Compaction of RFID Devices using Data Compression*

Yair Wiseman
Computer Science Department
Bar-Ilan University
Ramat-Gan, Israel
wiseman@cs.biu.ac.il

***Abstract*— When a coding system used to represent letters or numbers in transmitting messages has many unused codewords, it will be more than reasonable to compress the messages. EPC which is widely used by many RFID devices has plentiful unused codewords. As a matter of fact, the unused codewords are most by far of the codewords. In this paper we check the suitableness of a commonly used compression algorithm for EPC of RFID devices.**

Keywords — RFID; Electronic Product Code (EPC); Compression; Arithmetic Coding

I. Introduction

Nowadays, numerous products are labeled by two forms of automated data identification: Barcodes and RFID tags [1]. RFID is a newer technology and it contains a higher level of security that is to say data can be encrypted, password protection is enabled and besides it is even set to include a 'kill' feature to remove data permanently. The ability of code manipulation is very important for the objective of this paper – RFID coding compression [2]. *

Electronic Product Code (EPC) is a decided label specifies a unique code for any item everywhere [3]. The format of EPC can distinctly identify items [4]. As a matter of fact, practically, EPC standard's most significant functionality is enabling RFID tags having an applicable code for items utilizing RFID protocol. EPC and RFID actually put forward a method for uniqueness labeling of each single item in the world even in circumstances of plentiful of items.

RFID tags were deployed more than decade ago on several items at Wal-Mart's more than 3,750 USA branches [5]. In addition, some divisions of USA government like Federal Aviation Administration (FAA), Department of Homeland and Department of Defense Security also made the first move to put to use EPC tags more than a decade ago for a number of utilities like warehouse supply chain, tracking armaments, smart borders etc. [6]. As a result the increase prevalent usage of RFID devices, the cost of these devices has been reduced by many manufacturers [7].

In addition to the RFID tags, there are also several other functions for the EPC standard [8]. EPC data can be entirely produced based on analysis of optical data generators like several forms of linear bar codes; however traditional barcodes typically indicate just a producer and a set of products; whereas EPC marks by an unique label each single item anywhere in the world [9]. Customary RFID printers print both the bar code and the RFID circuit on the two sides of a sticker as can be seen in Figure 1.

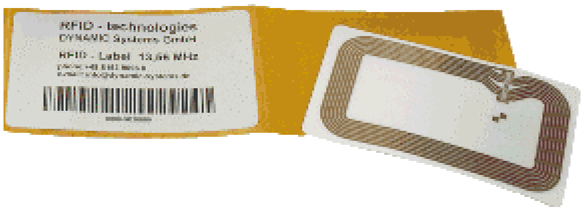

Fig. 1. Sticker of a RFID circuit and a bar code

II. Related Works

The need of power efficiency has led to a verity of studies on the subject of how the information of RFID equipment can be reduced [10]. Low-cost Passive RFID tags generate the electric energy by the interrogating radio waves emitted from the code transmitter and they do not contain an internal power source, so the energy of such passive RFID tags is very limited. When designing RFID tags, the memory and the communication bandwidth should be designed to utilize as few bits as possible because of the limited power resources. For that reason, smaller EPC is necessary so as to adjust the code to the limited power of RFID tags [11].

Adjustment of EPC has been already researched aiming at cryptography [12]. Similarly, we propose adjusting the EPC for RFID tags; however, rather than cryptography, our objective is compressing the EPC data. Actually, both cryptography and compression omit superfluous information, so the techniques are usually similar.

[*] A preliminary version of this paper appeared as Yair Wiseman, "Efficient RFID Devices", Proc. The 42nd Annual Conference of IEEE Industrial Electronics Society (IECON 2016), Firenze (Florence), Italy, pp. 4762-4766, 2016.

The necessity of RFID data compression has been realized as imperative by several previous works. In [13] the authors suggest using a very naive approach - looking for "the change of an object’s location or containment relationships with others" and because the majority of the items are without a change, saving only the changes will use up less memory space and the data will be compressed. Obviously, the current known data compression algorithms are aware of this attribute and they do not just take into account the changes, but the entire entropy of the data, which noticeably offers better compression ratio.

