histograms are illustrated with respect to each temporal level except for the highest temporal level that has no temporal distortion.

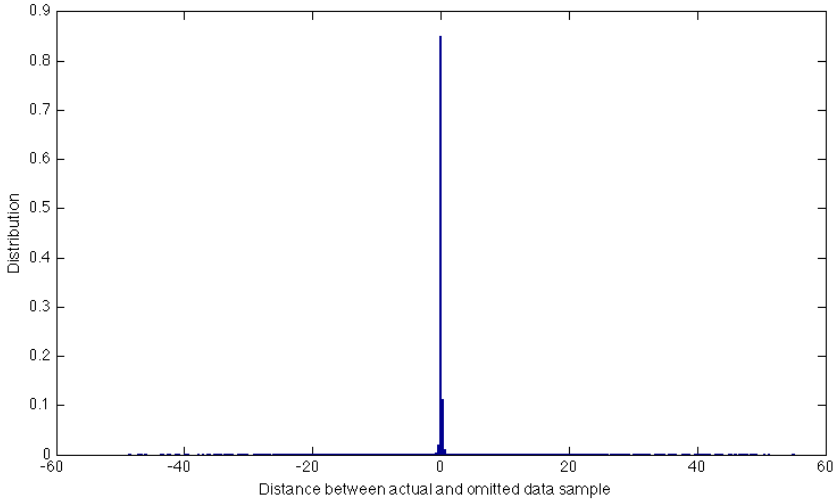



(a)

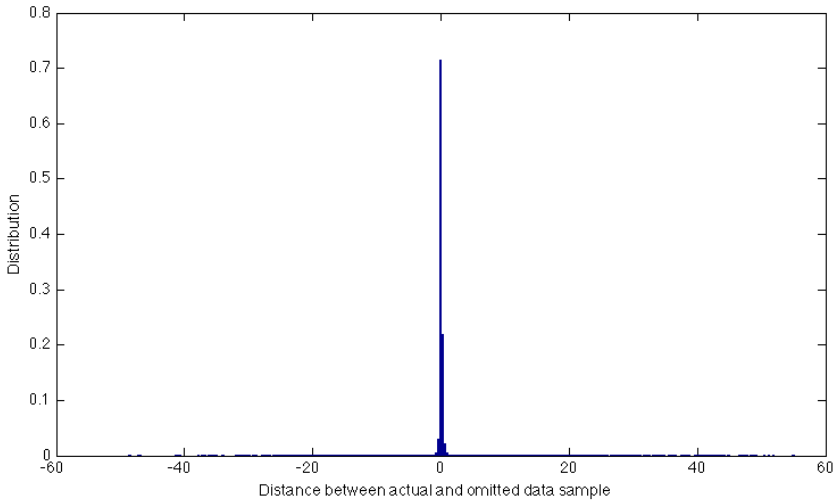



(b)

Figure 2.9 Distribution of distance between actual and omitted data samples for: (a) $T$=3; (b) $T$=2; (c) $T$=1; (d) $T$=0.




(c)




(d)

Figure 2.9 (Continued).

The distributions shown in Fig. 2.9 all have property that most of probability

masses are concentrated in zero, which demonstrates that there are excessive zeros in distance samples. This property can effectively be captured employing zero-inflated model [54]. We can model the distributions using the mixture of the Dirac delta function and Laplacian distribution. Let *p* denotes an inflation term that indicates point mass at zero, then the rest of probability mass (1-*p*) can be represented using the pdf of Laplacian. This zero-inflated Laplacian distribution is given by

$$f_{E_T}(e_T) = \begin{cases} p \cdot \delta(e_T) & e_T = 0 \\ (1-p) \cdot \dfrac{\lambda}{2} e^{-\lambda |e_T|} & e_T \neq 0 \end{cases}, \tag{2.14}$$

where $\lambda$ is the shape parameter of Laplacian distribution. We can identify that (2.14) follows the actual distributions properly in Fig. 2.10 where $f_{E_T}(e_T)$ was drawn over the histogram of Fig. 2.9d. Since the mean of $f_{E_T}(e_T)$ is zero, its variance $\sigma_{E_T}^2$ is equivalent to the distortion in temporal dimension. The range of distances between actual and omitted data samples that are replaced by previous data samples is widened as more data samples are dropped along the temporal dimension, which equates to decreasing *p* in (2.14).

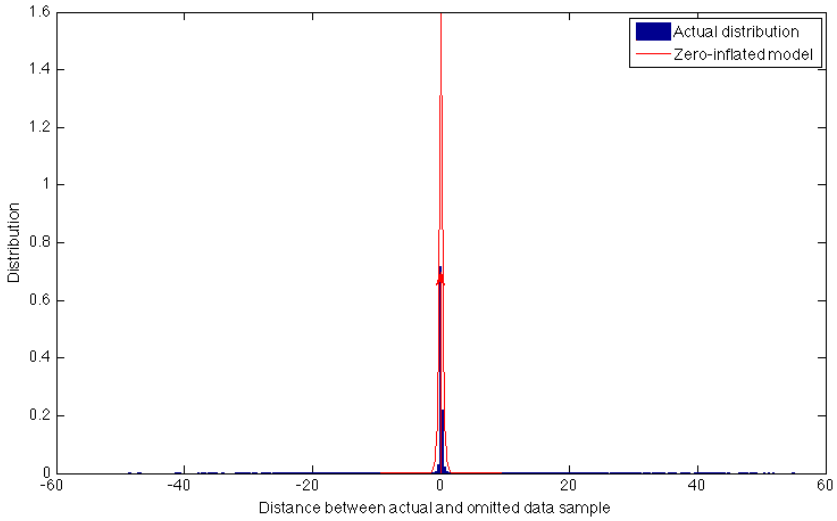



Figure 2.10 Distribution of distance fitted with zero-inflated Laplacian distribution.

With the lemma 2.1 and lemma 2.2 at hand, we can state the separability of all distortion sources with the following theorem.

*Theorem 2.1*: Every source of distortion is separable and can be analyzed independently.

*Proof*: Using the lemma 2.1 and lemma 2.2, the proof is straightforward. ■

Now we turn to the problem of estimating the temporal distortion model. Specifically, we have found that the temporal distortion $D_{temp}$ is a linear function of the temporal level $T$, which is given by

$$D_{temp} = a_{temp} \cdot T + b_{temp}, \tag{2.15}$$

where $a_{temp}$ and $b_{temp}$ are constants. The accuracy of (2.15) can also be verified by Fig. 2.11.

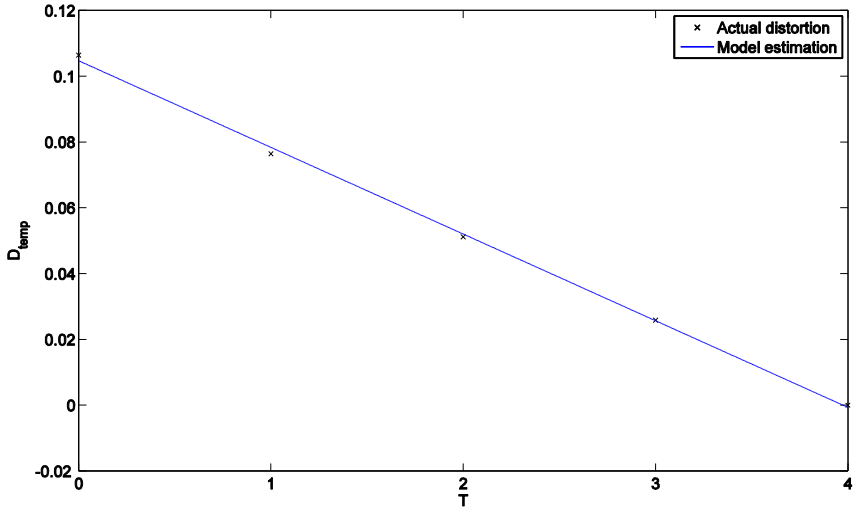



Figure 2.11 Temporal distortion as a function of *T* estimated by (2.15).

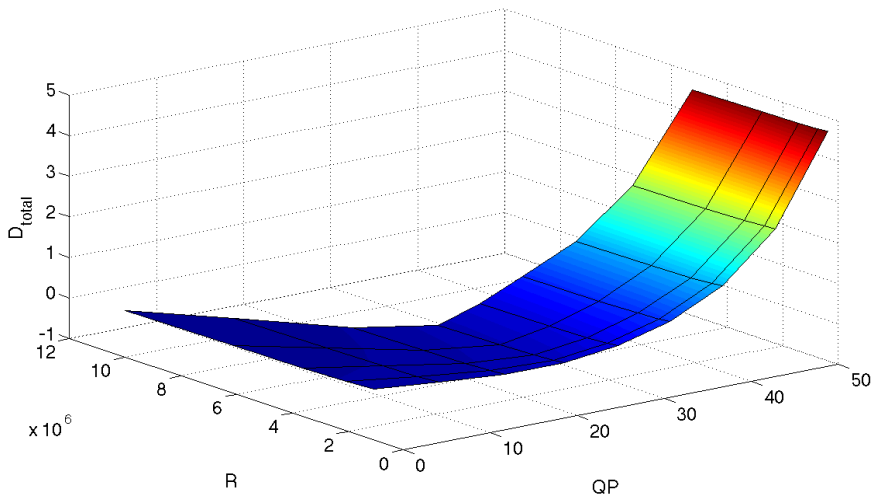
Thanks to the separation property proven in the theorem 2.1 and especially the lemma 2.2, we can combine both distortions in (2.9) and (2.15) to yield the joint distortion $D_{total}$ as follows:

$$D_{total}(QP,T) = D_{quant} + D_{temp} = a_{quant}\exp(b_{quant}QP) + a_{temp}T + a_{total}, \quad (2.16)$$

where $c_{quant}$ in (2.9) and $b_{temp}$ in (2.15) are absorbed into one constant.

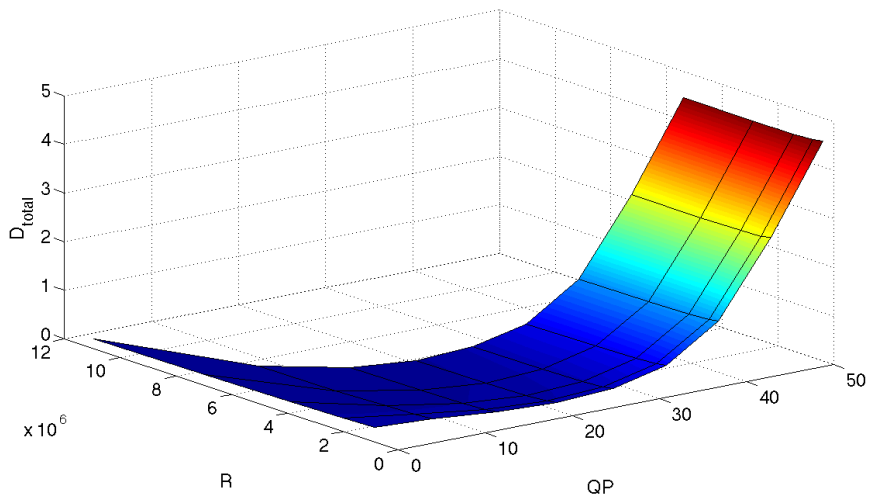
## 2.4 QP-Rate-Distortion Model

We now discuss the accuracy of our analytical model. Thus far, we have discussed the relationship between QP, temporal level, distortion, and rate, i.e., compressed data size. If we express the relationship without temporal level, we obtain the results shown in Fig. 2.12a, where the temporal change is implied in the variation of the rate, given a particular QP. The actual QP-Rate-Distortion surface graph is also shown in Fig. 2.12b for comparison. In Fig. 2.12, we can identify our model estimation is close to the actual result, which was confirmed for two other types of data as well.




(a)

Figure 2.12 (a) QP-Rate-Distortion surface of ambient temperature estimated by model (b) Actual QP-Rate-Distortion surface of ambient temperature data set.




(b)

Figure 2.12 (Continued).

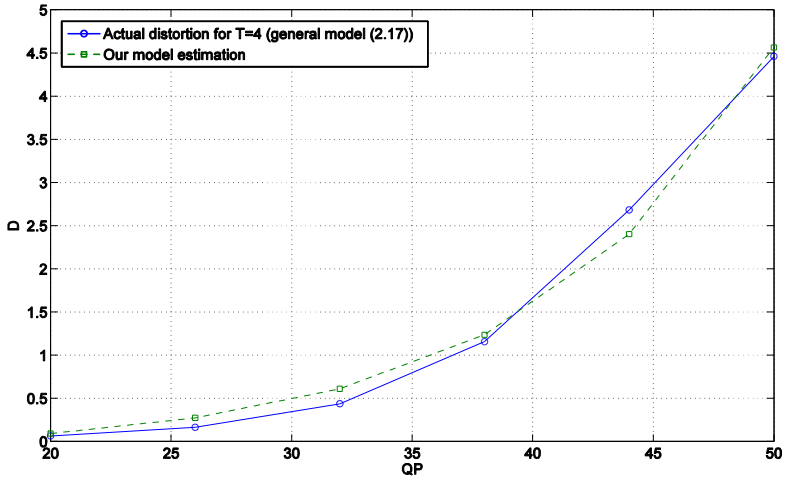
It is difficult to model our quality-adjustable archiving scheme using general rate-distortion models. For instance, a well-established modeling of rate and distortion for DCT-based video encoder is

$$D(\Delta) = \frac{\Delta^2}{\beta}, \qquad R(\Delta) = \frac{1}{2}\log_2\left(\frac{\varepsilon^2 \beta \sigma_x^2}{\Delta^2}\right), \tag{2.17}$$

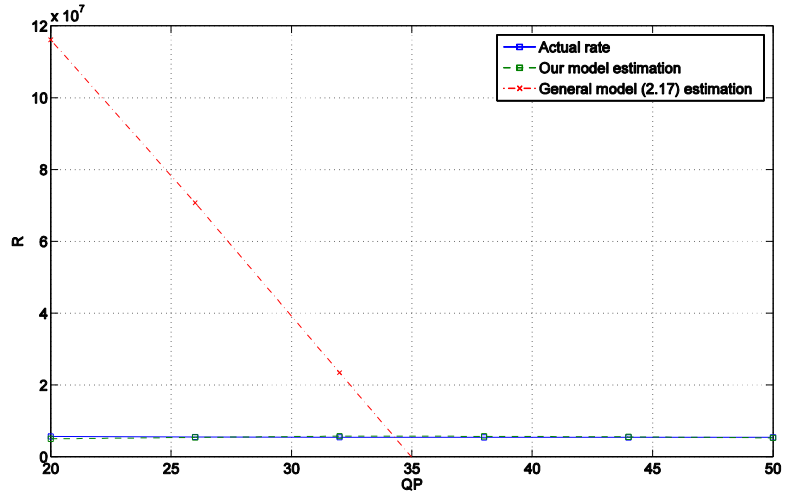
where $\beta$ is the same as in (2.5), and $\sigma_x^2$ is the variance of the source [47].

In (2.17), $\beta$ needs to be empirically adjusted to account for a wider range of $\Delta$. However, modeling our scheme with (2.17) yields discouraging results as shown in Fig. 2.13. In Fig. 2.13a, $\beta$ was adjusted to the actual distortion, which leads to the result identical to the actual distortion curve. On the contrary, the rate modeling of (2.17) with obtained $\beta$ is very far from the actual rate, as shown in Fig. 2.13b.

Furthermore, (2.17) has no provision for data fidelity control over temporal dimension, in contrast with our analytical model. Thus it is imperative that an accurate model is used in order to derive the optimal storage configuration strategy.




(a)




(b)

Figure 2.13 (a) Distortion curves comparison; (b) Rate curves comparison with (2.17) and our model. (Rate modeling of (2.17) yields negative values after *QP*=35.)

## 2.5 Optimal Rate Allocation

### 2.5.1 Rate Allocation Strategy

Using the analytical model derived in Section 2.3, our next concern is how to find the minimum distortion with a given specific rate $R_0$. The optimal rate allocation problem can then be formulated as follows:

$$\begin{aligned} &\min_{\{QP,T\}} D_{total}(QP,T) \\ &\text{s.t.} \quad R(QP,T) \le R_0, \end{aligned} \tag{2.18}$$

where $D_{total}(QP,T)$ and $R(QP,T)$ is the distortion and the rate function derived in (2.16) and (2.6), respectively.

Figure 2.14 shows the surface graph of $D_{total}(QP,T)$ derived in (2.16), where 10 contour plots, which are isolines of rate, are drawn together over the surface to reveal the contours of same rate over varying distortion. In Fig. 2.14, we can see that the minimum distortion can be obtained along the boundary of *QP* and *T*. Specifically, when there is available rate, it has to be first spent on reducing *QP*, and only after arriving at the minimum *QP* can the rate be spent on increasing the temporal level. This allocation strategy can also be explained by deriving the gradient of the distortion function, which is given by

$$\nabla D_{total}(QP,T) = (a_{quant} b_{quant} e^{b_{quant} QP}, a_{temp}). \tag{2.19}$$

In (2.19), the magnitude of $a_{temp}$ is much smaller than that of the *QP* component of

the gradient, which means it is more advantageous to adjust *QP* than temporal level in order to reach the minimum distortion quickly.

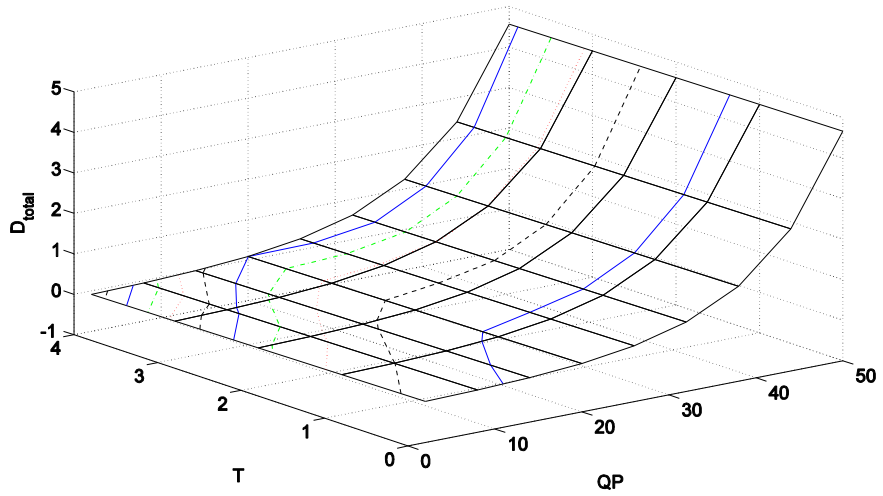



Figure 2.14 Isolines of rate over distortion surface.

