MODELLING THE STORAGE CAPACITY OF 2D PIXEL MOSAICS

SIMONE GÖTTLICH AND THORSTEN SICKENBERGER

INTRODUCTION

This contribution presents an interdisciplinary modelling problem focussed on the improvement of so-called QR-codes. A successful approach is being introduced, worked out by students as well as their teachers during the Modelling Week 2008 in Lambrecht/Germany. The task definition is particularly suitable for the project-related MINT[i] class on AP level. We conclude with a reference to extension possibilities and take a didactical look at the processing of this real-life problem.

MATHEMATICAL MODELLING WITH STUDENTS

Mathematical modelling belongs to the core competencies of today's knowledge-based society. Description and abstraction of real problems by using the mathematical language enables the simulation and optimisation of extensive systems with mathematical tools and IT capabilities. Besides, mathematical modelling with students provides new directions in motivation, knowledge transfer as well as problem solving. Therefore, it should be integrated into the interdisciplinary MINT education.

For 16 years, the Department of Mathematics at the University of Kaiserslautern has been organising mathematical modelling weeks for selected students of 10th – 12th form. Teachers, too, can participate in this event to broaden their knowledge through further education. In total, 40 students and 16 teachers had been engaged for the duration of one week in 8 different and realistic problems from industry, economy, society, sports, IT and physics. Scientific assistants from universities and research centres supported each group by offering advice and help, whenever it was needed.

During the 2008 modelling week, a team consisting of 5 students and 2 teachers from different schools worked on the optimisation of new two-dimensional (2D) bar codes, the so-called *pixel mosaics*. The problem was specially developed for students interested in mathematical modelling as well as in IT with experience in any programming language. Due to the complexity of the problem, it is recommended to implement such a task in interdisciplinary mathematics and IT class on AP level rather than in regular class.

A.N. Other, B.N. Other (eds.), Title of Book, 00–00.

CAN PAPER TALK?

Take a coke bottle, a biscuit tin or a shoe box: nowadays, black and white bar codes are printed on nearly all consumer goods and scanned at the supermarket checkout, replacing manual typing of prices. Bar codes accompany us anywhere, every day. For example, suitcases at airports can be sorted with these codes; libraries have introduced the bar code for their library cards to borrowing books. Therefore, for many students, it is very exciting to understand the functionality of these codes and how much mathematics it may contain [6].

Figure 1. Price tag for fruit with an EAN bar code

The bar code on product codes the 13-digit „European Article Number“ (EAN), classifying each item, and consists of a 12-digit item number as well as a control number. The control number indicates whether the item number was correctly read during the scanning process or if transmission problems have occurred. For the control number, EAN uses a digit, which is calculated by using a specially weighted checksum of the item number. To enable this, the digits of the item number are multiplied alternately with factors 1 and 3 and summed up. The control number completes this sum to the next multiple of 10. Hence, single errors like e.g. the wrong input of digit „8“ as „3“ as well as a pair wise mix up of single digits like „41“ instead of „14“ can easily be detected. The example in Figure 1 shows a bar code for yellow nectarines with the EAN 2404105001722. The weighted checksum of the first 12 digits of EAN is 48. Control number 2 completes this sum to the value 50 (see [2] for a detailed description).

When we refer to codes or coding in this context, it does not mean coding a message or its translation into a code talk, but developing a code or password from a message for its electronic transmission. Errors that may occur are identified and ideally corrected during the transmission process. The bar codes in question only recognise transmission errors, but are unable to localise or even correct it. The development of error-correcting codes which, similar to bar codes, are optically readable but save more data volume and, consequently, information about

correction modes. Black and white bar codes would require too much space for this.