In [14] the authors suggest an attempt to carry out the RFID data compression; yet again they propose to look for changes and if the changes are small, they will ignore them in order to make the compression more efficient. This procedure actually loses some data, but the authors of this paper believe it can be reasonable in some cases. In their conclusion they write "An interesting direction for future work is that of making the compression more adaptive to the analysis performed by the users on the data", but there are known data compression algorithms that already do that and we will make use of them.

III. Compression of EPC codes

Compression techniques are usually divided into two main categories: the first category is compression techniques for data with strong correlation between the bits [15] whereas the second category is compression techniques for data with weak correlation between the bits [16,17,18]. In this paper, we obviously employ a compression technique from the second category which are usually called entropy encodes. The most widespread entropy encoders are Huffman Coding [19] and Arithmetic Coding [20].Arithmetic Coding compression efficiency is always not worse than Huffman Coding compression efficiency; therefore many applications use versions of Arithmetic Coding; however, Huffman Coding has another benefit. Huffman Coding is able to spontaneity recover if an error in the transmission occurs, whereas Arithmetic Coding does not recover from an error at all. So, if the transmission technique is not so reliable and errors occurs from time to time, Huffman Coding ability of recovering can be an important consideration [21].

An optimal compression algorithm assigns an item with a probability P, a codeword with length of $-\log_2 P$ bits [22]. Huffman Coding rounds the codeword lengths in bits to an integer and as a result nearly always, Huffman coding will generate non-optimal set of codewords. Just in a very rare case where the lengths of all the codewords are integer anyway, the rounding will not change the length of the codewords and Huffman Coding will be optimal. Such a distribution is called Dyadic Distribution [23]; however such circumstances are very infrequent; therefore, Huffman Coding is typically less optimal than Arithmetic Coding. Unlike Huffman Coding, Arithmetic Coding does not have need of codewords in an integer length; however, Arithmetic Coding execution time is always more time-consuming than Huffman Coding.

Over the past few years many fast processors have been constructed, so although Arithmetic Coding is fairly slow, these processors can execute the algorithm in a reasonable time even in Real Time systems [24]. Additionally, many researches on reducing the time-consuming difficulty of the Arithmetic Coding have been conducted [25], so the Arithmetic Coding has turned out to be more attractive.

Arithmetic Coding usually assigns codewords in length of fractions of bits [26] whereas Huffman Coding assign codewords in length of only integer number of bits. This feature of the Arithmetic Coding ensues because in Arithmetic Coding several items can share the "ownership" of one bit. Accordingly, codewords of Arithmetic Coding can be in length of non-integer numbers of bits (e.g. in length of 2.9 bits). A sketch of the Arithmetic Coding algorithm is described in this pseudocode:

```
    Let A be a set of items.
    Each item i in A has a probability Pi within [0,1], such that:
    $\sum_{i \in A} P_i = 1$
    Each item is symbolized by the interval:
    $[ \sum_{j < i} P_j , \sum_{j \le i} P_j )$
Repeat until EOF:
        The current interval is divided into sub-intervals according to the items‘ probabilities.
        Replace the current interval by the sub-interval of the items that were read.
    Write into the compressed file the shortest obtainable binary fraction in the current interval.
```

IV. Adapting of Arithmetic Coding for EPC Compression

EPC standard form is 96 bits long consisting of four fields:

1. Header – length, type, structure, version and generation of the EPC.
2. EPC Manager Number – Manufacturer, business or company of a set of items.
3. Object Class – Category of items.
4. Serial Number – Specific item.

The fields and their lengths are summarized in table 1.

Header	EPC Manager Number	Object Class	Serial Number
8 bits	28 bits	24 bits	36 bits

Table 1. EPC fields and lengths

For instance the EPC tag of a specific Herbal Essences Shampoo of Clairol is

The Header field is 0011 0000 for Serialized Global Trade Item Number (SGTIN-96)

The EPC Manager Number field is 0000 0000 0000 0000 1011 0100 0111 for Clairol Company.

The Object Class field is 0000 0000 0000 0010 0110 1111 for Herbal Essences Shampoo.

The Serial Number field is 0000 0000 0000 0010 0111 1000 1011 0110 0000 for a specific bottle of Herbal Essences Shampoo.