### 2.5.2 Optimal Storage Configuration

We can furthermore extend the rate allocation problem of single sensor data block to accommodate more general case of storage configuration problem where multiple data blocks have to be stored efficiently. In our scheme, five temporal levels are supported with a fixed QP, which can be utilized as supplementary layers that can be gradually discarded as time elapses to handle less frequent data access. Figure 2.15 illustrates how incoming sensor data input is handled and archived with our archiving scheme. The quality management module first compresses raw sensor data block with a selected QP, which is then stored on the highest fidelity cluster, i.e. cluster 4. When a certain amount of time passes, the quality management module

discards the top layer and shift the data block to the next cluster. This process continues until the data block finally reaches cluster 0, where the data block is archived for a long time.

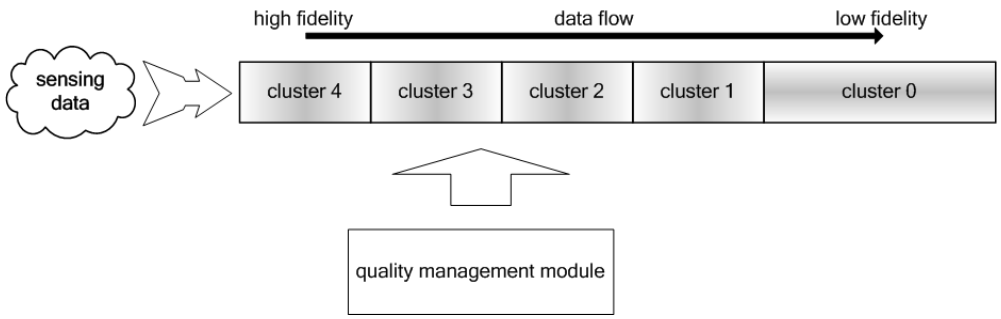



Figure 2.15 Data flow using our quality-adjustable archiving scheme.

Considering total storage efficiency, we are interested in how to allocate storage to each fidelity cluster and how to determine QP of each data block. Since each data block occupies less storage space in lower fidelity clusters than higher fidelity clusters, lower fidelity clusters can hold more data blocks given the same capacity. Besides, it is more natural to retain lower fidelity data longer than higher fidelity data. Assuming single sensor data type, the optimal storage configuration problem can then be formulated as follows:

$$\min_{\{QP_i, R_j\}} \sum_{j=0}^{j=4} \varphi_j \sum_{i=1}^{N} D_{total}(QP_i, j) \quad (\varphi_0 \gg \varphi_1 > \varphi_2 > \varphi_3 > \varphi_4 = 1)$$

$$\text{s.t.} \quad \varphi_j \sum_{i=1}^{N} R(QP_i, j) \le R_j \qquad (2.20)$$

$$\sum_{j=0}^{4} R_j \le R_{total},$$

where $QP_i$ denotes QP of each data block, $N$ is the number of data blocks in cluster 4, and $\varphi_j$ is a natural number denoting the proportion of data block numbers with respect to $N$. This equation describes a storage configuration at a certain instant where data blocks in lower fidelity clusters inherited QP's from data blocks in higher fidelity clusters. When the total rate budget $R_{total}$ is given, the optimal storage configuration should yield the overall minimum distortion.

The solution to (2.20) is an equal QP for each data block such that $\sum_{j=0}^{4} R_j \le R_{total}$, which no longer constrains $\varphi_j$ to be a natural number: $\varphi_j$ could be any positive rational number not less than 1. Hence the relationship between $R_j$'s is given by

$$\frac{R_j}{R_i} = \frac{\varphi_j}{\varphi_i} \cdot \exp(\beta(\Delta_l) \cdot (j - i)) \quad (j \ge i). \tag{2.21}$$

$N$ and $\varphi_j$ are system parameters that can be appropriately adjusted according to the target duration of retaining sensor data for each cluster.

The same result applies to a case when there are multiple sensor data types: an equal QP for each data block between the same types. However different sensor data types imply different model parameters, which leads to different QP's for different data types. In particular, the relationship between two different sensor data types using $QP_A$ and $QP_B$ can be represented as follows:

$$\frac{\sum_{j=0}^{4} \varphi_j D'_{total_A}(QP_A, j)}{\sum_{j=0}^{4} \varphi_j R'_A(QP_A, j)} = \frac{\sum_{j=0}^{4} \varphi_j D'_{total_B}(QP_B, j)}{\sum_{j=0}^{4} \varphi_j R'_B(QP_B, j)}, \tag{2.22}$$

where we used separate distortion and rate function for each QP. In (2.22), the ratio of the weighted sum of distortion slopes for each temporal level to the weighted sum of rate slopes for each temporal level is fixed. This result is a case of constant slope optimization [55, 56]: we obtain same marginal return for an extra rate spent on either sensor data type.

Utilizing the results, the optimal storage configuration strategy is first to determine proper QP's for each sensor data type in proportion to available storage, and then to encode sensing data input with the maximum temporal level. As time elapses, aged data blocks are shifted to next lower clusters till they reach the cluster 0. The gradually decreasing access pattern of sensor data is exploited using this scalable archiving scheme, resulting in efficient management of storage space.

### 2.5.3 Experimental Results

Although the solutions to (2.20) given in Section 2.5.2 are the optimal in analytical sense, we further want to show their optimality for selecting actual operating points of our archiving scheme. Given $N$, $\varphi_j$, and $R_{total}$, we first find the optimal QP's for each sensor data type using our analytical model, then actual operating points corresponding to the optimal QP's are selected to give overall distortion. We compare this overall distortion with other selection criteria: (i) uniform selection of arbitrary QP's even in the same sensor types; (ii) equal QP's for the same sensor types, but ignoring their relationship in (2.22).

Experimental results are shown in Table 2.2, where all of three storage

configuration strategies occupy the same storage space. However they exhibit dramatic difference in terms of overall distortion: the *arbitrary QP selection strategy* is the worst as expected, the *equal QP for the same sensor types strategy* shows better result, but neither of two strategies is comparable to *our optimal configuration strategy*. In other words, we spend the same amount of storage space for poorer overall data fidelity, which is equivalent to maintaining the same quality of data blocks while spending more amount of storage space. In addition, since the results in Table 2.2 are distortion ratios normalized by our optimal distortion, cumulative distortion will increase as $N$ increases to practical values for storage configuration. This result clearly shows the importance of the optimal storage configuration that has to be derived from proper analytical models.

Table 2.2 Distortion ratios of three strategies normalized by our strategy ( $N = 10$; $\varphi_0 = 10$, $\varphi_1 = 4$, $\varphi_2 = 3$, $\varphi_3 = 2$, $\varphi_4 = 1$)

| **Storage Configuration Strategy** | **Distortion Ratio** |
|---|---|
| Our Optimal Configuration | 1 |
| Arbitrary QP Selection | 8.3947 |
| Equal QP for the Same Sensor Types | 5.5941 |

# Chapter 3

# Scalable Management of Storage

In the previous chapter, we have seen the quality-adjustable sensor data archiving and its application for the optimal storage configuration. In this chapter, we add another quality dimension: the *quality enhancement layer*. This added quality dimension offers more options for controlling data fidelity, which should be advantageous to a scalable management of storage space. We derive analytical models that capture the characteristic of the added quality dimension, and study the optimal storage configuration strategy.

## 3.1 Scalable Quality Management

In Section 2.5, we discussed about utilizing the temporal levels as supplementary layers that can be gradually discarded to handle decreasing access pattern. In addition to adjusting temporal levels, another dimension can be employed to control the fidelity of data as well. This dimension is directly related to the management of quality of data, thus named as the quality enhancement layer.

In contrast to the QP adjustment that has to be determined prior to an encoding process, adding the quality dimension is close to the successive refinement concept

discussed in Section 1.3 apart from the fact that we reverse the refinement process such that discarding the highest layer one by one results in the efficient usage of storage space. In other words, the subset bitstream can be derived by dropping packets from the larger bitstream.

Few studies have embodied the quality adjustability of sensor data in their schemes [18-20, 41]. Since individual sensor nodes have limited storage space, fidelity control of aged data for storage efficiency may not be crucial in distributed environment. On the contrary, when we have to handle huge data from various sensor types, simply storing all data with the same fidelity is unacceptable considering storage efficiency and decreasing access pattern, which makes the quality adjustability essential in archiving sensor data.

Moreover, previous studies [18-20] supported the graceful degradation of quality by retaining multi-versions of fidelity blocks, which is contrary to the case of our scalable management where the graceful degradation is supported via one scalable data block. This difference in particular affects the coding efficiency since our scheme is designed to utilize the correlation among multiple fidelity levels, which yields better coding efficiency than previous studies.

### 3.1.1 Archiving Architecture

For the quality enhancement layer, we combine two layers each with different QP’s to enable the quality enhancement layer, which is illustrated in Fig. 3.1 where the base layer and the enhancement layer are encoded with different QP's. In Fig. 3.1, the temporal coding structure in Fig. 2.6 is extended to incorporate the quality enhancement layer, where the base layer represents the coding structure we have described in Section 2.3.1. Again, the range of temporal levels can be extended or reduced depending on applications, which entails modification of the temporal

coding and prediction structure. But we use this structure throughout the thesis.

Figure 3.1 shows temporal coding and prediction structure of the enhancement layer not only exploits the temporal correlation, but it benefits from the correlation inherent in the texture information of the base layer.

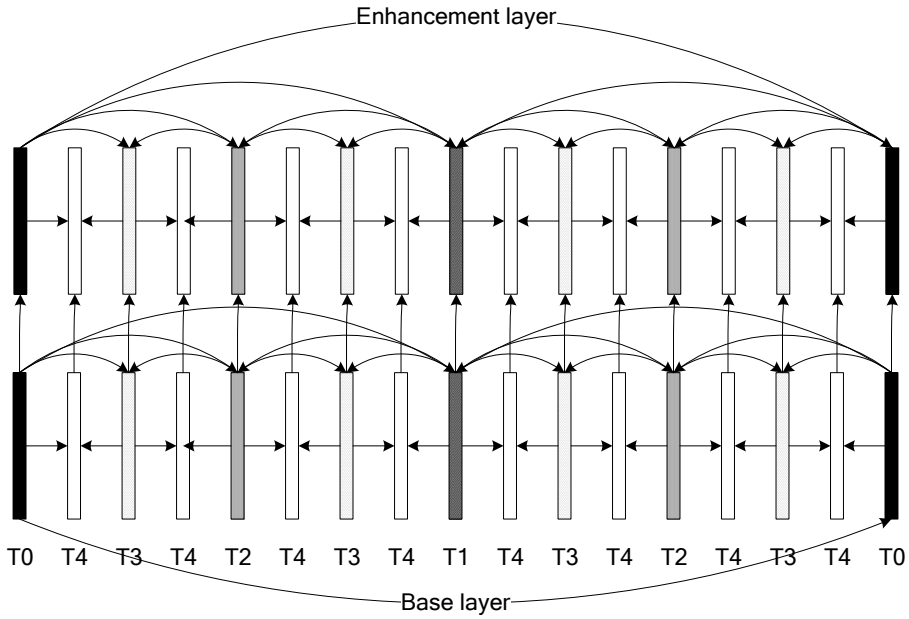



Figure 3.1 Temporal coding and prediction structure including quality enhancement layer.

Figure 3.2 shows an overview of *scalable quality management module* that is an extension of the quality management module in Fig. 2.2. Similar to the quality management module, incoming sensor input is first filtered through the spatio-temporal decorrelation module where most of correlation inherent in input data is removed, which in turn undergoes DCT for signal compaction. After that, the transformed data is subject to scalable quantization and entropy encode processes for lossy compaction of multiple layers. Again, the decoding process is the reverse

of this process. Here the scalable quantization process accounts for combining two layers: the base layer and the enhancement layer.

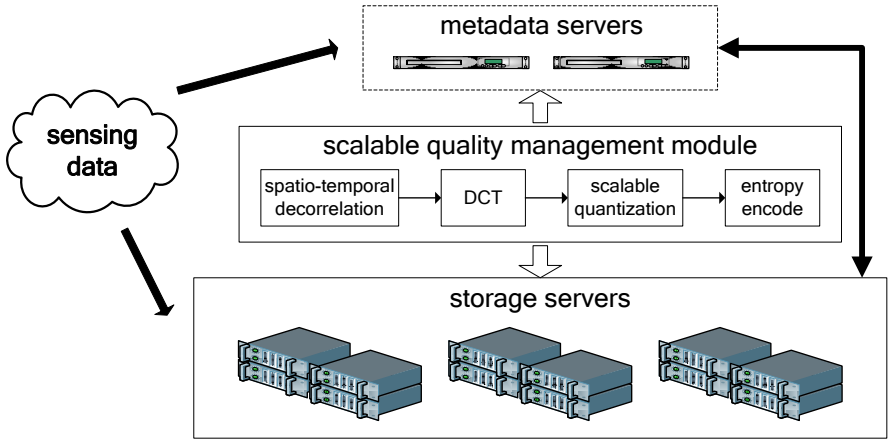



Figure 3.2 Scalable quality management module operating with distributed file system.

### 3.1.2 Compression Ratio Comparison

Figure 3.1 shows the combining of two layers each with different QP's. An ideal data block in accordance with the successive refinement concept should have data size equivalent to the size of non-scalable data block with lower QP. However, the actual combined data block using the scalable quality management module in Fig. 3.2 shows suboptimal data size due to the side information overhead as explained in Section 1.3. Nevertheless, the size of combined data block is still smaller than the sum of both data block with different QP's. Figure 3.3 shows the average compression ratios of sensor data sets with reference to the aggregate bitstream, where average ratio of aggregate bitstream to scalable bitstream and aggregate

bitstream to ideal bitstream are illustrated. The result of scalable bitstreams, which is our concern, is not very convincing when compared with the result of ideal bitstreams, showing ratios merely around 1.12. However a scalable data block is designed to be stripped down layer by layer and eventually to the base layer that has higher QP, so as to efficiently use the storage space. When the data block reaches the base layer, its coding efficiency is no more compromised. In Fig. 3.3, the QP difference between two layers is 12, which is a reasonable value when we consider a typical quality-enhanced data block.

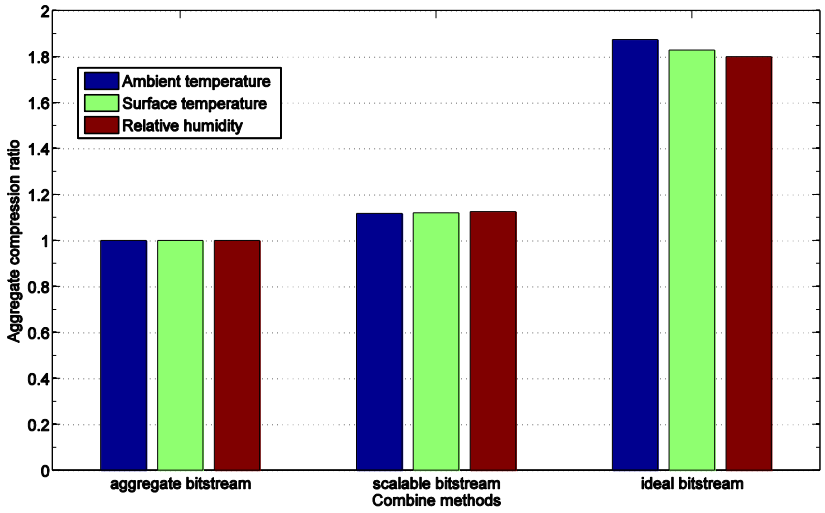



Figure 3.3 Average compression ratios of quality-scalable and ideal bitstream.

In order to present the performance of the quality enhancement layer, we compared our archiving scheme with lossless coding methods similar to Fig. 2.5a. Figure 3.4 shows the performance of our scheme employing the quality enhancement layer compared with lossless coding methods, where the compression

ratios are again expressed by the original raw data size divided by the compressed size. As in Fig. 2.5a, distortion incurred is still comparable to the order of sensor error margin $e^2$. In Fig. 3.4, our scheme exhibits some penalty in coding efficiency due to the overhead. Nevertheless, our scheme still shows impressive results and this penalty becomes insignificant, considering more data fidelity control options and vanishing penalty as layers are discarded.

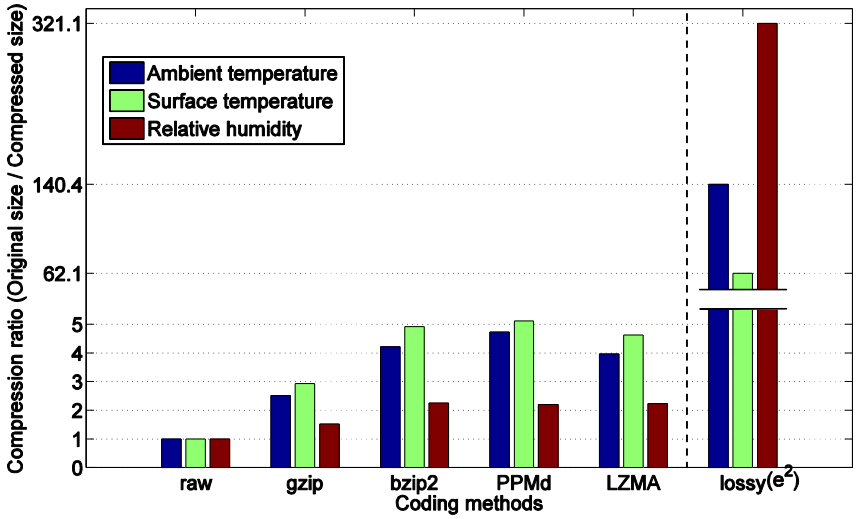



Figure 3.4 Compression ratios of our quality-scalable archiving scheme compared with lossless coding methods.

## 3.2 Enhancing Quality Adjustability

The scalable video coding, which is a notable realization of the successive refinement discussed in Section 1.3, has been a popular research area since its adoption to various video coding standards. In particular, quality scalability and its analytical modeling have been studied in several literatures [46, 57-60]. Although a

certain mechanism in scalable coding [61] can be well analyzed [46], others are difficult to model analytically due to different mechanisms [28, 29], which leads to solutions resorting to approximate estimation of quality scalable model [57-59]. In addition, these studies focus on the peak signal to noise ratio (PSNR) rather than MSE distortion itself since their target application is the scalable video. As a result, we should derive our own analytic models that precisely reflect the enhanced quality adjustability of our archiving scheme.

