



- (51) **International Patent Classification:**
G06K 7/14 (2006.01)
- (21) **International Application Number:**
PCT/EP2013/059187
- (22) **International Filing Date:**
2 May 2013 (02.05.2013)
- (25) **Filing Language:** English
- (26) **Publication Language:** English
- (71) **Applicant:** SICPA HOLDING SA [CH/CH]; Avenue de Florissant 41, CH-1008 Prilly (CH).
- (72) **Inventors:** BEGARD, Julien; Chemin du Suchet 17B, CH-1350 Orbe (CH). SAITTA, Sandro; Avenue Dapples 19, CH-1006 Lausanne (CH).
- (74) **Agents:** SCHEIBLE, Dominik et al.; Hoffmann · Eitle, Arabellastrasse 4, 81925 Munich (DE).
- (81) **Designated States** (*unless otherwise indicated, for every kind of national protection available*): AE, AG, AL, AM, AO, AT, AU, AZ, BA, BB, BG, BH, BN, BR, BW, BY,

BZ, CA, CH, CL, CN, CO, CR, CU, CZ, DE, DK, DM, DO, DZ, EC, EE, EG, ES, FI, GB, GD, GE, GH, GM, GT, HN, HR, HU, ID, IL, IN, IS, JP, KE, KG, KM, KN, KP, KR, KZ, LA, LC, LK, LR, LS, LT, LU, LY, MA, MD, ME, MG, MK, MN, MW, MX, MY, MZ, NA, NG, NI, NO, NZ, OM, PA, PE, PG, PH, PL, PT, QA, RO, RS, RU, RW, SC, SD, SE, SG, SK, SL, SM, ST, SV, SY, TH, TJ, TM, TN, TR, TT, TZ, UA, UG, US, UZ, VC, VN, ZA, ZM, ZW.

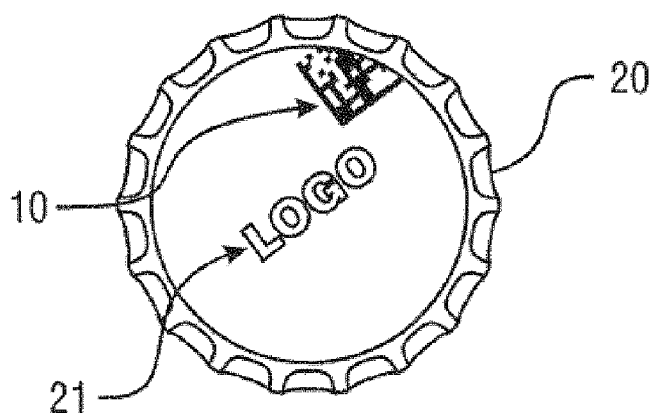
- (84) **Designated States** (*unless otherwise indicated, for every kind of regional protection available*): ARIPO (BW, GH, GM, KE, LR, LS, MW, MZ, NA, RW, SD, SL, SZ, TZ, UG, ZM, ZW), Eurasian (AM, AZ, BY, KG, KZ, RU, TJ, TM), European (AL, AT, BE, BG, CH, CY, CZ, DE, DK, EE, ES, FI, FR, GB, GR, HR, HU, IE, IS, IT, LT, LU, LV, MC, MK, MT, NL, NO, PL, PT, RO, RS, SE, SI, SK, SM, TR), OAPI (BF, BJ, CF, CG, CI, CM, GA, GN, GQ, GW, ML, MR, NE, SN, TD, TG).

Published:

— with international search report (Art. 21(3))

- (54) **Title:** DETECTION OF PRESENCE OF A BAR CODE IN IMAGE DATA

Fig. 2B



- (57) **Abstract:** A method of detecting a bar code in image data, the method comprising the steps of calculating a plurality of characteristic figures from the image data; evaluating a decision tree built for the plurality of characteristic figures; and making a determination of whether or not a bar code is present in the image data based on the evaluating of the decision tree, wherein the plurality of characteristic figures indicate a probability of whether or not a bar code is present in the image data.

DETECTION OF PRESENCE OF A BAR CODE IN IMAGE DATA

Technical Field

The present invention relates to detecting a bar code in image data. Specifically, the present invention relates to a method of detecting a bar code in image data, a method of building a decision tree used for detecting a bar code in image data, and a method of decoding a bar code in image data. The present invention also relates to an apparatus configured to detect and/or decode a bar code, an apparatus configured to build a decision tree, and to corresponding computer programs and computer program products.

Background

Nowadays, one-dimensional and two-dimensional bar codes are ubiquitous in the form of so-called "tags". Specifically, such bar codes can be found on consumer products, electronic and non-electronic devices, machines, vehicles, and also on documents, such as tickets, papers, currency bills etc. Besides the one-dimensional bar code also the two-dimensional bar code has gained in significance during recent years. Whereas the one-dimensional bar code is classically limited to encoding information only in one linear dimension, the two-dimensional bar code, comprising at least first and second type elements arranged in an ordered grid, allows for substantially increasing amount of data amount that can be encoded.

Figures 1A and 1B show schematic views of conventional bar codes. In figure 1A a common one-dimensional bar code is shown that comprises an arrangement of, for example, black lines 1' and white lines 2'. Information is encoded in a one-dimensional bar code by concatenating pre-defined groups of black and white lines 1', 2' of varying thickness. These groups are usually

associated to a specific character or meaning by some kind of industry standard. Fig. 1B shows a common two-dimensional bar code that encodes information by means of arranging, in general terms, first type elements 1" and second type elements 2", such as rectangles, dots, triangles and the like, along two dimensions. In both cases of one-dimensional and two-dimensional bar codes, at least two distinguishable types of elements/lines are used. For example, a white-printed square as a first type element may represent the information 0, whereas a black-printed square as a second type element represents the information 1.

The decoding of a bar code usually begins with taking a photographic image of the bar code affixed on a given item, such as a consumer product or document. This image is then obtained as digital image data defining respective pixel values for the pixels of the image. This digital image data is then subject to image processing by means of a processing unit (e.g., CPU, computer, server, embedded system, ASIC, etc.). Such processing may be divided into various individual steps for eventually decoding the data that is encoded in the one- or two-dimensional bar code.

Various factors determine the quality of the digital input image data - or in short just "image data" - and these factors have naturally also a more or less pronounced impact on the image processing, and, in turn, ultimately to the decoding efficiency and decoding reliability. To begin with, the image of the bar code is usually taken with an arbitrary perspective, meaning that the precise distance, angle, and general orientation of the imaging device (e.g. a CCD camera) relative to the bar code is unknown. Further external factors include illumination conditions, contaminations on the acquisition optics, speed of the bar code carrying item, etc.

Considering consumer goods, such as beverage cans/bottles, product packages, product containers and housings, and the

like, it is clear that the bar code is usually printed or affixed onto the item at a considerable speed in a production line or packaging line. For example, beverage bottles are filled and sealed at rates that exceed hundreds of bottles per minute. Usually, a crown cap can be embossed onto the bottle mouth for sealing the bottle.

In the following, a bar code can be printed/affixed to the cap for identifying, authenticating or tracing the bottle. As an example, the bar code can carry information that allows a determination of whether the beverage bottle is authentic, genuine or legitimate in the sense of its origin, content, selling point, tax deduction, etc. As a further example, a consumer or a government officer can read the bar code on an item for verifying that the product is authentic, or that the product fulfills the regulations required for that the product can be legally distributed at a given location. In this way, consumers can avoid counterfeit or adulterated beverages, and authorities can verify whether the required tax was paid for the respective product in circulation.