Therefore, the entire EPC tag for this specific Herbal Essences Shampoo of Clairol is:

0011 0000 0000 0000 0000 0000 1011 0100 0111 0000 0000 0000 0010 0110 1111 0000 0000 0000 0010 0111 1000 1011 0110 0000

Similarly, the EPC tag for a specific Diet Coca Cola can of 330ml will be:

The Header field is 0011 0000 for Serialized Global Trade Item Number (SGTIN-96)

The EPC Manager Number field is 0000 0000 0000 0000 1010 1000 1001 for Coca Cola Company.

The Object Class field is 0000 0000 0000 0001 1011 1111 for Diet Coca Cola can of 330ml.

The Serial Number field is 0000 0000 0000 0000 0110 1011 0011 1010 0111 for a specific individual 330ml can of Diet Coca Cola.

Therefore, the entire EPC tag for this specific 330ml can of Diet Coca Cola is:

0011 0000 0000 0000 0000 0000 1010 1000 1001 0000 0000 0000 0001 1011 1111 0000 0000 0000 0000 0110 1011 0011 1010 0111.

Provided that EPC reserves 28 bits for the manufacturer/business/company which can create 2^{28} combinations for manufacturer/business/company, certainly most of the potential combinations will be actually useless. The Object Class field also reserves too many bits for its codes. Potentially the Object Class field has 2^{24} objects; however, in reality manufacturers/businesses/companies do not sell so many kinds of items. The Serial Number field is no exception and most of its potential codes are also useless. The Serial Number field has 2^{36} potential items which is also certainly much more even for the most common items; however, it should be noted that the codes of Serial Number field do not have a tendency to be clustered, because most of the manufacturers/businesses/companies begin a new sequence of codes in a new year, a new line or other similar reasons.

The EPC tag codes have some features resulting from the many unused potential codes:

- Most of the code tables for their most part are vacant.
- The codes in use have a tendency to cluster together within several sets of codes.

Quite a few compression algorithms are able to effectively handle data with such attributes e.g. JPEG [27] and MP3 [28]. We have taken the model of these techniques and have tweaked it to EPC codes. JPEG and MP3 split the data into pairs of:

1. Recurring sequences.
2. The rest of the data.

Subsequently of this split, JPEG and MP3 assign each of the pairs a unique codeword (if Huffman Coding is used) or a unique interval (if Arithmetic Coding is used).

In this paper, we adapted this compression tactic but adjusted it to EPC tags. We employed the Arithmetic Coding, so we calculated the probabilities of any potential pair in the EPC data and make tables of intervals for each one of them.

EPC Gen2v2[29] tags consist of four fields:

1. Reserved field
2. PC (Process Control) field including the length of the EPC data.
3. EPC 96 bits
4. User memory

The Gen2v2 information is made up of 16-bit slices. Each such a slice is called "Word". The PC field length is two words. The first word contains the CRC-16 (Cyclic Redundancy Check) error detection code which is not connected to this paper. The first five bits of the second word contains the length of the EPC data. Usually its value is six indicating six words which are 96 bits.

When the data is compressed, there is usually a need to change these five bits in the PC field containing the length of the EPC data, because the size of the compressed data can be different for different information. Only a number of the PC fields are writable, so the change of only these five bits should be executed as a read-modify-write operation [30].

There is also another attribute of the PC field that should be taken into account. The EPC data cannot be accessed while writing to the PC field. As a result, two operations are needed to write EPC data with a changeable length - One operation for writing the EPC data and then another operation for writing the PC field.

V. Results

We have taken a dataset of a retail corporation selling during 2015 with the aim of evaluating our compression algorithm.

We have made arrays of intervals for each segment of the RFID tags. The use of the specific statistics clearly has created arrays of intervals that are suitable only for them. This statistics will probably be changed in the future and this change will make our arrays of intervals outdated. If such a change will take place, the compression will be harmed; however, typically such a statistics is not drastically changed, so the compression efficiency will not be significantly harmed. In any case, a modification in the arrays of intervals can be done from time to time if needed.

Nowadays, no more than several tens of thousands of manufacturers/businesses/companies make use of RFID tags with EPC. EPC standard stipulates a length of 28 bits for EPC Manager Number segment; however, for the current tens of thousands of manufacturers/businesses/companies in fact just 15 bits will be necessitated.