### 3.2.1 Data Fidelity Model: Rate

We found that the side information overhead of a quality-scalable bitstream can be effectively modeled by introducing additive scaling factors to the model parameters $\alpha(\Delta_l)$ and $\beta(\Delta_l)$ in (2.6), which represent the overhead of the quality enhancement layer as follows:

$$R(QP,T) = \left(\alpha(\Delta_l) + S_\alpha(\Delta_l)\right) \cdot \exp(\left(\beta(\Delta_l) + S_\beta\right) \cdot T), \tag{3.1}$$

where $S_\beta$ is a data-dependent constant and $S_\alpha(\Delta_l)$ is a model parameter that is a function of $\Delta_l$, which is given by

$$S_\alpha(\Delta_l) = a_S \Delta_l^{b_S} + c_S, \tag{3.2}$$

where $a_S$, $b_S$, and $c_S$ denote data-dependant constants. In (3.1), we can observe that the overhead incurred by the quality enhancement layer affects every temporal level of a quality-enhanced data block. Figure 3.5 shows actual data points and the model function of (3.1), which confirms accuracy of the modeling.

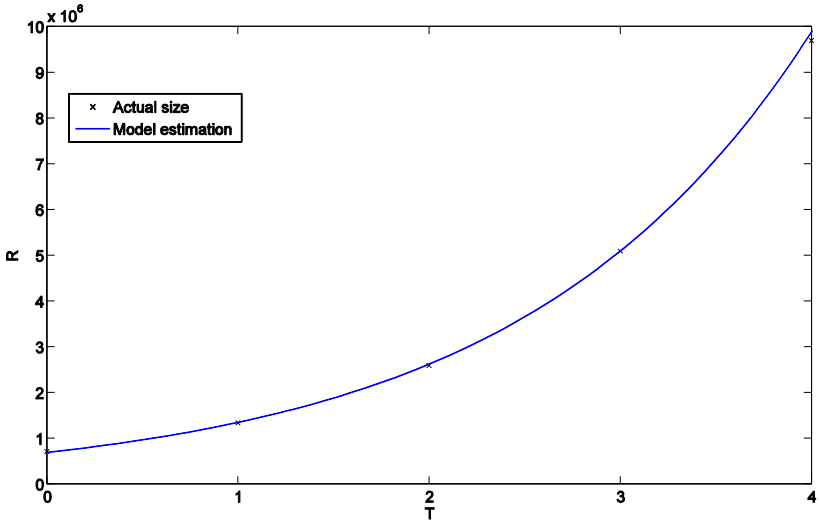



Figure 3.5 Rate curve of quality-enhanced data block as a function of temporal level estimated by (3.1) with *QP*=26.

In our coding structure, the number of quality enhancement points depends on the temporal coding structure of the base layer: given the base layer temporal level $T \in \{0,1,2,3,4\}$, there are *T*+1 quality enhancement points and one base layer point with no quality enhancement. For instance, if the base layer temporal level is 2, the enhancement layer temporal level can also increase up to 2, which is described in Fig. 3.6.

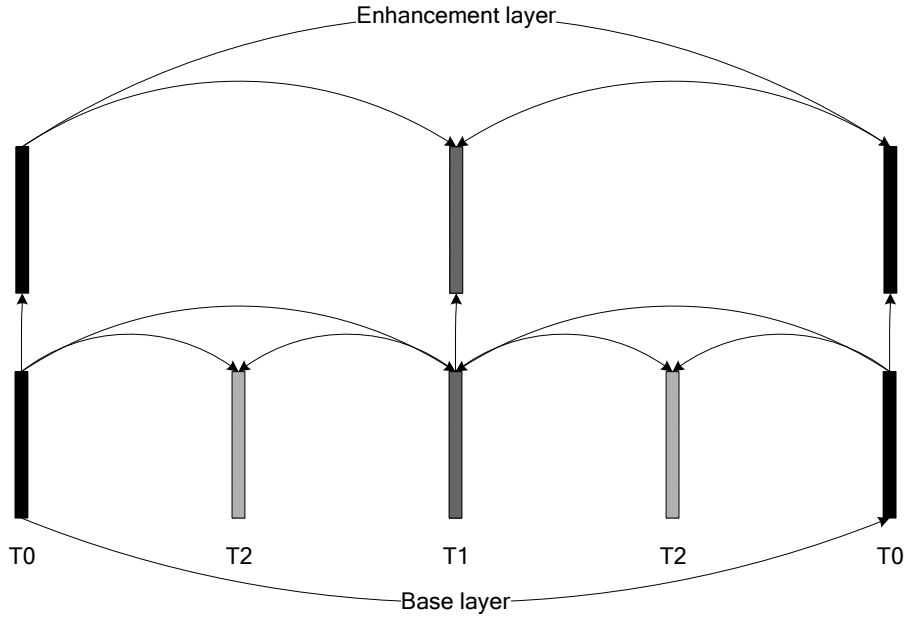



Figure 3.6 Temporal prediction structure with $T_{BASE} = 2$ and $T_{ENH} = 1$.

Since the base layer rate can be expressed using (2.6), we can accurately model these quality enhancement points as well. Specifically, if we let $T_{BASE}$ denote the base layer temporal level and $T_{ENH}$ the enhancement layer temporal level where $T_{ENH} \le T_{BASE}$, we can express the rate of every quality enhancement points $R_{ENH}$ as follows:

$$\begin{aligned} &R_{ENH}(QP_{ENH}, QP_{diff}, T_{ENH}, T_{BASE}) = R_{BASE}((QP_{ENH} + QP_{diff}), T_{BASE}) \\ &+\left(R_{TOT}(QP_{ENH}, T_{ENH}) - R_{BASE}((QP_{ENH} + QP_{diff}), T_{ENH})\right), \end{aligned} \tag{3.3}$$

where $R_{BASE}$ and $R_{TOT}$ represent the base layer rate and the total rate of the enhancement and base layers that are expressed by (2.6) and (3.1), respectively; $QP_{ENH}$ denote the QP of the enhancement layer and $QP_{diff}$ the QP difference of

base and enhancement layer. Thus (3.3) can be specified in detail as follows:

$$\begin{aligned}&\alpha(\Delta_{l_{BASE}})\exp(\beta(\Delta_{l_{BASE}})T_{ENH})\left(\exp(\beta(\Delta_{l_{BASE}})(T_{BASE}-T_{ENH}))-1\right)\\&+\left(\alpha(\Delta_{l_{ENH}})+S_{\alpha}(\Delta_{l_{ENH}})\right)\exp(\left(\beta(\Delta_{l_{ENH}})+S_{\beta}\right)T_{ENH}),\end{aligned} \tag{3.4}$$

where $\Delta_{l_{BASE}}$ and $\Delta_{l_{ENH}}$ denote the quantization step size of the base and enhancement layer. It should be noted that in (3.4), if $T_{ENH}$ equals $T_{BASE}$, $R_{ENH}$ becomes equivalent to $R_{TOT}$ in (3.1).

### 3.2.2 Data Fidelity Model: Distortion

Now that we found the quality scalable layer incurs a certain amount of overhead that can be incorporated in our rate model, our next question is how to properly model distortion at diverse quality enhancement points. We observed that distortion remains almost unchanged for the base layer and even for the enhancement layer as compared to the case of non-scalable data block. Moreover, the separation property of the quantization and temporal distortion proven in the lemma 2.2 also holds for the quality scalability: the temporal distortion only depends on the base layer temporal level regardless of the enhancement layer temporal level. This can be recognized by looking at Fig. 3.6, where we can find that a variation in the enhancement layer temporal level does not affect underlying base layer temporal encoding structure. In other words, the enhancement layer temporal level is solely related to the distortion of the quantization from quality control.

The above observation for the distortion in quality scalable dimension leads to the conclusion that we can use the distortion model derived in (2.16) in order to estimate the distortion of quality-enhanced data block. However this observation is valid only if the base layer temporal level is identical to the enhancement layer

temporal level such that $T_{ENH} = T_{BASE}$. In case of the quality enhancement points not being full temporal level such that $T_{ENH} < T_{BASE}$, we have to find another way of estimating distortion in these quality enhancement points.

By the separation lemma 2.2, we already know the temporal distortion can be modeled as a linear function of the base layer temporal level as in (2.15). Hence we can concentrate on the relationship between quality enhancement points when $T_{BASE}$ is equal to 4. We empirically derived the linear relationship between ratios of distortions of quality enhancement points with $T_{ENH} \le T_{BASE} = 4$ to the distortion of full temporal enhancement level with $T_{ENH} = T_{BASE} = 4$, which is given by

$$\begin{aligned} S_D(QP_{ENH}, QP_{diff}, T_{ENH}) &= \frac{D_{quality}(QP_{ENH}, QP_{diff}, T_{ENH})}{D_{total}(QP_{ENH}, 4)} \\ &= \rho(QP_{ENH}, QP_{diff}) \cdot (4 - T_{ENH}) + 1, \end{aligned} \tag{3.5}$$

where $S_D(QP_{ENH}, QP_{diff}, T_{ENH})$ is the ratio of distortion, $D_{quality}(QP_{ENH}, QP_{diff}, T_{ENH})$ is the distortion of quality enhancement points, $\rho(QP_{ENH}, QP_{diff})$ is a model parameter dependent on $QP_{ENH}$ and $QP_{diff}$. $\rho(QP_{ENH}, QP_{diff})$ can be derived by the following formula:

$$\rho(QP_{ENH}, QP_{diff}) = \alpha_\rho(QP_{diff}) \cdot \exp(\beta_\rho(QP_{diff}) \cdot QP_{ENH}), \tag{3.6}$$

where $\alpha_\rho(QP_{diff})$ and $\beta_\rho(QP_{diff})$ are model parameters dependent on $QP_{diff}$ that are also given by

$$\alpha_\rho(QP_{diff}) = a_{\alpha_\rho} QP_{diff} + b_{\alpha_\rho}, \tag{3.7}$$

$$\beta_\rho(QP_{diff}) = a_{\beta_\rho} QP_{diff} + b_{\beta_\rho}, \tag{3.8}$$

where $a_{\alpha_\rho}$, $b_{\alpha_\rho}$, $a_{\beta_\rho}$, and $b_{\beta_\rho}$ denote data-dependent constants.

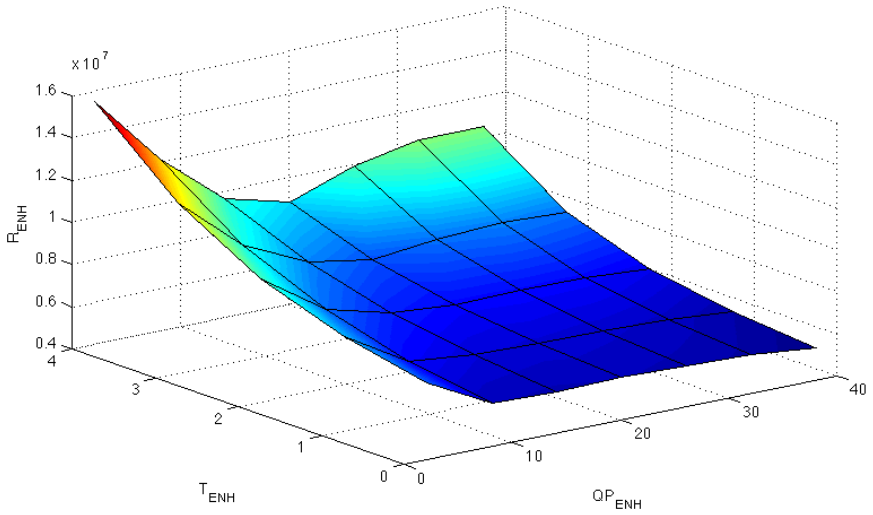
We can now combine the multiplicative scaling factor in (3.5) and the joint distortion $D_{total}$ in (2.16) as follows:

$$D_{quality}(QP_{ENH}, QP_{diff}, T_{ENH}) = S_D(QP_{ENH}, QP_{diff}, T_{ENH}) \cdot D_{total}(QP_{ENH}, 4). \tag{3.9}$$

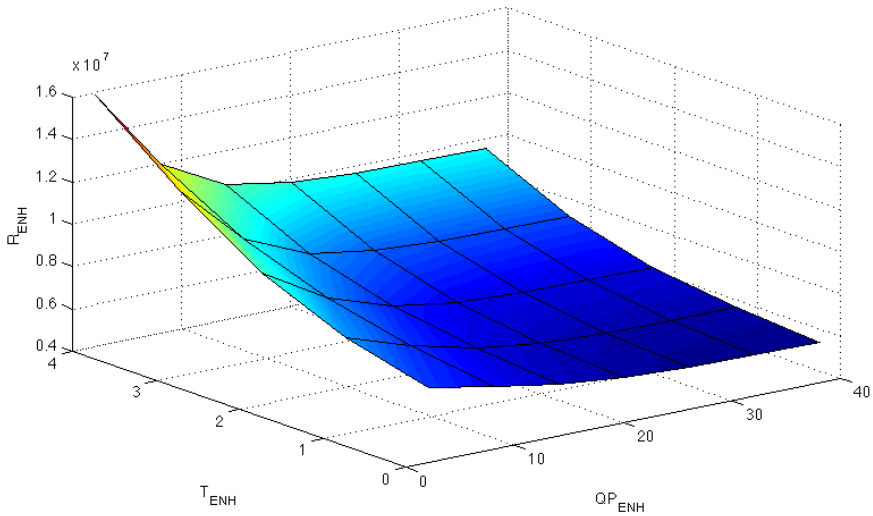
Finally, we include the temporal distortion in (2.15) and obtain

$$\begin{aligned} &D_{quality\_total}(QP_{ENH}, QP_{diff}, T_{ENH}, T_{BASE}) \\ &= D_{quality}(QP_{ENH}, QP_{diff}, T_{ENH}) + D_{temp}(T_{BASE}). \end{aligned} \tag{3.10}$$

The accuracy of the rate and distortion in quality enhancement points expressed by (3.3) and (3.9) can be verified by Fig. 3.7 and Fig. 3.8. Figure 3.7a shows the result of our model using (3.4), and Fig. 3.7b the surface graph describing the actual rate points. In Fig. 3.7, the rate points of seven different quality-enhanced data blocks with $T_{BASE} = 4$ and each having base and enhancement layer QP differences of 12 are shown. Similarly, Fig. 3.8a illustrates the result of our model using (3.9) and Fig. 3.8b actual distortion points, where we have the same configuration as in the case of Fig. 3.7. From Fig. 3.7 and Fig. 3.8, we can see that our model closely approximates actual rate and distortion points, which was confirmed for two other types of data as well.

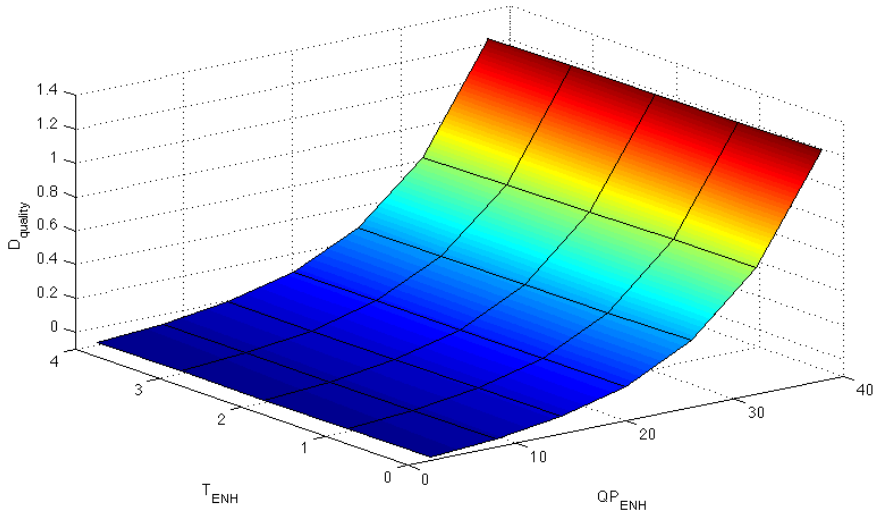



(a)

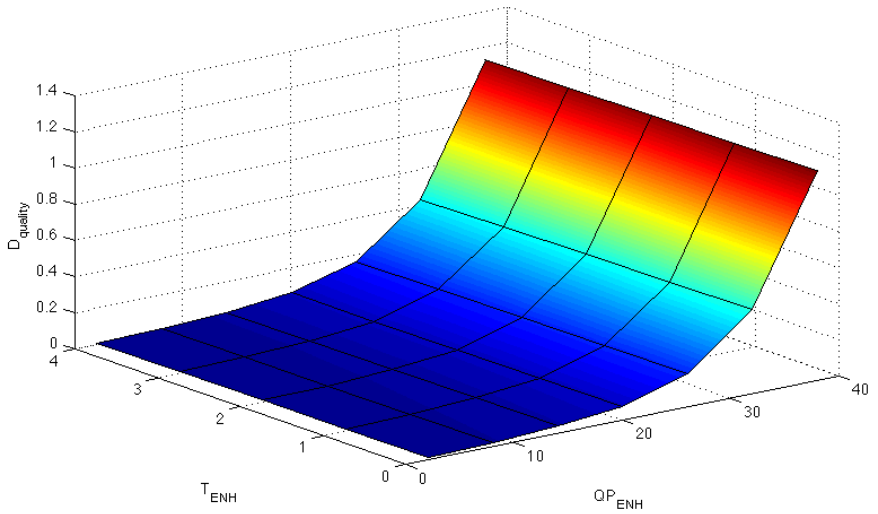



(b)

Figure 3.7 (a) Rate surface of quality-enhanced data block for ambient temperature as the function of $QP_{ENH}$ and $T_{ENH}$ estimated by (3.4) (b) Actual rate surface of quality-enhanced data block for ambient temperature data set.




(a)




(b)

Figure 3.8 (a) Distortion surface of quality-enhanced data block for ambient temperature as the function of $QP_{ENH}$ and $T_{ENH}$ estimated by (3.9) (b) Actual distortion surface of quality-enhanced data block for ambient temperature data set.

## 3.3 Optimal Rate Allocation

In Section 2.5, we have discussed how to allocate rate optimally in order to minimize distortion when we can control the quantization parameter *QP* and the temporal level *T*, where we have reached to a conclusion that the rate should be first spent on reducing *QP* rather than increasing *T*. Moreover, we extended our analysis to the case of multiple sensor data blocks from various sensor types, which led to the optimal storage configuration strategy.

Now that we have added another quality dimension, i.e., the quality enhancement layer, we have also added other controllable parameters, which are the quantization parameter and temporal level of enhancement layer, namely $QP_{ENH}$ and $T_{ENH}$ ; and the QP difference of base and enhancement layer $QP_{diff}$ . Therefore we have to take these controllable parameters into account when we consider the optimal rate allocation problems.

### 3.3.1 Rate Allocation Strategy

Using the analytical models derived in Section 3.2, we are now interested in how to allocate rate optimally in order to minimize distortion with a given specific rate $R_0$, adjusting various parameters. If we focus on the minimum distortion of the enhancement layer, then the rate allocation problem would be formulated as follows:

$$\begin{aligned} \min_{\{QP_{ENH},QP_{diff},T_{ENH},T_{BASE}\}} \quad & D_{quality\_total}(QP_{ENH},QP_{diff},T_{ENH},T_{BASE}) \\ \text{s.t.} \quad & R_{ENH}(QP_{ENH},QP_{diff},T_{ENH},T_{BASE}) \le R_0 \end{aligned}, \tag{3.11}$$

where $D_{quality\_total}(QP_{ENH},QP_{diff},T_{ENH},T_{BASE})$ and $R_{ENH}(QP_{ENH},QP_{diff},T_{ENH},T_{BASE})$ are the distortion and the rate function derived in (3.10) and (3.3), respectively.