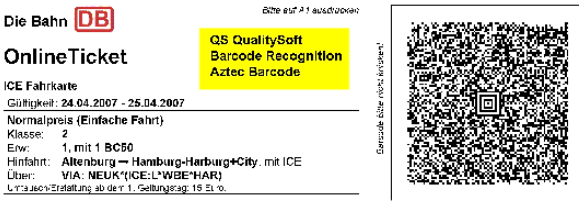


Figure 2. Online-Ticket of Deutsche Bahn (sample)

The new bar code generation consists of quadratic pixel mosaics. They are printed on Deutsche Bahn's online ticket (see Figure 2) or on online stamps named „StampIt" issued by the Deutsche Post. A mosaic like that of black and white pixels can be read through an optic camera or even with a mobile camera. The photographed mosaics are then read out by a specific software programme and finally decoded. This method enables the transmission of a higher data volume compared to those bar codes being used so far. Besides, there are completely new applications emerging from it: A weblink directing to a company or advertising homepage can be translated into a space-saving pixel mosaic and placed as magazine ad. The mosaic is photographed via mobile, the link behind to be downloaded. Personal weblinks of profile pages such as *Facebook* in a pixel mosaic, too, could be coded and printed on the back of a t-shirt. Anyone photographing the mosaic at the next party can view the web profile of the shirt wearer via internet browser of his mobile.

In the course of the modelling week, the students had to investigate the maximum data volume that could be saved in a 2D pixel mosaic and whether it was even possible to extend this storage capacity by developing a new concept. A particularly exciting question for students was raised, whether it would be possible to make paper talk, if a language was coded into a pixel mosaic, photographed with a mobile and played via an integrated loudspeaker.

BACKGROUND INFORMATION: 2D PIXEL-MOSAICS

The following background information regarding construction and functionality of pixel mosaic variations, were compiled by the team via internet research.

2-dimensional bar codes

Another description for pixel mosaics is 2-dimensional bar codes, as they code data horizontally as well as vertically. Since the late 1980's, they have been developed and primarily used in production departments of the vehicle and electronic industry for the clear identification of single elements. Since then, various types of pixel

mosaics have been developed, applied in different areas, competing with each other and a varying distribution. The Deutsche Bahn for example uses the *Aztec Code* for coding the data of their online tickets; the Deutsche Post implemented *Datamatrix* for its online stamp business whereas the *QR-Code* (Quick Response Code) is most widely spread. Numerous QR codes can be found in the public sector which can be recorded with a camera mobile and evaluated inside the mobile phone through special software („Code Reader"). They contain advertising slogans, internet links or save useful information on sights[ii]. This trend will probably soon find its way towards Europe.

Figure 3. The word „Modelling Week" in Aztec Code, Datamatrix and QR-Code

After a quick discussion, the team decided to analyse the QR code more thoroughly, followed by the search for an improvement approach concerning the storage capacity. The single improvement steps as well as the results achieved ought to apply to other mosaic types in a similar way.

The structure of the QR-Code[iii]

The QR-Code consists of black and white pixels and can save a message with up to 7089 signs, depending on the mode that is used. The mode determines the type of the data to be saved. A distinction is drawn between
- *numerical*, i.e. only digits,
- *alphanumerical*, i.e. digits as well as Arabic letters, and
- *special letters*, e.g. Japanese or Chinese letters.

Initially, the data is converted into binary numbers, coded through an error detection process, provided with a special mask, and then saved as pixel mosaic. This process is called coding. The inverse process, that is the read-out of a message from a pixel mosaic, is referred to as decoding. We will now put our main focus on the single steps of decoding.

At first, each character of the message is classified in a binary number. Often, for alphanumerical messages, the extended ASCII Table[iv] is being used, which uses one Byte (which equals 8 Bit) on information for the illustration of a character, thus picturing exact 256 different characters. The binary message produced is now coded through an error correction process. At this, the QR code reverts to the so-called Reed-Solomon-Process (see [11]), which is also used for coding data from a compact disc or DVD. Coding data by using this process enables the localisation of

an error and the recovery of messages even if the QR code can no longer be properly read. The Reed-Solomon algorithm also allows for the regulation of the correction level making it possible to choose between minimum storage space requirement and best possible error correction, depending on the application. The reconstruction of a message can be realised on maximum correction level, even if up to 30% of surface of the pixel mosaic are damaged.

The binary code word which is produced via error correction process is now transmitted into a pixel mosaic. Black pixels represent a 1, whereas white pixels mark a 0. Finally, a so-called mask is covering the code word, consisting of a short and repeating sequence of zeros and ones added pixel- and bitwise to the code word. It means that, for each 1 in the mask, the respective pixel of the code word changes its colour: black pixel turn to white and vice versa. It avoids large and monochrome surfaces which could easily cause problems if data is imported with an optical camera.

