Universität des Saarlandes
Naturwissenschaftlich-Technische Fakultät I
Fachrichtung Informatik

Invisible Codes as Enhancement of Retailing Processes Information Overlaying using Invisible Inks

Bachelorthesis

vorgelegt von

Marvin Barth

am 06.11.2015

Angefertigt und betreut unter der Leitung von
Christian Bürckert

Begutachtet von
Prof. Dr. Antonio Krüger
Dr. Michael Feld

Eidesstattliche Erklärung

Ich erkläre hiermit an Eides Statt, dass ich die vorliegende Arbeit selbstständig verfasst und keine anderen als die angegebenen Quellen und Hilfsmittel verwendet habe.

Statement in Lieu of an Oath

I hereby confirm that I have written this thesis on my own and that I have not used any other media or materials than the ones referred to in this thesis.

Einverständniserklärung

Ich bin damit einverstanden, dass meine (bestandene) Arbeit in beiden Versionen in die Bibliothek der Informatik aufgenommen und damit veröffentlicht wird.

Declaration of Consent

I agree to make both versions of my thesis (with a passing grade) accessible to the public by having them added to the library of the Computer Science Department.

Saarbrücken, ...

(Datum / Date) (Unterschrift/ Signature)

Abstract

In retail, identification of articles is often implemented by placing a barcode with a certain identification number onto the article. However, there are also other approaches to article identification. Some of these might even allow improvements in existing processes. The goal of this work is to lay the foundation for invisible QR overlaying (IQR overlaying), a method, which could potentially be used to aim for enhancements in retail processes as the checkout. In this work we design and implement IQR overlaying, which is an alternative approach to identifying articles that uses QR codes printed with invisible UV fluorescent ink. By radiating the codes with ultraviolet light it is then possible to read the printed code. We will research the effects of different inks in combination with different UV light sources and choose an appropriate pair of ink and light for working IQR overlaying. In a second step, we will look into ways of printing and reading invisible UV ink and implement a prototype by stamping the codes onto a variety of surfaces. Then we will find a way to read overlayed QR codes. First, we take a picture of the radiated code using a camera and then apply image filters to extract the QR code from the picture. Lastly, we use a QR reader to read the extracted code. During implementation we face a set of factors that influence IQR overlaying. We research the effects of a subset of these factors, namely the surface color and the surface motive. In addition to developing the IQR overlaying method we also provide hardware and software for working with IQR overlaying. This includes building a reading device, as well as the software that is required to use the reading device. We will suggest concepts for retail processes, which we want to improve by introducing IQR overlaying into retail. Lastly, we will experiment with various surfaces by overlaying IQR codes onto them to find out that the IQR overlaying can be made a viable method by extending this work with more research.

Contents

1 Introduction

Information overlaying describes the process of extending a surface by adding one or more additional layers on top, which contain additional information. According to this definition, printing can be seen as a possible way of overlaying information onto a surface. Information overlaying is often used to visibly store data. Invisible information overlaying is a possible way of hiding data. The term refers to a special kind of information overlaying, where the overlayed information is not visible to the naked eye. This can be achieved by including information in an invisible representation [7].
In the European retailing industry, information overlaying is used by printing the European Article Number(EAN) onto article surfaces, as a matter of identification [1]. This EAN is usually used in processes such as the checkout to identify product instances and record the products that are bought. For this work, we will distinguish between the term *product*, which refers to a certain product from a certain producer, and the terms *instance of a product* or *article*, which refer to the actual item that can be purchased in a shop. EAN is usually printed onto articles in form of a barcode.
The state of the art might be considered to be unsatisfying or not optimal. Processes as the checkout or the stocktake, which could be considered essential in common retailing, might have potential for improvement. For this reason, recent works are trying to develop new techniques to change or improve retail [15] [2].
In this work, we will investigate a way of using invisible information overlaying in retail. We will design a method of invisible QR code overlaying, which we will call *IQR overlaying*. Invisibility will be achieved by printing QR codes with invisible ink, which can be made temporarily visible by radiaton through UV waves (UV ink). The invisible QR codes will be refered to as *IQR codes*. Figure 1.1 shows a sample picture of an IQR code. The IQR codes used for the overlaying will be adapted to retail and referred to as *RIQR codes*. Their content will be made of the EAN, as well as the expiration date and a unique article identifier, which will differ for each article. We will refer to the process of overlaying RIQR codes as *RIQR overlaying*. Accordingly, overlaying of IQR codes in general will be refered to as *IQR overlaying*.

The RIQR overlaying method is one approach to improve retail. We have two specific processes in mind when talking about improvement. Those are the checkout and the stocktake. Our aim is to automate or at least partially automate these processes. Today most checkout processes require a worker to manually scan each article's EAN. Many retailers realize the stocktake by having workers count the articles and record their findings more or less manually. Partial automization could be one way to reduce the effort and maybe also the time needed to perform these tasks.

[1]https://en.wikipedia.org/wiki/International_Article_Number_(EAN), accessed on 5th Octobre 2015, information on EAN that is often used in retail

[2]https://www.youtube.com/watch?v=5loxojJvy1k, accessed on 31st March 2015, shows the approach of Tesco to build virtual supermarkets that deliver shopped goods to peoples' homes

Figure 1.1: Picture of an IQR code

1.1 Motivation

Optimization often refers to different possible improvements as for example speedup, reduction of effort, reduction of costs and increase of benefit. With reduction of effort we refer to the reduction of human resources that are required to execute a process. With reduction of costs we refer to lowering the costs of the resources required for a process execution, including human and non-human resources. Increase of benefit refers to an improvement of the quality of a process's results, so that other processes can be optimized with these results.
RIQR overlaying could directly adress these improvements. Speedup and reduction of effort might be achieved by automating or partially automating the retail processes. Partial automation could allow simultaneous execution of multiple work steps as for example parallel scanning, which might result in less time consumption. The amount of tasks which would have to be executed manually, could decrease this way. As a result, less workers could be required to perform these tasks, which would reduce the resources that are needed to execute the retail processes. The increase of benefit could be adressed by the additional data that is stored in RIQR codes. Adding a unique article identifier to each article would allow distinction not only between different products, but between all instances of these products. This could be an advantage for duplicate detection when automating. Also stocktake might be positively affected, since every article could be tracked individually. The availability of an expiration date might then allow directed search for expired articles in stocktake. This way, localization of expired articles would be immediately possible without loss of time.
It might be suggested that reduction of costs could also be a result of the reduction of workers required for performing retailing tasks. Introducing a completely new method to retail is bound to raise costs. Since we cannot make a statement about the relation

of both costs and retrenchment, we will not claim reduction of costs to be one of the adressable goals.

1.2 Goals

The aim of this bachelor thesis is to lay the foundation for the use of RIQR overlaying in retail. This results in five goals for us to reach. The top priorities are the design and the verification of RIQR overlaying, such that the method could be used as an approach for improving checkout and stocktake processes in retail. The implementation of definite instance-recognition is not only the basis for individual tracking of articles but also an important feature, which will support automization. In order to provide a fitting testing environment, implementing software and the equipment to read RIQR codes, as well as implementing a program that can coordinate IQR overlaying processes is required. Therefore, the exact goals are as follows:

- **The design of the RIQR overlaying method.**
  This is the main goal of this work. Designing a RIQR overlaying method includes a concept of the overlaying process, as well as a concept of the reading process that are required to use overlayed RIQR codes. We will deliver a possible concept for the application in retail.
  In this work, we will present a possible implementation of our concept of RIQR overlaying.

- **The verification of the RIQR overlaying method.**
  After designing RIQR overlaying we will verify the method. In order to do so, we will research factors of the environment of RIQR overlaying that might interfere with the method.
  As a result, we want to present a list of influencing factors, as well as a description of the influence process. We want to present a list of setups of influencing factors, with which RIQR overlaying will work despite possible interferences.

- **Implementing definite instance-recognition.**
  Definite instance-recognition means that for every two instances of the same product, we are able to distinguish between them with the help of our RIQR code.
  In this work, we will present a possible way to implement definite instance-recognition that is adjusted to its use in retailing industry.

- **Providing software and equipment that can read RIQR codes.**
  In order to actually use RIQR overlaying, we need appropriate software and equipment to read RIQR codes. We will create such equipment as a prototype version and provide the respective software.
  We want to exemplary display the processes of overlaying and reading RIQR codes, using the provided software and equipment.

- **Providing a program coordinates the overlaying process.**
  It is possible that there are factors, which will interfere with RIQR overlaying. Since it is not possible to test every possible setup of these factors, we will write a program, which will be able to run simulations on whether RIQR overlaying will succeed for a certain setup.
  We want to provide a program that, given a surface, can tell whether it is possible to apply RIQR overlaying and if so, how to apply the overlaying for it to succeed.

1.3 Dissociation

Since a bachelor thesis is linked with a time limit, some maybe desirable goals are not achievable within the scope of this work. In the following, we will explain what those goals are and why we cannot achieve them.
In this work, it will not be possible to research actual effects of RIQR overlaying on retail. Such research would require time and resources that would go beyond a bachelor thesis. Equipping a retailer with the setup for RIQR overlaying would also be expensive and should only be attempted after confirming the applicability of RIQR overlaying, which will be attempted in this work.
RIQR overlaying requires a method for overlaying RIQR codes onto a surface. Generally, we can achieve this by printing RIQR codes onto the surfaces. We will implement the printing process by stamping the RIQR codes onto the surface. This way of printing is chosen for simplicity reason, since the actual implementation of a printing process would require major resources and research, while only yielding results that are of minor importance for the overlaying method itself.
Our purpose is the verification of RIQR overlaying. We will perform experiments in order to research setups of environments, where the overlaying will work. However, we will limit the number of setups to those we consider interesting for the actual use of RIQR overlaying. A setup is considered as interesting, if it is likely to show up in retail.

1.4 Outline

In chapter 2, we will introduce related works, which cover research about the foundation of RIQR overlaying and research of similar techniques to the ones used for IQR overlaying. In chapter 3, we will introduce concepts of how interaction with RIQR codes in practice could look like. We will also provide concepts for the handling of RIQR codes in checkout and stocktake with the aim of improving named processes. In chapter 4, we will introduce and explain our approach to the implementation of RIQR overlaying. We will introduce the design of our codes, as well as our methods of printing and reading them. We will explain the verification of IQR overlaying and design a simulation method that will help judging, whether IQR overlaying is applicable in a certain case or not. In chapter 5, we will present the results of our research. This includes issues encountered while implementing, functioning ways of implementation and the evaluation of RIQR overlaying. Chapter 6 concludes, why RIQR overlaying is a method with potential for actual improvements. In chapter 7, we will give an outlook on possible future work that can be performed in order to extend this research.

2 Related work

In this chapter, we want to introduce works that cover the research of invisible inks and similar or alternative methods of information hiding. We will introduce alternative approaches to implement information overlaying and explain, why we did choose to not use these approaches.