Unfortunately, most of the current tens of thousands of manufacturers/businesses/companies do not appear in the dataset we have employed because this dataset shows the data of just one retail corporation. In fact, we have in the dataset just 978 manufacturers/businesses/companies. We have made the arrays of intervals based on this dataset and we have considered the manufacturers/businesses/companies that do not appear in the dataset as very infrequent. Certainly, if we employed a more comprehensive dataset, the compression and the arrays of intervals could be quite dissimilar.

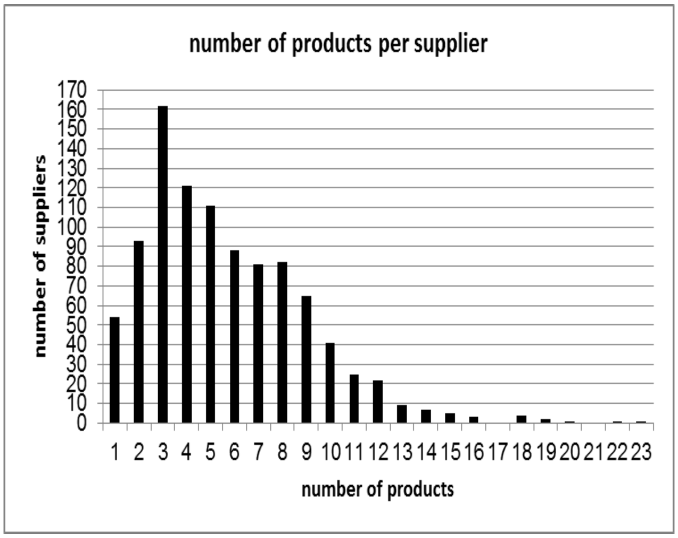

Fig. 2. Number of items per manufacturer/business/company

Typically, most manufacturers/businesses/companies sell just few items to retail corporations. Figure 2 show the number of manufacturers/businesses/companies that sold N items in 2015 to the retail corporation that we have its dataset, where the number of products was range from 1 to 23, because there was no manufacturer/business/company that sold more than 23 types of products to this retail corporation.

EPC standard reserves 24 bits for the Object Class field. This field length grants more than 16 million classes of items for each manufacturers/businesses/companies; however, obviously this quantity is much more than what is really needed for them. The excess length of the Object Class field can be effectively resolved by the Arithmetic Coding.

The Serial Number field is a unique number for each particular item. The length of 36 bits reserved by the EPC standard for this field is also always excesses the field actual requirements. A length of 36 bits stipulates that there can be more than 68 billion of the very items in each class which is at all times much more than needed.

In the retail corporation we checked, there were 127 different types of Clairol shampoo bottles. We counted the number of the bottles of each type. Figure 3 lists the number of Clairol shampoo bottles for each Object Class sold in 2015. Each bar in the chart stands for one type of Clairol shampoo. The type names of the Clairol shampoos are not written within the figure because there will be no room for 127 type names, but actually the type names are unimportant for understanding the EPC compression algorithm.

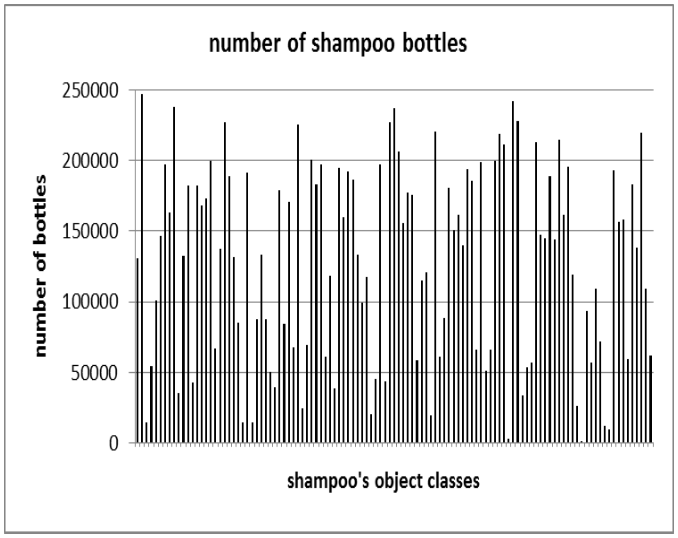

Fig. 3. Number of Clairol shampoo bottles sold one year.