Figure 3.9 illustrates 10 contour plots that are isolines of rate, drawn over the surface graph of distortion shown in Fig. 3.8a to display the contours of same rate

over differing distortion. Comparing Fig 3.9 to Fig. 2.14, we can see similar results as distortion can be minimized along the boundary of $QP_{ENH}$ and $T_{ENH}$ given a certain rate. In other words, available rate has to be first spent on minimizing $QP_{ENH}$, and only after arriving at the minimum $QP_{ENH}$ can the rate be spent on increasing $T_{ENH}$.

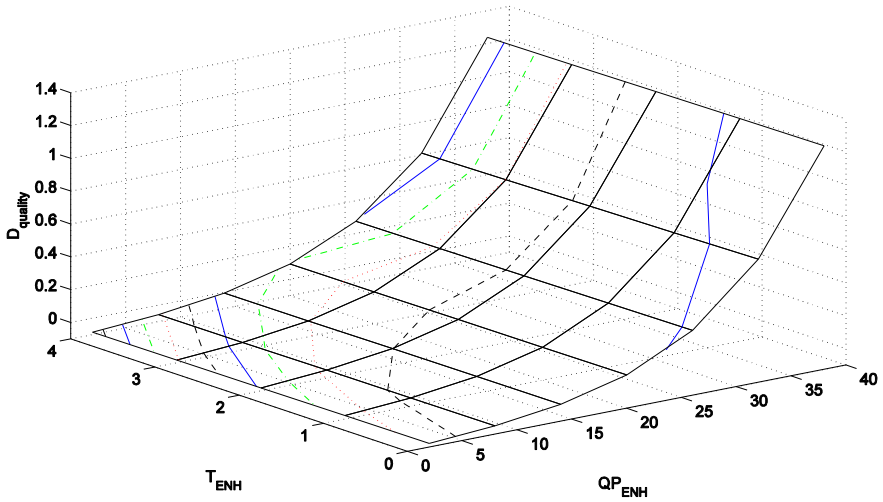



Figure 3.9 Isolines of rate over distortion surface of quality-enhanced data block.

In addition, $T_{BASE}$ and $T_{ENH}$ should be both integer values between 0 and 4, along with the condition $T_{ENH} \le T_{BASE}$. Figure 3.10 shows varying distortion with respect to both temporal levels, where we can identify $T_{BASE}$ governs most distortion while $T_{ENH}$ accounts for linear enhancement of quality.

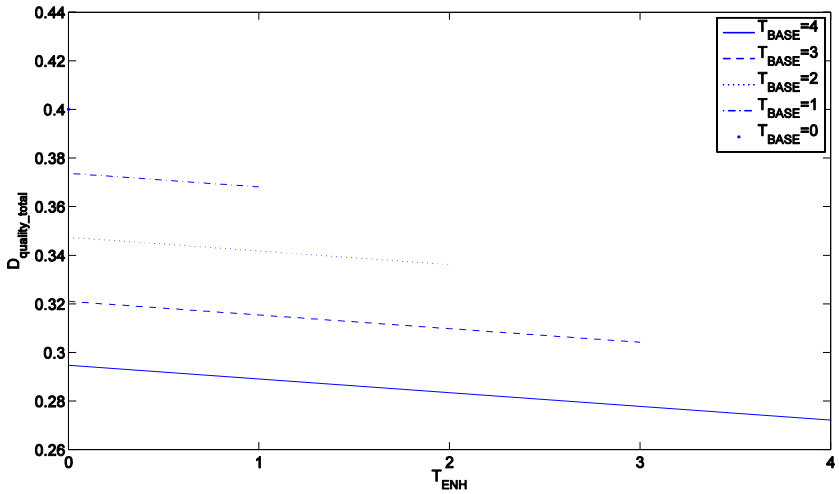



Figure 3.10 Distortion graphs as a function of $T_{ENH}$ for different $T_{BASE}$'s estimated by (3.10) with $QP_{ENH} = 26$ and $QP_{diff} = 12$.

We are also interested in the effect of varying $QP_{diff}$ on distortion, which is illustrated in Fig. 3.11. In contrast to the result in Fig. 3.10, no dominant factor is found between $T_{ENH}$ and $QP_{diff}$, which can also be explained by wider range of possible values $QP_{diff}$ can have unlike the limited range of $T_{ENH}$.

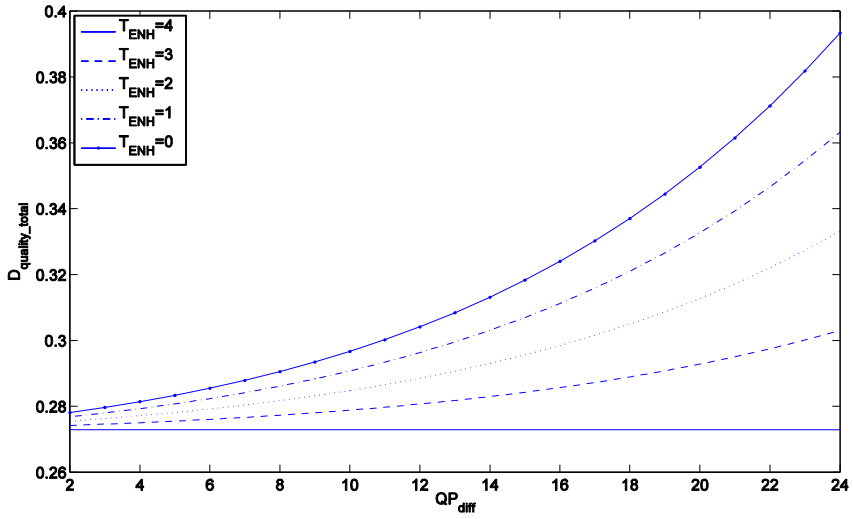



Figure 3.11 Distortion graphs as functions of $QP_{diff}$ 's for different $T_{ENH}$ 's estimated by (3.10) with $QP_{ENH} = 26$ and $T_{BASE} = 4$.

Since $QP_{ENH}$, $QP_{diff}$, $T_{ENH}$, and $T_{BASE}$ are all non-negative integer values, now (3.11) turns into the problem of nonlinear integer programming [62]. In addition, we can draw a general rule of thumb from above results: the priority of four variables is the following order $QP_{ENH}$, $T_{BASE}$, $T_{ENH}$, and $QP_{diff}$. In other words, $QP_{ENH}$ is generally the most important factor for the enhancement layer distortion under given specific rate, and $T_{BASE}$ precedes $T_{ENH}$, which $QP_{diff}$ follows.

### 3.3.2 Optimal Storage Configuration

The rate allocation problem of single sensor data block in the previous subsection can be extended to more general case of storage configuration problem where multiple data blocks have to be stored efficiently. With fixed $QP_{ENH}$ and $QP_{diff}$, nine quality enhancement points are supported that can be utilized as supplementary

layers, which can be gradually discarded as time elapses to handle less frequent data access.

Figure 3.12 illustrates how incoming sensor data input is handled and archived with our scalable quality management scheme. The scalable quality management module first compresses raw sensor data block with selected $QP_{ENH}$ and $QP_{diff}$, which is then stored on the highest fidelity scalable cluster, i.e. scalable cluster 8. When a certain amount of time passes, the scalable quality management module discards the top layer and shift the data block to the next cluster. This process continues until the data block finally reaches the scalable cluster 0, after which the final quality enhancement point, i.e., $T_{ENH} = 0$, is discarded and the sensor data block is permanently archived with $T_{BASE} = 0$ only.

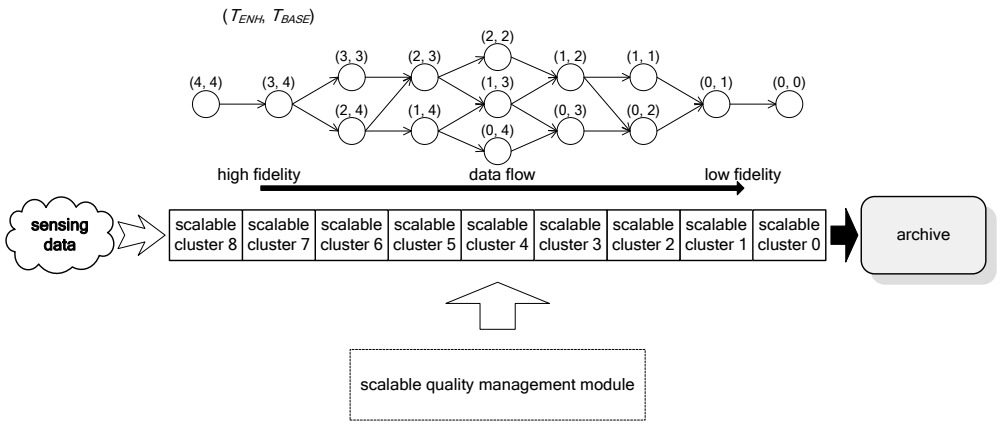



Figure 3.12 Data flow using our scalable quality management scheme.

Concerning total storage efficiency, we are interested in how to allocate storage to each fidelity scalable cluster and how to determine $QP_{ENH}$ and $QP_{diff}$ of each sensor data block. Since each block occupies less storage space in lower fidelity scalable clusters than higher fidelity scalable clusters, lower fidelity scalable

clusters can hold more sensor data blocks given the same capacity. Moreover, it is more natural to retain lower fidelity data longer than higher fidelity data. If we assume unique sensor data type, the optimal storage configuration problem can be formulated as follows:

$$\min_{\left\{QP_{ENH_i}, QP_{diff_i}, R_j\right\}} \quad \sum_{j=0}^{8} \varphi_j \sum_{i=1}^{N} D_{quality\_total}(QP_{ENH_i}, QP_{diff_i}, T_{ENH_j}, T_{BASE_j})$$

$$\text{s.t.} \quad \varphi_j \sum_{i=1}^{N} R_{ENH}(QP_{ENH_i}, QP_{diff_i}, T_{ENH_j}, T_{BASE_j}) \le R_j, \quad \sum_{j=0}^{8} R_j \le R_{total}, \quad (3.12)$$

$$\left\{T_{ENH_j} + T_{BASE_j} = j,\ \varphi_j > \varphi_i\ (j < i,\ \varphi_8 = 1)\right\}$$

where $QP_{ENH_i}$ and $QP_{diff_i}$ denote $QP_{ENH}$ and $QP_{diff}$ of each sensor data block, $N$ is the number of sensor data blocks in the scalable cluster 8, and $\varphi_j$ is a natural number denoting the proportion of block numbers with respect to $N$. A storage configuration at a specific instant is described by (3.12) where sensor data blocks in lower fidelity scalable clusters inherited $QP_{ENH}$'s and $QP_{diff}$'s from sensor data blocks in higher fidelity scalable clusters. Given the total rate budget $R_{total}$, the optimal storage configuration yields the system-wide minimum distortion.

The analytical solution to (3.12) is an equal $QP_{ENH}$ and an equal $QP_{diff}$ for each sensor data block such that $\sum_{j=0}^{8} R_j \le R_{total}$. This result no longer constrains $\varphi_j$ to be a natural number: $\varphi_j$ can be any positive rational number that is not less than 1. $N$ and $\varphi_j$ become system parameters that can be appropriately adjusted according to target duration of retaining sensor data for each scalable cluster.

A similar result to (3.12) applies to a case when multiple sensor data types should coexist on the storage: an equal $QP_{ENH}$ and an equal $QP_{diff}$ for each

sensor data block between the same sensor data type. But different sensor data types imply different model parameters, which results in different $QP_{ENH}$ 's and $QP_{diff}$ 's for different sensor data types. Specifically, the relationship between two different sensor data types A and B can be represented as follows:

$$\begin{aligned}
&\frac{\sum_{j=0}^{8}\varphi_j \cdot \frac{\partial D_{quality_total_A}(QP_{ENH_A}, QP_{diff_A}, T_{ENH_{A_j}}, T_{BASE_{A_j}})}{\partial QP_{ENH_A}}}{\sum_{j=0}^{8}\varphi_j \cdot \frac{\partial R_{ENH_A}(QP_{ENH_A}, QP_{diff_A}, T_{ENH_{A_j}}, T_{BASE_{A_j}})}{\partial QP_{ENH_A}}} \\
&= \frac{\sum_{j=0}^{8}\varphi_j \cdot \frac{\partial D_{quality_total_B}(QP_{ENH_B}, QP_{diff_B}, T_{ENH_{B_j}}, T_{BASE_{B_j}})}{\partial QP_{ENH_B}}}{\sum_{j=0}^{8}\varphi_j \cdot \frac{\partial R_{ENH_B}(QP_{ENH_B}, QP_{diff_B}, T_{ENH_{B_j}}, T_{BASE_{B_j}})}{\partial QP_{ENH_B}}} \\
&= \frac{\sum_{j=0}^{8}\varphi_j \cdot \frac{\partial D_{quality_total_A}(QP_{ENH_A}, QP_{diff_A}, T_{ENH_{A_j}}, T_{BASE_{A_j}})}{\partial QP_{diff_A}}}{\sum_{j=0}^{8}\varphi_j \cdot \frac{\partial R_{ENH_A}(QP_{ENH_A}, QP_{diff_A}, T_{ENH_{A_j}}, T_{BASE_{A_j}})}{\partial QP_{diff_A}}} \\
&= \frac{\sum_{j=0}^{8}\varphi_j \cdot \frac{\partial D_{quality_total_B}(QP_{ENH_B}, QP_{diff_B}, T_{ENH_{B_j}}, T_{BASE_{B_j}})}{\partial QP_{diff_B}}}{\sum_{j=0}^{8}\varphi_j \cdot \frac{\partial R_{ENH_B}(QP_{ENH_B}, QP_{diff_B}, T_{ENH_{B_j}}, T_{BASE_{B_j}})}{\partial QP_{diff_B}}},
\end{aligned} \tag{3.13}$$

where we used separate distortion and rate function for each sensor data type. In (3.13), $QP_{ENH_A}$ and $QP_{diff_A}$ denote $QP_{ENH}$ and $QP_{diff}$ for sensor data type A; $QP_{ENH_B}$ and $QP_{diff_B}$ for sensor data type B. From this result, we can deduce that the ratio of the weighted sum of partial derivative of distortion with respect to $QP_{ENH}$, to the weighted sum of partial derivative of rate with respect to $QP_{ENH}$ for

each sensor data type is fixed; in addition, this ratio applies to partial derivative with respect to $QP_{diff}$ in the same way. This result is another case of the constant slope optimization [55, 56]: in the optimal operating points, we obtain same marginal return for an extra rate spent with adjusting $QP_{ENH}$ or $QP_{diff}$ on either sensor data type.

As we know an equal $QP_{ENH}$ and an equal $QP_{diff}$ have to be selected between the same sensor types throughout entire scalable clusters, the next question is how to determine $T_{ENH}$ and $T_{BASE}$ of each sensor data block within particular scalable cluster *j*. This problem can be formulated as follows:

$$\begin{aligned} \min_{\{T_{ENH_i}, T_{BASE_i}\}} \quad & \sum_{i=1}^{N} D_{quality\_total}(QP_{ENH}, QP_{diff}, T_{ENH_i}, T_{BASE_i}) \\ \text{s.t.} \quad & \sum_{i=1}^{N} R_{ENH}(QP_{ENH}, QP_{diff}, T_{ENH_i}, T_{BASE_i}) \le R_j \\ & \{T_{ENH_i} + T_{BASE_i} = j\} \end{aligned} \tag{3.14}$$

Unsurprisingly, the solution to (3.14) is an equal $T_{ENH}$ and an equal $T_{BASE}$ for each sensor data block in particular scalable cluster. In fact, we can describe $(T_{ENH}, T_{BASE})$ pairs that belong to specific clusters using a graph as shown in Fig. 3.12, where a scalable cluster index corresponds to the sum of $T_{ENH}$ and $T_{BASE}$. A routing path from (4, 4) to (0, 0) in this graph represents a possible selection of $(T_{ENH}, T_{BASE})$ pairs in the course of entire data aging process. According to the solution of (3.14), each scalable cluster from 2 to 6 has equal $(T_{ENH}, T_{BASE})$ pairs for the same sensor type data blocks.

Thus we have to pick one selection of $(T_{ENH}, T_{BASE})$ pair for the scalable clusters from 2 to 6; the selection process should be based on possible routing paths

in the graph. If we count the number of routing paths in Fig. 3.12, we obtain 14 different paths. However, not all of them are eligible for quality enhancement paths: some paths yield suboptimal results in terms of the *Pareto efficiency* as shown in Fig. 3.13. Figure 3.13 represents possible storage configurations at a specific time instant, where storage consumption and system-wide distortion are displayed. Obviously we should choose configurations that yield the minimum system-wide distortion under a certain rate budget.

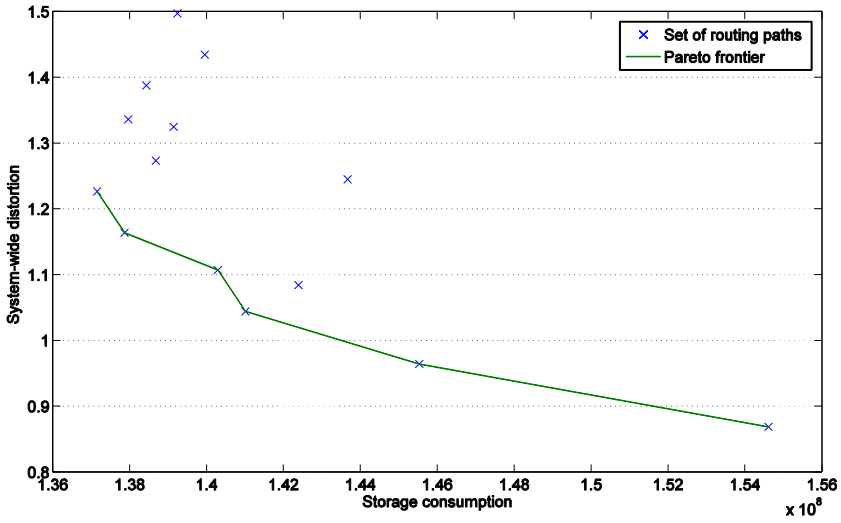



Figure 3.13 Possible storage configurations for ambient temperature data and their Pareto frontier. ( $N=1$, $QP_{ENH}=2$, $QP_{diff}=12$; $\varphi_0=9$, $\varphi_1=8$, $\varphi_2=7$, $\varphi_3=6$, $\varphi_4=5$, $\varphi_5=4$, $\varphi_6=3$, $\varphi_7=2$, $\varphi_8=1$)

Finding optimal quality enhancement paths with specific $QP_{ENH}$ and $QP_{diff}$ can be carried out using a deterministic dynamic programming with a trellis diagram that represents all possible solutions [63]. In the trellis diagram, each stage corresponds to sensor data blocks in particular scalable cluster with its index equal

to the sum of $T_{ENH}$ and $T_{BASE}$ in question, and each node of the trellis at a given stage represents a possible cumulative rate usage. In addition, each branch has a distortion corresponding to a particular choice of $(T_{ENH}, T_{BASE})$ pair.