2.1 Inks Fluorescenting Under Infrared or Ultraviolet Light

The term invisible inks refers to inks that are almost or completetly invisible to the human eye when applied to a surface. Applied inks can usually be turned visible. We divide invisible inks into groups depending on the process of making inks visible. There is research focusing on two big groups of invisible inks, the UV inks [3] and the infrared inks (IR inks) [16, 2]. Both, UV inks and IR inks contain particles, which will start fluorescenting, when being radiated with light of the right wavelength. For UV inks, this light is ultraviolet light and for IR inks, it is infrared light.
Invisible inks need to be applied to surfaces. Therefore, equipment and suitable methods are required. There are printing systems, which meet this requirement [11]. With these, it is possible to create lightfast prints that have a definition, which is high enough for easily reading or detecting the print. The printing system enables printing both, UV inks and IR inks.
Reading of tags that are printed with invisible inks, e.g. invisible barcodes, can be performed similarly to reading normal tags. This can be achieved by using a reading device, which emits light of the wavelength that makes the ink fluorescent. The device's reading mechanism must be adapted to the light that is returned by the tag, when fluorescenting. In [4] the reading device emits ultraviolet light of the wavelength between 230nm and 330nm. An invisible barcode is radiated with this light, making it fluorescent. The light that is emitted by that barcode is of a wavelength between 480nm and 1800nm. The reading device reads the emission, meaning that the barcode's elements that were printed are read, while normal barcode scanners read the gaps between the printed elements of a barcode.

2.2 Alternative Ways of Hiding Information

There are different approaches to information hiding. One example for an approach to hiding information in physical form is the use of disappearing ink or vanishing ink [1]. Vanishing ink can be used to write or print information in physical form, just as

Figure 2.1: Example of information embedding in an image from [6]

any other ink. As the name suggests, vanishing ink will vanish after a certain period of time though, leaving behind no traces of the information that was printed using the vanishing ink. While the information was originally not intended to be restorable, it is possible to reveal it again using some alkaline solution. This makes the use of vanishing ink a possible approach for hiding information.

Watermarking means embedding information into an object, which should usually not be removable, but which can be extracted in order to read the information. Since watermarks are often not directly visible to the human eye, we can consider watermarking to be a method for information hiding. Often watermarking is used for identification of an object or to identify ownership of the object [10]. Watermarking can be applied to both physical and digital objects. An example for physical watermarking are banknotes. Digital watermarking can be applied to any type of file, as images for example. Images or audio files can also be used as container for information by using the means of steganography [6]. This can, for example, be achieved by simply appending information into the file's extended file information. Another approach is hiding the information in the image itself, which can be achieved by altering the up to four least significant bits to encode the information. While the use of more least significant bits of each byte allows to store more information in an image of the same size, it also creates artifacts in the image and makes the hidden information more perceivable.

Similar to hiding information in pictures, information can also be encoded or hidden by using colors for objects that normally do not make use of colors. For example two dimensional codes like QR codes or similar codes could use colors to store additional information [9]. This would allow the extension of storage capacity immensely depending on the scope of colors that is used.

Another approach to hiding information would be the attachement of particles that cannot be perceived by humans. One way of doing this would be the creation of a covert thermal barcode system [7]. Thermal barcodes differ from barcodes in the sense that they are not visually read. The barcodes consist of special nanoparticles, which can be detected by thermal analysis. These nanoparticles are attached to an object in form of barcodes. By using thermal analysis, the barcode can then be detected and read. Since the nanoparticles for this approach have discrete and sharp melting points, they are extremely resistant, making the thermal barcode system fit for information hiding in extreme environments.

2.3 Information Hiding in Security

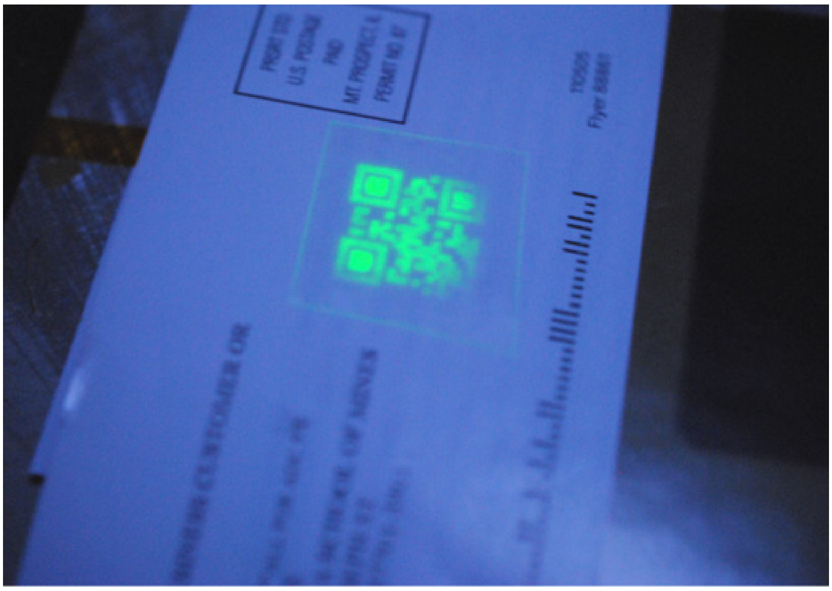

Figure 2.2: Sample of hidden watermark from [17]

Information hiding can be seen as one approach of achieving secrecy. Since simply hidden information can be found and read, it could be suggested that this notion of secrecy is not particularly strong. Still, hiding of information is a common practice that is used when communicating secret data [6]. The basic idea behind this practice is the fact that well hidden information is harder to counterfeit than plainly visible information. For this reason, hiding of information is often used for supplementing other approaches trying to achieve security.
2D barcodes and QR codes have become common matters of storing data in the real world [18] and the internet [21]. These mediums could be misused and might contain malicious content. Therefore, the security of 2D barcodes and QR codes is of interest. In order to achieve this, verification of the good intent of such a code is desirable. By checking wether a code is actually trustworthy in advance, damage by malicious content can be avoided. This can be achieved by hiding watermarks inside these codes [20].
Hiding of information is not only used to verify stored content. Another application is the verification of the legitimacy of objects in the real world. Examples are the protection of banknotes or ballots from being forgable. Money, as well as votes in an election

are objects that are illegal to counterfeit [1]. For this reason, ways to hide codes for verification on such objects have been researched. For banknotes, it is possible to print invisible QR codes as a tool for verification onto the note [17]. These QR codes contain special security characters that confirm the legitimacy of the banknote, and thus allow recognition of faked notes, which miss these characters. The ballots for voting can be protected by printing an invisible confirmation code in the fields for each possible option of the ballot that differs for each ballot [5]. Checking one of these options will be done with a special pen that reveals the confirmation code. The ballots and the respective confimation codes are only known to the organizers of the election. Without knowing the code for a certain option of a certain ballot it is thus not possible to fake the vote.

2.4 Advanced Printing Techniques

Figure 2.3: Sample picture of a printed touchscreen from [19]

Printing usually refers to a process of applying a material to a surface. The most common version of this process is the application of inks onto paper. One might think that printing is only meant for application of colors, in explicit liquid materials. Printing is not restricted though. For example, it is possible to print electronic circuits onto paper or plastic [13, 14]. This can be achieved by printing silver nanoparticles onto on a surface. These printed nanoparticles will then conduct electricity if printed in form of an electrical circuit.
With circuits being printable, it is possible to also print electronic devices. Sensate surfaces and displays are devices that often cover surfaces as one of the upper layers of that very surface. For this reason, printing of these devices might be of interest. It is possible to print sensate surfaces [8]. Sensate surfaces are based on a layer of capacitive sensor electrodes and different types of RF antennas that are printed on a flexible surface. Detection of interaction with the surface is mainly handled via passive and

[1] http://www.strafgesetzbuch-stgb.de/, accessed on 31st March, §107-108 and §146 state laws about counterfeiting

active capacitive coupling schemes. Display devices are also printable with the right means [12, 22]. This can for example be achieved by creating a matrix of capacitive light-emitting display elements. Display elements can then be controlled in order to display information.
Further research has found ways to combine the functionality of sensate surfaces with displays. The result is a process of printing touchscreens onto surfaces [19]. The display part of the touchscreen is printed similarly to a normal printed display, that is by printing display elements. These elements contain electrodes, which are in charge of displaying information, and which can also be used to implement capacitive touch-sensing. This way, interaction with the surface can be detected. By implementing software for these printed touchscreens it is then possible to use detected interaction as input, making the screens touch-sensitive.

3 Concept

In order to design a new method for the use in retailing industry we are bound to design a concept for the actual use of the method. RIQR overlaying is meant to optimize retailing processes, in explicit the checkout process and the stocktaking process. This should be achieved by introducing RIQR codes as a new tagging system for identification of articles in retail. The new tagging system will be refered to as *RIQR tagging system*. Tagging articles usually involves adding tags to each article, as well as reading tags that are attached to articles. The ways of adding tags to articles and reading them can be different depending on tagging system in use [7, 17]. RIQR codes and barcodes in retail differ in the fact that IQR codes are invisible, while barcodes are visible. For this reason, we will have to adapt processes that involve adding RIQR codes to articles or reading RIQR codes.
In this chapter, we will describe characteristics that have to be paid special attention to, when working with IQR codes. We will also analyze the current checkout and stocktake processes and provide a concept for the adapted processes that could be implemented to use the RIQR tagging system. Since the printing and reading of RIQR codes are of importance, we will provide concepts for these.

3.1 Handling of IQR Codes

IQR codes are different from QR codes and barcodes, which we know from everyday life. For this reason, we will provide features of IQR codes and characteristics that have to be looked out for, when working with them.
While IQR codes are invisible, they are still codes that are read visually. This rather paradox characteristic requires a method to turn IQR codes visible in order for us to be able to read the IQR codes' content. This can be achieved by radiating an IQR code with ultraviolet light, since IQR codes are printed with UV ink. Radiated IQR codes will fluorescent. The color of the IQR code depends on the ink that is used and the surface that the IQR code is printed onto, while the grade of fluorescence depends on the synergy between ink and ultraviolet light. Because of this, choosing the right types of ink for the use in certain areas is of utmost importance. For this reason, the applicability can be maximized by a suitable ink.
IQR overlaying overlays on top of a surface. This means, the most upper layer of the surface is the IQR code. Since IQR codes are a storage for data, damaging the codes and with them the data, is unwanted. Being on the most upper layer grants no protection from damage for the IQR code. For this reason, actions that might damage a surface with overlayed IQR codes should be avoided. Another possible approach to preserve the IQR code would be to add an additional protection layer. To not interfere with the readability of an IQR code, this additional layer should neither be visible nor affect the

UV waves passing through or darken the fluorescence of the IQR code beneath.
In our case, IQR codes are used in form of RIQR codes. RIQR codes come with the feature of definite instance recognition. Definite instance recognition means that there are no two RIQR codes on different articles that are equal. This is an important feature that can be used when working with RIQR codes. It for example allows the implementation of software that can automatically identify articles without the risk of duplicate detection. While IQR codes in general do not implement this feature, the use of special kinds of IQR codes, which provide definite instance recognition can be useful for uniquely identifying any type of object.

