An embodiment of the method results from the fact that, before step (b), the initial code is modified using predetermined parameters B, which in particular comprise one or more from the list:

a. Modification of the format version of the code,

b. Modification of the form of the modules of the code,

c. Modification of the mask of the code.

One embodiment of the method results from the fact that, before step (a), the base image is modified using predetermined parameters C, which in particular comprise one or more from the list:

a. Modification of the image section,

b. Modification of the resolution of the base image,

c. Modification of the brightness and/or contrast of the base image.

An embodiment of the method results from the fact that after step (e) the embedded code is either confirmed or rejected by an instance using an evaluation, wherein after the rejection steps d.1 and d.2 are repeated one or more times using parameters B and C until the further embedded code is confirmed by applying the evaluation.

An embodiment of the method results from the fact that the instance is a person or a AI module, wherein the AI module is trained on the basis of a data set of evaluations performed by people.

One embodiment of the method provides for the creation of a sequence of embedded codes from a sequence of base images and a common initial code, wherein the sequence of embedded codes is combined into a movie file. This makes it possible to extend the method proposed here to include movies as base images/base graphics.

The invention is explained using exemplary embodiments and the drawings, in which **Fig. 1b** shows an aesthetic QR code in black and white, **Fig. 2** shows a schematic section of an aesthetic QR code, **Fig. 3b** and **Fig. 3d** show the structure of an aesthetic QR code, **Fig. 4k** to **Fig. 4t** show simulated distortions of an aesthetic QR code in black and white, **Fig. 5** to **Fig. 8** show block diagrams of aspects of the method for generating an aesthetic QR code, **Fig. 9c** and **Fig. 9d** each show an aesthetic QR code in black and white.

The proposed method essentially involves generating an aesthetic two-dimensional barcode, such as a Data Matrix code (DM code) or a QR code, by building up layers. This method is applicable to all 2D barcodes. The following describes it using QR codes as an example.

Fig. 1b shows an example of an aesthetic QR code, which contains both the functionality of a conventional QR code and a graphic representation in black/white and grayscale.

According to **Fig. 2**, an aesthetic QR code is constructed from modules, or cells 11, where each cell contains or can contain a bit value 12 of the QR code and/or an image portion 13. It is now necessary to determine how each cell 11 is to be configured for a given QR code and a specific graphic representation as the base image.

The proposed method forms three layers according to **Fig., 3b** and **3d**, which together lead to the desired aesthetic QR code:

1. Bottom layer: Background - contains the image to be displayed.
2. Middle layer: QR code information (in usual black and white coding).
3. Top layer: Foreground – partially transparent image information that repeats parts of the image in the bottom layer and thus partially covers the QR code. The image in the top layer can be the same as the one in the bottom layer (examples: Fig. 3a, Fig. 3b) or consist of a different image (examples: Fig. 3c, Fig. 3d).

The bottom layer comprises an (initial) base image, which can be chosen almost arbitrarily. However, during the generation of the aesthetic QR code, it is possible that the base image will be adapted using graphical means. It is also possible that certain graphically extreme representations may be difficult or even impossible to combine with a QR code due to incompatibility with the QR code information.

The QR code information in the middle layer is designed so that not every cell 11 is completely encoded in black and white, but rather a portion (**Fig. 2**) remains transparent, thus allowing the underlying image layer to remain visible. For individual areas of the QR code that are particularly important for readability, cells 11 are completely filled.

The third layer conceals the QR code information in parts of the image, and in other parts, the black and white coding is transformed into differently colored light and dark elements using

partially transparent image elements. Thanks to the error tolerance built into the QR code standard, the QR code can still be read correctly even if parts of the pixels encoding the QR code information are no longer legible. In a first embodiment, the third layer repeats the image information from the lowest layer 1:1 in as many areas as possible, thus making it possible to display parts of the image entirely without any distracting QR pixels.