Figure 3.14 shows an example of dynamic programming using the trellis diagram with the same set-up as in Fig. 3.13. Since there is only one path from (4, 4) to (3, 4), we start the diagram from (3, 4). At the scalable cluster 3 that corresponds to (1, 2) and (0, 3) in Fig. 3.12, nine different routing paths are available as presented by nine separate nodes. In Fig. 3.14, cumulative distortion along different paths is listed in solid and dotted boxes. In particular, three dotted boxes represent suboptimal paths that yield more system-wide distortion than achievable.

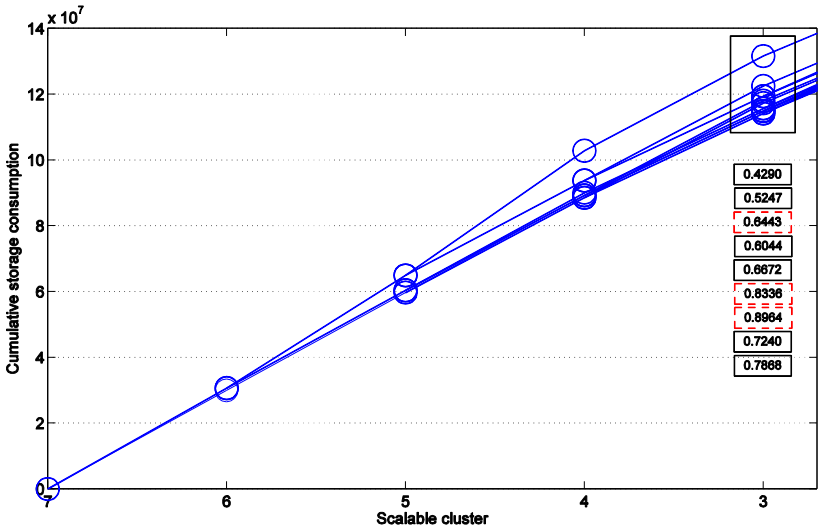



Figure 3.14 Dynamic programming using trellis diagram.

In fact, all of these suboptimal paths converge at (1, 2); but there are other paths that converge at (1, 2) and still provide Pareto optimal results. Thus we can

remove the three suboptimal routing paths without penalty and proceed to the next stage. Putting together, an algorithm for the optimal storage configuration strategy is described in Fig. 3.15.

**1)** *Determine sensor data block size and duration of retaining sensor data for each scalable cluster*
*- adjust system parameters* $N$ *and* $\varphi_j$ *accordingly*
**2)** *Find model parameters and data-dependent constants for different sensor data types*
**3)** *Determine proper* $QP_{ENH}$ *and* $QP_{diff}$ *for a specific sensor data type in proportion to available storage*
*- minimizing* $QP_{ENH}$ *is far more effective for decreasing system-wide distortion in data aging process than minimizing* $QP_{diff}$ *; minimizing* $QP_{diff}$ *is more responsible for quality of permanently archived sensor data*
**4)** *Encode sensing data input*
**5)** *Find optimal routing paths in quality enhancement points with dynamic programming using the result in Step 4*
*- determine proper routing path in proportion to available storage*
**6)** *Store encoded data block with the highest quality enhancement points* $(4, 4)$ *on the scalable cluster* $8$
**7)** *As the duration determined in Step 1 elapses, discard top layer and shift aged sensor data blocks to next lower scalable clusters following the routing path in Step 5*
**8)** *If aged sensor data blocks are set at the scalable cluster* $0$*, discard the last layer and permanently archive them*

Figure 3.15 Algorithm for the optimal storage configuration.

### 3.3.3 Experimental Results

We now show the optimality of solutions to (3.12) in Section 3.3.2 by selecting actual operating points of our scalable archiving scheme. Given $N$, $\varphi_j$, and $R_{total}$, we find the optimal pairs of $QP_{ENH}$ and $QP_{diff}$ for each sensor data type using the relationship in (3.13), then actual operating points corresponding to analytical solutions are selected for our configuration. We compare this result with a configuration of equal pairs of $QP_{ENH}$ and $QP_{diff}$ only for the same sensor types, but ignoring the relationship in (3.13).

In reality, it is difficult to satisfy (3.13) strictly since the amount of impact on derivatives may be mismatched between $QP_{ENH}$ and $QP_{diff}$. In this case, $QP_{ENH}$ and $QP_{diff}$ can be separately handled to satisfy (3.13). Table 3.1 shows experimental results of *our optimal configuration strategy* and *equal pairs of* $QP_{ENH}$ *and* $QP_{diff}$ *for the same sensor types strategy*. Both strategies exhibit the same storage consumption, while incurring distinct distortion ratios normalized by our optimal distortion. In Table 3.1, we can again verify the importance of determining optimal parameters as in Section 2.5.3.

Table 3.1 Distortion ratios of different strategies normalized by our strategy ( $N=1$; $\varphi_0=9$, $\varphi_1=8$, $\varphi_2=7$, $\varphi_3=6$, $\varphi_4=5$, $\varphi_5=4$, $\varphi_6=3$, $\varphi_7=2$, $\varphi_8=1$)

| **Storage Configuration Strategy** | **Distortion Ratio** |
|---|---|
| Our Optimal Configuration | 1 |
| Equal parameters for the Same Sensor Types | 6.8069 |

We can also quantify the importance of determining optimal routing paths by

comparing distortion ratios. In particular, we take the case shown in Fig. 3.13 for an example and show the result in Table 3.2, which compares the maximum difference of distortion ratios under a certain rate budget.

Table 3.2 Distortion ratio difference between routing paths in Fig. 3.13

| **Quality Enhancement Path** | **Distortion Ratio** |
|---|---|
| Our Optimal Path | 1 |
| Worst Case | 1.2865 |

Results in Table 3.1 and Table 3.2 provide a rationale for the use of the optimal storage configuration strategy in Fig. 3.15. The gradually decreasing access pattern of sensor data is effectively exploited using this scalable quality management strategy, resulting in efficient utilization of storage space.

# Chapter 4

# Quality-Adjustable Sensing

Thus far, we have seen how to optimally store massive collection of sensor data. Our next concern is the sensing environment: how to efficiently capture physical phenomena. This chapter addresses the quality-adjustable sensing that is suited for resource-limited sensors. To this end we adopt a different coding method called compressive sensing. We enhance the quality adjustability of the basic compressive sensing framework by introducing quantization and downsampling. We also discuss resource savings and coding efficiency improvement induced by the downsampling.

## 4.1 Compressive Sensing

Compressive sensing or compressed sampling (CS) is an emerging sensing/sampling paradigm that enables sampling of a signal under the Nyquist-Shannon sampling rate, where the signal must be sampled at least two times faster than the signal bandwidth [64-67].

A typical data acquisition scenario works as follows: massive amounts of data are collected and most part of them is discarded at the compression stage for storage and transmission needs. In particular, a signal is sampled at high frequency to

accommodate possible high frequency component inside that can be up to half the sampling frequency. The sampled signal is transformed using DCT or wavelet transform as explained in the previous chapters. The transformed signal in turn undergoes the quantization process, which inevitably involves discarding insignificant coefficients and keeping only a few largest coefficients.

This process of massive data acquisition followed by compression is wasteful, especially for resource-constrained devices. On the contrary, CS operates very differently as if it were possible to directly acquire just the important information about the object of interest.

Previous studies presented compression of data tailored for usage on mostly individual sensor node [18-20, 40-42, 68-74]. Among these studies, some of them adopted lossless coding schemes [40, 69, 71-74], while others adopted lossy coding schemes [18-20, 41, 42, 68, 70]. Whether their schemes were lossless or lossy, they tried to achieve two goals: energy savings in wireless transmission and storage usage savings, with the help of reduced sensor data size using compression.

Although these studies modified conventional coding schemes to adapt to resource-constrained sensor nodes, their computational complexity is still high as compared to that of CS, which hampers their wide adoption to various types of resource-limited devices such as wearable sensors [75]. In contrast, CS is well suited even for such limited devices.

### 4.1.1 Compressive Sensing Problem

In CS, a signal is projected onto random vectors whose cardinality is far below the dimension of the signal. For instance, consider a signal $\mathbf{x} \in \mathbb{R}^N$ that can be compactly represented in some orthogonal basis $\mathbf{\Psi}$ with only a few large coefficients and many small coefficients close to zero as follows:

$$\mathbf{x} = \mathbf{\Psi s}, \tag{4.1}$$

where $\mathbf{s} \in \mathbb{R}^N$ is the vector of transformed coefficients. In (4.1) $\mathbf{\Psi}$ could be any orthogonal basis that makes $\mathbf{x}$ sparse in transformed domain such as DCT and wavelet. The signal $\mathbf{x}$ is called *K*-sparse if it is a linear combination of only $K \ll N$ basis vectors in $\mathbf{\Psi}$: only *K* of the coefficients in $\mathbf{s}$ are significant.

CS projects $\mathbf{x}$ onto random sensing basis $\Phi \in \mathbb{R}^{M \times N}$ as follows ($M < N$):

$$\mathbf{y} = \mathbf{\Phi x} = \mathbf{\Phi \Psi s}, \tag{4.2}$$

where $\mathbf{\Phi}$ is generally constructed by sampling independent identically distributed (i.i.d.) entries from the Gaussian distribution with mean 0 and variance $1/M$. Instead of Gaussian, other sub-Gaussian distributions can be used such as Rademacher distribution, i.e., symmetric Bernoulli distribution [76]. (Sub-Gaussian is the distribution where moment-generating function is bounded by that of Gaussian, which has more uniform and shorter tail than Gaussian distributions.)

The system shown in (4.2) is ill-posed as the number of equations *M* is smaller than the number of variables *N*: there are infinitely many $\mathbf{x}$'s that satisfy $\mathbf{y} = \mathbf{\Phi x}$. Nevertheless, this system can be solved with overwhelming probability provided that $\mathbf{s}$ is sparse.

Here since $M < N$, the sampled (or measured) signal $\mathbf{y} \in \mathbb{R}^M$ is undersampled than the Nyquist-Shannon sampling rate. For a discrete-time signal such as $\mathbf{x}$, if a signal were dynamic at the granularity of each vector element, that is, if it had the highest frequency component, the Nyquist-Shannon sampling rate would force the length *N* of $\mathbf{x}$ preserved. In contrast, CS enables undersampling of the signal by

the length $M$ provided that $M = O(K\log(N/K))$.

### 4.1.2 General Signal Recovery

The signal recovery algorithm must take $\mathbf{y} \in \mathbb{R}^M$, the random sensing matrix $\mathbf{\Phi}$, and the orthogonal basis $\mathbf{\Psi}$. Then the recovery algorithm recovers $\mathbf{s}$ knowing that $\mathbf{s}$ is sparse; but it needs not know that $\mathbf{s}$ is exactly $K$-sparse. Once we recover $\mathbf{s}$, the original signal $\mathbf{x}$ can be recovered through (4.1).

It has been shown that the following linear program gives an accurate reconstruction of $\mathbf{s}$:

$$\min \|\tilde{\mathbf{s}}\|_1 \qquad \text{subject to} \qquad \mathbf{\Phi\Psi}\tilde{\mathbf{s}} = \mathbf{y}. \tag{4.3}$$

Apparently, there are many efficient linear programming algorithms that solve (4.3).

### 4.1.3 Noisy Signal Recovery

Suppose $\mathbf{y}$ were corrupted with a noise $\mathbf{z} \in \mathbb{R}^M$ that is a stochastic or deterministic unknown error term, which could be from communication channel or quantization. The corrupted $\hat{\mathbf{y}}$ can be represented as

$$\hat{\mathbf{y}} = \mathbf{\Phi\Psi s} + \mathbf{z}. \tag{4.4}$$

It has been shown that (4.4) can be solved using the following minimization problem with relaxed constraints for reconstruction:

$$\min \|\tilde{\mathbf{s}}\|_1 \qquad \text{subject to} \qquad \|\mathbf{\Phi\Psi}(\mathbf{s} - \tilde{\mathbf{s}}) + \mathbf{z}\|_2 \le \varepsilon, \tag{4.5}$$

where $\varepsilon$ bounds the amount of noise in the signal. Problem (4.5) is often called LASSO [77] and can also be solved efficiently.

## 4.2 Quality Adjustability in Sensing Environment

Various sensing devices from mobile phones to large scale sensor networks are essential in our daily lives. The near-optimal coding process we have seen in the previous chapters is not applicable to many resource-constrained devices due to its complexity. For instance, a sampled signal has to be transformed and quantized to discard insignificant coefficients. Since we are not aware of the exact positions of significant coefficients, the position information should be included as side information, as well as the coefficients themselves. These are in turn entropy-coded to yield the compact representation of the original signal. The entire chain of these processes is not an issue in storage environment we have discussed so far, where plenty of resources are available.

However, the sensing environment does not entirely consist of devices that are capable of this whole chain of coding processes. While some sensors such as mobile phones are fully fledged, others such as biosensors are severely resource-limited. At this point, CS comes into relief that can be used instead of conventional source coding schemes. CS shifts the complexity burden to the decoder where original signal is estimated in best-effort manner, which promotes its universal adoption among various types of sensing devices.

In Section 1.3, we discussed the quality adjustability of sensor data, from which we devised the efficient archiving scheme. Besides, the quality-adjustable nature of sensor data can also be leveraged in sensing environment. Sensing devices

may want to adjust data quality for various reasons: (i) energy, (ii) network bandwidth, and (iii) task overhead. If overall conditions get worse (e.g., device energy is low and CPU is loaded with other more important tasks), sensors can decrease data quality (reduce data rate); if conditions get better, vice versa.

Apart from its low complexity benefit, CS inherently supports a progressive refinement of data quality through the number of random measurements: the more measurements are received by the decoder, the better reconstruction of data is possible [78]. In other words, we get progressively better results as we compute more CS measurements. Therefore, CS is an ideal coding method for the sensing environment.

The progressive refinement feature in CS promises that the quality of recovered signal is as good as if one knew ahead of time the location of most significant pieces of information and decided to measure those directly [64]. This means that we need not send the side information that contains the position information of significant coefficients, which is automatically determined in decoding process.

Meanwhile, in Section 2.1.1, we briefly discussed the characteristics of DCT such as energy compaction and signal decorrelation, where we also mentioned that, being an approximation of KLT, it had similar performance to wavelets. In order to show the performance of three orthogonal transform bases (i.e., DCT, wavelet, and KLT) in terms of the progressive refinement feature, we present in Fig. 4.1 MSE results of three bases with varying percentage of significant transform coefficients included, where all other coefficients are set to zeros.

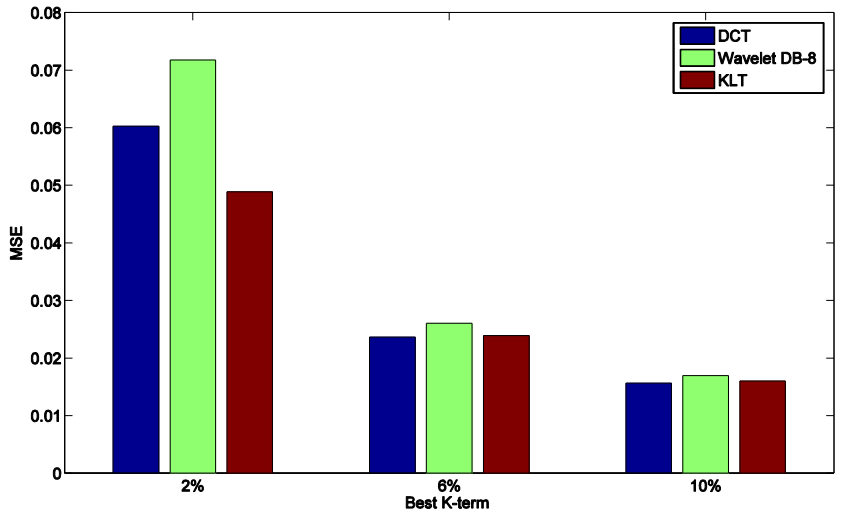



Figure 4.1 Best *K*-term approximations for three transform bases (Daubechies-8 wavelet used).

In Fig. 4.1, it is clear that three transform bases show similar performances. Since KLT is not ideal for actual implementations [32], we can select a particular $\Psi$ among DCT and various wavelet families. However, CS-equipped sensing devices need not decide which transform basis it will use; rather, a transform basis is decided at decoder. In other words, if a better transform basis is found in terms of signal energy compaction, the same random measurements can be used to reconstruct more accurate view of the original signal [78].

### 4.2.1 Quantization and Temporal Downsampling

Although CS supports the control of sensor data quality, simply relying on the adaptation of random measurements is insufficient for adjusting data quality; we need more options for the quality adjustment that can handle various context sensing devices are subject to.

Therefore, we employ (i) quantization and (ii) temporal downsampling into basic CS framework. This addition provides more rate-distortion operating points than basic CS framework, by which sensing data quality can be adapted in more efficient manner depending on various contexts.

If a *K*-sparse time-domain signal $\mathbf{x} \in \mathbb{R}^N$ captured by a sensing device is projected onto random sensing basis $\Phi \in \mathbb{R}^{M \times N}$ as in (4.2), it has been shown that $M \geq c \cdot K \log(N/K)$ random measurements is sufficient to recover the original signal [64]. In (4.2), each entry in $\mathbf{y} \in \mathbb{R}^M$ is usually represented in IEEE floating point format that is 32 bits (single precision) or 64 bits long (double precision). This measurement vector can be quantized to reduce length. In contrast to the quantization process we have seen in the previous chapters where the quantization occurred after the transform process, CS framework directly applies the quantization on the random measurements $\mathbf{y} \in \mathbb{R}^M$.

Thus the quantization on the measurement vector yields $\hat{\mathbf{y}} = \mathbf{\Phi\Psi s} + \mathbf{z}$ with a quantization noise $\mathbf{z}$. At decoder, this noise-corrupted signal can be reconstructed with the LASSO optimization problem as in (4.5), where we allow slack $\varepsilon$ in the constraint to account for the quantization noise.