3.2 Concept for Reading RIQR Codes

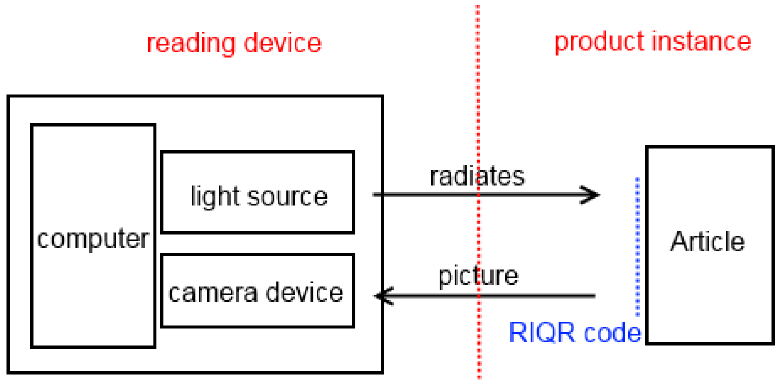

Figure 3.1: Concept scheme of a RIQR reading device

Special reading devices are needed in order to read RIQR codes. A concept scheme of such a device is shown in figure 3.1. In order to make RIQR codes visible, ultraviolet light is required. For this reason, the reading device includes a light source that is able to radiate the RIQR code with ultraviolet light. Since we need a picture of the RIQR code in order to read it, we also include a camera device. The camera of course needs to be able to take a picture of the RIQR code, while the light source radiates it. The reading device also needs to be connected to a computer of some sort. This computer will take care of the processing of the picture and the decoding of the QR code, which is the basis of our RIQR code. Since an image can contain multiple codes, it is possible to read multiple RIQR codes at the same time, as well by having the computer not only decode one, but all codes that are contained in the picture. The computer will also forward the content of the RIQR code to software that will process the scanned article.
Figure 3.2 shows a flowchart of the internal reading process of the earlier described reading device. The RIQR code can only be read while being radiated with ultraviolet light. This is the reason why the reading is bracketed with the activation and deactivation of the light source. The reading itself works by taking a picture of the fluorescenting barcode and analyzing that picture. The RIQR codes has a different color from its background. This enables the extraction of the code from the background to read it. Extracting the code means to apply image filters that divide the taken picture

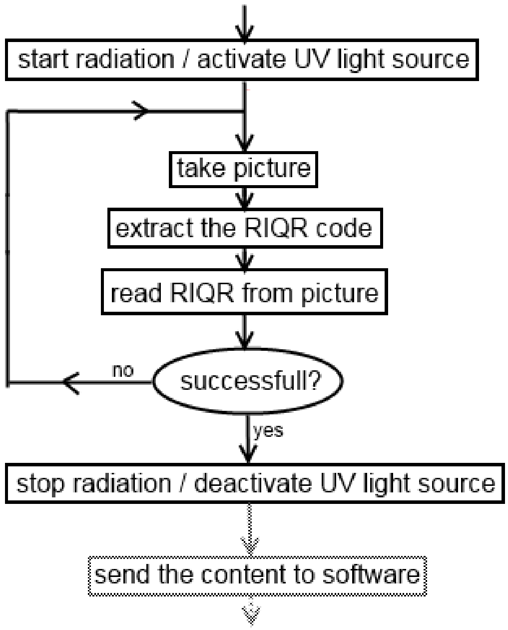

Figure 3.2: Flowchart of the internal reading process of the reading device

into a black code and white background. There are cases when the RIQR code cannot be read, for example when the code is too small, not in the picture, or the picture is taken with a bad angle. It is also possible that ambient light makes the code fluorescent without sufficient contrast to the background, so the code cannot be extracted. For this reason, if the RIQR code cannot be read from a picture, another picture is taken and so on. Afterwards, the content of the RIQR code can be processed by further software.

3.3 Concept for Printing RIQR Codes

In order for us to be able to use the RIQR codes, we require them to be on the packing of each article. Therefore, the codes need to be attached in some way that will not interfere with the reading. The basic idea is to overlay the RIQR code as a top layer onto each packing.
Currently, product packings are often produced by storing the actual article inside the blank packing. Then the packing is closed and put on a conveyer belt, where the packing will be printed on. The printing itself can happen in one or more steps, in which every step prints one layer onto the packings' surface [1]. The exact structure of the packing process differs for different kinds of product packings. For carton packings for example, the printing can be handled before the actual article is stored inside the packing. However, a layerwise printing onto the carton is still happening.
Our concept is to use the layerwise printing of the article packings. It could be altered into additionally printing the RIQR codes onto the packings without changing the rest

[1] https://www.youtube.com/watch?v=xmYht5dRAmU, accessed on 1st April, shows a packing machine with included ”printing” of labels onto the packing

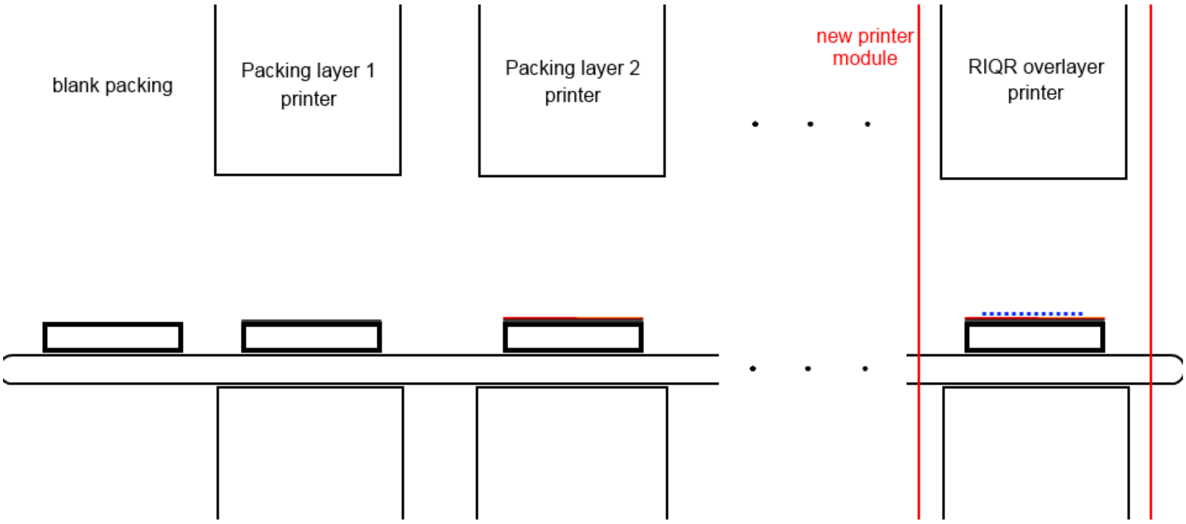

Figure 3.3: Sketch of an additional printer module for RIQR codes in packing printing

of the printing process. For every packing, we would have to add an additional printer module at the end, which prints an additional layer on top of the other packing layers. This new layer should be the RIQR code. Because the RIQR code is invisible, the packing would not visually differ from the original packing, which does not have the additional layer. But since the RIQR code can be made visible, it is possible to benefit from its advantages. In case a producer would not introduce this new printing module for RIQR codes, it would also be possible for the retailer to get such a printing module. This way the retailer would be able to overlay RIQR codes on the articles that miss the code layer. Then the retailer would not have to differ between articles that have RIQR codes and articles that do not. If this differation has to be considered, it might involve additional effort, which we want to circumvent.

In order for the RIQR codes that are printed this way to be fully pledged RIQR codes there is one factor to watch out for. Each RIQR code should be unique. This results in two requirements for the printer modules that print the RIQR codes. Printer modules must be able to efficiently print different codes and there must be no printer module that ever prints a code that has already been printed before. Efficient printing should be achievable by choosing suitable printers that fullfill this requirement. To guarantee the uniqueness of each RIQR code, it is possible to link the code content to the printer modules. Each printer module could for example get assigned a unique ID, which will be included into the RIQR code. This would ensure that there cannot be two printer modules that print the same code. In addition, a timestamp that indicates the date of the printing job, could be included to prevent printers from RIQR code duplication. This way, the uniqueness should be guaranteed as long as the uniqueness of the printing modules' IDs are guaranteed.

3.4 Adaption of the Checkout

The checkout refers to the process of identifying articles that customers want to buy and computing the price of those articles. An additional purpose of the checkout is to record which articles are sold and thus not available in stock anymore.
The checkstand can be divided into three parts. The conveyer belt, onto which customers can put the articles they want to purchase, the register area, in which the employee scans the articles, and the post register area, in which already registered articles are put (see figure 3.4).

In most cases, the identification of articles is done by an employee (see figure 3.5).

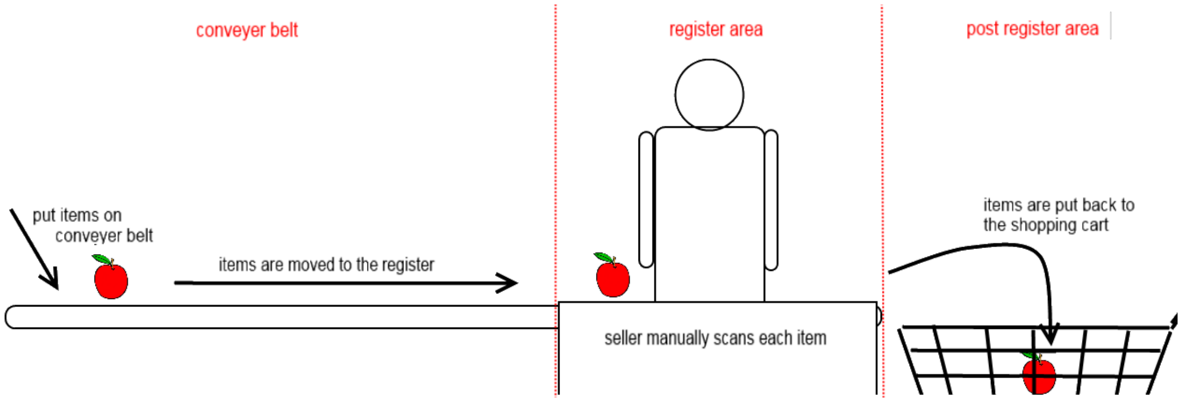

Figure 3.4: Sketch of the common structure of a checkstand

This is achieved by either scanning each article's barcode or scanning each product once and entering the amount into the register.

The scanning subprocess of the checkout is marked by a red rectangle in figure 3.5. We can see that the employee has to perform a scanning action for each article. This is where we approach improving the checkout process. Our aim is to reduce the time needed for the single scanning actions that need to be performed to register all articles. In order to achieve, this we will introduce the RIQR tagging system and add an additional part to the checkout. This part is called the black box. The blackbox is a device that can read RIQR codes on the surface of articles that are inside the blackbox (see figure 3.6). Single RIQR codes, as well as multiple RIQR codes can be read simultaneously.