The third step of overlaying is performed using conventional graphic overlay algorithms. This means that a field with a given graphic content is overlaid with a second graphic content at a ratio of 0% to 100%. Various approaches to this overlay (pixel-by-pixel, area-by-area, color change of areas or pixels) are possible and can either be selected or implemented predefined by the graphics program.

The base image is first divided into a grid. The grid corresponds to the grid format that corresponds to the QR code to be used. The subdivision of the base image results in a matrix of, in most cases, square cells 11, where each cell 11 is then to be combined with the corresponding module of the QR code. One partial area remains unchanged and shows the image part 13 from cell 11 of the base image. The remainder receives the bit value 12 of the corresponding module of the QR code. The area share, the area shape, and the exact position of the bit value 12 can be varied. The area share can be between 30% and 70% of the area of the entire cell 11; it can also be expressed graphically in the number of pixels and can be as small as just one pixel. The shape can be round, angular, square with sharp or rounded corners, or even heart-shaped. The position is, for example, centrally in a cell 11 or, as shown in Fig. 2, in a corner. **Fig. 9c** shows an aesthetic QR code with partially round bits, and Fig. 9d shows one with partially heart-shaped bits in black and white.

In contrast to the state of the art, the QR code layer can be completely partially transparent except for the functional areas, thus displaying the aesthetic background image across the entire surface of the QR code. The border with the "classic" QR code pattern, which is sometimes mandatory in state-of-the-art processes, is omitted here.

The aesthetic QR code, initially constructed in three layers, is first subjected to quality assurance, i.e. a test for scannability, i.e. readability of the coded information, before it is issued as the final result of the process. This is achieved by simulating realistic scannability impairments in real-world environments, thus determining an average scannability rate in real-world environments.

By iterating over the QR code generation, this method allows the creation of aesthetic QR codes that meet definable minimum requirements for technical scannability.

Specifically, the process consists of the following steps, demonstrated using the example of **Fig. 4k** to **Fig. 4t**. Under real-world conditions, a QR code may appear distorted and/or visually impaired permanently or only at the moment of scanning due to:

Fig. 4k: perspective distortions,

Fig. 4l: Curvature of the entire QR code or parts of it,

Fig. 4m: (artificial) reflections, e.g. for codes printed on or presented behind reflective material,

Fig. 4n: (artificial) shadow,

Fig. 4o: Soiling/damage of the QR code,

Fig. 4p: Covering (here e.g. by a fingertip)/ tear/ damage to parts of the code.

Changes can also occur due to the resolution and color spectrum of a physical printer, as well as the characteristics of the print or display medium, which can be simulated. Contamination can be simulated by adjusting the so-called seed value:

Fig. 4q: Contamination with a seed value of 4023.

As shown in Fig. 4r and Fig. 4s, combinations and overlaps of distortion effects can also occur. These are also simulated during quality assurance. According to Fig. 4t, different print results can occur depending on any intermediate image processing using an RGB or CMYK color profile, which can also lead to altered scannability and is therefore tested.

In contrast to state-of-the-art methods, scannability testing follows a "top-down" approach, as a finished QR code is tested and subjected to many realistic distortions. In the state of the art, however, a "bottom-up" approach is taught, where, starting from an understanding of how scanning algorithms work, When creating the QR code, we try to avoid elements that could impair readability. However, this severely limits the freedom to choose base images and adapt them.

Each distortion can be determined by several parameters (e.g., angle of perspective distortion, shape, intensity, and gradient of a shadow). An initially generated aesthetic QR code therefore

serves as the starting point for a series of tests. The quality assurance process then includes the following steps:

Creation of a test series using different combinations of distortions, with:

a. specifying the number of test images;

b. each test image is created by applying randomly selected distortions with randomly selected parameters to the original aesthetic QR code (because many of the parameters are numeric, there are virtually infinite possibilities for test images for each aesthetic QR code);

c. the random selection can also be controlled depending on requirements (e.g., increased selection of a specific printing medium on which the code should primarily be printed).