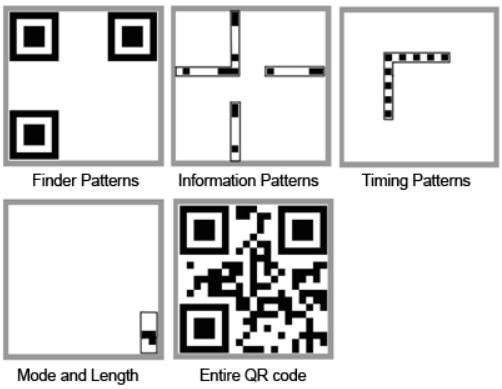


Figure 4. Structure of the QR-Code

Figure 4 shows the detailed structure of the QR code of the message „HALLO“:
– The three *Finder Patterns* serve as alignment of the QR Code, enabling the decoding of a message even if the QR code is photographed in profile, distorted or twisted.
– The *Information Patterns* indicate the mask that is used as well as information about further dimensions.
– The *Timing Patterns* serve as coordinate axis and scale reference. Therefore, black and white pixels are alternating continuously.
– The four pixels in the bottom right corner of the mosaic indicate the *mode* that is used.
– *The message length* is coded directly above the mode, i.e. the number of signs in the binary system.

The code word connects to the message length information, consisting of message and error correction data. The single pixels are populated from bottom to top as well as from right to left. Of course, the actual dimension of a mosaic depends on the length of the message, and is therefore calculated in advance. Pixels that may not be used, are valued as 0. Another pattern of black and white pixel emerges with the use of this mask, identified as "zero information" when it is read out.

MODELLING AND OPTIMISATION: THE QUATTRO-CODE

After completion of a thorough analysis of structure and functionality of the QR code, it was the team's aim to develop their own model of a pixel mosaic with higher storage capacity. The new code was to memorise more information with the same number of pixels, but to also enable the transmission of compressed audio files such as e.g. short voice messages. Following approaches were developed and discussed (see Figure 5):

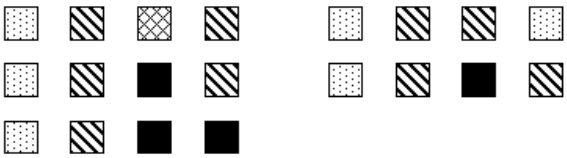
Figure 5. Three different proposals for memory optimisation:Shapes, grey shades or colours.

The first two approaches, however, had soon been dismissed. The students realised, that using different shapes would require a high-definition camera as well as an algorithm for the identification of patterns. Both were graded as additional difficulty and source of error. The readout of pixel mosaics using several shades instead of black and white is very much dependent on the lighting conditions during the shot. The team eventually concentrated on the third approach, as there is no guarantee for ideal and persistent lighting in every situation.

The use of different colours initially raises the question as to how many colour shades should be applied. The team chose four different colours, which are assigned to the binary numbers 00, 01, 10 and 11. In contrast to the QR code which requires 8 pixels, 1 Byte (= 8 Bits) of information was coded into 4 colour pixels. For their work, the students opted for white, red, green and blue, as these colour shades are in a sharp contrast to each other.

Below, we are exemplifying the illustration of the message "HALLO" with coloured pixels, the error correction not being taken into account:

– First, the ASCII numerical code is assigned to every single letter of the message:
  H = 72, A = 65, L = 76, O = 79,
– the ASCII numerical codes are displayed in the binary numerical system (8 Bit):
  72 = 01001000, 65 = 01000001,
  76 = 01001100, 79 = 01001111,
– the binary numbers are now being combined:
  „H A L L O" = 01001000 01000001 01001100 01001100 01001111
– and finally subdivided into pairs and coded in the corresponding „colours":



00 =  , 01 =  , 10 =  and 11 =  to be applied. The pixel mosaic being generated this way was named *Quattro code* by the team.