A change of the checkout process requires an adaption of the task that employees have to carry out, when working at the checkstand. A flowchart of the adapted process is shown in figure 3.7. The changes only affect the scanning subprocess. This is based on the fact that we want approach the improvement by only improving the scanning subprocess. We want to partially automize the scanning. The introduced blackbox will take care of that. Every article is directly moved from the conveyer belt to the inside of the blackbox to be automatically scanned. Since our RIQR tagging system implements definite instance recognition, each RIQR code only exists once. For this reason, no article will be recorded more than once.
However, articles might not be scanned successfully by the blackbox. This might happen if, for some reason, the article or the code were damaged. As a result, the code

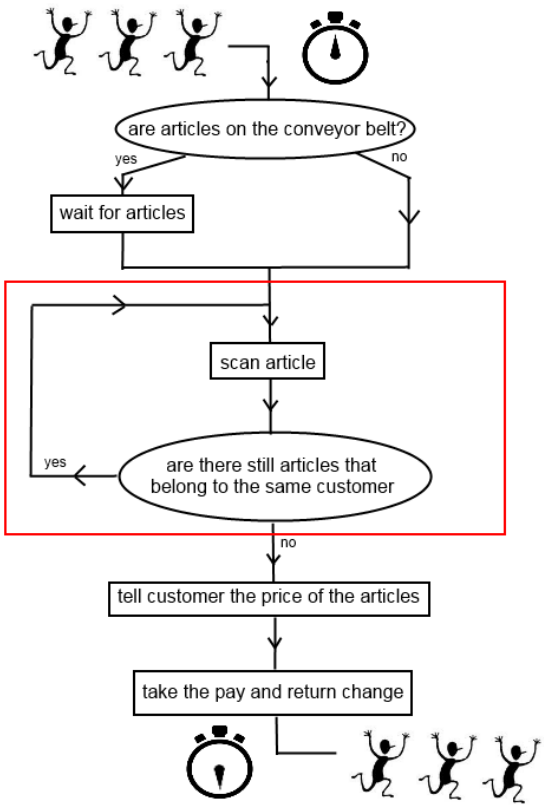

Figure 3.5: Flowchart for an employee's task at checkout

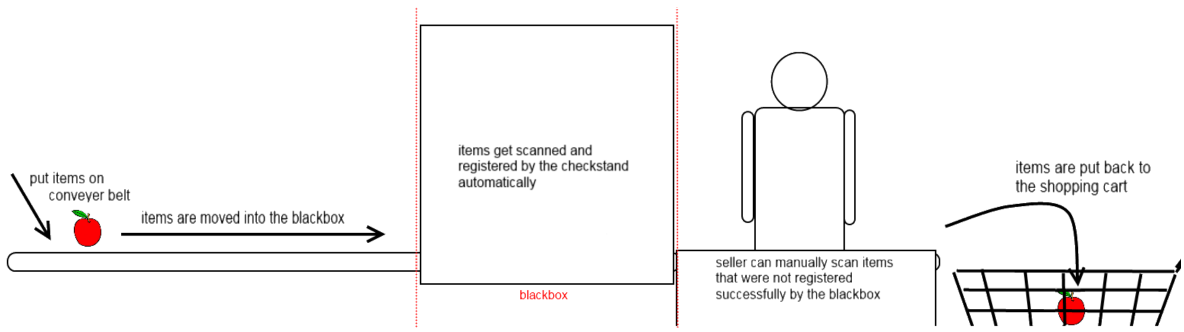

Figure 3.6: Sketch of the checkstand with blackbox

itself might not be readable anymore or the surface on which the code is overlayed might be in a shape, in which the blackbox cannot read the code successfully. For this reason, the employee is still needed. Articles will leave the blackbox and enter the register area, where the employee has to register the articles that the blackbox could not scan. In order to show the employee which items were not scanned yet, there will be a display that shows all the articles in the box and marks the ones that were already scanned. The employee will scan articles as he used to do. Since we only extend the current identification system using EAN, the EAN barcodes are still available on the articles packing and can be used for identification of the product. Simply using the current EAN barcodes will create data inconsistency. The checkout is in charge of reg-

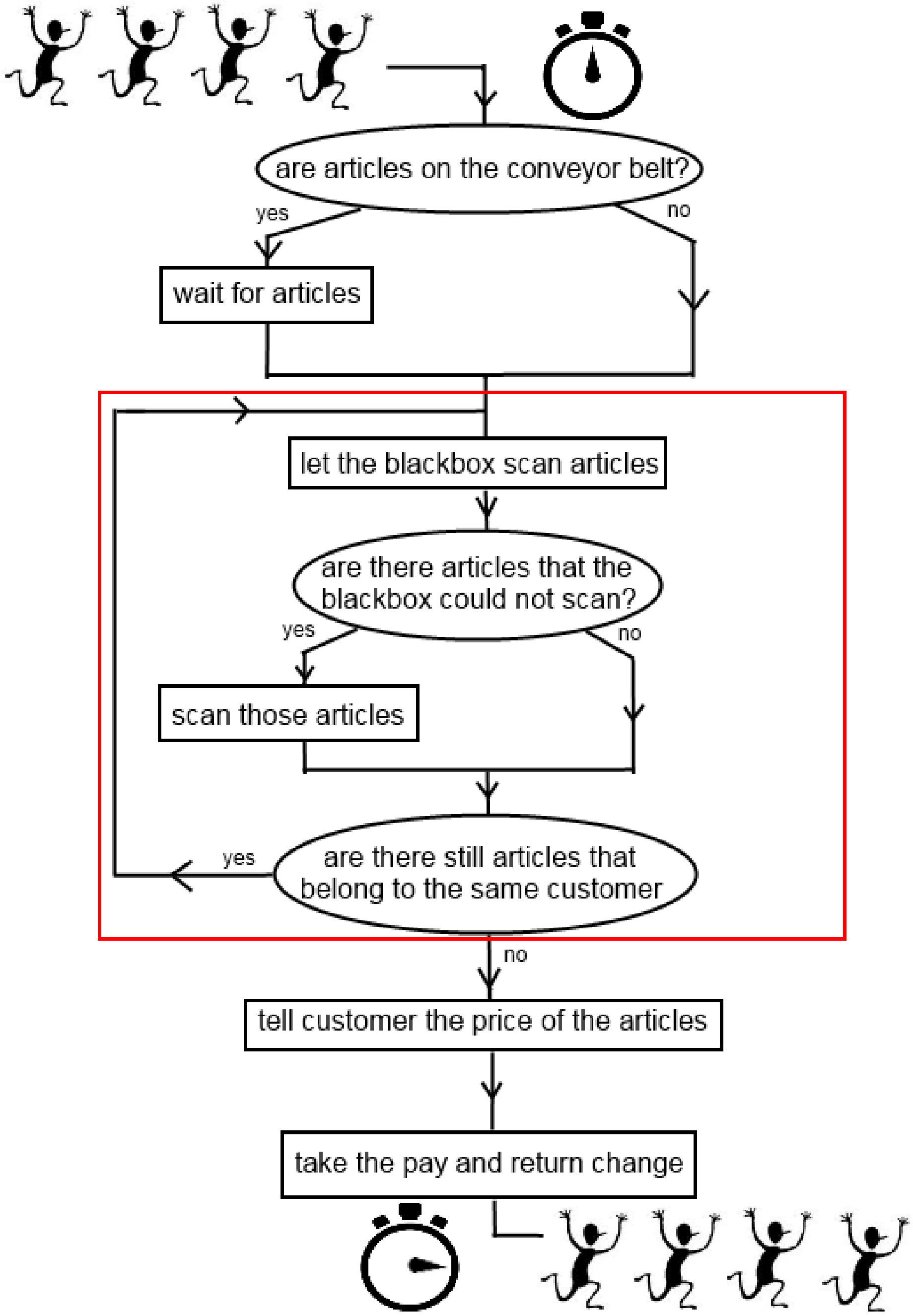

Figure 3.7: Flowchart of the adapted task for employees at checkout

istering the articles that leave the stock. Since EAN alone is not enough for definite instance recognition, there will be no information about which exact instance is leaving the stock. For these cases, it might be considered to replace the normal EAN barcodes with barcodes that additionally contain the unique article identifier. This is a subject beyond this work's scope.
The actual improvement in this concept of the checkout process is based on the partial automization. Since the blackbox scans the articles while they are moving through, the time for taking each article, scanning each article and putting back each article will be saved. The blackbox's ability to scan multiple articles at the same time will decrease the amount of scanning actions that have to be performed. This will also result in less time consumption. The time that is needed to scan articles that could not be scanned by the blackbox will result in additional time consumption. Those articles need to be taken, scanned manually and put back. Those actions match exactly with the actions that an employee needs to perform without this concept. There is no improvement but also no worsening of time consumption in case of an unsuccessful scan. We can approximate this gain of time with the following definition.

Definition. *Let A be the set of all articles that customers purchase and let $A_s \subseteq A$ be the set of all articles that can be scanned successfully. Further let $t(a)$ be the time that is needed to take an article a, scan it and put it back. Then the gain of time t_{gain} by introducing the described concept is:*

$$t_{gain} = \Sigma_{a \in A_s} t(a) = \Sigma_{a \in A} t(a) - \Sigma_{a \in A \setminus A_s} t(a)$$

3.5 Adaption of the Stocktake

The stocktake refers to the process of counting the entire inventory of a business. Its purpose is determining the capital and debts of that very business at a certain time and creating an inventory.
The stocktake process is performed by one or more employees. Often the employees

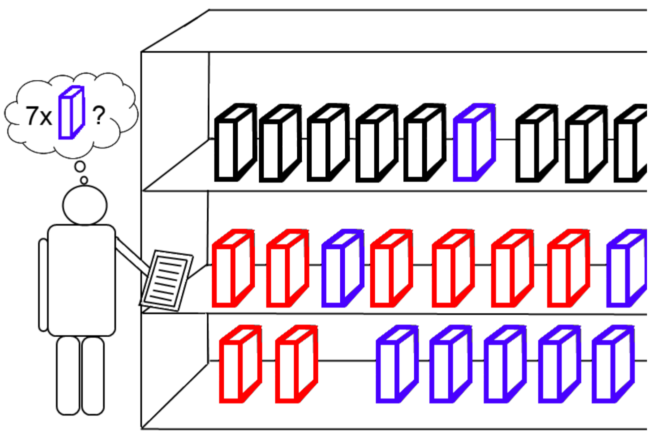

Figure 3.8: Sketch of an employee performing a stocktake

have to locate articles of different products, count and record them manually (see figure

3.8.
The stocktake scenario, which we base our concept on, is actually a simplified version of the present scenario in retail. As shown in the figure, our scenario assumes that the articles are lined up next to one another in a way that for each article, a part of its surface is clearly visible to us. This means we can easily scan the RIQR code on each article's surface. This is often not the case in retail, where articles in the shelf are usually also stored in depth, one article behind another. This means only the surface of the article in the very front is visible to us. Thus, scanning each article might not be possible without investing the unwanted effort of taking each article out of the shelf and then putting it back. Still, there might be cases in the future, where it actually is possible to arrange the articles in the assumed way. One might suggest that for example boxlike articles could be arranged next to one another. However, if this assumption does not hold at all, there still is no disadvantage for the stocktake. In this case, the present stocktaking process can be completely adopted without modifying it. This will not result in any improvements, but at the same time, it will not result in any loss of performance either. We could say that there might be potential for improvement of the stocktake, depending on how article arrangement in retail will evolve.
A possible flowchart for our concept is shown in figure 3.9. We can divide this stocktaking process into steps, where counting the amount of each product can be considered as one step.