In general, it is to be noted that a bar code is not necessarily printed by using, for example, a black dye on a metal cap. Rather, the bar code can be well implemented also by means of color dyes or inks, thermo printing, milling, embossing, grinding, laser etching, acid etching, etc. In general, any type of implementation is possible as long as the elements of the bar code (i.e. the lines, dots, rectangles, etc.) can be distinguished into their respective type in the obtained image data.

Both the variety of printing methods and the above aspects during production/packaging show that also the quality of the bar code as such can vary. Following the example of the beverage bottles, various scenarios can be contemplated as shown in conjunction with Figures 2A to 2F, depicting schematic top views of bottle caps 20. In this way, a bar code 10 can be

printed in a desired fashion (bar code 10 and logo 21 in Fig. 2A), printed only in part (bar code 10 in Fig.s 2B and 2C), at a low contrast (bar code 10' in Fig. 2D), overlapping graphical features of the product (bar code 10 and logo 21 in Fig. 2E), or the bar code can also be completely missing (Fig. 2F).

However, the (sufficient) presence of a bar code on an item may determine its further fate, in the sense that only items with a bar code are authorized for distribution, whereas items with no bar code are rejected as waste. Following the example of beverage bottles, official regulations may require that only bottles with bar code (identification) are authorized for distribution. It would be thus illegal to feed bottles without a bar code into the distribution chain. Given the above-outlined uncertainties in printing/affixing a bar code to an item, it is practice to verify the presence of a bar code on an item before the item can proceed in the distribution chain: For example, a bottle with a bar code is allowed to leave the factory and to be distributed, whereas a bottle without a bar code is either destroyed or re-fed into the production line.

As already mentioned, consumer products are manufactured, packed etc. at considerable speed. Therefore, also the verification process of determining whether an individual product was provided with a bar code may need to meet the speed requirements set by the production/packing line. In other words, the processes associated with the verification of a product bar code should interfere with the processes of production and/or packaging to a least possible extend. With the high speed of production/packaging as such, also the speed of the image processing for detecting a presence of a bar code needs to be high enough.

Although conventional image processing techniques for decoding bar codes in image data may be adapted for fast result output, these techniques then suffer from a lowered reliability. In a way, these techniques may provide determination results at a

high speed but these results contain too many false positive hits and/or false negative hits. Depending on the definition, a false positive hit occurs when image data is found to comprise a bar code whilst this is actually not the case, i.e. the imaged bottle cap does not have a bar code. Similarly, a false negative hit occurs when the bottle cap is found not to have a bar code although it has one. Both these cases can cause significant problems in that a high number of false positives cause an increased number of items to appear in the field that cannot be authenticated/traced (no bar code), and in that a high number of false negatives results in an increased waste and/or rework at the point of production/packaging.

There are several known methods to locate barcodes. One of these methods is the localization part of the reference decode algorithm for data matrix (ISO/IEC 16022:2006, Information technology - Automatic identification and data capture techniques - Data Matrix bar code symbology specification). This algorithm uses the finder pattern (i.e. two solid adjacent borders in an L shape) to locate the data matrix. However the performance of this reference algorithm can be low in case of low quality image data.

There is therefore a need for improvements in detecting a bar code in image data. Specifically, it is desirable to have image processing at hand that is both fast and reliable. In other words, the involved methods, apparatuses, computer programs and related products should allow for speed compatibility with the production/packaging equipment, a satisfactory share of authenticable and traceable items in the field, and an acceptably low cost involved with waste and/or re-working. Preferably, the solution is fast enough in order not to decrease the performances of the accompanying decoding in terms of decoding time, and the solution should also be easy to implement within the decoding.

Summary

The above-mentioned objects and problems are solved by the subject matter of the independent claims. Further preferred embodiments are defined in the dependent claims.

According to one aspect of the present invention, a method of detecting a bar code in image data is provided, the method comprising the steps of calculating a plurality of characteristic figures from the image data; evaluating a decision tree built for the plurality of characteristic figures; and making a determination of whether or not a bar code is present in the image data based on the evaluating of the decision tree, wherein the plurality of characteristic figures indicate a probability of whether or not a bar code is present in the image data.

According to one aspect of the present invention, a method of building a decision tree used for identifying a bar code in image data is provided, the method comprising the steps of: separating template image data into positive-hit image data with a bar code being present and negative-hit image data with a bar code being absent; calculating a plurality of characteristic figures for the positive-hit image data and the negative-hit image data; and building the decision tree based on the calculated characteristic figures for the positive-hit image data and the calculated characteristic figures for the negative-hit image data, wherein the plurality of characteristic figures indicate a probability of whether or not a bar code is present in the positive-hit image data and the negative-hit image data.

According to one aspect of the present invention, a method of decoding a bar code in image data is provided, comprising the steps of the method of detecting a bar code in image data according to a disclosed embodiment, wherein processing of

decoding the bar code is aborted upon making a determining of no bar code being present in the image data.

According to further aspects, a related device, a related computer program, and a related computer program product are provided.

Brief Description of the Drawings

Embodiments of the present invention, which are presented for better understanding the inventive concepts and which are not the same as limiting the invention, will now be described with reference to the figures in which:

Figures 1A and 1B show schematic views of exemplary conventional bar codes;

Figures 2A to 2F show schematic views of various scenarios of a bar code being printed on an exemplary item;

Figure 3 shows a schematic view of a system architecture according to an embodiment of the present invention;

Figures 4A and 4B show flow charts of method embodiments of the present invention;

Figures 5A to 5C show schematic views of a decision tree during building according to an embodiment of the present invention;

Figures 6A and 6B show flow charts of further method embodiments of the present invention;

Figures 7A to 7C show schematic views of exemplary data graphs;

Figures 8A and 8B show schematic views of further scenarios of a bar code being printed on an exemplary item;

and

Figures 9A to 9C show schematic views of apparatus and device embodiments of the present invention.

Detailed Description

Generally, image processing for decoding a bar code includes various sub-processes. Such image processing usually starts from a raw image (image data) of the bar code, possibly attached to an item or document. The image is obtained as said image data from the bar code by means of, for example, conventional digital cameras, video cameras, CCD cameras, scanners, and the like.

A possible image processing is explained for the example of the two-dimensional bar code shown in Figure 1B. There are already various standards available that define the details and the decoding and encoding of two-dimensional bar codes (e.g. ISO IEC 16022 standard). One example is one according to the GS1 (Trademark) DataMatrix ECC 200 standard (GS1 being an international association providing standards for two-dimensional barcodes). This two-dimensional bar code comprises two main areas, namely the so-called "finder pattern" and the data, which carries the actual payload data of the bar code and comprises first and second type elements 1" and 2". The finder pattern are the elements framing the bar code, and is, in turn, divided into the so-called "L finder pattern" and the so-called "clock track". In the example of Fig. 1B, the clock track are the

upper most and right most elements of alternating type and the L finder pattern consists of the elements at respective opposite locations (bottom and left).

The input data is firstly subject to a thresholding operation for the purpose of binarization of the input image comprising only first and second type pixels (e.g. black and white). The purpose of thresholding is to provide some means to distinguish in digital image data objects of possible interest from background. Optionally, data is rescaled for downscaling the image by a given factor. This may substantially speed up the approximate localization of the bar code in the digital image data, since, as long as no important features are lost, a reduction of the data may substantially contribute to accelerate any following sub-process.