Error correction

An error correction algorithm followed as a next step: errors in the read data should be identified, localised and corrected. Note that for the QR Code, the Reed-Solomon error correction is being used. This algorithm divides the binary message into units of 8 Bits each and calculates their error correction information. Depending on the correction level, the calculated codeword is getting longer by 3 to 8 Bits compared to the original message since both message and error correction information must be stored. The algorithm is based on the construction of defined polynomials and its evaluation followed by interpolation into predefined nodes. In order to avoid large function values, the calculation of all numbers is carried out in the finite field of integers. This additional difficulty prevented the students from getting deeper into the theory of this procedure.

Instead, they put their focus on the development of their own error correction procedure, based on that of conventional bar codes on simple check sums. For this, the group were pursuing several approaches, of which a suitable one was selected.

A linear system of equations for error correction. The first approach the students developed was not designed for the use of several colours. In fact, they focussed on a message composed of eight bits (e.g. 10011011). The individual bits are assigned to the variables *a, b, c, d, e, f, g* and *h*. Therefore, the variables correspond to the values *a=1, b=0, c=0, … , h=1.*

Next, six control numbers are generated which to be calculated as solutions of six linear equations. The aim is to choose this system of linear equations in such a way that differences in the message between read and calculated control numbers can be uniquely identified. It depends on the smart combination of the variables in the equations; in particular, two variables must not occur exclusively in the same set of equations. Furthermore, each variable is to appear in at least two and a maximum of three equations. For instance, the following equations meet the desired criteria:

$$a+c+e=x_1,$$
$$a+d+h=x_2,$$
$$a+c+f=x_3,$$

$$b + d + g = x_4,$$
$$b + f + h = x_5,$$
$$b + e + g = x_6.$$

The (binary) addition of each left hand side produces a 6-bit control number $(x_1, x_2, x_3, x_4, x_5, x_6)$. Here, the check number is 011010.

Having this tool at hand, the error correction will work for this system as follows: If there is an error in the 8-bit message at one position, the result of the linear system will change in at least two equations; hence the recalculated control number is incorrect, i.e. it differs from at least two positions from the original control number. Thus, conclusions regarding the incorrect bit are possible and corrections can be made. In case the message is 11011011, the calculated control number is 011101, which differs from the original control number at the points x_4, x_5 and x_6. This is due to a read error in the variable b, since only this variable occurs in the appropriate equations. Thus, the correct message is 10011011.

The disadvantage of this system is, that it can merely detect and correct single errors. Several errors in the message or errors while reading the control numbers could cancel each other or lead to more than three different digits in the control number. These errors can neither be recognised nor corrected.

Check sums for the entire message. Another possibility for the generation of check sums is to consider the entire message where all pixels are arranged line by line starting from top left to bottom right. This results in a big square, which should be as small as possible; void space may be filled with zero values. In a second step check sums can be introduced and calculated, e.g., check sums for rows, columns and diagonal elements. Together with these check sums the original message is encoded to the final codeword. The check sums are used to detect possible errors in the transmission of the codeword. This is done by comparison between the stored check sums calculated from the original message and the post-calculated check sums of the read in codeword.

Using this approach the students demonstrated an error detection and error correction of an incorrect codeword "by hand". However it caused them difficulties to design an algorithm whilst implementing this approach, such that this way of error correction was not used in the end. But the group follow up the idea of introducing check sums for error correction and adapt this procedure locally to the Quattro code.

Check sums for 2 x 2 pixels

The ultimately implemented error correction approach was specially developed for pixel mosaics using four different colours. It enables the algorithm to detect and correct a single error within four pixels of information.

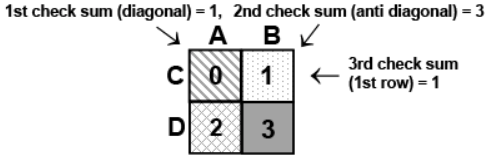


Figure 6: Sketched representation of own error correction procedure.

One byte of information represented by four coloured pixels is arranged as a square (see Figure 6) and the different colours depict a different value in the four number system (= 0, = 1, = 2, = 3). Three additional check sums are used for error detection and correction: the first and second check sums give the sum of the diagonal and anti-diagonal elements, respectively, and the third check sum is the sum of the upper row of that square. The three check sums extend the message to a codeword, they are represented as additional information in coloured pixels and displayed in the pixel mosaic. As an example we consider the letter "H": The ASCII code 01 00 10 00 results in the Quattro code and we get the following square for the calculation of the additional check sums.

