Compaction of RFID Devices Using Data Compression

Yair Wiseman

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. Electronic product code (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.**

Index Terms—**RFID, electronic product code (EPC), compression, arithmetic coding.**

Fig. 1. Sticker of a 13.56 MHz RFID tag circuit and a bar code.

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 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 Security and Department of Defense 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 prevalence of RFID devices has reduced the cost of these devices [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.

II. Related Works

The need of power efficiency has led to a variety 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].

The limit power of RFID tags is of the essence and several approaches of taking care of this limitation have been suggested [12]. Indeed, optimization of power consumption is a significant issue that influences also other uses [13] and RFID limit power is no exception.

Adjustment of EPC has been already researched aiming at cryptography [14]. 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.

The necessity of RFID data compression has been realized as imperative by several previous works. Nie *et al.* [15] 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

Manuscript received April 24, 2017; revised June 5, 2017, September 13, 2017, and October 20, 2017; accepted November 3, 2017. Date of publication November 10, 2017; date of current version January 31, 2018. 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 author is with the Computer Science Department, Bar-Ilan University, Ramat-Gan 52900, Israel (e-mail: wiseman@cs.biu.ac.il).

Digital Object Identifier 10.1109/JRFID.2017.2771348

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.

Fazzinga *et al.* [16] 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.

In [17], a patent of RFID information compression that has been implemented by Symbol Technologies Inc. is described. The inventers of this patent use variable-length code for compression; however, they write in this patent "Additional gains may be had by combining a general purpose compression method (multi-base, LZW, etc.)". This additional gain is exactly the core of our work.

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 [18] whereas the second category is compression techniques for data with weak correlation between the bits [19]–[21]. In this paper, we employ a compression technique from the second category, which are usually called entropy encodes. The most widespread entropy encoders are Huffman Coding [22] and Arithmetic Coding [23]. 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 [24].

An optimal compression algorithm assigns an item with a probability P, a codeword with length of -$\log_2$P bits [25]. 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 [26]; 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.

TABLE I
EPC Fields and Lengths

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

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 [27]. Additionally, many researches on reducing the time-consuming difficulty of the Arithmetic Coding have been conducted [28], so the Arithmetic Coding has turned out to be more attractive.

Arithmetic Coding usually assigns codewords in length of fractions of bits [29] 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:

$$\sum_{i\in A} P_i = 1,$$

where each item is symbolized by the interval:

$$\left[\sum_{j<i} P_j, \sum_{j\le i} P_j\right),$$

and then repeat until EOF:

- Divide the current interval 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 I.

For instance the EPC tag of a specific Herbal Essences Shampoo of Clairol is shown in Table II and the EPC tag of a specific 330ml can of Diet Coca Cola is shown in Table III.

Provided that EPC reserves 28 bits for the manufacturer/business/company which can create 2^{28} combinations for

TABLE II
EPC OF A SPECIFIC HERBAL ESSENCES SHAMPOO OF CLAIROL

Field	Value	Stands for
Header	0011 0000	Serialized Global Trade Item Number (SGTIN-96)
EPC Manager Number	0000 0000 0000 0000 1011 0100 0111	Clairol Company
Object Class	0000 0000 0000 0010 0110 1111	Herbal Essences Shampoo
Serial Number	0000 0000 0000 0010 0111 1000 1011 0110 0000	Specific bottle
Entire EPC tag 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		

TABLE III
EPC OF A SPECIFIC 330ml CAN OF DIET COCA COLA

Field	Value	Stands for
Header	0011 0000	Serialized Global Trade Item Number (SGTIN-96)
EPC Manager Number	0000 0000 0000 0000 1010 1000 1001	Coca Cola Company
Object Class	0000 0000 0000 0001 1011 1111	Diet Coca Cola can of 330ml
Serial Number	0000 0000 0000 0000 0110 1011 0011 1010 0111	Specific bottle
Entire EPC tag 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		

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.

It should be noted that the assumption about these many unused codes is valid in the area of retail. It can be valid for many other areas as well; however, there are areas where RFID is used and the assumption about these unused codes is invalid, e.g., Telemetry [30].

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 [31] and MP3 [32]. 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.

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.

EPC Gen2v2 [33] 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 which mean that our suggested compression must be on a 16-bit boundary. 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 [34].