Subsequently, the image data is subject to dilatation and hole filling, during which adjacent pixels of a given value are merged and filled until a number of blocks in the image is constant. At this stage, the image data may, nevertheless, comprise more than one form that can be interpreted as a potential candidate of the two-dimensional bar code. This may involve searching and enumerating of all forms in the image and erasing all the forms with a low area so that small objects and noise are removed. It may then be considered to keep the biggest form which is then supposed to be the best candidate for the two-dimensional bar code within the image data. Then, erosion is carried out in order to retrieve the original contours of the image. For this purpose it can be considered to apply the same number of the erosion steps as the number of steps that have been applied using the preceding dilatation.

In said eroded image data the borders of the two-dimensional bar code (candidate) are highlighted. This may involve the application of a filter on the image data that detects pixel intensity variation. In general, a conventional edge detection algorithm may be employed. Then, so-called border equations are

calculated by, for example, detecting parametric objects (lines, circles, etc., using the so-called Hough transform). Subsequently, the corners can be extracted. Optionally, since the Hough lines intersections are only approximation corners of the two-dimensional bar code, better and more accurate coordinates may have to be computed by, for example, a line-by-line or column-by-column scan in a small region around the approximate corners until a black pixel is reached.

The process can now identify the parameters of the two-dimensional bar code. This may specifically apply to the identification of the L finder pattern and the clock track. This may involve the counting of the number of black pixels along each segment so as to obtain a so-called score value. The scanned segment that has obtained the highest score value is identified as the longest segment of the L finder pattern, the scanned segment that has obtained the second highest score value and that is substantially perpendicular to the identified longest segment of the L-shape solid line is identified as being the second segment of the pattern. The two remaining segments form the clock track.

If optional rescaling applied previously, then in a subsequent step the two-dimensional bar code and the initial grid location are recovered. This may involve the scanning of the clock line borders and counting the number of transitions so as to determine the size of the grid. Then sampling and decoding is performed taking into account that the initial grid location may not be perfect in terms of sampling quality. The result is then the data being encoded in the bar code. Although the one-dimensional bar code of Figure 1A differs quite substantially from the bar code of Figure 1B, the involved image processing may share at least some of the above-outlined aspects. It is to be noted, however, that the one-dimension bar code can also be decoded by laser scanners.

Decoding processes, such as the above, can thus read well-formed bar codes (e.g. a DataMatrix) in the sense of the bar code appearing in the image data in a fashion that meets some given quality characteristics such as contrast, size, etc. However, the involved algorithms may encounter difficulties on bad quality image data in which the bar code does not clearly appear (e.g. situation in Figure 2B to 2E). As mentioned above, inaccurate decoding may result in that marked items (items with a bar code) are ejected from the production line or inadvertently released to the distribution chain. Therefore, besides the above decoding as such, also the mere information on whether a bar code is present or absent can be of importance.

In this way, detecting the presence/absence of a bar code in image data may be either performed as part of the decoding or also as a "stand alone" application. In the former case, decoding can be rendered more efficient, since a determining that no bar code is present can be used to abort any further image processing. The latter stand-alone application can be used independent from printing and/or decoding a bar code. In an embodiment, the detection of a presence/absence can be employed to verifying the correct product marking at final stages of production/packaging before the release of the products. In this way, it can be ensured that only items with a bar code (marked product) are released to the distribution chain, whereas unmarked products are dismissed or reworked. Furthermore, a further implementation may foresee to (at least) attest the presence of a bar code when it is not decodable.

According to the embodiments of the present invention a decision tree is employed. In general terms, a decision tree is a sequence of one or more decisions to be evaluated based on input figures. A graphical representation of a decision tree is shown with an exemplary configuration in conjunction with Figures 5C. This decision tree involves three decisions comparing three input figures P1, P2, and P3 to respective

reference values. Each decision either yields a result (e.g. C=1 or C=2) or leads to a next decision considering other input and/or other reference figures. In the meaning of the present disclosure, a decision tree thus defines any sequence of decisions evaluated for a given input relative to given reference figures so as to obtain a decision result. Naturally, an implementation of a decision tree can be in form of rules or code which, in turn, can be defined using a command language, programming language, query language, etc.

According to an embodiment of the present invention, the decision tree is built based on collected data. In this way, "first" input data may serve to build the decision tree, which is then applied to evaluate "second" input data. Said first input data can be accompanied with known results, so that the decision tree can be learned from the first input data. As regards actual implementations in the field of product marking, the diversity of possible background images (brand name, logo, substrate, etc.) on a production line can be considered when compiling the first input data. Also, input data with a known low decodability rate can be of use for building a decision tree.

In other words, a decision tree is a data mining algorithm for constructing a prediction model from data. The final model is obtained by recursively partitioning the data space and fitting a simple prediction model within each partition. As a result, the partitioning can be represented graphically as a decision tree (cf. Figures 5A to 5C). Decision trees can use a measured prediction error as a misclassification cost. Once a model (tree) is obtained, corresponding rules can be extracted. They facilitate model understanding and allow fast implementation.

Figure 3 shows a schematic view of a system architecture according to an embodiment of the present invention. The architecture involves an application 303 coupled to a printer 301 and a camera 302. The application 303 is coupled to a

decoding system 304, which, in turn is coupled to a bar code detection algorithm.

Figure 4A shows a flow chart of a method embodiment of the present invention. This embodiment is a method of building a decision tree used for identifying a bar code in image data, and involves a step 411 of separating template image data into positive-hit image data with a bar code being present and negative-hit image data with a bar code being absent, a step 412 of calculating a plurality of characteristic figures for the positive-hit image data and the negative-hit image data, and a step 413 of building the decision tree based on the calculated characteristic figures for the positive-hit image data and the calculated characteristic figures for the negative-hit image data. The characteristic figures indicate a probability of whether or not a bar code is present in the positive-hit image data and the negative-hit image data.

The template image data can be a collection of images that could not be decoded in a previous image processing. In a way, the template image data can be a repository of images, which - preferably - includes images from different products, variants, and/or taken under varying imaging conditions. This may particularly serve to increase data variability. In other words, the higher the diversity of collected images, the better the decision tree will perform on unseen images. Images can thus be collected with varying parameters, such as production/packaging line, location, product, time, date, season, etc. Since it may be difficult to collect a representative set of all products available in a given distribution chain (e.g. country), a manual selection can be made.

The separation into positive-hit image data and negative-hit image data can be effected by, for example, high-accuracy image processing. Such processing may be able to detect a bar code with satisfying accuracy, but may not be suitable for online

(live) decoding due to the increased demand on processing resources.

Also, an expert can visually separate the images into the two classes (absence/presence of a bar code). In this alternative, collected images are visually classified as containing a bar code (cf. Figures 2A, 2B, 2D, and 2E) or not containing a bar code (Figures 2C and 2F). It is noted, however, that the separation - either automated or manual - needs only to be effected for building the decision tree. Once the model (tree) is ready, no additional visual inspection is needed.

Possible scenarios of marked products have already been mentioned in conjunction with Figures 2A to 2F. More specifically, Figure 2E shows a situation in which the bar code printing overlaps with a logo 21 on the cap 20. This situation may be frequent since most brands have their logo on bottle caps. In general, if a bar code is present on the cap, the bar code ink can produce saturated values in the image data. In such a situation the gradient magnitude Otsu threshold parameter (P3) based on the image gradient can be employed. When a logo is overlapping a bar code, this figure is not significantly influenced (as shown, for example, in conjunction with Figures 7B and 7C). Since this parameter can be given as an input characteristic figure to the decision tree, the bar code overlap with the brand logo can be managed. The decision tree is not influenced by the slight variation of this parameter, which, in turn, allows for the decision tree to discriminate between presence and absence of a bar code even in the overlapping case.