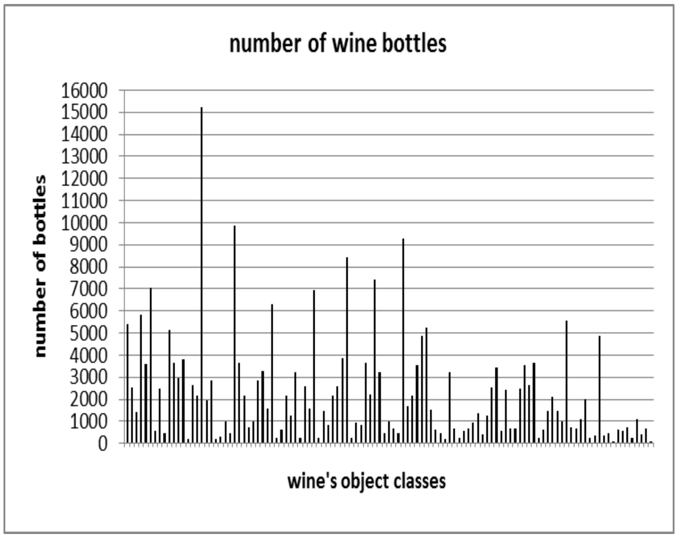

Fig. 4. Number of wine bottles sold in one year.

Yet another example is shown in Figure 4. This example looks at several types of 750ml wine bottles. There were 113 different kinds of wines in this category. The distribution of the wine bottles selling was in some way different from distribution of the shampoo bottles selling. There were few low-priced wines that were the most sold wines, whereas there were many other expensive wines which the retail corporation sold just few of them.

Figure 4 shows the number of wine bottles for each Object Class sold in one year i.e. each bar in the chart represents one type of wine. As in figure 3, the names of the wines are not written within the Figure because of the same reasons.

As can be seen in this figure, most of the wines require only just few bits. Even if we take the most sold wines (15248 items) and we take just the half of the bits suggested by EPC – 18 bits, we will have 262144 combinations which are enough for more than 17 years of selling of this type of wine.

Similarly, the diversity of the codes for the shampoo bottles is also excessive. Even the most sold shampoo requires only 18 bits in order to indicate how many times this shampoo has been sold.

In both of the cases - the shampoo bottles and the wine bottles, more than 68 billion items are far too much as can be seen in Figure 3 and Figure 4.

The Arithmetic Coding is an entropy encoder; consequently, if most of the data is identical, the entropy of the data will be at a low level and the Arithmetic Coding will compress the data very efficiently.

Over all, as was mentioned EPC tag length is 96 bits and it has been averagely compressed to 32.94 bits which are 34.3% of its original length.

This result was obtained as follow:

The header of the EPC has been completely omitted because just SGTIN-96 has been employed in our scheme.

The EPC Manager Number field length has been averagely compressed from 28 bits to 9.41 bits.

The Arithmetic Coding average code length is the sum of the compressed items' entropies. This entropy length can be calculated by this formula:

$$\sum_{i=1}^{n} P_i Log(P_i)$$

Where n is the number of the compressed items and P_i are the items' probabilities. Placing the data of figure 2 in this formula will yield 6.34 or in other words, Object Class field has been compressed from 24 bits to 6.34 bits

Similarly, the Serial Number filed has been calculated according to the same formula, but from the data detailed in figure 3. The result of this calculation was 17.19 bits which means the 36 bits of the Serial Number filed have been compressed to just 17.19 bits. The Serial Number of the wines was compressed even better – from 36 bits to 11.78 bits, because there are many expensive wines that are sold only once in a while.

Figure 5 details the compression ratio of the EPC tag fields. The bar of the Serial Number represents the calculations of the shampoo bottles selling.