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

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

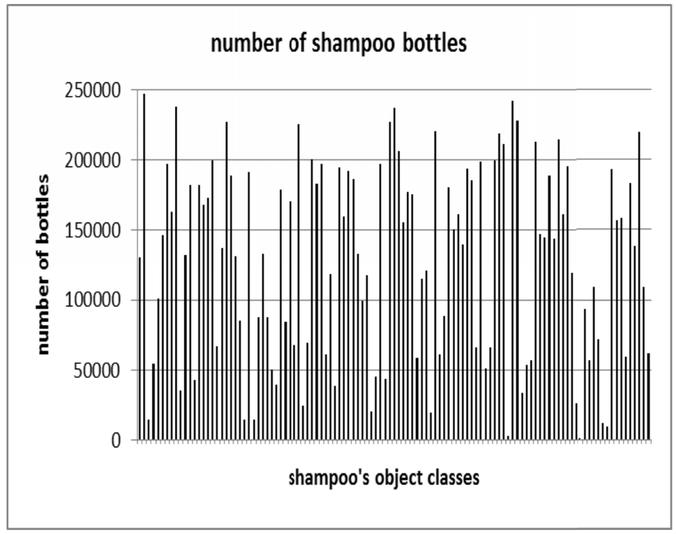


Fig. 3. Number of Clairol shampoo bottles sold one year.

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.

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.

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.

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.

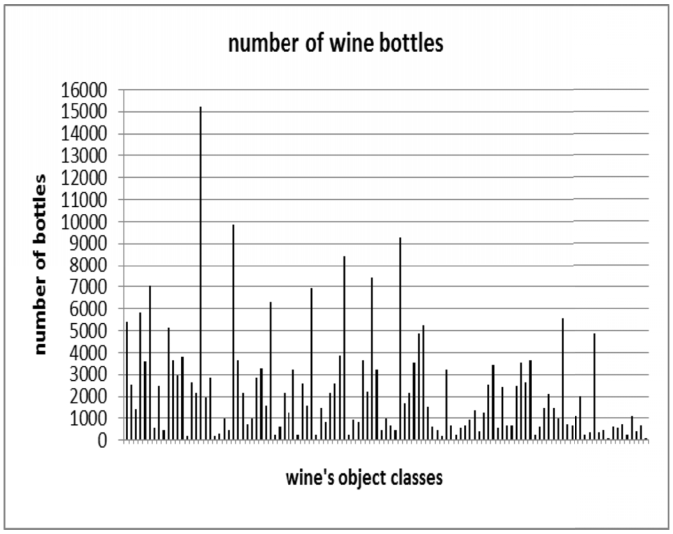


Fig. 4. Number of wine bottles sold in one year.

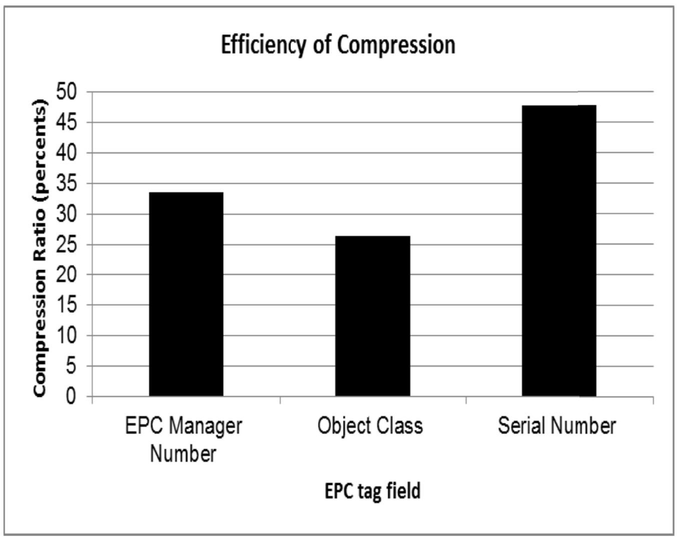


Fig. 5. Compression ratio percent for each of EPC tag fields.

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 [35] 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:

$$\sum_{i=1}^{n} PiLog(Pi)$$

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.

VI. Conclusion

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 [36].

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 [37]:

- 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," *J. Inf. Inf. Technol. Org.*, vol. 20, pp. 119–132, Jan. 2007.