The solution $\mathbf{s}^*$ to (4.5) obeys the following reconstruction error bound [64]:

$$\left\|\mathbf{s}-\mathbf{s}^*\right\|_2 \leq \underbrace{C_0 \cdot \left\|\mathbf{s}-\mathbf{s}_K\right\|_1 / \sqrt{K}}_{\text{measurement error}} + \underbrace{C_1 \cdot \varepsilon}_{\text{quantization error}}, \tag{4.6}$$

where $\mathbf{s}_K$ is the vector $\mathbf{s}$ with all but the largest *K* components set to 0; and $C_0$ and $C_1$ are constants depending on data. Although (4.6) provides us with an upper bound on the reconstruction error, the separable nature between measurement error (due to insufficient *M*) and quantization error (due to quantization noise) indicates

that both errors are also separable when calculating the expected error, which was indeed confirmed through our several experiments.

We now introduce a powerful tuning knob for adjusting data quality: *temporal downsampling*. The signal **x** is down-sampled and goes through CS with quantization. Figure 4.2 presents our quality-adjustable sensing architecture that extends CS with both quantization and temporal downsampling. Here the encoder and decoder use same pseudo-random matrix, which can be periodically updated using seed, e.g., combination of global seed and device ID. Note that this is a common practice in CS literatures [78-81].

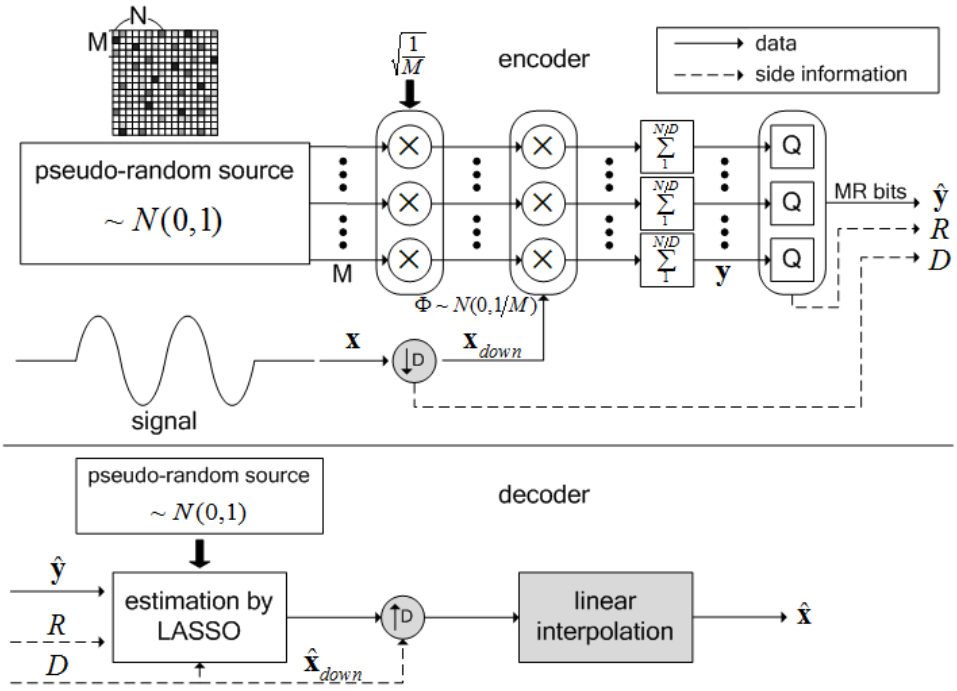



Figure 4.2 Quality-adjustable sensing architecture incorporating downsampling.

In Fig. 4.2, downsampling is performed without prior low-pass filtering at the

encoder, which should cause the aliasing of signal that are generally deemed to be undesirable. However, real low-pass filters (LPFs) are not comparable with an ideal filter in terms of sharp cutoff between passband and stopband. We found in experiments that using LPFs worsen the reconstruction quality most of times. LPFs incur much distortion when up-sampled and linear-interpolated although they reduce artifacts on down-sampled signal.

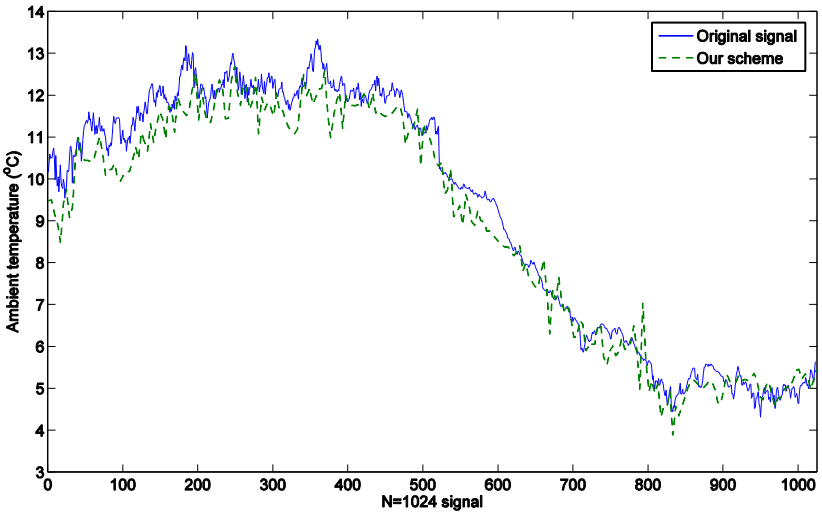
Three quality adjusting parameters affect overall performance of a sensing device as shown in Table 4.1: we need $O(MN)$ multiplication and summation operation (Computation) by default, which corresponds to $O(MR_{\max})$ total rate (Bandwidth) where $R_{\max}$ is the number of bits used by raw vector $\mathbf{y}$ without quantization. Among three parameters, in order to examine the effect of downsampling by factor of $D$, consider $M = c \cdot K \log(N/K)$ random measurements without the downsampling. The downsampling without LPF leaves $K$ large coefficients mostly concentrated on low frequency intact, and reduces $N$ to $N/D$. Thus the number of random measurements with downsampling involved is given by

$$M_{down} = M - c \cdot K \log(D). \tag{4.7}$$

Table 4.1 Effect of adjusting parameters on overall performance (marginal difference)

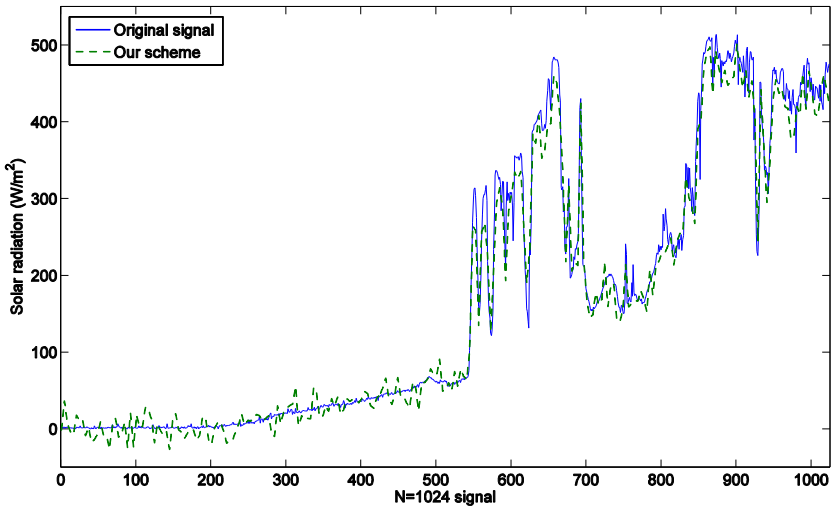
| | **Computation (Energy)** | **Bandwidth** |
|---|---|---|
| **Measurements** | $\downarrow (N)$ | $\downarrow (R_{\max})$ |
| **Quantization** | $\uparrow$ (on initial operation) | $\downarrow\downarrow (M)$ |
| **Downsampling** | $\downarrow\downarrow\downarrow \ ((DM - M_{down}) \cdot \frac{N}{D})$ | $\downarrow\downarrow\downarrow \ (c \cdot K \log(D) R_{\max})$ |

Furthermore, temporal downsampling surprisingly yields better coding performance under the same condition of random measurements and quantization; enough number of random measurements for down-sampled signal and interpolating between nonzero values approximate to original signal more closely. Figure 4.3 depicts approximations of two different signal types with downsampling and quantization.




(a)

Figure 4.3 (a) Ambient temperature data; (b) solar radiation data [22] and their approximations using downsampling by factor of 4 ($N/4=256$) and 16 bits quantization (Daubechies-8 wavelet for $\Psi$).




(b)

Figure 4.3 (Continued).

The advantage of our quality-adjustable sensing scheme can be identified by Pareto optimal frontier (the best achievable points) in Fig. 4.4, which shows the improvement of average MSE by 84.5% when the downsampling is employed. This better coding efficiency from temporal downsampling and linear interpolation can be a remedy for CS that suffers optimality in coding efficiency compared to conventional source coding we have seen in the previous chapters [82]. Moreover, the downsampling reduces both computational complexity and network bandwidth as can be identified in Table 4.1, demonstrating it can be a compelling add-on to CS framework.

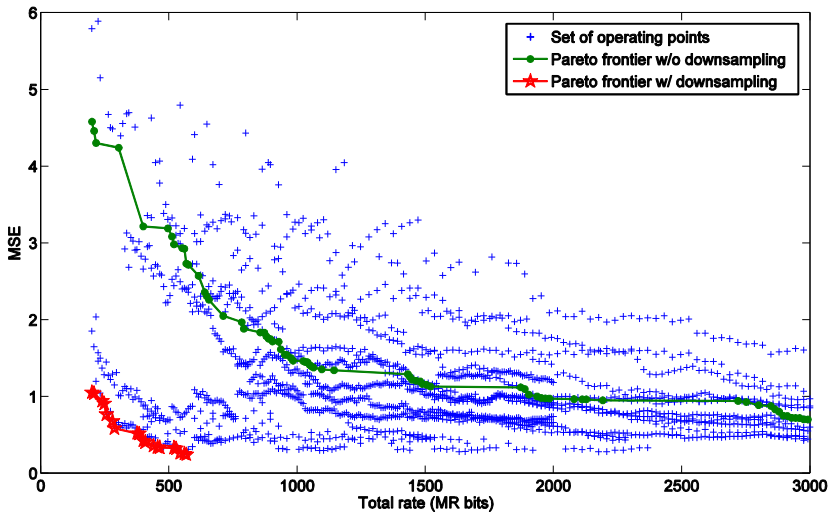



Figure 4.4 Set of operating points for ambient temperature data and their Pareto frontiers using CS with only quantization; and our scheme with both quantization and downsampling where $\mathrm{MSE} = \|\mathbf{x} - \hat{\mathbf{x}}\|_2^2 / N$. (Similar results obtained for the solar radiation data)

### 4.2.2 Optimization with Error Model

The effect of linear interpolation is almost unrelated with (4.6), which was also identified through experiments. Besides, the norm of error in transform coefficients is equivalent to the norm of error in signal since we use orthogonal basis $\mathbf{\Psi}$. Thus total reconstruction error is given by

$$D \sim C_0 \cdot \|\mathbf{s} - \mathbf{s}_K\|_1 / \sqrt{K} + C_1 \cdot \sigma_{source} \sqrt{N} 2^{-R} - \mathrm{Refinement}_{\mathrm{interpolation}}, \qquad (4.8)$$

where $R$ is bits spent on quantizing each entry in $\mathbf{y}$, and $\sigma_{source}$ is the standard

deviation of the original signal

In (4.8), we replaced $\varepsilon$ in (4.6) with $c \cdot \sigma_{source} \sqrt{N} 2^{-R}$, which can be derived from the following theorem.

*Theorem 4.1*: Quantization error $\varepsilon$ can be explained by $c \cdot \sigma_{source} \sqrt{N} 2^{-R}$.

*Proof*: Note that $\varepsilon = \|\mathbf{z}\|_2$. We want to analyze its approximation in statistical sense: especially we want to find its expected value $\mathbf{E}\left[\|\mathbf{z}\|_2\right]$. To this end, we start with the following result [82]:

$$\frac{\mathbf{E}\left[|y_i - \hat{y}_i|^2\right]}{\mathbf{E}\left[|y_i|^2\right]} = c \cdot 2^{-2R}, \tag{4.9}$$

where $y_i$ is each entry in $\mathbf{y}$ and $c$ is a constant.

In the right-hand side of (4.9), the numerator can be denoted as follows:

$$\mathbf{E}\left[|y_i - \hat{y}_i|^2\right] = \mathbf{E}\left[|z_i|^2\right] = \frac{\mathbf{E}\left[\|\mathbf{z}\|_2^2\right]}{M}, \tag{4.10}$$

where $z_i$ is again each entry in $\mathbf{z}$. Similarly, the denominator can be denoted as follows:

$$\mathbf{E}\left[|y_i|^2\right] = \frac{\mathbf{E}\left[\|\mathbf{y}\|_2^2\right]}{M} \approx \frac{\mathbf{E}\left[\|\mathbf{s}\|_2^2\right]}{M}, \tag{4.11}$$

where the last approximate equality is due to the *restricted isometry property* (RIP) [64]. Thus (4.9) can be reformulated as

$$\frac{\mathbf{E}\left[\left\|\mathbf{z}\right\|_2^2\right]}{\mathbf{E}\left[\left\|\mathbf{s}\right\|_2^2\right]} = c \cdot 2^{-2R}. \tag{4.12}$$

Meanwhile, we can find that the following approximate equality holds for sufficiently large *M*:

$$\mathbf{E}\left[\left\|\mathbf{z}\right\|_2\right] \approx \sqrt{\mathbf{E}\left[\left\|\mathbf{z}\right\|_2^2\right]}, \tag{4.13}$$

which can be also validated through numerical experiments, assuming $z_i$ follows various distributions such as Gaussian, uniform, and beta. Moreover, we know that $\mathbf{E}\left[\left\|\mathbf{s}\right\|_2^2\right] = N(\sigma_{source}^2 + \mu_{source}^2)$, where we can assume that source mean is zero without loss of generality. Therefore, combining this with (4.12) and (4.13), we obtain the result $\mathbf{E}\left[\left\|\mathbf{z}\right\|_2\right] \approx c \cdot \sigma_{source}\sqrt{N}\,2^{-R}$. ■

This error model can be utilized to find the optimal operating points with given rate budget $R_{total}$. We here consider the optimization solely in rate-distortion sense. Specifically, if we approximate $\left\|\mathbf{s}-\mathbf{s}_K\right\|_1$ using $\alpha K^{\beta} + \gamma$ ($\alpha$, $\beta$, and $\gamma$ are model constants), we can cast the problem as follows:

$$\min D \qquad \text{subject to} \qquad R_{total} \le MR. \tag{4.14}$$

The resulting optimization is shown in Fig. 4.5, where four actual rate-distortion curves with different quantizations are obtained from a down-sampled signal. We can also identify our error model in (4.8) follows the actual curves well. It is quickly apparent that in Fig. 4.5, the optimal operating points appear in quantization-granularity increasing order, that is, $R$=4, 8, 12, 16.

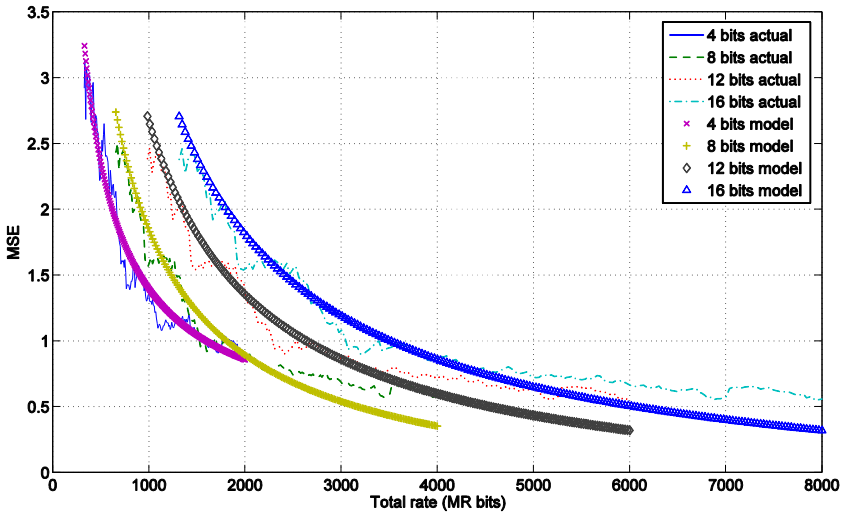



Figure 4.5 Our model following the rate-distortion curves of down-sampled signal ($N$/2=512) with different quantizations.

## 4.3 Low-Complexity Sensing

In CS, a signal is projected onto random sensing basis as in (4.2), which is essentially the computation of inner products, that is, multiplication and summation operations. Since the random sensing basis $\mathbf{\Phi}$ is generally constructed from dense random matrices such as Gaussian and sub-Gaussian distributions, the inner product

computation is the key factor in overall encoding complexity.

Meanwhile, if we can construct $\mathbf{\Phi}$ from sparse matrix, we can dramatically reduce the encoding complexity. This indeed is possible through the use of sparse random matrix while assuring the same performance as dense random matrices [83, 84]. This sparse matrix is binary and very sparse, which apparently reduces the multiplication and summation operations. Combined with the downsampling we already discussed, the sparse random matrix can significantly benefit resource-limited sensing devices.

### 4.3.1 Sparse Random Matrix

Suppose a dense random sensing matrix $\mathbf{\Phi}$ of which virtually every entry is set to non-zero real numbers. This leads to $O(MN)$ multiplication and summation operations; however, this might sometimes be costly to resource-limited sensors without specific CS-supporting architectures [64].

The sparse random matrix turns out to be a solution to this complexity issue. The random sensing matrix $\mathbf{\Phi}$ now has $d$ ones for each column; and all other entries are zeros. (Each column has roughly the same number of ones: slight unbalance in the number does not affect overall results [84].) It was shown that this matrix construction could be deemed an adjacency matrix of an *unbalanced expander graph* [85], which at the same time satisfies RIP-1 [83, 84]:

$$(1-\delta)\left\|\mathbf{s}\right\|_1 \le \left\|\mathbf{\Phi\Psi s}\right\|_1 \le (1+\delta)\left\|\mathbf{s}\right\|_1, \tag{4.15}$$

where $\delta > 0$ should not be close to one [64]. Note that $\mathbf{\Phi}$ constructed using the Gaussian or sub-Gaussian distributions satisfies RIP-2, i.e., the $\ell_2$ norm instead of the $\ell_1$ norm in (4.15). It was also shown that the sparse random matrix satisfying

RIP-1 was essentially as good as dense matrix satisfying RIP-2 [84]. Furthermore, a decoder with the RIP-1 matrix can recover the original signal using linear programming as in the case of RIP-2 matrix, which is given by (4.3).