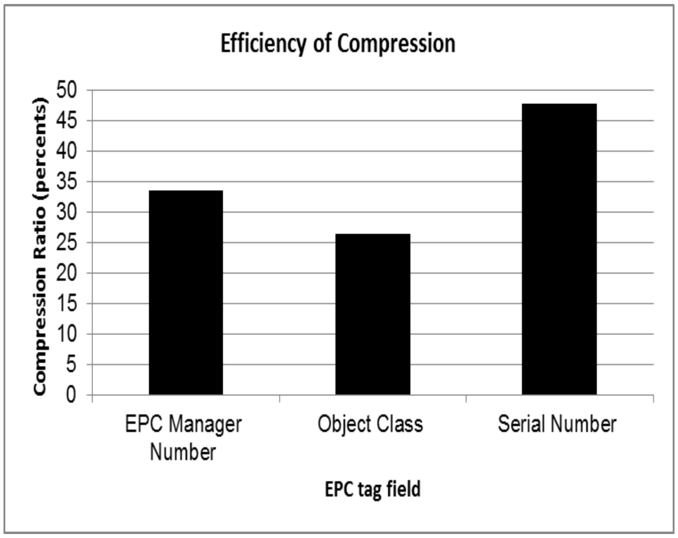

Fig. 5. Compression ratio percent for each of EPC tag fields

VI. Conclusions and Future work

The results of this paper are encouraging. EPC average tag length has been compressed to 34.3% of the original length. Such results can facilitate the use of passive RFID tags in many more utilities [31].

In the literature, two main policies have been proposed with the aim of encountering the restricted communication bandwidth and the restricted electric power of passive RFID devices [32]:

- Decreasing the average EPC tag length.
- Constructing low power energy consuming passive tags.

This paper implements the first policy. The suggested compression procedure efficiently decreases EPC tag lengths. This decrease can improve the propagation of the RFID devices in the market because production of RFID devices that consume less electric power is easier and the devices will be less expensive.

References

[1] G. White, G. Gardiner, G. P. Prabhakar and A. A. Razak. "A comparison of barcoding and RFID technologies in practice", Journal of information, information technology and organizations, Vol 2, pp. 119-132, 2007.

[2] Y. Wiseman, "Compression Scheme for RFID Equipment", Proceedinds of IEEE International Conference on Electro Information Technology (EIT 2016), Grand Forks, North Dakota, USA, pp. 387-392, 2016.

[3] K. Myny, S. Steudel, S. Smout, P. Vicca, F. Furthner, B. van der Putten, A. K. Tripathi, G. H. Gelinck, J. Genoe, W. Dehaene, and P. Heremans, "Organic RFID transponder chip with data rate compatible with electronic product coding." Organic Electronics Vol. 11, no. 7, pp. 1176-1179, 2010.

[4] H. Hada, and J. Mitsugi, "EPC based internet of things architecture", In Proceedins of IEEE International Conference on RFID-Technologies and Applications (RFID-TA), pp. 527-532, 2011.

[5] M. Bustillo, "Wal-Mart radio tags to track clothing", Wall Street Journal, July, 2010.

[6] US. Government Accountability Office, “Information Security: Radio Frequency Identification Technology in the Federal Government”, Report to Congressional Requesters, GAO-05-551, 2005.

[7] A. Sarac, N. Absi and S. Dauzère-Pérès, "A literature review on the impact of RFID technologies on supply chain management", Int'l Journal of Production Economics, Vol. 128(1), pp. 77-95, 2010.

[8] J. Mitsugi, Y. Sato, T. Yokoishi, T. Tashiro, T. Eda, K. Sato and Y. Kishi, "Regional purchase orders dissemination and shipments aggregation of agricultural products with interworking EPC network and EDI system" Proceedings of IEEE International Conference on RFID-Technologies and Applications, pp. 1-6, Johor Bahru, Malaysia, 2013.

[9] A. Milne, "The rise and success of the barcode: Some lessons for financial services", Journal of Banking Regulation, Vol. 14, no. 3, pp. 241-254, 2013.

[10] Y. Wiseman, "Compression scheme for RFID equipment", Proceedings of IEEE International Conference on Electro/Information Technology (EIT 2016), Grand Forks, North Dakota, USA, 2016.

[11] L. Xie, B. Sheng, C. C. Tan, H. Han, Q. Li and D. Chen, "Efficient tag identification in mobile RFID systems", In Proceedings of IEEE International Conference on Computer Communications, pp. 1-9, San Diego, California, 2010.