The counting subprocess for articles of one product is marked by a red rectangle

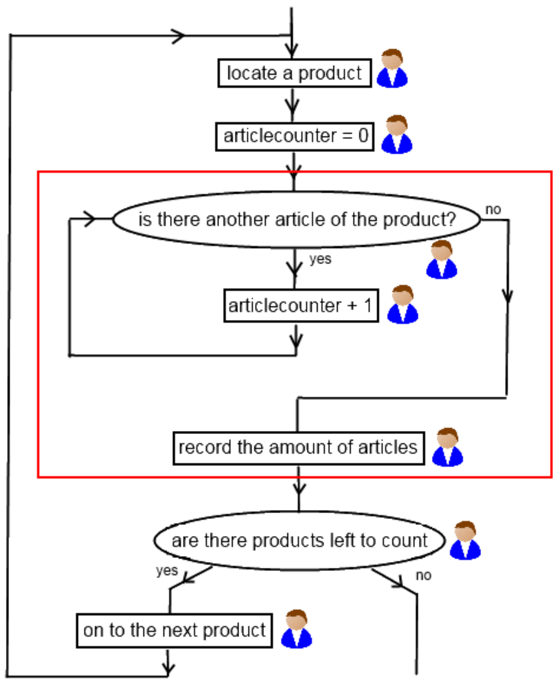

Figure 3.9: Flowchart of an employee's task at stocktake

in figure 3.9. We can see that the employee has to locate a product and then count the articles manually in this process. Manual counting has the possible disadvantage of misscounting. An article might not be with the other articles of its kind or the employee might simply make a mistake. Figure 3.8 shows such a case. The employee might not have noticed that one of the purple boxes is on the upper shelf and thus misscounted the amount of the purple boxes. Since the articles might be placed in a mixed up order, this mistake might not be noticed later, when counting the black boxes.

Our aim is to decrease this error potential by introducing the RIQR tagging system

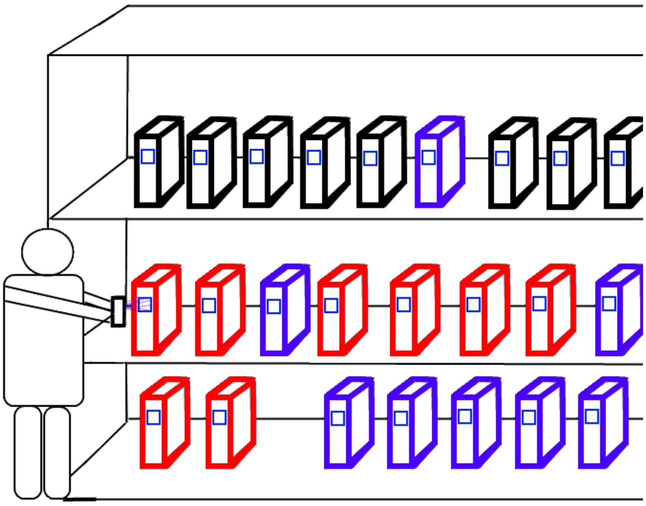

Figure 3.10: Sketch of an employee performing the stocktake with RIQR tagging system

and simplifying the process itself. With the RIQR tagging system each article will have an RIQR code. This RIQR code can then be scanned by a portable scanning device (see figure 3.10). The scanning device will identify and record each item that it scanned. By enabling the employee to scan the items he does not have to count them manually anymore. This should already decreases the error potential due to simple misscounting. However, if articles are not scanned because they are not next to other articles of their kind, scanning them in steps according to the their product affinity will still bear the same risk. Since the RIQR tagging system implements definite instance recognition, it is possible to scan an article countless times without recording its existence multiple times. This means we can break up the single steps of stocktaking and merge them into one big step. Articles would not have to be scanned in product groups anymore. We could now scan the articles in any prefered order without the risk of scanning the article multiple times. This way, it should be less likely to miss articles that have to be scanned, since the employee can simply go along the shelf and scan one article after another. In case the employee is not sure wether an article has already been scanned, he can scan it again. The flowchart to this version of the process can be seen in figure 3.11.

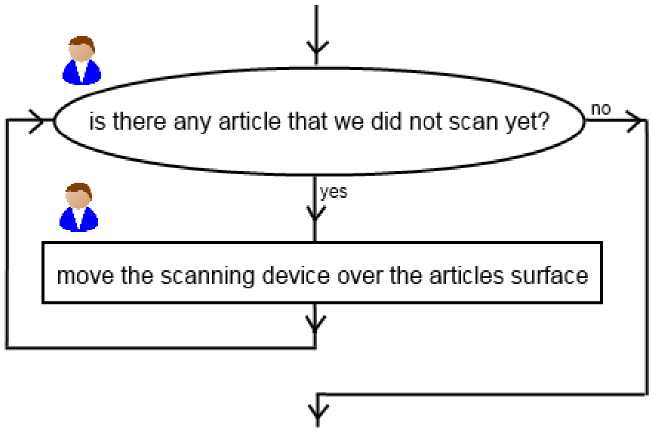

Figure 3.11: Flowchart of the adapted task for an employee performing the stocktake

4 Implementation

In this chapter we will talk about the implementation of the RIQR tagging system, as well as the implementation of IQR overlaying. These topics include our design of the RIQR codes, the means of printing that we used in order to overlay the RIQR codes, the construction of a reading device for RIQR codes and the verification of the IQR overlaying method. Additionally, we will provide a program, which can run a simulation of wether and where to apply IQR overlaying for it to work. We will evaluate the results of this simulation and compare it to the actual results. It should be kept in mind that the implementations in this work are prototypes. They are not meant to be applied like this in real scenarios. Yet, they suffice to reach a proof of concept.

4.1 Content and Structure of RIQR Codes

RIQR codes are meant to allow the identification of the product instance that they are printed on. Each RIQR code should be unique to implement definite instance recognition. Thus we want RIQR codes to contain a product identifier, as well as a unique article identifier. Figure 4.1 shows sample content of a RIQR code and its structure.

For the product identifier, we chose to adopt the EAN, since every article is already

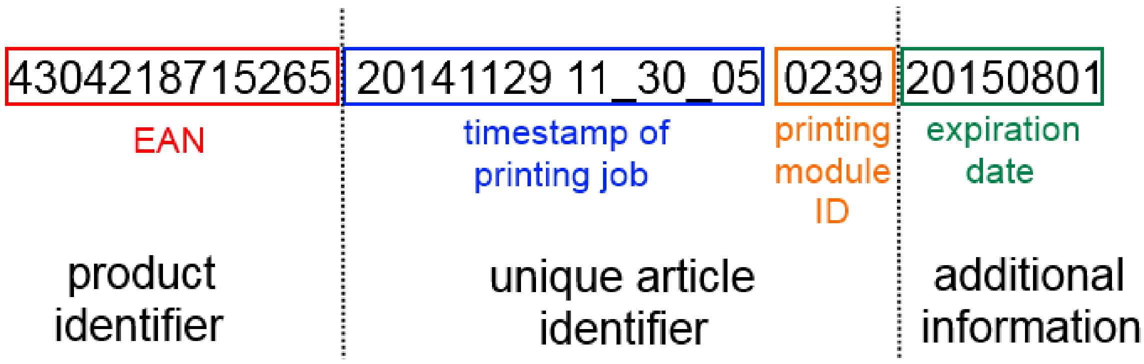

Figure 4.1: Sample of a RIQR code's content and structure

assigned an EAN that clearly identifies the product. This way, all mapping of product identifiers to article names and prices can remain the same.

As for the unique article identifier, we decided to include two kinds of information. We assign each printing module that prints RIQR codes a unique ID. This ID is included into the RIQR code together with the timestamp of the print of the RIQR code. These two values together with the EAN make a unique key under the assumption that there are, at most, ten thousand printing modules that print RIQR codes onto articles with the same EAN and that no printing module can print more than one RIQR code in a

Ink name	Producer	Color under UV
UV-BA	I.P Printing	blue
UV-B	I.P Printing	blue
UV TexJet	I.P Printing	blue
uv aktive Stempelfarbe rot	uv-elements	red
uv aktive Stempelfarbe gelb	uv-elements	yellow
uv aktive Stempelfarbe blau	uv-elements	blue
UV-aktive Leuchtfarbe (UV varnish)	eurolite	blue

Figure 4.2: List of the different UV inks that were tested during research

second. During our research, these assumptions can be made, since we work towards a proof of the concept. However, in a real scenario one cannot be sure that the assumptions will hold. In this case, the design would have to be slightly altered. Additional digits could be added to the printing module ID in order to enable multiple printing modules. The obstacle of printing too many codes in a short time can be circumvented by using a more precise timestamp. If we use for example a timestamp, whose precision is accurate to the millisecond it would be even more unlikely to create any RIQR codes with the same unique article identifier. By checking the printing speed of a printing module, the creation of equal RIQR codes can be avoided.
At the end of the RIQR content, we included the expiration date as additional information. Additional information can be any information and there is theoretically no limit to how much information can be additionally stored. We chose to store the expiration, since it could be an advantage for stocktake to know the expiration date of each article. This way, we might be able to improve the stocktake process, which is one of our motivations.

4.2 Implementing a Printing Method

Choosing a Composition of Ink and Light

In order to print IQR codes, we first decided on which ink we wanted to use and with which light we would make the ink visible. We have tested different inks to find the ink that is most suitable to our need. In figure 4.2 a list of all tested inks is shown.