In a second step, statistics on the scannability of the test series are determined:

d. each image of the test series is scanned using standard algorithms for QR code scanning,

e. the scan result is recorded for each image of the test series with the two possible values 0/1 for success/failure,

f. determination of success statistics (e.g. average of the individual scan results).

If the statistics shown in **Fig. 5** reach a predetermined minimum value ("minimum requirement met"), the initially generated aesthetic QR code is considered usable and is issued as a "finished QR code," or final aesthetic QR code. Otherwise, the process for generating a three-layer aesthetic QR code is repeated, with the image processing steps being performed for each of the three layers under modified parameters B and C.

In **Fig. 5**, “QR code generation” represents a suitable method for creating an aesthetic QR code from three layers, which can create a new aesthetic QR code based on image and QR code information, thus making changes to a given aesthetic QR code is possible. The data in an aesthetic QR code must be technically separated into image information and QR code information, which allows for separate revision of a design. "Determining scannability" is done using the process described above.

The repeated creation of an aesthetic QR code with different parameters can be performed by a single person. Alternatively, it can be automated by an AI-based artificial intelligence that learns the scannability. In addition, AI can also learn user preferences, which can be objectively determined, for example, through A/B testing.

Fig. 6 shows a machine or program logic for generating aesthetic QR codes (images with embedded QR codes). Hyperparameters are used here that affect non-graphical properties of image files and QR codes, i.e., not brightness, color values, shape of cells and areas, etc. These include, for example, selecting from a variety of QR code versions as a template, file type, and the like.

The method from **Fig. 6** requires as input 102 data to be integrated into the QR code as well as the input 104 of a base image. Context data, such as descriptive keywords (names of individual people, companies, or events for which a specific code was created, or the like), can also be entered as optional input 122. From these inputs and hyperparameters 106 and 112, the QR generator generates in step 108 a QR code 114 optimized for embedding in the image. A first image generator 110 generates 116 a variant of the base image optimized for the integration of the QR code. From these inputs, as well as further hyperparameters and optional context 122 provided as input parameters, the generator 120 generates an image with an integrated QR code, i.e., a preliminary aesthetic QR code (so-called QR image combination 124). A discriminator 126 checks this image for criteria such as the correct readability of the QR code and the learned human preferences. These preferences are learned by the discriminator 126 providing one or more example images 130 to the human evaluator 132. The example images 130 are evaluated by the human evaluator 132 using an evaluation input 128. Based on this evaluation data and the evaluation of the technical quality criteria, the discriminator 128 again adjusts the hyperparameters 106, 112, and 118 of the generators 108, 110, and 120. This loop repeats until the human evaluator 132 signals the end of the process via a special stop signal. Changing the context 122 during runtime is generally possible.

Optionally, this aesthetic QR code generation machine can be accompanied by a second machine as shown in **Fig. 7** for generating an image 210 from the QR data 202 and a context 204, such as keyword-like descriptions. In addition to the QR coding and aesthetic image components, the final aesthetic QR code can also include visible written information, such as a name. This makes

the aesthetic QR code personalizable and individualizable and can be used for applications with an additional security aspect. For example, third parties can determine for whom a specific aesthetic QR code was created without the need for technical aids.

The image 210 created by the machine of **Fig. 7** can be used as the base image 104 for the machine 100 from **Fig. 6**. This results in several design possibilities: the machine of Fig. 7 can be linked to the machine of Fig. 6 and the 206 hyperparameters can be adjusted by the discriminator 126 together with the other hyperparameters 106, 112 and 118.

As an alternative design option, a separate feedback loop with a human evaluator can be used for image generation, as shown in Fig. 8. The machine in Fig. 8 shows how this works: in step 218, the discriminator 212 presents one or more example images to the human evaluator 216, and the human evaluator provides evaluations to the discriminator 212 in step 214. Based on these evaluations, the discriminator 212 adjusts the hyperparameters 206 of the generator 208. This loop repeats until the human evaluator 218 signals the end of the process via a special stop signal.

The same benefits are achieved for data matrix codes and other two-dimensional barcodes. The process is not limited to QR codes.