In this case, the solution $\mathbf{s}^*$ to (4.3) obeys

$$\left\|\mathbf{s}^* - \mathbf{s}\right\|_1 \le C\left\|\mathbf{s} - \mathbf{s}_K\right\|_1 \tag{4.16}$$

for some constant *C*, where $\mathbf{s}_K$ is again the vector $\mathbf{s}$ with all but the largest *K* components set to 0: the quality of recovered signal is as good as that with the *K* most significant pieces of information [64, 83, 84]. We get progressively better results as we compute more measurements *M*, as discussed in Section 4.2.

Because of the selective nature of the sparse random matrix, computational complexity is reduced to $O(dN)$, where $d = O(\log(N/K))$ [84, 86]. This is a considerable saving compared to the general case of $O(MN)$, where $M = O(K\log(N/K))$. In fact, we found that *d* could be decreased as small as 2 without noticeable loss in coding efficiency from our experiments where two different signal types were used. (If *d* = 1, a subset of *K* columns taken from $\mathbf{\Phi}$ can be linearly dependent when *M* < *N* since there can be at most $\begin{pmatrix} M \\ 1 \end{pmatrix}$ unique columns.)

We here introduce the downsampling scheme to further reduce the computational complexity and increase the coding performance at the same time. Figure 4.6 presents our low-complexity CS architecture. The downsampling process takes every *L*th sample and the upsampling process inserts *L* - 1 zeros between samples, where *L* is a downsampling factor. Note that the LPF was not used and the sparse random matrix generation can be synchronized between encoder and decoder

using pseudorandom number generator as in Fig. 4.2.

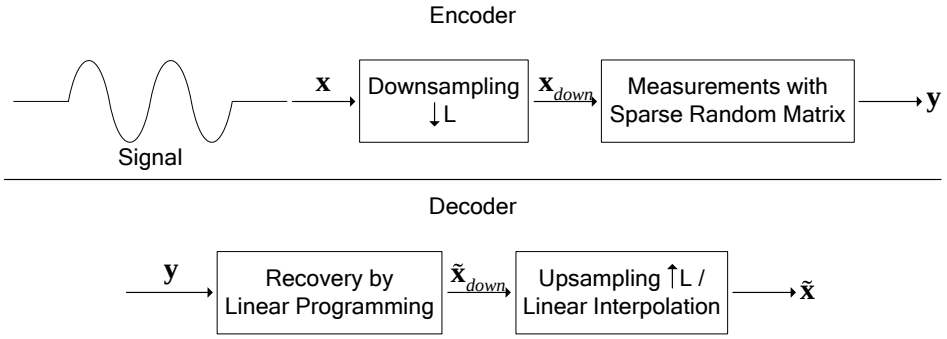



Figure 4.6 Low-complexity CS architecture incorporating downsampling.

The downsampling in Fig. 4.6, combined with upsampling and linear interpolation, again yields better coding performance than general CS framework. The rationale behind the better coding performance with downsampling is illustrated in Fig. 4.7, where original sensor data and two approximations using CS and CS with downsampling are drawn together. We can identify that down-sampled approximation is smoother than general CS approximation, resulting in less distortion. In other words, CS recovery tries to approximate the original signal while incurring distortion bounded by (4.16), which can be mitigated by less sample points recovery and smoothing out fluctuations using linear interpolation.

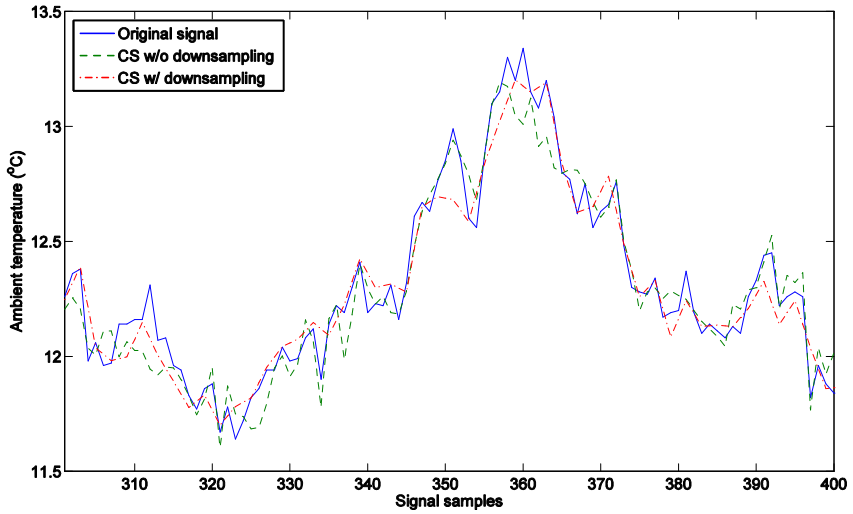



Figure 4.7 Ambient temperature data [22] and their approximations using CS with and without downsampling (Daubechies-8 wavelet used).

The downsampling scheme further reduces the encoding complexity to $O(dN/L)$. We classify overall encoder complexities in Table 4.2.

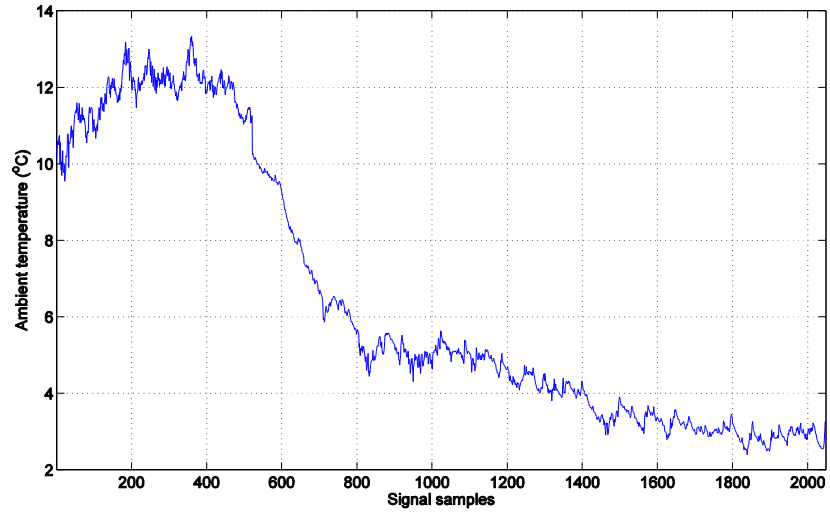
Table 4.2 Overview of encoder complexities

| General CS | Sparse Random Matrix | Our Scheme |
|---|---|---|
| $O(NK\log(N/K))$ | $O(N\log(N/K))$ | $O((N/L)\log(N/K))$ |

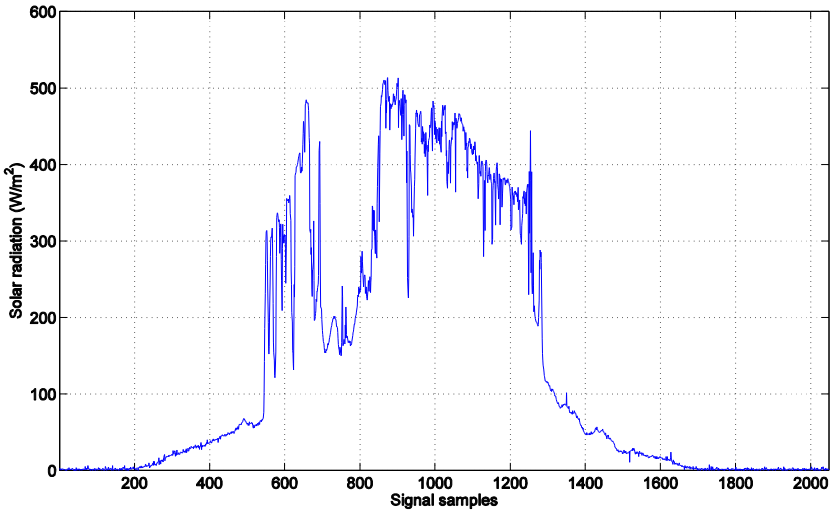
### 4.3.2 Resource Savings

Two different signal types from environmental sensor data set shown in Fig. 4.8 were selected for our experiments. In Fig. 4.9, we show averaged results of our downsampling scheme and the baseline scheme without downsampling. We here consider sum of squared error (SSE) distortion; parameters of the sparse random

matrix are $M = 1024$, $N = 2048$, and $d = 2$. It should be noted that in Fig. 4.9, the performance of baseline scheme is equivalent to general CS framework that uses dense Gaussian matrix for random sensing basis.

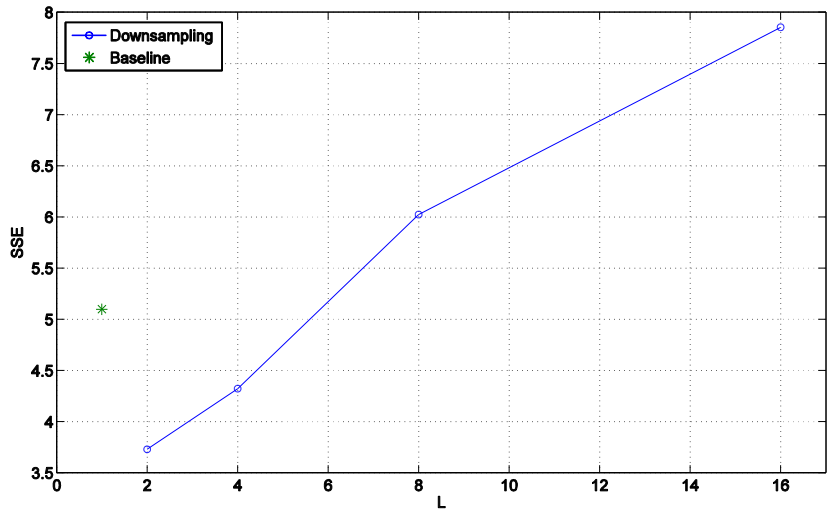



(a)

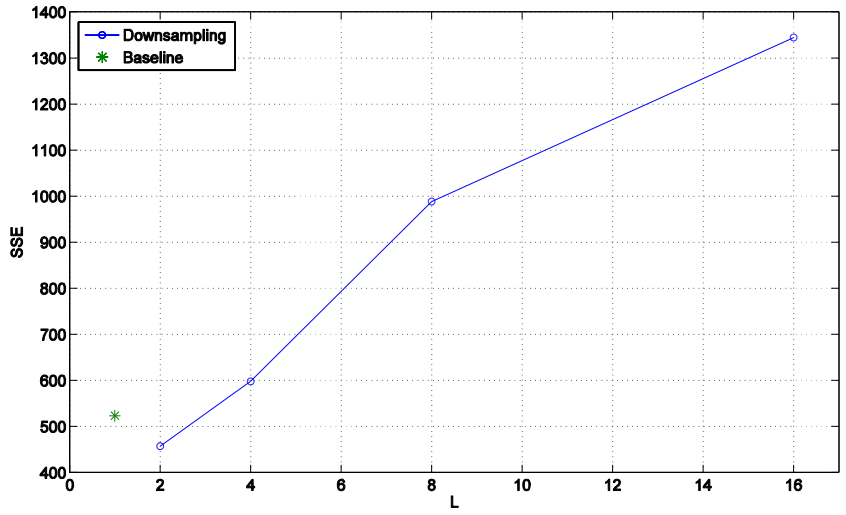



(b)

Figure 4.8 Environmental sensor data of: (a) static; (b) dynamic types.




(a)




(b)

Figure 4.9 SSE comparison with several downsampling factors for: (a) ambient temperature; (b) solar radiation data.

In Fig. 4.9, the extra benefit of our scheme appears at $L = 2$ (and 4); however SSE increases after this point, which means too few sample points and interpolation between them oversimplify approximations. Obviously, if reducing the computational burden is the utmost importance, a sensing device can increase the downsampling factor while sacrificing data quality.

Meanwhile, we obtain these results using $dN/L$ computations as compared to $MN$ computations in general CS framework, which especially is 99.95% of reduction at $L = 4$. (This is equivalent to 75% of reduction as compared to $dN$ computations using solely the sparse random matrix.) Furthermore, we can leverage this benefit to reduce the length of vector $\mathbf{y}$, which corresponds to rate and bandwidth usage of sensors. Therefore, we can find the minimum number of measurements that allows the same SSE as the baseline measurements. The resulting rate savings were 46.29% for ambient temperature data and 32.62% for solar radiation data.

# Chapter 5

# Conclusions

This thesis has focused on the quality adjustability of sensor data, thereby making two major contributions: quality-adjustable sensor data archiving and quality-adjustable sensing. We now summarize what we have discussed thus far and present future research directions.

## 5.1 Summary

We first discussed a new archiving technique for huge volume of sensor data that leverages large spatio-temporal correlation inherent in the collection of sensor data. In particular, we adopted lossy coding scheme in order to take into account the quality adjustability of sensor data. Experimental results showed that our archiving scheme could efficiently handle massive volume of sensor data with remarkable compression ratio under tolerable amount of distortion concerning sensor accuracy, compared to the performance that popular state-of-the-art compression schemes exhibit. We could also identify the importance of utilizing both spatio-temporal correlation.

Furthermore, the quality adjustability was considered in progressive manner at

archival level. A progressive data fidelity control is reasonable because of gradually decreasing access pattern to sensor data collection that user exhibits. To this end, we made our archiving scheme scalable by adding two dimensions for the fidelity control: temporal dimension and quality dimension. Through discarding enhancement layers as time elapses, the quality of sensor data can be progressively adjusted, which is essential in archival of sensor data collection. Our scheme allows an efficient management of storage space through graceful degradation of data fidelity, while retaining key features of sensor data.

In addition, the archiving of massive data generated from various types of sensors should be regulated such that we make the best use of storage space: data fidelity of various sensor data blocks has to be maximized under given storage space. Thus we investigated the optimal management strategy of storage space. In this regard, we derived analytical models that reflected the characteristics of our quality-adjustable archiving scheme. We confirmed our model closely followed actual operation of the archiving scheme, from which the optimal storage configuration problem could be explored. Experiments showed the importance of the optimal storage configuration by comparing overall distortion under the same amount of storage space, where it was demonstrated that any arbitrary strategy could result in a waste of storage space.

Next, we focused on the sensing environment and proposed an efficient sensing scheme that exploits the quality-adjustable nature of sensor data. In order to support resource-constrained sensing devices, we adopted compressive sensing (CS) that is computationally less complex than conventional source coding schemes. A sensing device may need to adapt data quality depending on context to meet the requirements of overall performance. CS is well suited to this case since it naturally provides for quality adjustability that can be utilized by sensors.

We enhanced the basic progressive refinement feature in CS by employing both quantization and temporal downsampling, and provided more rate-distortion operating points than basic CS framework. This enhancement made CS more adaptive to various conditions sensing devices would be subject to. The temporal downsampling, along with linear interpolation at decoder, was shown to significantly improve overall coding efficiency.

Besides, the downsampling approach also reduced encoding complexity of sensing devices. This effect culminated in sheer decrease of complexity when combined with sparse random sensing matrix. As a result, our quality-adjustable sensing scheme can deliver significant gains to a wide variety of resource-constrained sensing techniques.

## 5.2 Future Research Directions

In the quality-adjustable sensor data archiving, we envisaged our archiving scheme working with conventional distributed file systems. Currently this scheme is implemented by modifying popular scalable video encoder [25, 28, 29, 53], which is apparently optimized for image and video data. Although the scalable video coding has made its way to commercial success that proves its encoding and decoding ability in real time [31], it still has complex features that are unsuitable for the purpose of archiving sensor data. In addition, sensor-data-specific properties, which could be exploited in our archiving scheme to further improve the compression efficiency, have not been considered in our scheme.

Now that we have shown feasibility of our archiving scheme with outstanding compression efficiency, we can reduce the complexity and possibly improve the compression efficiency of our archiving scheme, by closely inspecting which

property of sensor data is exploited to increase the compression efficiency; and which property is overlooked to yield suboptimal results. Moreover, in order to ensure its practical operation within distributed file system, we have to also consider the aspect of replica management and user retrieval of archival data blocks. Therefore we need to implement our archiving scheme as the quality management module in distributed file system and validate its practicality.

In addition, we need to further investigate various aspects of the quality-adjustable sensing. We first have to delve into the effect of downsampling and refine our rate-distortion analysis. Then we can extend the analysis of the three tuning parameters (measurements, quantization, and downsampling) to take account of overall system performance such as energy consumption and computational overhead. In this regard, the research aims to embrace real-time scheduling theory to address time constraint issues in quality-adjustable sensing. Specifically, the *imprecise computation technique* can be employed to account for the computational perspective of quality-adjustable sensing [87].

Meanwhile, the downsampling approach has been only considered temporally within individual sensing device. We need to extend this approach to spatially distributed sensing domain as well, which would yield distinctive performance especially in large scale domain. This research naturally connects with more general problem of big data management and analysis. While analyzing trend and pattern of massive data with various types, probing all the data would be impossible or very expensive. Low-complexity CS with downsampling could be a solution to this problem.

# Bibliography


[1] M. Satyanarayanan, “Mobile computing: the next decade,” in *Proceedings of the 1st ACM Workshop on Mobile Cloud Computing & Services: Social Networks and Beyond*, San Francisco, CA, USA, 2010, pp. 1–6.

[2] J. K. Laurila, D. Gatica-Perez, I. Aad, J. Blom, O. Bornet, T.-M.-T. Do, O. Dousse, J. Eberle, and M. Miettinen, “The mobile data challenge: Big data for mobile computing research,” in *Proceedings of Workshop on the Nokia Mobile Data Challenge, in Conjunction with the 10th International Conference on Pervasive Computing*, Newcastle, UK, 2012, pp. 1–8.

[3] A. T. Campbell, S. B. Eisenman, N. D. Lane, E. Miluzzo, and R. A. Peterson, “People-centric urban sensing,” in *Proceedings of the 2nd Annual International Workshop on Wireless Internet*, Boston, MA, USA, 2006, pp. 18.

[4] M. Naphade, G. Banavar, C. Harrison, J. Paraszczak, and R. Morris, “Smarter cities and their innovation challenges,” *Computer*, vol. 44, no. 6, pp. 32–39, 2011.

[5] J. Lee, S. Baik, and C. Lee, “Building an integrated service management platform for ubiquitous cities,” *Computer*, vol. 44, no. 6, pp. 56–63, 2011.

[6] R. J. Honicky, “Understanding and using rendezvous to enhance mobile crowdsourcing applications,” *Computer*, vol. 44, no. 6, pp. 22–28, 2011.

[7] S. Helal, “IT footprinting - groundwork for future smart cities,” *Computer*, vol. 44, no. 6, pp. 30–31, 2011.

[8] F. Gil-Castineira, E. Costa-Montenegro, F. J. Gonzalez-Castano, C. Lopez-Bravo, T. Ojala, and R. Bose, “Experiences inside the ubiquitous Oulu smart city,” *Computer*, vol. 44, no. 6, pp. 48–55, 2011.