Since our objective is to read RIQR codes that are printed with the UV ink, we want the ink to fluorescent as bright as possible. This should result in a high contrast to the background, which would allow us to extract the RIQR code from the image and make it readable. For this reason, we applied the different inks to paper in order to check their fluorescence. After the ink had dried, we radiated it with a UV LEDs and took pictures of the fluorescenting marks. The wavelength of the used ultraviolet light is between 380nm and 400nm. Sample pictures can be seen in figure 4.3. We then rated the inks depending on their brightness. The inks that were most fitting at this point were the UV BA ink and the UV varnish. However, since the UV varnish is visible even when not radiated with ultraviolet light, we decided that we would use the UV BA ink

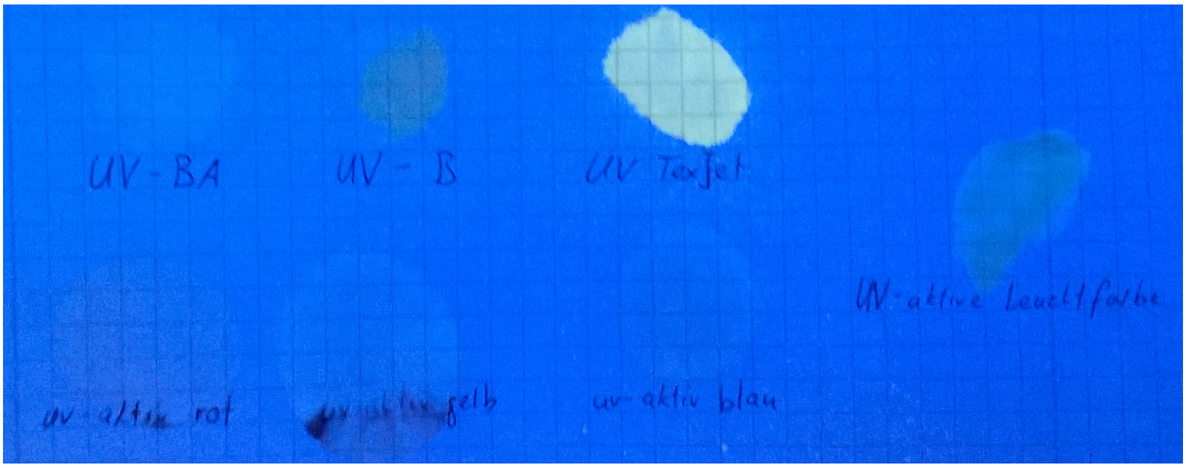

Figure 4.3: Samples of different inks applied to paper and radiated with UV LEDs

from this point on.
After printing the first RIQR codes with UV BA ink, we radiated them with ultraviolet light from the LEDs and took a picture. Analysis of the picture revealed that the LEDs created a spotlight on the RIQR code's center, which caused the center to fluorescent brightly, while the corners of the RIQR code fluorescented only lightly. This resulted in the color of the code's corners to equal the color of the background in the spotlight's center. Because of this, it was extremely hard to extract the code from the background. The top pictures of figure 4.4 show RIQR codes with the spotlight. For this reason, we decided to use a light source that would equally distribute the UV waves and make every part of the code fluorescent with the same brightness.
Our approach to equally distributing the UV waves was the use of UV tubes. The used UV tubes send out a light of the wavelength between 280nm and 400nm. Since the UV inks fluorescent differently under this light, we had to repeat the process of choosing a fitting ink. With the UV tubes the UV TexJet ink looked promising, since the applied ink fluorescented equally for every part of the code. The contrast of the UV BA ink and UV TexJet ink, as well as the difference between LEDs and UV tubes can be seen in figure 4.4. After taking a picture and confirming that the RIQR code could be extracted easier this way, we decided to use UV TexJet ink together with UV tubes to print and radiate the codes.

Attempting to Print RIQR Codes With Inkjet Printers

Our first idea of how printing could be achieved was printing the RIQR codes with an inkjet printer. We know that UV inks contain nanoparticles that make the ink fluorescent. This bears the risk of plugging the print head with these particles in case they are not fine-grained enough. For this reason, we wanted to use printers that have a low resolution and a robust print head.
The printer we tried was the Hewlett Packard PaintJet XL. Since this printer was built in 1989 the resolution is only 180dpi and its parts are very robust, which qualified the printer for tests with invisible ink.

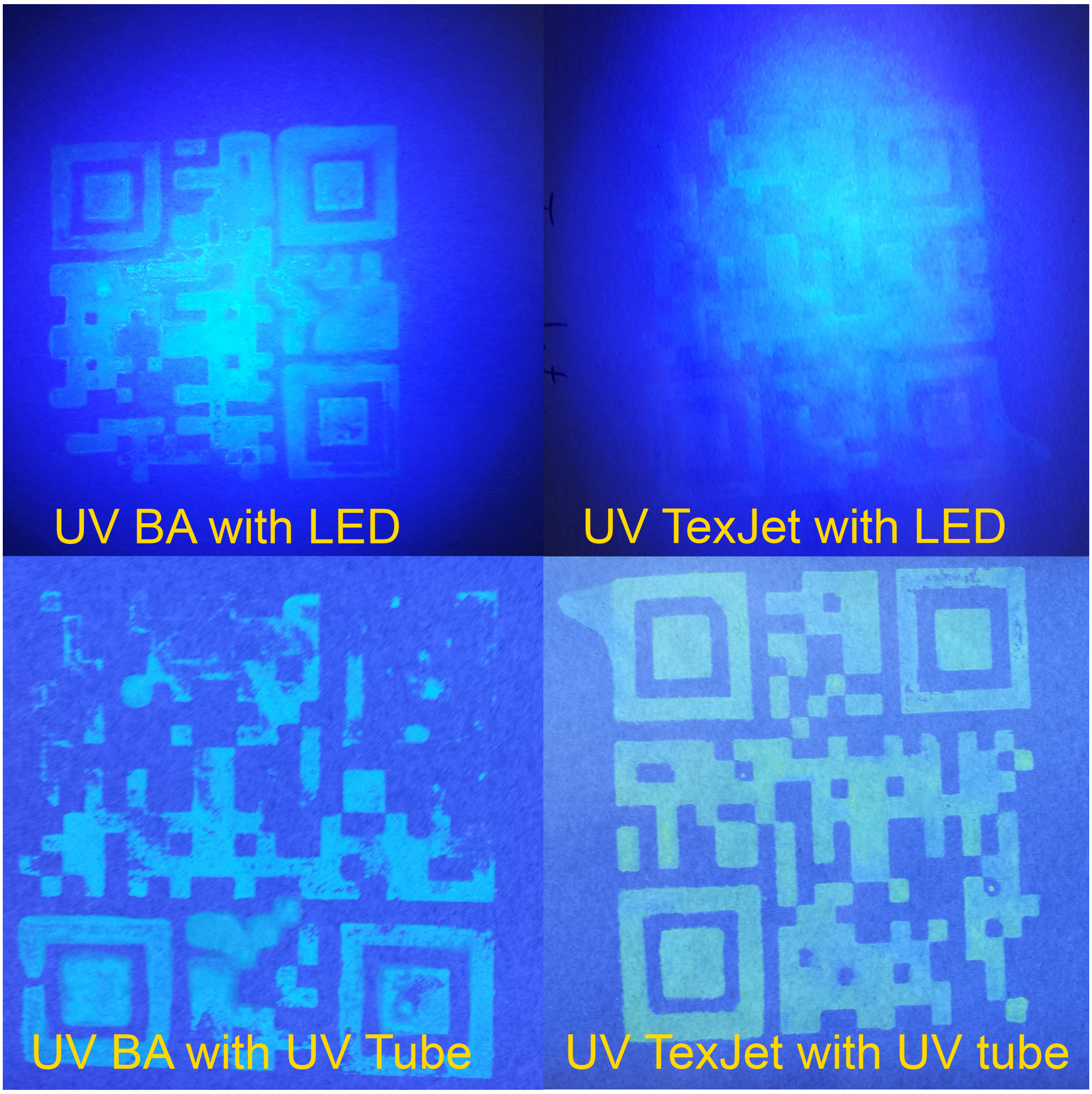

Figure 4.4: Samples of UV BA and UV Texjet ink radiated with ultraviolet light from LED and UV tubes

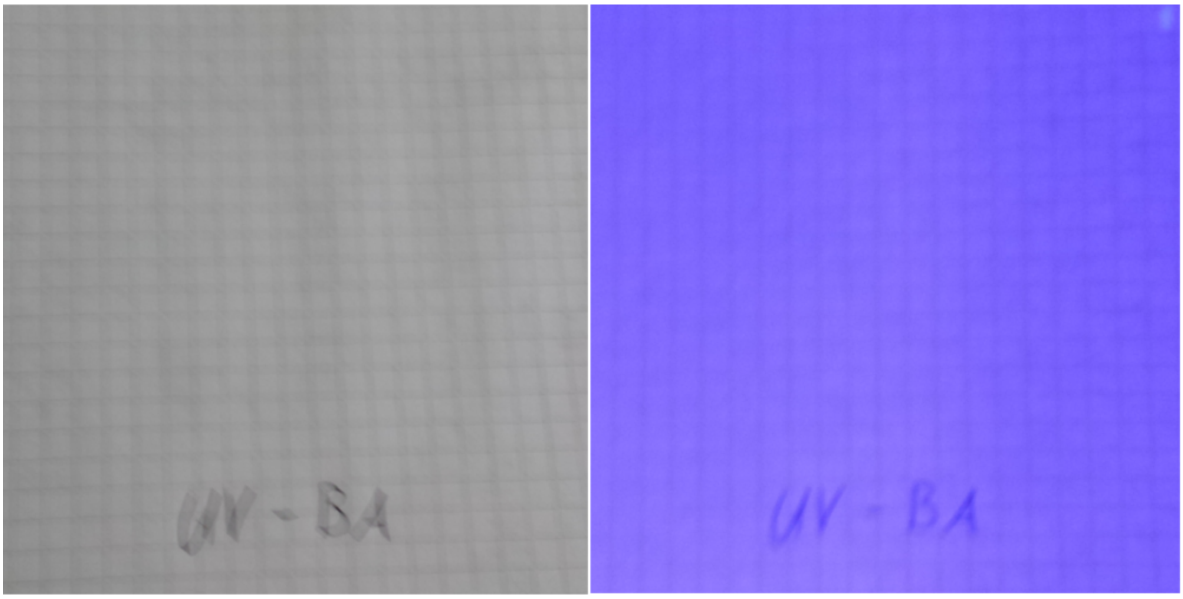

Figure 4.5: Picture of the failed print with and without radiation by ultraviolet light

In order to prepare the printing, we first removed the cartridge for black ink and cleaned it from any ink leftovers. Then, we filled the UV ink into the cartridge and placed it back into the printer. After that, we created a black RIQR code and printed it using the printer. Since we replaced the black ink with our UV ink, the code was printed with the UV ink. The resulting print was invisible, as expected. However, radiating the paper, on which we printed the RIQR code, with ultraviolet light did not make the ink fluorescent. A picture of this failed print can be seen in figure 4.5.

The missing fluorescence means that somehow the fluorescenting nanoparticles were not printed onto the paper. There must have been some filter that removed the nanoparticles from the ink, so that only the ink without the particles was applied to the paper. After this failed attempt, we decided to try stamping the ink using a ink pad. We poured ink onto a new ink pad and used some stamp to see whether the ink could be applied this way. The behaviour of the ink in this case was exactly the same as with the printing. This means that the pad must have filtered the nanoparticles from the ink. Inside the printer's cartridge, there also was a sponge which was similar to the ink pad. It is likely that sponges do filter the fluorescenting nanoparticles from ink.
Knowing this, we attempted to open one of the cartridges to remove the sponge and circumvent the filtering of the nanoparticles. After opening the cartridge, removing the sponge and closing the cartridge again, we retried printing an RIQR code with the UV ink. Because we were not able to sufficiently seal the cartridge after removing the sponge, no ink left the cartridge and we were not able to apply ink via printing.

Stamping RIQR Codes

The issue with stamping the UV ink was the ink pad, which filtered the nanoparticles from the ink, so that the fluorescence was lost. If we were able to apply the ink to the

stamp directly, it should be possible to apply the ink together with the fluorescenting nanoparticles.
For this reason, we created a 3D model of a RIQR code stamp and printed it with a 3D printer using plastic. We applied the ink by painting it onto the stamp with a brush. Then, we stamped the RIQR code onto paper, which we had placed on a table. The result can be seen in 4.6. The ink fluorescented when radiated with ultraviolet light, but the quality of the stamp was rather bad and the RIQR code was not printed completely. The problem was that the print area's surface was not even enough. This way, parts of the print area were not properly affixed and the ink was not applied properly to the paper on these spots. We confirmed this by trying to stamp on slightly soft surfaces like skin or paper towels for example that would adapt their surface to the stamp when stamping.