Likewise, as shown in Figure 2B, the bar code 10 may only be partially printed on the cap 20. This situation is frequent since conveyors are set to high speed which make bottles collide and move. In such a situation, the characteristic figure of the maximum gradient magnitude parameter (P2) can be based on the highest gradient on the image. Whether a bar code

10 is fully or partially printed on the cap 20, P2 has nearly the same value. This quasi invariance of P2 allows the decision tree to detect the bar code even when it is partially printed. The position of the highest gradient on the image is shown in Figures 8A and 8B (reference numeral 81), which again stress the quasi invariance of this characteristic figure in related situations.

The template image data may optionally subject to data randomization prior to the separation 421. In this way, individual images are shuffled so as to destroy, or at least minimize, any sequential or statistical similarities. Furthermore, only a fraction of the template data may be subject to separating and calculating. This "training" fraction, for example 70 % or $2/3$, is used for building the decision tree in step 423, whereas the remainder, for example 30 % or $1/3$, is used for testing the obtained model (decision tree). In this way, a training set and a test set can be formed. This can allow for minimizing the risk of building a model that performs well only on collected data.

Broadly speaking, a decision tree is used to classify images into two classes, i.e. whether a bar code is present or absent. One advantage of a decision tree is its readability: Once built, it is possible to extract rules that predict the target class.

The step 412 may involve image processing as mentioned. While decoding the image data one or more characteristic figures are obtained, calculated, compared, and evaluated. These figures can be obtained by a dedicated algorithm or simply as part of the above described image processing used for decoding a bar code. Exemplary characteristic figures include the parameters minimum gradient magnitude (P1, minimum value of the magnitude of the gradient computed on the image), maximum gradient magnitude (P2, maximum value of the magnitude of the gradient computed on the image), gradient magnitude Otsu threshold (P3,

threshold value returned by an Otsu algorithm computed on the image in gradient space), number of edges (P4, number of edges or segments detected in the image), L finder pattern candidates size (P5, number of L finder-pattern candidates formed by to segments that are almost perpendicular).

The calculated parameters are obtained in form of a table, wherein each line can represent one image, having a set of corresponding parameters (e. g. P1 to P5) and being classified into one of the class $C = 1$ (absence of bar code) or $C = 2$ (presence of bar code). An example of input data for building a decision tree may thus look like:

P1	P2	P3	P4	P5	Class (C)
10	15	...	...	...	1
10	15	...	...	...	1
10	25	...	...	...	1

In general, however, the set of parameters depends on the decoding algorithm. While decoding, images are processed to obtain some decoding parameters, i.e. some characteristic figure that can be compared to some given reference value in order to make determination on either a decoding result or the further procedure in the image processing. Also, additional parameters may be temporarily considered but chosen to be meaningless in the absence of a bar code in the image so as to be deselected. A final selection of parameters and characteristic figures can be left to an implicit selection during building the decision tree.

For the sake of simplicity the above data is reduced as providing an example of building the decision tree. This example is given in conjunction with Figures 5A to 5C and explained as follows. As a starting point, the calculating in step 412 yielded the following values for the characteristic figures P1 and P2 for the positive-hit image data with a bar

code present (class C = 2) and the negative-hit image data with a bar code absent (class C = 1)

P1	P2	C
10	15	1
10	15	1
10	25	1
20	15	2
20	25	2
20	15	1

From this exemplary data, it is apparent that selecting P1 is a better choice than P2 to discriminate between class 1 and 2, since P2 alone does not provide sufficient information about the class, while P1 does. Hence a first part of the decision tree can be built as shown in Fig. 5A: If P1 is lower than 15, then the left branch determines the class 1 (C = 1). If P1 is greater than 15 then evaluation proceeds along the right branch. The following data remain to be classified:

P1	P2	C
20	15	2
20	25	2
20	15	1

In a second step, the remaining parameter P2 is chosen so as to build the next part as shown in Figure 5B: If P2 is greater than 20, then the right branch determines class 2 (C = 2). If P2 is lower than 20, then the class is not yet known. In an actual implementation, of course, the data set is much larger than in the above example. In general, however, the tree is preferably kept as small and simple as possible to avoid unnecessary complexity.

Using the above concepts, a decision tree can be built by successively setting threshold on chosen parameters in order to divide data into sub-groups that contain less disorder measured

by the so-called Gini's Diversity Index (GDI) which is defined as follows:

$$\text{GDI}(t) = 1 - \sum [p(i|t)]^2, \text{ with } i = 1, 2,$$

and where $p(i|t)$ is the probability of belonging to class i (i.e. class 1 or class 2) at node t .

Further, in order to build the model (tree), a training set as explained above can be used and a 10-fold cross-validation procedure is performed within this training set. The cross-validation serves to tune the model parameters (in this case the depth of the tree). Several trees can be built with varying depth. The best tree can then be selected as the one with minimum cross-validation error. The obtained model can then be tested using the test set (1/3 of the data) to avoid overfitting.

A complete decision tree, for example the one considering also a further parameter/figure "P3" as shown in Figure 5C, can then be transformed into rules and the obtained model can be implemented into the decoding library (as part for example of the decoding system 304 of Figure 3). Following the above example the resulting rules may look in an instruction/command language as follows:

```
Node 1:  If P1 < 15 then node 2 else if P1 >= 15 then node 3
Node 2:  Class = 1
Node 3:  If P2 < 20 then node 4 else if P2 >= 20 then node 5
Node 4:  If P3 < 5 then node 6 else node 7
Node 5:  Class = 2
Node 6:  Class = 1
Node 7:  Class = 2
```

Such obtained rules can be both easy to implement and fast to test in an online production environment. As can be seen from the above, the requirements on processing resources are

limited, especially as compared to the requirements of the remaining image processing. Further, this allows for detecting the presence of a bar code in new images (image data), and at the same time a degree of confidence ("Accuracy") can be given.

The above process is schematically reproduced in Figure 6A: Decoding parameters P1, ..., Pn are provided as characteristic figures 611 together with class information (i.e. C = 1 or 2) 612 to the decision tree building 613. The decoding parameters are obtained by applying the decoding algorithm on a set of images, and the class determine the presence/absence of a bar code. The built decision tree is output in form of rules 614.

A classification accuracy can be defined as the ratio of successes on the test set size:

$$\text{Accuracy} = 100 \times (N_{TP} + N_{TN}) / N_{\text{nondec}}$$

where N_{TP} is the number of true positives, N_{TN} is the number of true negatives and N_{nondec} is the total number of non decoded images, wherein all these figures now refer to the test set. The above classification accuracy may also allow for selecting the best model.