[2] Y. Wiseman, "Compression scheme for RFID equipment," in *Proc. IEEE Int. Conf. Electro Inf. Technol. (EIT)*, Grand Forks, ND, USA, 2016, pp. 0382–0386.

[3] K. Myny *et al.*, "Organic RFID transponder chip with data rate compatible with electronic product coding," *Organic Electron.*, vol. 11, no. 7, pp. 1176–1179, 2010.

[4] H. Hada and J. Mitsugi, "EPC based Internet of Things architecture," in *Proc. IEEE Int. Conf. RFID Technol. Appl. (RFID TA)*, Sitges, Spain, 2011, pp. 527–532.

[5] M. Bustillo, "Wal-Mart radio tags to track clothing," *Wall Street J.*, Jul. 2010. [Online]. Available: https://www.wsj.com/articles/SB10001424052748704421304575383213061198090

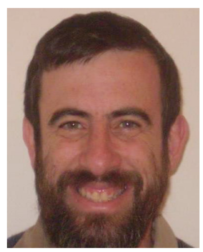
[6] "Information security: Radio frequency identification technology in the federal government," Rep. Congressional Requesters, U.S. Govt. Accountability Office, Washington, DC, USA Tech. Rep. GAO-05-551, 2005.

[7] A. Sarac, N. Absi, and S. Dauzère-Pérès, "A literature review on the impact of RFID technologies on supply chain management," *Int. J. Prod. Econ.*, vol. 128, no. 1, pp. 77–95, 2010.

[8] J. Mitsugi *et al.*, "Regional purchase orders dissemination and shipments aggregation of agricultural products with interworking EPC network and EDI system," in *Proc. IEEE Int. Conf. RFID Technol. Appl.*, Johor Bahru, Malaysia, 2013, pp. 1–6.

[9] A. Milne, "The rise and success of the barcode: Some lessons for financial services," *J. Bank. Regul.*, vol. 14, nos. 3–4, pp. 241–254, 2013.