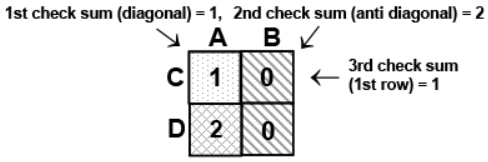
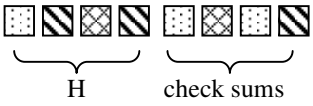


Figure 7. Diagram of the Quattro Code of "H" in a square and the additional calculated check sums.

With these three check sums we can now identify and correct a single error within the four pixels. Let us assume that a pixel of the original square was read in incorrectly, the newly calculated check sums will not match with the stored check sum. We take a closer look at Figure 6 and refer to the columns as A and B and to the rows as C and D, respectively. If for instance the pixel (A/D) is read in incorrectly, the newly calculated check sum of the diagonal does not match with the stored check sum. However the first and the third check sums confirm the correctness of the pixels (A/C), (B/C) and (B/D). Hence the single error can appear in pixel (A/D) only and have to be corrected. A single error among the other pixels can be detected and corrected analogously.

In our error correction we so far have assumed, that at least the three check sums have been read in correctly. However, if one of these is read in incorrectly then this check sum indicates an error in the message also the message itself was read in correctly. To detect errors in the check sums, a fourth check sum is added as the sum of the first three check sums. With view to our example of the letter "H" of the

message "HALLO" the fourth check sum is the sum of the first, second and third check sum: 1 + 2 + 1 = 0 mod 4 (see Figure 7). In this way the codeword of the letter “H” is given by: H = 01 00 10 00 plus four additional check sums = 01 10 01 00:



Note that we had 00 = , 01 = , 10 = and 11 = . It enables to first check the correctness of the check sums first, then locate and correct an error in the message afterwards: If the fourth check sum is correct, we can also assume that the first, second and third check sums, too, are correct and go on with the error localisation and correction of the message itself as described above.

If the fourth check sum does not match with the sum of the first three check sums, the first three check sums will be recalculated from the read in pixels of the message. In a second step the fourth check sum is recalculated and again compared to the stored one. If now the fourth check sum is correct, it can be assumed, that an error occured in one of the first three check sums, but the the message itself was read incorrectly.

If the recalculation of the check sums do not result in a correct message, multiple error occurred in the four pixels of the message and/or the four pixels of the corresponding check sums. In that case the error detection and correction is no longer possible and this byte of information is lost. In that case a question mark is used at the affected place to indicate the failure. In most cases the message can still interpreted by the reader, e.g., if instead of the original message "HALLO" only "H?LLO "is decoded.

The developed error correction for the Quattro code requires additional 8 bits for 8 bits of information (equivalent to 4 pixels). So the required storage space could also used to store the message twice instead of storing the message and data for error correction. In doing so, errors can still be detected and localised, but one criterion is missing to decide, which of the two sources contains the correct message. Here, an approach based on check sums is more efficient.

The layout of the Quattro code

Compared to the QR code the layout of the Quattro code is slightly different. The Finder and Timing Patterns (see Section 3.2) have been merged and will be also used as coordinate system and scaling reference. Beginning in the upper left the pixel mosaic is filled up with information. The first eight pixels represent the length of the message. This is followed from the left to the right by the coded message and the error correction data. Four coloured pixel are needed to store one byte (=8 bit) of information. In total up to 4^8 = 65536 bit of information can be stored in a pixel mosaic.

The Quattro code can store information much more efficient and smallish than the QR code. Figure 8 shows the message "Modelling Week" encoded as QR Code and as Quattro code. It can be seen that the newly developed Quattro code stores the same message in significantly fewer pixels. However, the information for error correction data requires more space compared to the QR-code. This fact was identified by the students and suggested for the next generation the use of the Reed-Solomon method for error correction, which is more efficient.