Likewise, a so-called false positive (FP) can be a non-marked product that is believed to be marked and thus distributed. A false negative (FN) is a marked product that is believed to be non-marked and thus rejected. With FP, non-marked products are present on the market, and, since they are not marked, audit is not feasible. With FN, marked products but perhaps not decodable bar codes are ejected from the production line. The objective is to minimize FP and FN values since they both have negative impacts. In this way, a performance metric can be defined in form of a matrix:

		Predicted value	
		Absence of bar code	Presence of bar code
actual value	Absence of bar code	True Negative (TN)	False Positive (FP)
	Presence of bar code	False Negative (FN)	True Positive (TP)

In specific situations, FP and FN can have different impacts on the implementation. When a decision tree is built, it is possible to set a different cost for FP and FN. In this way, a given type of error (FP or FN) can be penalized more in order to bias the model (tree) towards reducing this error. Both measures are strongly related. An increase on the cost of FP will reduce them, but generally at the price of increasing FN. Since reducing FP affects FN, the first one can be specified as a function of the second one. For varying FP costs, one can see the effect on both FP and FN. This allows selecting the best situation according to business needs.

An example of the relationship between FP and FN for varying FP costs is given in Figure 7A showing a graph of 704 representing false negatives and a graph 703 representing false positives for the cost of false positive compared to false negative (axis 701) versus percentage (axis 702).

Figure 4B shows a flow chart of a method embodiment of the present invention. This embodiment is a method of detecting a bar code in image data and involves a step 421 of calculating a plurality of characteristic figures from the image data, a step 422 of evaluating a decision tree built for the plurality of characteristic figures, and a step 423 of making a determination of whether or not a bar code is present in the

image data based on the evaluating of the decision tree. The characteristic figures indicate a probability of whether or not a bar code is present in the image data.

The above process is schematically reproduced in Figure 6B: Image data 621 is input to the bar code decoding 622 which outputs code 623 and the decoding parameters 624 obtained during the decoding process. Bar code detection 625 inputs the parameters 624 and the decision rules 627. Detection 625 outputs the predicted class C 626 of the image (i.e., value of C, presence/absence of bar code).

In this way, the information returned by the bar code detection 625 is the class 626. In addition to this binary information, a confidence percentage can be provided. Each rule can be associated with a given probability of producing a correct result. This probability can be obtained from the training set. In the model obtained from training data (tree), terminal nodes (that represent a class) may not be completely pure. When class 1 is predicted, it means that more than 50% of the data from the training set that are in this node belong to class 1. However, there is usually a non-null proportion of class 2 as well. The confidence at node t can be given as the ratio of majority class on the minority class as follows:

$$\text{Confidence} = 100 \times N_{\text{majorityClass}} / N_{\text{minorityClass}},$$

where $N_{\text{majorityClass}}$ is the number of data belonging to the most represented class in node t and $N_{\text{minorityClass}}$ is the number elements belonging to the least represented class in node t.

Figure 9A shows a general apparatus embodiment of the present invention. Specifically, there is shown an apparatus 100 comprising a processing unit 110 and a memory unit 120. The memory unit 120 may store code that, when executed on the processing unit 110, implements one or more method embodiments of the present invention. Optionally, the apparatus 100 may

comprise an imaging unit 131 for acquiring image data. Further, the apparatus 100 may comprise a communication unit 132 for communicating an detection and/or decoding result to other entities, such as servers, controllers and the like. The communication may be effected over a network, such as a local area network (LAN), wireless network (WLAN), the internet, and the like. Further, also bus systems, such as CAN, can be employed for data exchange.

Fig. 9B shows a schematic view of a handheld embodiment of an apparatus for taking an image of the bar code and detecting and/or decoding the same. The apparatus 200 comprises a window 201 through which a digital image of an item 210 can be acquired. A two-dimensional bar code 10 is applied to the item 210 by means of any mentioned printing, mechanical, physical, or chemical method. The apparatus 200 may further comprise an integrated processing unit (not shown) for performing one or more method of embodiments of the present invention. An additional operation element 202 may be provided for switching on and off the apparatus 200 and/or initiating the taking of a picture, acquiring/obtaining respective image data, and/or processing of the digital image data so as to detect and/or decode the two-dimensional barcode 10 on the item 210. The device may, of course, take other forms and may be wire-bound or wireless.

Fig. 9C shows a schematic view of a fixed type embodiment of an apparatus for taking an image of the bar code and detecting and/or decoding the same. Like, for example, a module operable to be mounted on a production/distribution line for detecting bar codes disposed on items transported on said line. Again, the apparatus 200' comprises a window 201' through which a digital image of an item 210 can be acquired with a two-dimensional bar code 10. The apparatus 200' may further comprise an integrated processing unit (not shown) for performing one or more method of embodiments of the present invention. An additional fixation element 202' may be provided

for mounting the apparatus 200' on, for example, a production line in which a plurality of items 210 pass by the apparatus 200' for detection. The device may, of course, take other forms and may be wire-bound or wireless.

Benefits of the present disclosure and the embodiments of the present invention include that a lower rate of false negative can be obtained. Further embodiments of the present invention uses past decoding parameters to infer a decision tree and predict the presence/absence of pattern in the current image.

Although detailed embodiments have been described, these only serve to provide a better understanding of the invention defined by the independent claims, and are not to be seen as limiting.

Claims

1. A method of detecting a bar code in image data, the method comprising the steps of:

- calculating a plurality of characteristic figures from the image data;
- evaluating a decision tree built for the plurality of characteristic figures; and
- making a determination of whether or not a bar code is present in the image data based on the evaluating of the decision tree,

wherein the plurality of characteristic figures indicate a probability of whether or not a bar code is present in the image data.

2. The method of claim 1, further comprising a step of calculating a confidence parameter indicating correctness of the determination.

3. A method of building a decision tree used for identifying a bar code in image data, the method comprising the steps of:

- separating template image data into positive-hit image data with a bar code being present and negative-hit image data with a bar code being absent;
- calculating a plurality of characteristic figures for the positive-hit image data and the negative-hit image data; and
- building the decision tree based on the calculated characteristic figures for the positive-hit image data and the calculated characteristic figures for the negative-hit image data,

- wherein the plurality of characteristic figures indicate a probability of whether or not a bar code is present in the positive-hit image data and the negative-hit image data.
4. The method according to any one of claims 1 to 3, wherein the plurality of characteristic figures include at least one of a minimum gradient magnitude, a maximum gradient magnitude, a gradient magnitude Otsu threshold, a number of edges, a size of an L finder pattern.
 5. The method of claim 3, further including a step of randomizing the template image data.
 6. The method of claim 3, further including a step of splitting the template image data from collected image data, and a step of testing the built decision tree using the remainder of the collected image data.
 7. The method of claim 6, further including a step of randomizing the collected image data before splitting.
 8. The method of any one of claims 3 and 5 to 7, further including a step of generating rules based on the built decision tree.
 9. The method of any one of claims 3 and 5 to 8, further including a step of calculating a classification accuracy.
 10. The method of any one of claims 3 and 5 to 9, further including a step of biasing the building of the decision tree, by means of penalizing false positives and/or false negatives.
 11. A method of decoding a bar code in image data, comprising the steps of the method of detecting a bar code in image data according to claim 1 or 2, wherein processing of

decoding the bar code is aborted upon making a determining of no bar code being present in the image data.

12. The method of any one of claims 1 to 11, wherein the bar code is a two-dimensional data matrix provided to a product.
13. A device comprising a processing resource configured to perform a method according to any one of claims 1 to 12.
14. A computer program being executable on a processing resource and comprising code implementing a method according to any one of claims 1 to 12.
15. A computer program product storing the code according to claim 14.