We circumvented this issue by printing RIQR stamps with higher quality and finer

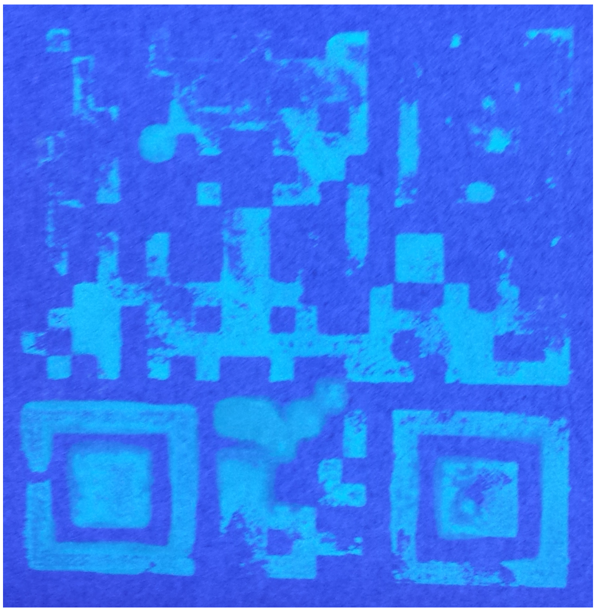

Figure 4.6: Result of the first stamping attempt

print area. This way, the stamp would adjust its print area to the paper's surface and the ink would be applied correctly. After applying ink to the print area, we again tried to stamp the RIQR code onto paper. This time the printed RIQR code was completely printed. However, there were still spots, on which we applied too much or too little ink to the stamp, which prevented us from stamping a well readable code. The print can be seen in figure 4.7 on the left side. To solve this issue, we needed to find a way to equally apply the ink to the print area without using too much ink. We achieved this by filling the ink into a spray can and spraying the ink onto the stamp. It takes about spraying three to four times to apply sufficient ink equally over the print area. A stamp that was printed with this technique can be seen in figure 4.7 on the right side.

Stamping is not an optimal solution for printing RIQR codes. There are two reasons for this. One reason is that all RIQR codes are unique, meaning that according to our concept each stamp would only be needed once for the stamping of a single article. We need printers that are able to print different codes efficiently. Creating a new stamp

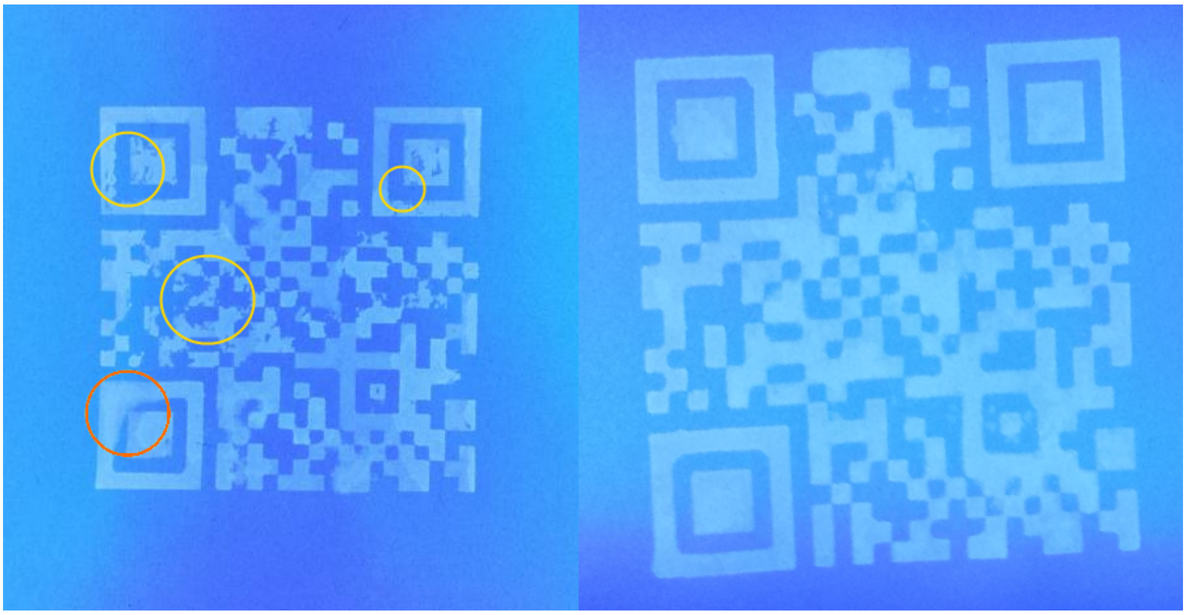

Figure 4.7: Stamp with ink applied via brush (left) and via spraying (right); left code blurs (orange) or is not stamped correctly (yellow)

for each print cannot be considered efficient though. The other reason is that we were not able to print on normal article packings. Because most articles packings are coated, the ink would not stick to the packing and the stamp quality would be reduced drastically. Since stamping RIQR codes is only the very first prototype for printing of RIQR codes, it is sufficient because it allows us to print codes, which we can work with when researching the readability of RIQR codes in general.

4.3 Creating the RIQR Reading Device

Approaching the Reading Process

Before writing software that would automatically drive a device to reading a RIQR code, we started by manually performing this task. This way we could collect experiences about factors that would influence the readability in order to adapt the process in a way it would work.
After printing a RIQR code, we started by taking a picture of that code with a mobile phone camera. When taking the picture, we made sure to take it in a dark room with only the UV tubes serving as a light source because we expected this to deliever the maximal contrast between the code and the background. Then, we opened the picture in GIMP and tried processing it with image filters in order to extract the RIQR code. After processing, we inputted the image into a QR code reading program to find out whether the code was readable.
We tried different sequences of filters for the purpose of extracting the RIQR code. Testing these sequences for different pictures with slightly different ambient light and orientation of the code revealed that one sequence delievered a satisfying result for the

most pictures of the tests. This filter sequence is described in figure 4.8. An original picture and the extracted RIQR code for the described sequence can be seen in figure 4.9. It should be noticed that the judgement of whether a result was good or bad was based on the intuition wether a code looked well readable.

We tried reading the RIQR codes by inputting the processed image into a simple pro-

Filter	Parameters	Effect
Color Balance	midtones magenta green 40, preserve luminosity	make code brighter
Colorize	Hue 180 Saturation 100 Lightness -45	increase contrast between code and background
Threshold	Lower Bound 175 Upper Bound 255	extract white code from black background
Invert	none	make code black and background white
Posterize	none	make sure that black and white are only colors

Figure 4.8: Filter sequence that succeeded in extracting the RIQR code for most pictures

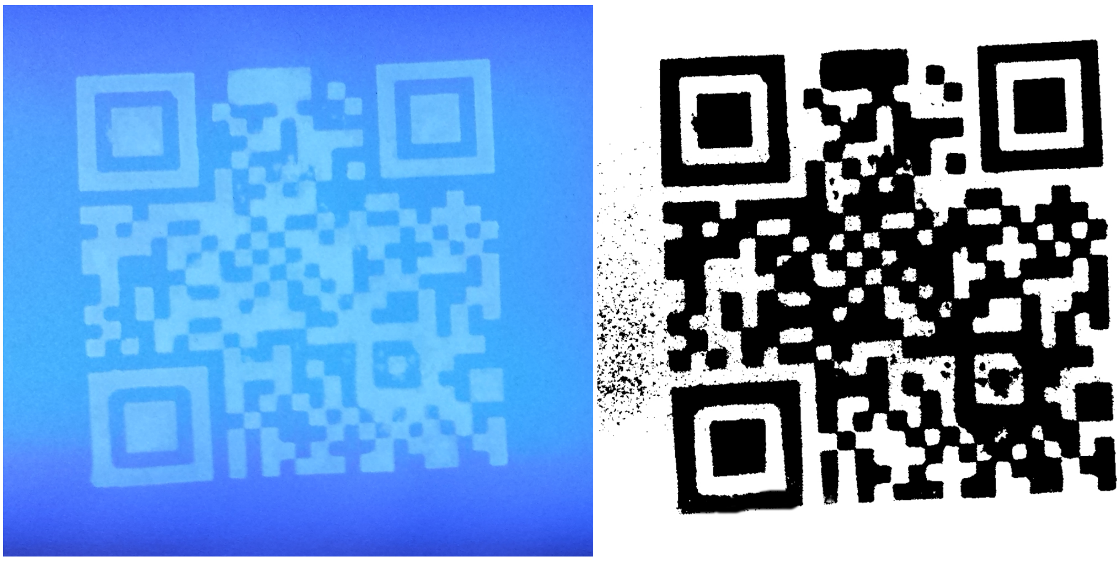

Figure 4.9: Picture of RIQR code before and after extraction of RIQR code

gram. The program uses google's QR code libraries in order to detect the RIQR code in the image and decode it. Whether a printed RIQR code was considered readable, thus depended on whether the program detected a QR code in its processed image.

The result was very negative at first. We took single pictures of RIQR codes that looked well printed and processed them according to the described filters. Even though the resulting images looked well readable the program never detected any code. We tried fixing critical parts in the processed images that seemed slightly incorrect and after

some minor fixes the codes were usually detected. Because of this, we tried reading one of the images that was processend with the filter sequence off the screen via some QR code application for smart phones. After a short time the application succeded in reading the RIQR code without issues. We figured that the problem was that single pictures of a RIQR code do not suffice for reading, since there might be details that are not always captured correctly. Smart phone applications for reading QR codes usually keep capturing images and analyzing them until they find a QR code. Because of the hand motion, which affects the phone and with it the camera, multiple slightly different pictures of the same code are taken, until one is found, on which the code is well readable.
After solving this issue, our aim was to create a reading device that behaves similar to the smart phone application. We would not approach reading by using a single picture but rather keep capturing slightly different pictures of the same RIQR code until it could be read. The use of UV tubes has been advantageous for this. When carefully inspecting the light that is created by the tubes, it was revealed that the UV tubes do not create continous light, but rather waves of brighter and darker ultraviolet light. This would allow taking pictures with slightly different ambient light, depending on when the picture is taken. This way, the variety of the different possibly taken pictures would increase, making a good image more likely, where the code can be successfully read, to be available.