[10] Y. Wiseman, “Compression scheme for RFID equipment,” in *Proc. IEEE Int. Conf. Electro/Inf. Technol. (EIT)*, Grand Forks, ND, USA, 2016, pp. 0382–0386.
[11] L. Xie *et al.*, “Efficient tag identification in mobile RFID systems,” in *Proc. IEEE Int. Conf. Comput. Commun.*, San Diego, CA, USA, 2010, pp. 1–9.
[12] R. Want, “An introduction to RFID technology,” *IEEE Pervasive Comput.*, vol. 5, no. 1, pp. 25–33, Jan./Mar. 2006.
[13] M. Dreyfuss and Y. Giat, “Optimizing spare battery allocation in an electric vehicle battery swapping system,” in *Proc. 6th Int. Conf. Oper. Res. Enterprise Syst.*, Porto, Portugal, 2017, pp. 38–46.
[14] 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*. New York, NY, USA: Springer, 2013, pp. 175–187.
[15] Y. Nie, R. Cocci, Z. Cao, Y. Diao, and P. Shenoy, “SPIRE: Efficient data inference and compression over RFID streams,” *IEEE Trans. Knowl. Data Eng.*, vol. 24, no. 1, pp. 141–155, Jan. 2012.
[16] B. Fazzinga, S. Flesca, F. Furfaro, and E. Masciari, “RFID-data compression for supporting aggregate queries,” *ACM Trans. Database Syst.*, vol. 38, no. 2, 2013, Art. no. 11.
[17] F. Schuessler and A. Schuessler, “Truncation, compression, and encryption of RFID tag communications,” U.S. Patent 8 188 839, 2012.
[18] Y. Wiseman, “The relative efficiency of data compression by LZW and LZSS,” *Data Sci. J.*, vol. 6, pp. 1–6, Jan. 2007.
[19] Y. Wiseman, “Enhancement of JPEG compression for GPS images,” *Int. J. Multimedia Ubiquitous Eng.*, vol. 10, no. 7, pp. 255–264, 2015.
[20] Y. Wiseman and E. Fredj, “Contour extraction of compressed JPEG images,” *J. Graph. Tools*, vol. 6, no. 3, pp. 37–43, 2001.
[21] Y. Wiseman, “Alleviation of JPEG inaccuracy appearance,” *Int. J. Multimedia Ubiquitous Eng.*, vol. 11, no. 3, pp. 133–142, 2016.
[22] D. A. Huffman, “A method for the construction of minimum-redundancy codes,” *Proc. Inst. Radio Eng.*, vol. 40, no. 9, pp. 1098–1101, Sep. 1952.
[23] Y. Wiseman, “A pipeline chip for quasi arithmetic coding,” *IEICE J. Trans. Fundam.*, vol. E84-A, no. 4, pp. 1034–1041, 2001.
[24] S. T. Klein and Y. Wiseman, “Parallel Huffman decoding with applications to JPEG files,” *Comput. J.*, vol. 46, no. 5, pp. 487–497, 2003.
[25] C. E. Shannon, “A mathematical theory of communication,” *Bell Syst. Tech. J.*, vol. 27, pp. 398–423, Oct. 1948.
[26] G. Bocherer and R. Mathar, “Matching dyadic distributions to channels,” in *Proc. IEEE Data Compression Conf.*, Snowbird, UT, USA, 2011, pp. 23–32.
[27] G. Fuchs, V. Subbaraman, and M. Multrus, “Efficient context adaptive entropy coding for real-time applications,” in *Proc. IEEE Int. Conf. Acoust. Speech Signal Process. (ICASSP)*, Prague, Czechia, 2011, pp. 493–496.
[28] T. Ma, M. Hempel, D. Peng, and H. Sharif, “A survey of energy-efficient compression and communication techniques for multimedia in resource constrained systems,” *IEEE Commun. Surveys Tuts.*, vol. 15, no. 3, pp. 963–972, 3rd Quart., 2013.
[29] Y. Wiseman, K. Schwan, and P. Widener, “Efficient end to end data exchange using configurable compression,” *Oper. Syst. Rev.*, vol. 39, no. 3, pp. 4–23, 2005.
[30] N. J. Seidl and L. Zielinski, “Telemetry system and method,” U.S. Patent 12/338 512, 2008.
[31] Y. Wiseman, “The still image lossy compression standard—JPEG,” *Encyclopedia of Information and Science Technology*, vol. 1, 3rd ed. Hershey, PA, USA: IGI Glob., 2014, ch. 28, pp. 295–305.
[32] J. Sterne, *MP3: The Meaning of a Format*. Durham, NC, USA: Duke Univ. Press, 2012.
[33] *EPC^{TM} Radio-Frequency Identity Protocols Generation-2 UHF RFID, Version 2.0.1 Ratified*, EPCglob., Boston, MA, USA, 2015.
[34] Impinj. (2017). *How Do I Change the Length of Backscattered EPC Data?* [Online]. Available: https://support.impinj.com/hc/en-us/articles/202756008-How-do-I-change-the-length-of-backscattered-EPC-data
[35] (2017). *SGTIN Info*. [Online]. Available: http://www.epc-rfid.info/sgtin
[36] Y. Wiseman, “Efficient RFID devices,” in *Proc. 42nd Annu. Conf. IEEE Ind. Electron. Soc. (IECON)*, Florence, Italy, 2016, pp. 4762–4766.
[37] C. R. Valenta and G. D. Durgin, “Harvesting wireless power: Survey of energy-harvester conversion efficiency in far-field, wireless power transfer systems,” *IEEE Microw. Mag.*, vol. 15, no. 4, pp. 108–120, Jun. 2014.



Yair Wiseman received the M.Sc. *(summa cum laude)* and Ph.D. degrees from Bar-Ilan University, the first Post-Doctoral degree with the Hebrew University of Jerusalem, and the second Post-Doctoral degree with the Georgia Institute of Technology.

His research interests include hardware-software codesign, autonomous vehicle, embedded systems, real-time systems, intelligent transportation systems, computational transportation science, operating systems, process scheduling, memory management, computer clusters, data compression, and JPEG.

Dr. Wiseman has authored two books. He is on the editorial board of several journals, a member of dozens of conference committees and a reviewer of many scholarly journals.

He has been teaching in many institutes including Bar-Ilan University, The Hebrew University of Jerusalem, and Israel Aircraft Industry.

He has been supervising many graduate students.

He has collaborated with other partners and received research grants to run an active laboratory from inter alia Sun Microsystems, Intel, and Polak Foundation.

He is an international expert who has reviewed and evaluated several large projects of the European Union, Israel Science Foundation, Marie Skłodowska-Curie actions in Ireland, Ministry of Education and Science of Kazakhstan. His papers have been published in many venues around the world.