Fig. 1A

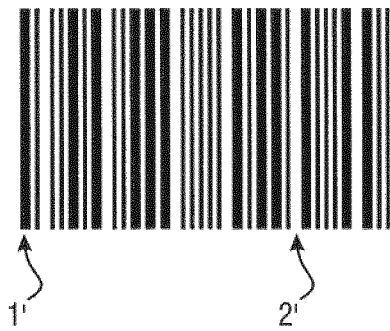


Fig. 1B

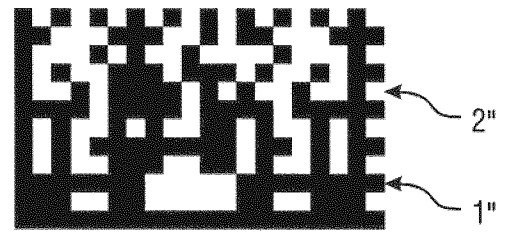


Fig. 2A

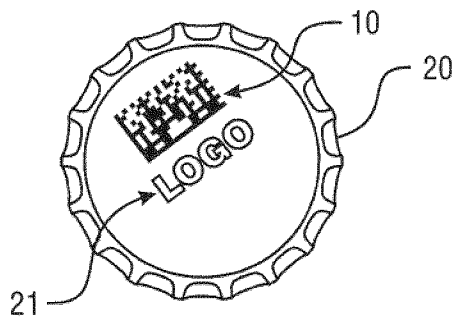


Fig. 2B

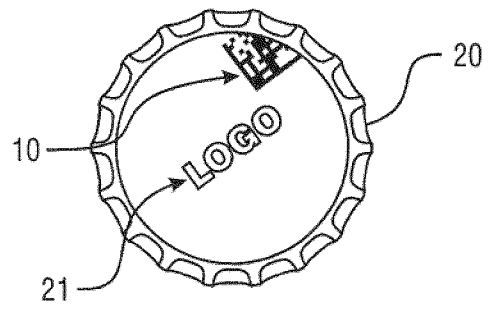


Fig. 2C

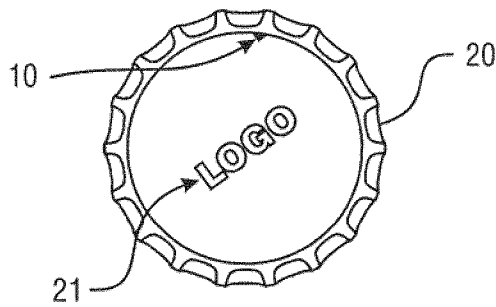


Fig. 2D

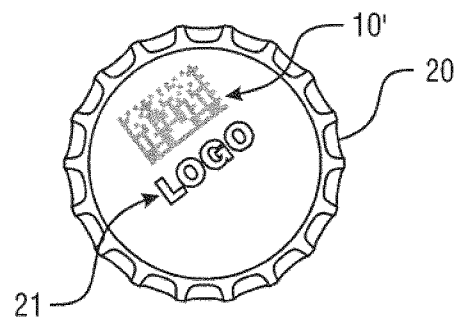


Fig. 2E

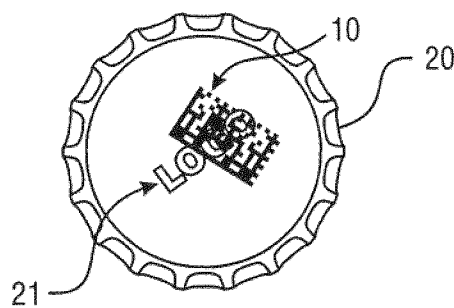


Fig. 2F

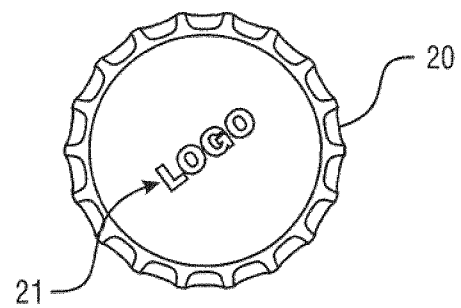


Fig. 3

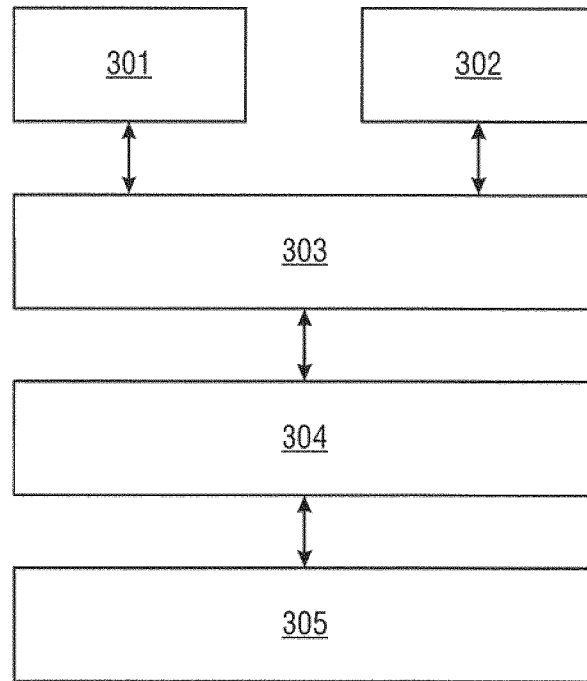


Fig. 4A