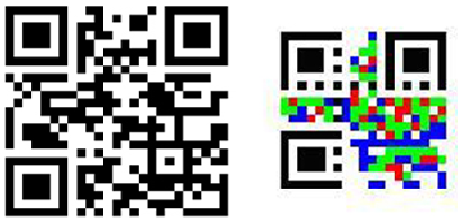


Figure 8: Comparison of a QR code (left) and a black/white print of the coloured Quattro code (right).

The Quattro code uses less space to code a message, but there are two main drawbacks: On the one hand, it becomes necessary to print the pixel mosaic coloured; however, on the other hand, the colours used need to be distinguishable. However, this additional sources of errors can be avoided by suitable printers and camera lenses.

Implementation

The team have developed two computer programs to encode a text message into a Quattro code and to decode a Quattro code into the original message, respectively.

The encoding program transforms a word, a text phrase or even an audio file into a Quattro code which can be displayed on the screen or printed on paper.

Independent of that, the decoding program reads a Quattro code as an image file, analyses that image and decodes the original data. If the original data was a text message, the decode message is displayed on the screen – if the original data was an audio file, the decoded file are played on the loudspeaker.

FURTHER DEVELOPMENTS OF THE QUATTRO CODE

The good project results encouraged the group to discuss and to document potential further developments and optimisation of the Quattro code.

One improvement could be to use more than four different colours. Ideal would be the use of eight different colours, so that 3 bits (instead of 2 bits of information) could be stored in a single coloured pixel. These colours should be chosen out of the basic colour map (e.g. for the Quattro code the RGB colour map was used), such that the difference in the contrast of the colours is as large as possible.

Another potential for optimizing the storage capacity is the error correction method. Here, the relation between the length of the codeword and the length of the message might be improved by integrating the Reed-Solomon error correcting method. Regarding the use of mobile phones and their build-in cameras the decoding algorithm of the Quattro code should be implemented in Java to run on a Java-enabled mobile phone.

Finally, the group discussed the idea of pixel mosaics which change in time. These kind of pixel mosaics could easily be represented on displays consisting of a large number of small LEDs. Instead of making photos one have to use the video function of a mobile phone camera to read in ten or more different configurations of a Quattro code per second. However, time animated LED pixel mosaics can no longer be printed on T-shirts or posters, but could be integrated in an electronic chip or badge.

DIDACTIC REMARKS

From the didactical point of view, the presented modelling task requires competencies in different areas. In the following we analyse and clarify the theoretical framework of this complex problem-solving process.

Modelling process: The individual steps and milestones in mathematical modelling are described by the already known modelling process (see, eg, [1]). Simplifying and structurising, mathematical modelling, working with algorithms and finding a mathematical solution, interpretation and validation of the mathematical results were the main topics for the students' work. All those steps have been considered during the team work. Further modelling tasks in the field of secondary school are found in [3, 5, 7, 8, 10, 12]. But mathematical modelling can also be applied in primary school (see, e.g. [4, 7, 9]).

Interdisciplinary application-oriented education: The presented modelling task requires the study of new mathematical concepts (in particular the calculation using binary numbers) and in-depth experience in computer sciences (such as PHP and Java programming). Hence, it is recommended to implement such a modelling task in interdisciplinary mathematics. For many students the above-indicated relation to everyday life (bar codes printed on nearly all consumer goods, possibility to use pixel mosaics on T-shirts, etc.) may additionally have a positive impact on their motivation.

Knowledge transfer: The task requires a high level of knowledge transfer and an overall good student performance. First, alphanumeric messages must be converted into binary numbers and arranged in pairs. Secondly, these pairs of numbers are translated into colour boxes, furnished with some error correction data and arranged in square form. To enable the error decoding, these algorithm must be bijective, which requires concentrated work, so as to not lose track of the structure of the code.

Knowledge documentation and presentation skills: The team work during the week included the documentation of acquired knowledge by means of a project report and a 25-minute talk on the project results at the last project day. The

presentation is followed by a brief discussion on success and aberration of the team work. The group members are faced with questions of all participants as well as with questions of external scientists in charge. Although, the processes of putting down the project results on paper and creating slides for the final presentation were initially regarded as tedious or even disturbing, but looking back the students are proud to present their own project results in front of a large audience and the positive feedback they were given. Therefore the presentation of project results should be an integral part in the modelling with students.