[12] H. J. Chae, M. Salajegheh, D. J. Yeager, J. R. Smith, and K. Fu, "Maximalist cryptography and computation on the WISP UHF RFID tag", In Wirelessly Powered Sensor Networks and Computational RFID, pp. 175-187, Springer New York, 2013.

[13] Y. Nie, R. Cocci, Z. Cao, Y. Diao and P. Shenoy, "SPIRE: efficient data inference and compression over RFID streams", IEEE Transactions on Knowledge and Data Engineering, Vol. 24, No. 1 , pp. 141-155, 2012.

[14] B. Fazzinga, S. Flesca, F. Furfaro and E. Masciari, "RFID-data compression for supporting aggregate queries", ACM Transactions on Database Systems (TODS), Vol. 38, No. 2, Article no. 11, 2013.

[15] Y. Wiseman, "The Relative Efficiency of LZW and LZSS", Data Science Journal, Vol. 6, pp. 1-6, 2007.

[16] Y. Wiseman, "Enhancement of JPEG Compression for GPS Images", International Journal of Multimedia and Ubiquitous Engineering, Vol. 10, No. 7, pp. 255-264, 2015.

[17] Y. Wiseman and E. Fredj, "Contour Extraction of Compressed JPEG Images", Journal of Graphic Tools, Taylor & Francis publications, Vol. 6, No. 3, pp. 37-43, 2001.

[18] Y. Wiseman, "Alleviation of JPEG Inaccuracy Appearance", International Journal of Multimedia and Ubiquitous Engineering, Vol. 11, No. 3, pp. 133-142, 2016.

[19] D. A. Huffman, "A method for the Construction of Minimum Redundancy Codes", In Proceedings of the Institute of Radio Engineers, Vol. 40, pp. 1098-1101, 1952.

[20] Y. Wiseman, "A Pipeline Chip for Quasi Arithmetic Coding", IEICE Journal - Trans. Fundamentals, Tokyo, Japan, Vol. E84-A No.4, pp. 1034-1041, 2001.

[21] S. T. Klein and Y. Wiseman, "Parallel Huffman Decoding with Applications to JPEG Files", The Computer Journal, Oxford University Press, Swindon, UK, Vol. 46(5), pp. 487-497, 2003.

[22] C. E. Shannon, "A Mathematical Theory of Communication, Bell System", Tech. Journal, Vol. 27, pp. 398-403, 1948.

[23] G. Bocherer and R. Mathar, "Matching Dyadic Distributions to Channels" In proceedins of IEEE Data Compression Conference, Snowbird, Utah, pp. 23-32, 2011.

[24] G. Fuchs, V. Subbaraman and M. Multrus, "Efficient context adaptive entropy coding for real-time applications" In proceedings of IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), pp. 493-496, 2011..

[25] M. Tao, M. Hempel, D. Peng, and H. Sharif. "A survey of energy-efficient compression and communication techniques for multimedia in resource constrained systems", IEEE Communications Surveys & Tutorials, Vol. 15, No. 3, pp. 963-972, 2013.

[26] Y. Wiseman, K. Schwan and P. Widener, "Efficient End to End Data Exchange Using Configurable Compression", Operating Systems Review, Vol. 39, No. 3, pp. 4-23, 2005.

[27] Y. Wiseman, "The still image lossy compression standard – JPEG", Encyclopedia of Information and Science Technology, Third Edition, Vol. 1, Chapter 28, pp. 295-305, 2014.

[28] J. Sterne, "MP3: The meaning of a format", Duke University Press, 2012.

[29] EPCglobal, "EPC™ Radio-Frequency Identity Protocols Generation-2 UHF RFID", Version 2.0.1 Ratified, 2015.

[30] Impinj, "How do I change the length of backscattered EPC data?", https://support.impinj.com/hc/en-us/articles/202756008-How-do-I-change-the-length-of-backscattered-EPC-data- , 2017.

[31] Y. Wiseman, "Efficient RFID Devices", Proceedings of The 42nd Annual Conference of IEEE Industrial Electronics Society (IECON 2016), Firenze (Florence), Italy, 2016.

[32] C. R. Valenta and G. D. Durgin, "Harvesting wireless power: Survey of energy-harvester conversion efficiency in far-field, wireless power transfer systems", IEEE Microwave Magazine Vol. 15(4), pp. 108-120, 2014.