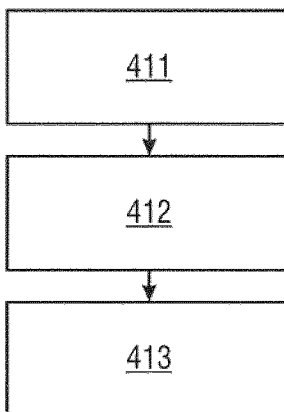


Fig. 4B

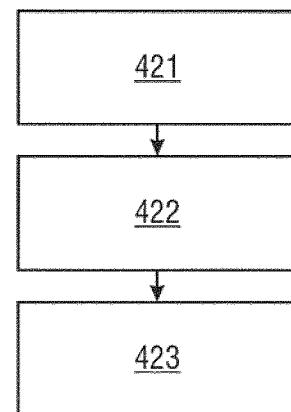


Fig. 5A

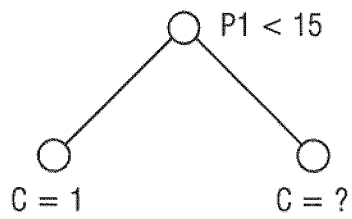


Fig. 5B

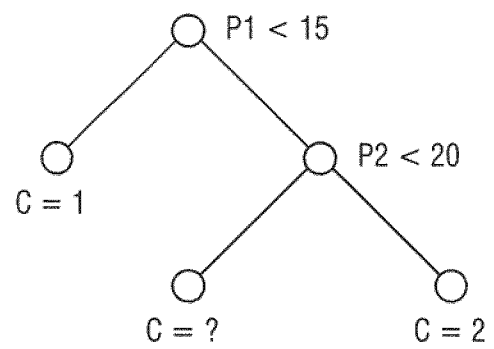


Fig. 5C

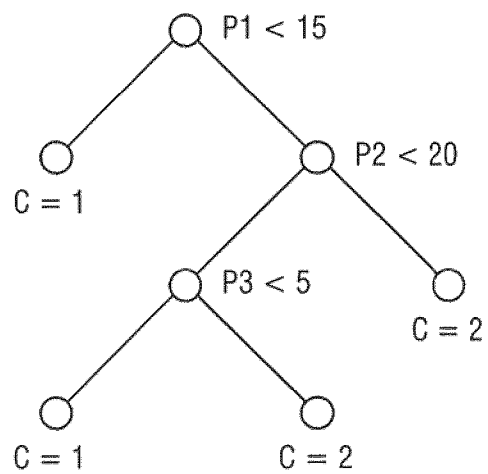


Fig. 6A

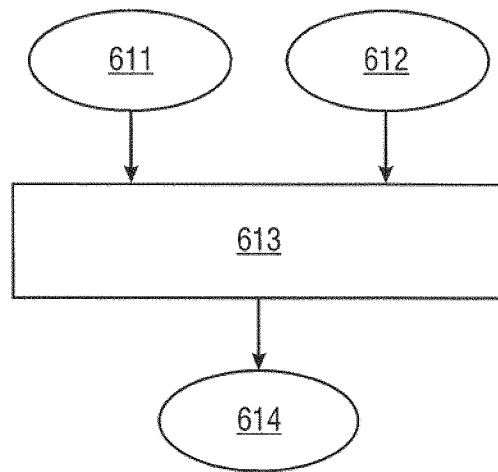


Fig. 6B

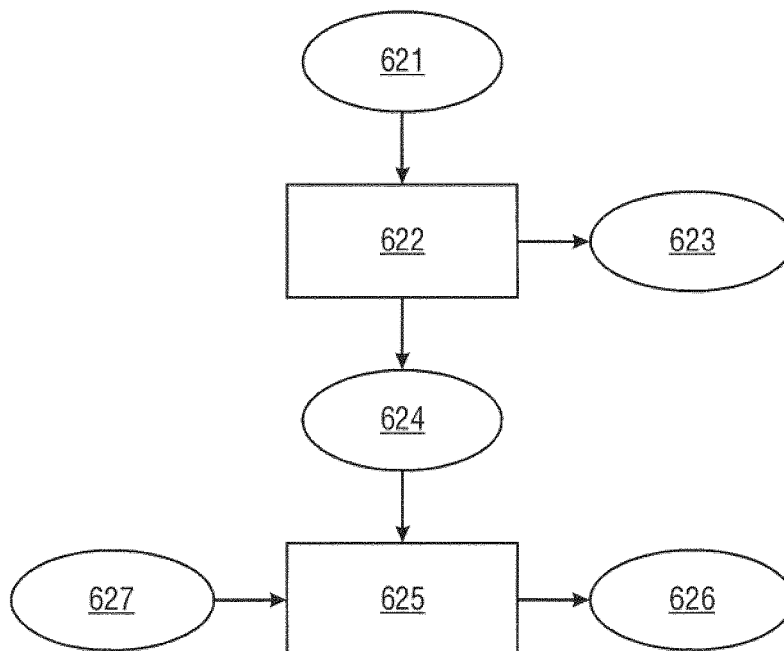


Fig. 7A

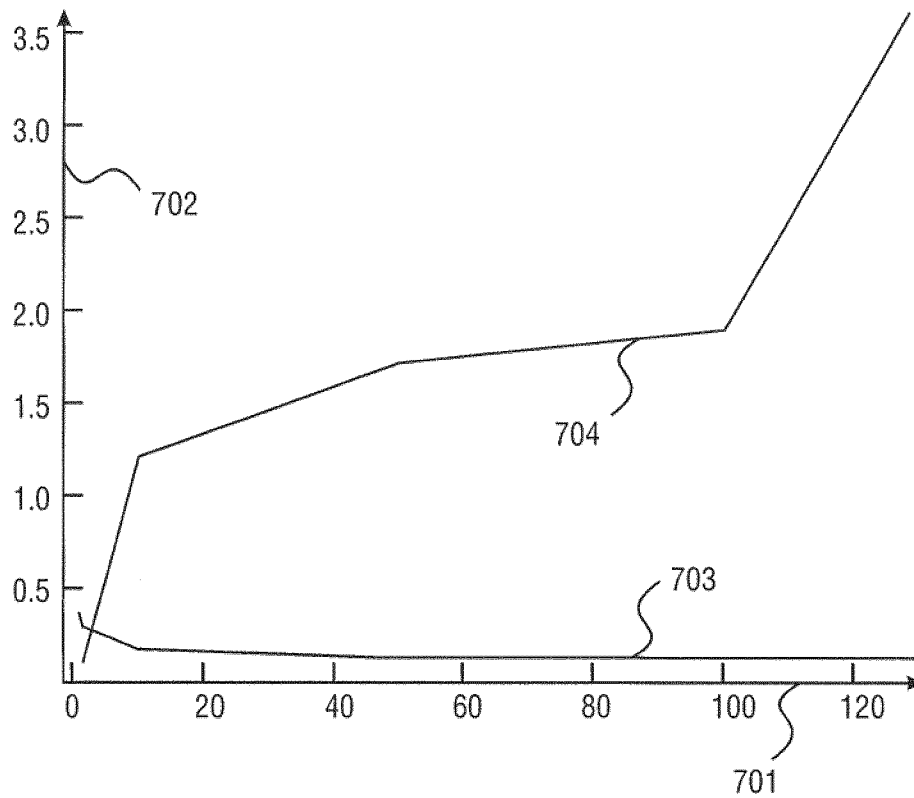


Fig. 7B

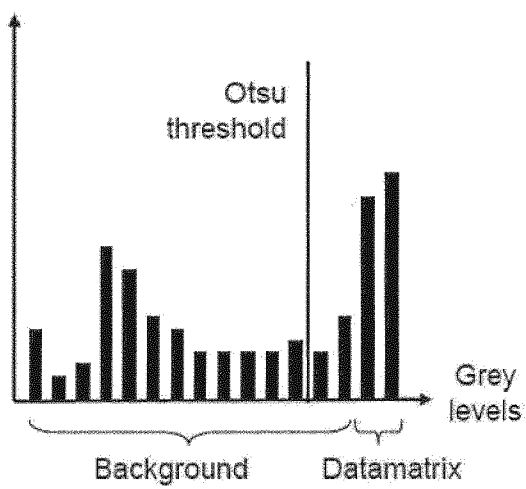
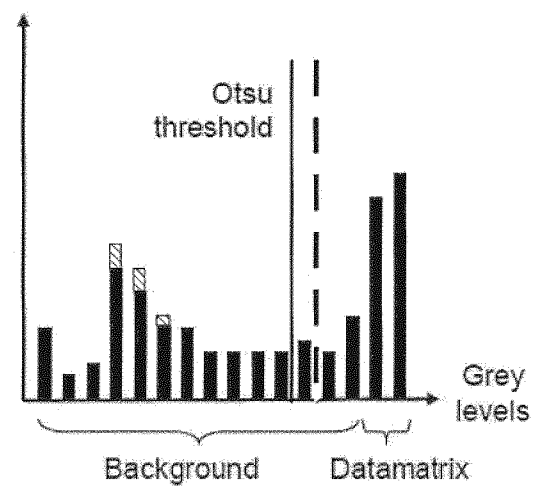