All participants called the Modelling Week 2008 in Lambrecht a big success, as reflected by the questionnaires: The students were highly motivated to work on their modelling task, because they could work out authentic, complex and open problems which applications in real life. Also, the description of the modelling problem gave no information about the field of mathematics which is required to solve the problem. In particular this fact is a big challenge for the teachers in the team. For a start they are on the same level of knowledge, but after analysing and structuring the problem they can benefit form their mathematical knowledge and point the team to the most useful mathematical tool to solve the problem.

We hope that the interdisciplinary nature of mathematical modelling and problem solving will find integration into day-to-day school. However, the working atmosphere during the Modelling Week in Lambrecht was extremely good and the modelling task was really fun.

REFERENCES/BIBLIOGRAPHY

[1] Blum, Werner (1996). Anwendungsbezüge im Mathematikunterricht - Trends und Perspektiven. *Schriftenreihe Didaktik der Mathematik* 23, 15-38.

[2] Dorfmayr, Anita (2007). Von Strichcode bis ASCII - Codierungstheorie in der Sekundarstufe I. In G. Greefrath, J. Maaß (Eds.) *Materialen für einen realitätsbezogenen Mathematikunterricht 11*, ISTRON-Schriftenreihe, 9-17.

[3] Eck, Christof; Garcke, Harald, & Knabner, Peter (2008). *Mathematische Modellierung.* Springer Lehrbuch, Springer Verlag, Berlin.

[4] Göttlich, Simone (2007). Mathematische Modellierung in der Mittelstufe: Personalausweis für Schildkröten. In: *Beiträge zum Mathematikunterricht 2007,* Verlag Franzbecker, Hildesheim, Berlin, 324-327.

[5] Hamacher, Horst; Korn, Elke; Korn, Ralf, & Schwarze, Silvia (2004). *Mathe und Ökonomie: Neue Ideen für einen projektorientierten Unterricht.* Universum Verlag, Wiesbaden.

[6] Herget, Wilfried (1994). Artikelnummern und Zebrastreifen, Balkencode und Prüfziffern – Mathematik im Alltag. In W. Blum, H.-W. Henn, M. Klika, & J. Maaß (Eds.) *Materialen für einen realitätsbezogenen Mathematikunterricht 1*, ISTRONSchriftenreihe, 69-84.

[7] Hinrichs, Gerd (2008). *Modellierung im Mathematikunterricht: Mathematik Primar- und Sekundarstufe*. Spektrum Verlag, Heidelberg.

[8] Kiehl, Martin (2006). *Mathematisches Modellieren für die Sekundarstufe II.* Cornelsen Verlag, Berlin.

[9] Maaß, Katja (2007). *Praxisbuch Mathematisches Modellieren, Aufgaben für die Sekundarstufe I.* Cornelsen Verlag, Berlin.

[10] Pesch, Hans Josef (2002). *Schlüsseltechnologie Mathematik - Einblicke in aktuelle Anwendungen der Mathematik.* Teubner Verlag, Wiesbaden.

[11] Reed, Irving & Solomon, Gustave (1960). Polynomial codes over certain finite fields. *SIAM Journal on Applied Mathematics* 8(2), 300-304.

[12] Sonar, Thomas (2001). *Angewandte Mathematik, Modellbildung und Informatik.* Vieweg Verlag,Wiesbaden.

AFFILIATIONS

Simone Göttlich
Department of Mathematics,
University of Kaiserslautern
goettlich@mathematik.uni-kl.de

Thorsten Sickenberger
Department of Mathematics,
Heriot-Watt University Edinburgh
t.sickenberger@hw.ac.uk

---

[i] MINT is an abbreviation for the subjects "Mathematics, Computer Science, Natural Science and Technology".

[ii] http://www.youtube.com/watch?v=OxFR6r-Dqk4 (last viewed on 19 December 2008).

[iii] The QR-Code was developed by the Japanese company *Denso Wave*. It is now standardized under ISO 18004.

[iv] ASCII is an abbreviation for "American Standard Code for Information Interchange".