[9] J. F. Roddick, E. Hoel, M. J. Egenhofer, D. Papadias, and B. Salzberg, “Spatial, temporal and spatio-temporal databases - hot issues and directions for phd research,” *SIGMOD Record*, vol. 33, no. 2, pp. 126–131, 2004.

[10] P. Ranganathan, “From microprocessors to nanostores: Rethinking data-centric systems,” *Computer*, vol. 44, no. 1, pp. 39–48, 2011.

[11] M. Kitsuregawa and T. Nishida, “Special issue on information explosion,” *New Generation Computing*, vol. 28, no. 3, pp. 207–215, 2010.

[12] L. Sweeney, "Information explosion," *Confidentiality, Disclosure, and Data Access: Theory and Practical Applications for Statistical Agencies*, L. Zayatz, P. Doyle, J. Theeuwes, and J. Lane (eds): Urban Institute, Washington, DC, USA, 2001.

[13] M. Hilbert and P. López, “The world’s technological capacity to store, communicate, and compute information,” *Science*, vol. 332, no. 6025, pp. 60–65, 2011.

[14] J. F. Gantz and C. Chute, "The diverse and exploding digital universe: An updated forecast of worldwide information growth through 2011," IDC, 2008.

[15] M. C. Vuran, Ö. B. Akan, and I. F. Akyildiz, “Spatio-temporal correlation: theory and applications for wireless sensor networks,” *Computer Networks*, vol. 45, no. 3, pp. 245–259, 2004.

[16] T. Srisooksai, K. Keamarungsi, P. Lamsrichan, and K. Araki, “Practical data compression in wireless sensor networks: A survey,” *Journal of Network and Computer Applications*, vol. 35, no. 1, pp. 37–59, 2012.

[17] K. Chakrabarti, M. Garofalakis, R. Rastogi, and K. Shim, “Approximate query processing using wavelets,” *The VLDB Journal*, vol. 10, no. 2-3, pp. 199–223, 2001.

[18] D. Ganesan, B. Greenstein, D. Estrin, J. Heidemann, and R. Govindan, "Multiresolution storage and search in sensor networks," *ACM Transactions on Storage*, vol. 1, no. 3, pp. 277–315, 2005.

[19] D. Ganesan, B. Greenstein, D. Perelyubskiy, D. Estrin, and J. Heidemann, "An evaluation of multi-resolution storage for sensor networks," in *Proceedings of the 1st International Conference on Embedded Networked Sensor Systems*, Los Angeles, CA, USA, 2003, pp. 89–102.

[20] D. Ganesan, D. Estrin, and J. Heidemann, "Dimensions: why do we need a new data handling architecture for sensor networks?," *SIGCOMM Computer Communication Review*, vol. 33, no. 1, pp. 143–148, 2003.

[21] I. Yoon, D. K. Noh, D. Lee, R. Teguh, T. Honma, and H. Shin, "Reliable wildfire monitoring with sparsely deployed wireless sensor networks," in *Proceedings of the 26th International Conference on Advanced Information Networking and Applications*, Fukuoka, Japan, 2012, pp. 460–466.

[22] SensorScope: Sensor Networks for Environmental Monitoring, http://lcav.epfl.ch/op/edit/sensorscope-en

[23] D. Lee, J. Lee, Y. Lee, H. Lee, and H. Shin, "Low-complexity aggregation of collected images with correlated fields of view in wireless video sensor networks," in *Proceedings of IEEE Symposium on Computers and Communications*, Riccione, Italy, 2010, pp. 765–771.

[24] B. G. Haskell, "Entropy measurements for nonadaptive and adaptive, frame-to-frame, linear predictive coding of videotelephone signals," *Bell System Technical Journal*, vol. 54, no. 6, pp. 1155–1174, 1975.

[25] I. E. Richardson, *H.264 and MPEG-4 Video Compression: Video Coding for Next-Generation Multimedia*, John Wiley & Sons, 2003.

[26] B. Rimoldi, "Successive refinement of information: characterization of the achievable rates," *IEEE Transactions on Information Theory*, vol. 40, no. 1, pp. 253–259, 1994.

[27] W. H. R. Equitz and T. M. Cover, “Successive refinement of information,” *IEEE Transactions on Information Theory*, vol. 37, no. 2, pp. 269–275, 1991.

[28] H. Schwarz, D. Marpe, and T. Wiegand, “Overview of the scalable video coding extension of the H.264/AVC standard,” *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 17, no. 9, pp. 1103–1120, 2007.

[29] H.-C. Huang, W.-H. Peng, T. Chiang, and H.-M. Hang, “Advances in the scalable amendment of H.264/AVC,” *IEEE Communications Magazine*, vol. 45, no. 1, pp. 68–76, 2007.

[30] H. Schwarz, D. Marpe, and T. Wiegand, “Overview of the scalable H.264/MPEG4-AVC extension,” in *Proceedings of IEEE International Conference on Image Processing*, Atlanta, GA, USA, 2006, pp. 161–164.

[31] Vidyo, http://www.vidyo.com

[32] K. Sayood, *Introduction to Data Compression*, 3rd ed., Morgan Kaufmann, San Francisco, CA, USA, 2005.

[33] N. Ahmed, T. Natarajan, and K. R. Rao, “Discrete cosine transform,” *IEEE Transactions on Computers*, vol. 100, no. 1, pp. 90–93, 1974.

[34] Z. Xiong, K. Ramchandran, M. T. Orchard, and Y.-Q. Zhang, “A comparative study of DCT- and wavelet-based image coding,” *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 9, no. 5, pp. 692–695, 1999.

[35] G. J. Sullivan and T. Wiegand, “Rate-distortion optimization for video compression,” *IEEE Signal Processing Magazine*, vol. 15, no. 6, pp. 74–90, 1998.

[36] A. Ortega and K. Ramchandran, “Rate-distortion methods for image and video compression,” *IEEE Signal Processing Magazine*, vol. 15, no. 6, pp. 23–50, 1998.

[37] D. Pan, "Efficient data compression techniques for weather data," Alabama Univ. in Huntsville Dept. of Electrical and Computer Engineering, 2011.

[38] S. Goel and T. Imielinski, “Prediction-based monitoring in sensor networks: taking lessons from MPEG,” *SIGCOMM Computer Communication Review*, vol. 31, no. 5, pp. 82–98, 2001.

[39] B. E. Usevitch, “JPEG2000 compatible lossless coding of floating-point data,” *Journal on Image and Video Processing*, vol. 2007, no. 1, pp. 85385:1–85385:8, 2007.

[40] C.-H. Wu and Y.-C. Tseng, “Data compression by temporal and spatial correlations in a body-area sensor network: A case study in pilates motion recognition,” *IEEE Transactions on Mobile Computing*, vol. 10, no. 10, pp. 1459–1472, 2011.

[41] Y.-C. Wang, Y.-Y. Hsieh, and Y.-C. Tseng, “Multiresolution spatial and temporal coding in a wireless sensor network for long-term monitoring applications,” *IEEE Transactions on Computers*, vol. 58, no. 6, pp. 827–838, 2009.

[42] Y.-H. Oh, P. Ning, Y. Liu, and M. K. Reiter, “Authenticated data compression in delay tolerant wireless sensor networks,” in *Proceedings of the 6th International Conference on Networked Sensing Systems*, Pittsburgh, PA, USA, 2009, pp. 1–8.

[43] P. Ratanaworabhan, J. Ke, and M. Burtscher, “Fast lossless compression of scientific floating-point data,” in *Proceedings of Data Compression Conference*, Snowbird, UT, USA, 2006, pp. 133–142.

[44] P. Lindstrom and M. Isenburg, “Fast and efficient compression of floating-point data,” *IEEE Transactions on Visualization and Computer Graphics*, vol. 12, no. 5, pp. 1245–1250, 2006.

[45] F. Ghido, “An efficient algorithm for lossless compression of IEEE float audio,” in *Proceedings of Data Compression Conference*, Snowbird, UT, USA, 2004, pp. 429–438.

[46] M. Dai, D. Loguinov, and H. M. Radha, “Rate-distortion analysis and quality control in scalable internet streaming,” *IEEE Transactions on Multimedia*, vol. 8, no. 6, pp. 1135–1146, 2006.

[47] H.-M. Hang and J.-J. Chen, “Source model for transform video coder and its application. I. Fundamental theory,” *IEEE Transactions on Circuits and Systems for Video Technology,* vol. 7, no. 2, pp. 287–298, 1997.

[48] gzip, http://www.gzip.org

[49] bzip2, http://www.bzip.org

[50] A. Moffat, “Implementing the PPM data compression scheme,” *IEEE Transactions on Communications,* vol. 38, no. 11, pp. 1917–1921, 1990.

[51] 7-Zip, http://www.7-zip.org

[52] Sensirion, http://www.sensirion.com/en/home/

[53] ISO/IEC 14496-10 and ITU-T Recommendation H.264, Coding of Audiovisual Objects - Part 10: Advanced Video Coding, 2003.

[54] D. Lambert, “Zero-inflated Poisson regression, with an application to defects in manufacturing,” *Technometrics,* vol. 34, no. 1, pp. 1–14, 1992.

[55] Y. Shoham and A. Gersho, “Efficient bit allocation for an arbitrary set of quantizers,” *IEEE Transactions on Acoustics, Speech and Signal Processing,* vol. 36, no. 9, pp. 1445–1453, 1988.

[56] H. Everett, “Generalized Lagrange multiplier method for solving problems of optimum allocation of resources,” *Operations Research,* vol. 11, no. 3, pp. 399–417, 1963.

[57] H. Mansour, P. Nasiopoulos, and V. Krishnamurthy, “Rate and distortion modeling of CGS coded scalable video content,” *IEEE Transactions on Multimedia,* vol. 13, no. 2, pp. 165–180, 2011.

[58] M. Cesari, L. Favalli, and M. Folli, “Quality modeling for the medium grain scalability option of H.264/SVC,” in *Proceedings of the 5th International ICST Mobile Multimedia Communications Conference,* London, UK, 2009, pp. 1–6.

[59] H. Mansour, V. Krishnamurthy, and P. Nasiopoulos, “Rate and distortion modeling of medium grain scalable video coding,” in *Proceedings of the 15th IEEE International Conference on Image Processing*, San Diego, CA, USA, 2008, pp. 2564–2567.

[60] D. Lee, Y. Lee, H. Lee, J. Lee, and H. Shin, “Determining efficient bit stream extraction paths in H.264/AVC scalable video coding,” in *Proceedings of the 42nd Asilomar Conference on Signals, Systems and Computers*, Pacific Grove, CA, USA, 2008, pp. 2233–2237.

[61] ISO/IEC 14496-2, Coding of Audiovisual Objects - Part 2: Visual, 2001.

[62] D. Li and X. Sun, *Nonlinear Integer Programming*, Springer, New York, NY, USA, 2006.

[63] G. D. Forney Jr., “The viterbi algorithm,” *Proceedings of the IEEE*, vol. 61, no. 3, pp. 268–278, 1973.

[64] E. J. Candès and M. B. Wakin, “An introduction to compressive sampling,” *IEEE Signal Processing Magazine*, vol. 25, no. 2, pp. 21–30, 2008.

[65] R. G. Baraniuk, “Compressive sensing [lecture notes],” *IEEE Signal Processing Magazine*, vol. 24, no. 4, pp. 118–121, 2007.

[66] D. L. Donoho, “Compressed sensing,” *IEEE Transactions on Information Theory*, vol. 52, no. 4, pp. 1289–1306, 2006.

[67] E. J. Candès and T. Tao, “Near-optimal signal recovery from random projections: Universal encoding strategies?,” *IEEE Transactions on Information Theory*, vol. 52, no. 12, pp. 5406–5425, 2006.

[68] J. Min, J. Kim, and Y. Kwon, "Data compression technique for wireless sensor networks," *Convergence and Hybrid Information Technology*, pp. 9–16: Springer, 2012.

[69] A. K. Maurya and D. Singh, “Median predictor based data compression algorithm for wireless sensor network,” *International Journal of Computer Applications*, vol. 24, no. 6, pp. 15–18, 2011.

[70] F. Marcelloni and M. Vecchio, “Enabling energy-efficient and lossy-aware data compression in wireless sensor networks by multi-objective evolutionary optimization,” *Information Sciences*, vol. 180, no. 10, pp. 1924–1941, 2010.

[71] F. Marcelloni and M. Vecchio, “An efficient lossless compression algorithm for tiny nodes of monitoring wireless sensor networks,” *The Computer Journal*, vol. 52, no. 8, pp. 969–987, 2009.

[72] F. Marcelloni and M. Vecchio, “A simple algorithm for data compression in wireless sensor networks,” *IEEE Communications Letters*, vol. 12, no. 6, pp. 411–413, 2008.

[73] C. M. Sadler and M. Martonosi, “Data compression algorithms for energy-constrained devices in delay tolerant networks,” in *Proceedings of the 4th International Conference on Embedded Networked Sensor Systems*, Boulder, CO, USA, 2006, pp. 265–278.

[74] K. C. Barr and K. Asanović, “Energy-aware lossless data compression,” *ACM Transactions on Computer Systems*, vol. 24, no. 3, pp. 250–291, 2006.

[75] A. Pantelopoulos and N. G. Bourbakis, “A survey on wearable sensor-based systems for health monitoring and prognosis,” *IEEE Transactions on Systems, Man, and Cybernetics, Part C: Applications and Reviews*, vol. 40, no. 1, pp. 1–12, 2010.

[76] D. Takhar, J. N. Laska, M. B. Wakin, M. F. Duarte, D. Baron, S. Sarvotham, K. F. Kelly, and R. G. Baraniuk, “A new compressive imaging camera architecture using optical-domain compression,” in *Proceedings of Computational Imaging IV at SPIE Electronic Imaging*, San Jose, CA, USA, 2006, pp. 43–52.

[77] R. Tibshirani, “Regression shrinkage and selection via the lasso,” *Journal of the Royal Statistical Society. Series B (Methodological)*, pp. 267–288, 1996.

[78] M. F. Duarte, M. B. Wakin, D. Baron, and R. G. Baraniuk, “Universal distributed sensing via random projections,” in *Proceedings of the 5th International Conference on Information Processing in Sensor Networks*, Nashville, TN, USA, 2006, pp. 177–185.

[79] G. Quer, R. Masiero, D. Munaretto, M. Rossi, J. Widmer, and M. Zorzi, “On the interplay between routing and signal representation for compressive sensing in wireless sensor networks,” in *Proceedings of Information Theory and Applications Workshop*, San Diego, CA, USA, 2009, pp. 206–215.

[80] C. Luo, F. Wu, J. Sun, and C. W. Chen, “Compressive data gathering for large-scale wireless sensor networks,” in *Proceedings of the 15th International Conference on Mobile Computing and Networking*, Beijing, China, 2009, pp. 145–156.

[81] W. Bajwa, J. Haupt, A. Sayeed, and R. Nowak, “Compressive wireless sensing,” in *Proceedings of the 5th International Conference on Information Processing in Sensor Networks*, Nashville, TN, USA, 2006, pp. 134–142.

[82] V. K. Goyal, A. K. Fletcher, and S. Rangan, “Compressive sampling and lossy compression,” *IEEE Signal Processing Magazine*, vol. 25, no. 2, pp. 48–56, 2008.

[83] A. Gilbert and P. Indyk, “Sparse recovery using sparse matrices,” *Proceedings of the IEEE*, vol. 98, no. 6, pp. 937–947, 2010.

[84] R. Berinde, A. C. Gilbert, P. Indyk, H. Karloff, and M. J. Strauss, “Combining geometry and combinatorics: A unified approach to sparse signal recovery,” in *Proceedings of the 46th Allerton Conference on Communication, Control, and Computing*, Urbana-Champaign, IL, USA, 2008, pp. 798–805.

[85] M. A. Nielsen, "Introduction to expander graphs," 2005.

[86] R. Berinde and P. Indyk, “Sparse recovery using sparse random matrices,” *preprint*, 2008.

[87] J. W. Liu, W.-K. Shih, K.-J. Lin, R. Bettati, and J.-Y. Chung, “Imprecise computations,” *Proceedings of the IEEE*, vol. 82, no. 1, pp. 83–94, 1994.

# 요약

현재 센서 데이터를 비롯하여 장치들이 생성한 데이터들이 전체 데이터 중 상당한 양을 차지하고 있다. 본 논문에서는 두 가지 중요한 문제인 대량의 센서 데이터의 저장과 효율적인 센싱에 대해 고찰한다. 먼저 우리는 '품질 조절이 가능한 센서 데이터 보관 기법'을 제안하며 이 기법을 사용하면 중요한 특성을 훼손하지 않으면서 전체 센서 데이터 집합을 효율적으로 압축할 수 있다.

본 논문에서 제안하는 보관 기법은 데이터의 노화를 고려, 사용자의 감소하는 접근 경향을 반영하여 데이터의 품질을 점차적으로 조절할 수 있도록 설계하였으며 이는 저장 장치의 공간을 효율적으로 사용하는데 큰 도움을 준다. 다양한 센서 종류들에서 발생하는 데이터를 효과적으로 저장하기 위해 우리는 보관 기법에 대한 모델을 도출해 내고 이를 활용한 최적의 저장 품질 구성 전략에 대해 논의한다. 이는 다양한 종류의 센서 데이터 블록들을 주어진 저장 공간 하에서 최적의 품질로 저장하는 데에 도움을 준다.

다음으로 우리는 효율적인 센싱 기법에 착안하여 '품질 조절이 가능한 센싱 기법'을 제안한다. 이를 위해 낮은 계산 복잡도의 특성을 가지는 '압축 센싱' 방법을 도입한다. 이는 성능에 제약이 있는 센서 장치들에 효과적인 방법이다. 우리는 압축 센싱 방법에서 본질적으로 지원되는 품질 조절을 양자화와 특히 시간 차원의 다운샘플링 기법을 적용하여 개선하였으며, 이전의 방법들에 비해 더 많은 비트량-왜곡 동작 지점을 제공한다. 이러한 방법은 센서 장치들이 자신들이 처한 전체적인 성능을 고려하여 데이터의 품질을 더욱 효율적으로 조절할 수 있도록 한다. 더욱이 제안하는 다운샘플링 기법은 기존의 압축 센싱 방법에 있어서 단점이던 부호화 효율을 향상 시킨다. 그와 동시에 다운샘플링 기법은 희소 확률 행렬과 함께 사용하면 센서 장치의 계산 복잡도를 더욱 낮출 수 있기 때문에, 다양한 종류의 성능 제약하의 센싱 기법들에 유리하다.

**주요어 :** 품질 조절이 가능한 센서 데이터, 데이터 보관, 데이터 노화, 최적 저장 공간 관리, 압축 센싱, 다운샘플링

**학 번 :** 2006-21240