Fig. 7C



6/7

Fig. 8A

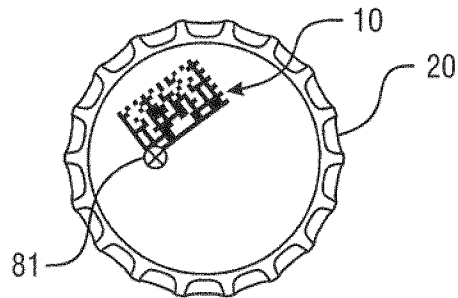


Fig. 8B

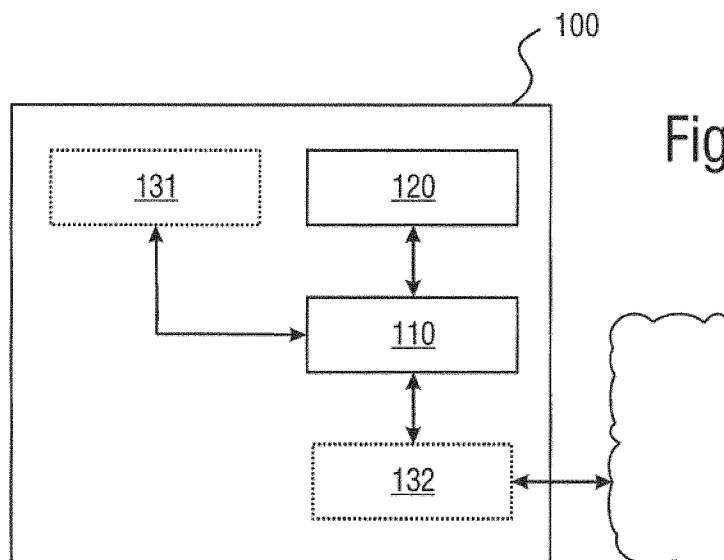
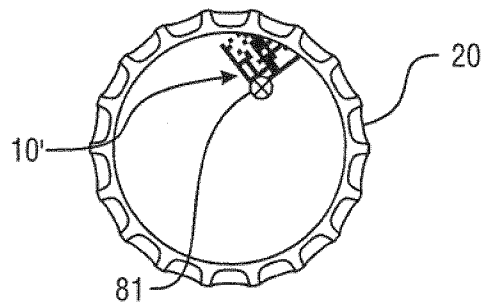


Fig. 9A

Fig. 9B

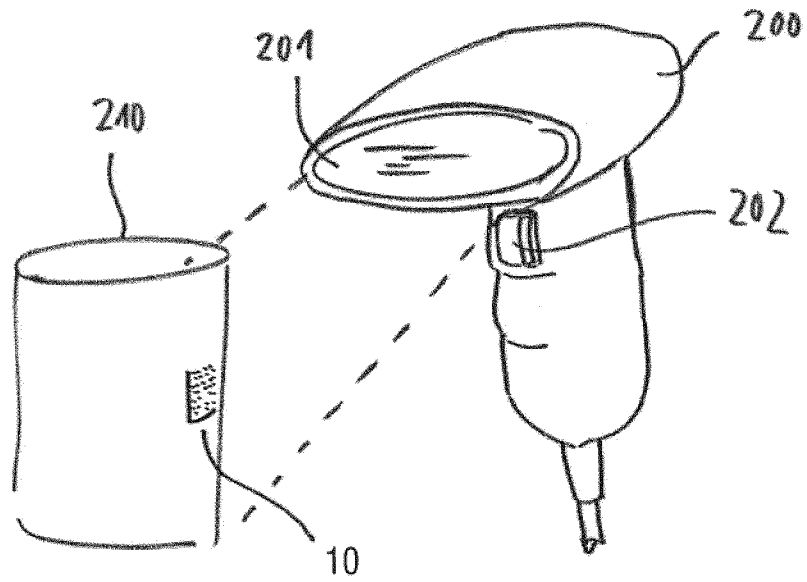
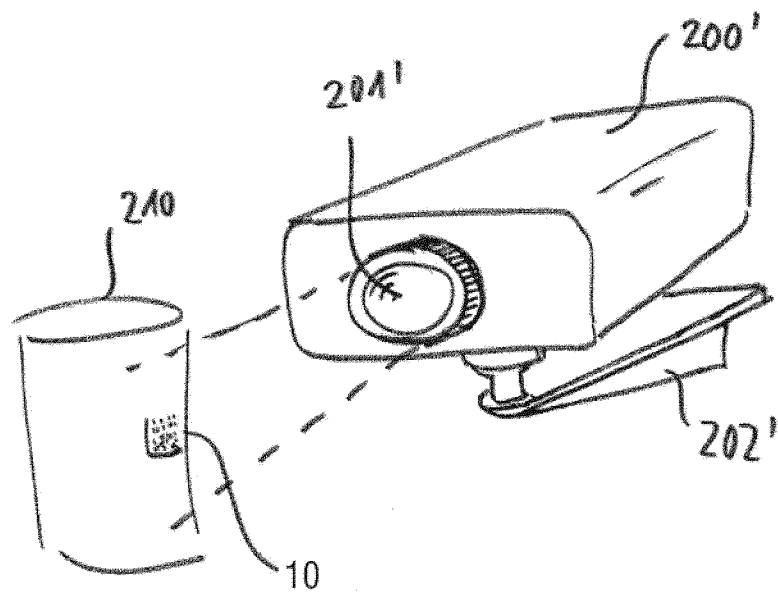


Fig. 9C



INTERNATIONAL SEARCH REPORT

International application No
PCT/EP2013/059187

A. CLASSIFICATION OF SUBJECT MATTER
INV. G06K7/14
ADD.

According to International Patent Classification (IPC) or to both national classification and IPC

B. FIELDS SEARCHED

Minimum documentation searched (classification system followed by classification symbols)
G06K

Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched

Electronic data base consulted during the international search (name of data base and, where practicable, search terms used)

EP0-Internal, WPI Data

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
X	EP 0 164 012 A1 (IBM [US]) 11 December 1985 (1985-12-11) page 2, paragraph 3 - page 3, paragraph 3 claims 11-15	1-15
X	US 2007/278306 A1 (BROCK CHRISTOPHER WARREN [US] ET AL BROCK CHRISTOPHER [US] ET AL) 6 December 2007 (2007-12-06) paragraph [0022] - paragraph [0024] figure 4 claims 8-13	1-15

☐

Further documents are listed in the continuation of Box C.

☒

See patent family annex.

* Special categories of cited documents :

"A" document defining the general state of the art which is not considered to be of particular relevance

"E" earlier application or patent but published on or after the international filing date

"L" document which may throw doubts on priority claim(s) or which is cited to establish the publication date of another citation or other special reason (as specified)

"O" document referring to an oral disclosure, use, exhibition or other means

"P" document published prior to the international filing date but later than the priority date claimed

"T" later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention

"X" document of particular relevance; the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone

"Y" document of particular relevance; the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious to a person skilled in the art

"&" document member of the same patent family

Date of the actual completion of the international search

2 October 2013

Date of mailing of the international search report

09/10/2013

Name and mailing address of the ISA/

European Patent Office, P.B. 5818 Patentlaan 2
NL - 2280 HV Rijswijk
Tel. (+31-70) 340-2040,
Fax: (+31-70) 340-3016

Authorized officer

Al-Hazam, Lorens

INTERNATIONAL SEARCH REPORT

Information on patent family members

International application No

PCT/EP2013/059187

Patent document cited in search report	Publication date	Patent family member(s)	Publication date
EP 0164012	A1	11-12-1985	CA 1222322 A1 26-05-1987
			DE 3572554 D1 28-09-1989
			EP 0164012 A1 11-12-1985
			JP H0157394 B2 05-12-1989
			JP S60262284 A 25-12-1985
			US 4608489 A 26-08-1986

US 2007278306	A1	06-12-2007	EP 2021974 A2 11-02-2009
			US 2007278306 A1 06-12-2007
			WO 2007143136 A2 13-12-2007