Building the Reading Device

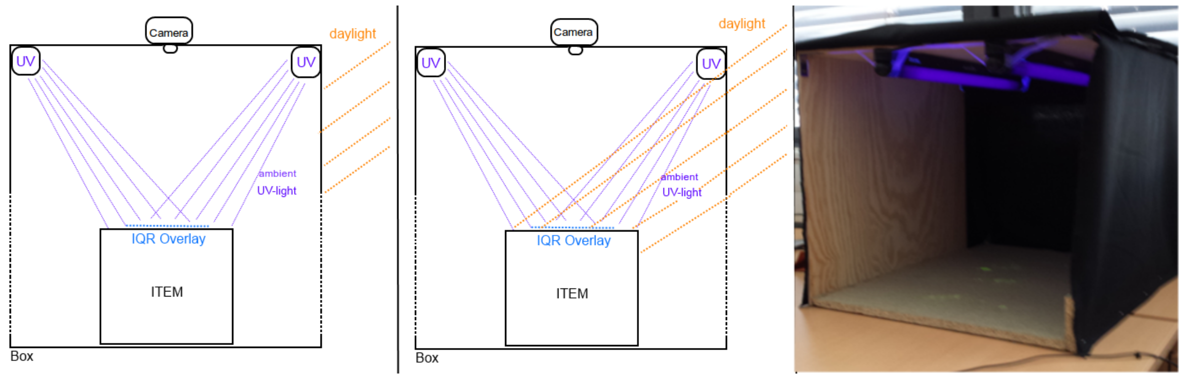

Figure 4.10: Sketches of the reading device and picture of the first prototype

We require a reading device to be able to take a picture of a RIQR code, while radiatiating the code with ultraviolet light. Additionally, there needs to be some computer, which needs to be able to process the image and decode the RIQR code.
For our implementation of the reading device, we built the blackbox from the checkout concept. For usability reasons, we splitted off the computer from the main reading device by using a laptop that is connected via USB to the blackbox. Sketches of the blackbox, as well as a picture can be seen in figure 4.10.
Our blackbox contains about 0.25m^2 of empty space so we can easily fit articles inside and move them around. The walls, as well as the ceiling are made of wood. The front

and the back are left open in order to enable placing articles inside the box. We have covered the front and the back with black cotton, so that the interior can be darkened if necessary (4.10 left). We chose relatively thick cotton so that only little to no light could enter the blackbox. The cotton can be lifted up in order to let in the ambient light from outside (4.10 middle). The UV tubes were affixed on the ceiling as light source. Between them, we created a small hole. On top of the box we placed a webcam or a smart phone with the camera lens directed through the hole. This way, the whole interior can be radiated with ultraviolet light, while the camera takes pictures.

Developing the Checkout Software

The software we implemented for reading RIQR codes is for the software required for the checkout use case. The code that is necessary for general reading of RIQR codes is capsuled in a seperate library, which is used in the checkout software. A software that would fulfill the requirements of the stocktake use case could be created using this library as well. Since one use case is enough for the desired proof of concept, we did not implement a software for the stocktake.

The general function of our checkout software is to start registering articles for a customer and list them together with their prices until it is the next customer's turn. To achieve this, the program takes a picture of the blackbox's inside, through which the articles that customers want to purchase will move. For taking pictures, we used the Java Media Framework API. Webcam or smart phone camera are both used as a normal webcam for this. The pictures are analyzed by applying the filters described and scanning the resulting image for RIQR codes. The scanning is done by using google xzing's QR code libraries, which provide methods for extracting multiple QR codes from pictures. If one or more codes are found, it is checked for each code whether the belonging article registered already. This can be determined because we have implemented definite instance-recognition for RIQR codes. By simply remembering the codes we have already read, we can avoid registering articles twice. If an article was not registered already, it is added to the already scanned articles. After identifying all codes that are contained in an image we go on by taking another one. This process is repeated until all articles of a customer have been registered. This information is given via a simple click input. After that, the list of registered articles and their prices is displayed and the program moves on to the next customer. A flow chart of this program is displayed in figure 4.11.

The proper use of filters and filter sequences is essential for our purpose. For this reason, we used JH LAB's image filter libraries, which looked promising for our needs. The filters are not optimized, but all filters we required are implemented. Since time itself is not a relevant factor for the proof of our concept, there is no issue with this implementation.

The application of our filter sequences is described in figure 4.12. First, the program takes the taken picture as input. For each filter sequence, the program first configures the filters according to our needs. In this step, parameters as for example the exact threshold values for threshold image filters are set. Then each filter of the sequence is applied in the defined order. The duration of this step, as well as the numbers of filters applied vary depending on which filter sequence is applied. The result of this process

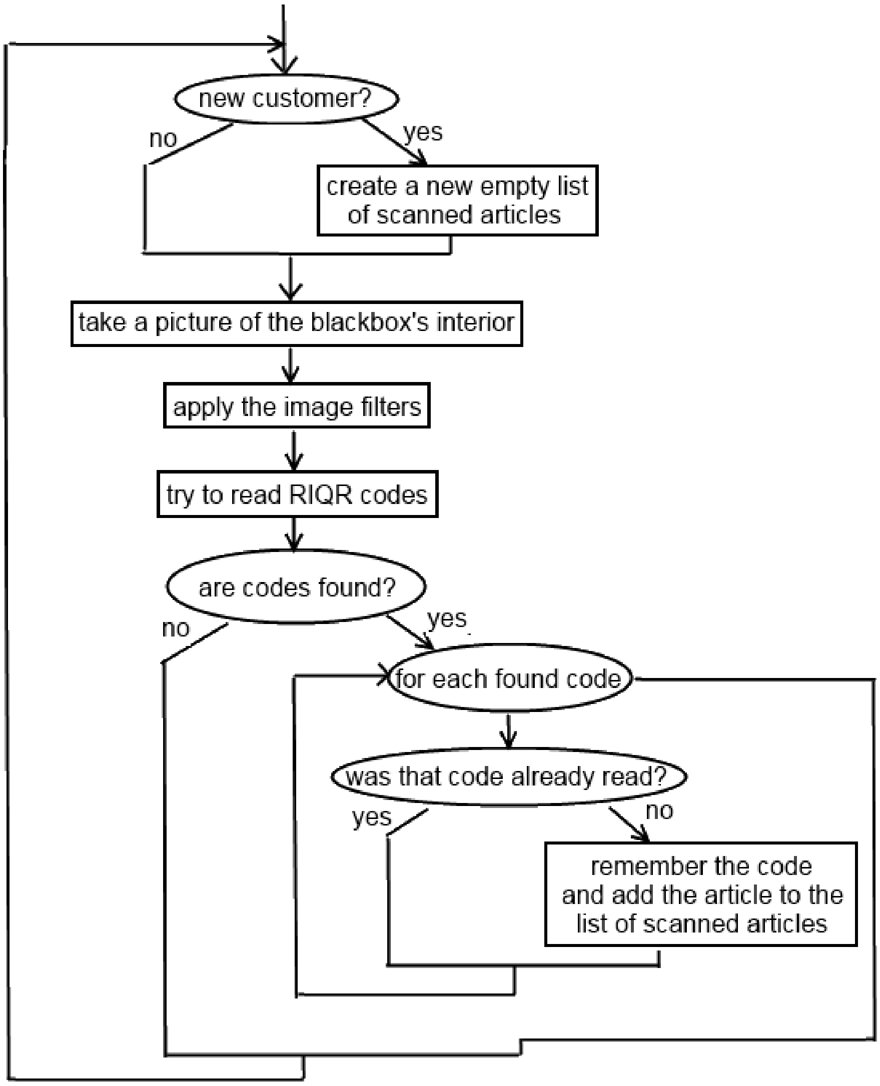

Figure 4.11: Flowchart of the reading program for the checkout use case

is the processed image, from which we hope to be able to extract a RIQR code.

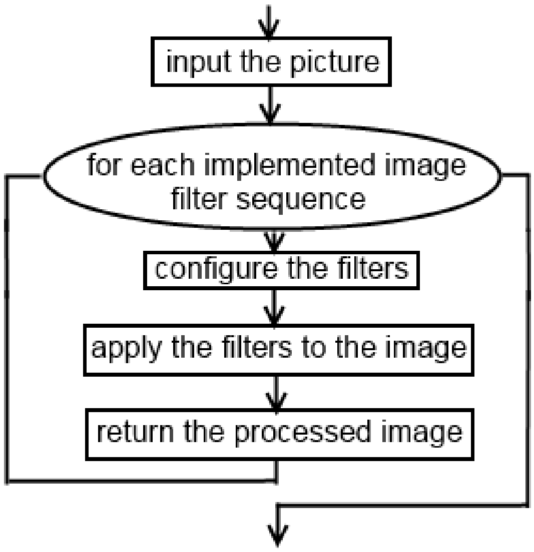

Figure 4.12: Flowchart for the implemented application of image filters

4.4 Verification of IQR Overlaying

Since we want to reach a proof of our concept, we want to verify our developed method of IQR overlaying. For this reason, we wanted to print RIQR codes onto surfaces with different properties in order to find out on which surfaces the overlaying would work. We were not able to print on different materials, like for example coated cardboard, which is the basis of many product packings. In contrary, we were able to print on paper, normal cardboard and cloth. After some observation of product packings under ultraviolet light, we came to the conclusion that the coating of the packing might only be relevant for printing. That is, because the coated surface seemed to behave just like an other not coated cardboard surface under ultraviolet light. For this reason, we tried verifying the overlaying method with uncoated surfaces instead of coated surfaces, since we expect the result to be the same.

The verification itself was performed by putting the surface that we printed on into the blackbox, while our checkout software was running. A well printed code should be readable if the surface is suited for IQR overlaying. This means that if the software recognized an article inside the blackbox, the overlaying must have worked on the particular surface. By showing the applicability of IQR overlaying on different surfaces, we then want to reach a proof of concept. A table of the surfaces we tested can be seen in table 4.13.

For paper and cardboard, we can say that IQR overlaying is generally applicable according to our test results. Most surface colors did not influence the reading quality. There were some exceptions, but it might be argued that these tests failed due to a lack of printing quality. A sample picture can be seen in figure 4.14 on the left side.

Material	Color	Motive	Reading successfull?
paper	white	none	yes
paper	gray	none	yes
paper	green	none	yes
paper	blue	none	yes
paper	light green	none	yes
paper	yellow	none	yes
paper	orange	none	yes
paper	brown	none	no
paper	black	none	no
paper	gold	none	yes
paper	pink	none	yes
paper	sand	none	yes
paper	red	none	yes
paper	purple	none	no
paper	white with red dots	none	no
paper	green with white dots	none	no
cardboard	gray	none	yes
cardboard	brown	none	yes
cloth	white	none	yes
cloth	red	none	no
cloth	green	none	no
cloth	blue	none	no

Figure 4.13: Surfaces that were tested for verification of IQR overlaying

Our results give strong evidence that printing on paper with a motive poses more of an issue. We were not able to read RIQR codes from these surfaces. A sample can be seen in figure 4.14 in the middle. As for the reason why we could not read the RIQR codes, we believe that the high contrast between the dots and the ground color made it hard for the image filters to distinguish between the fluorescenting code and the background. In order to be able to read IQR codes from such surfaces, we would need to invent new image filters that can handle such surfaces. Regarding cloth, all the tests failed. However, we believe this to be a consequence of bad printing quality. Even though the prints seem to look good, the edges of the printed codes are not clear enough. This is a result of the cloth soaking up the ink and thus not keeping the original form in which it was applied. Sample pictures can be seen in 4.14 on the right.
It can be concluded that IQR overlaying can be a valid method for information hiding.

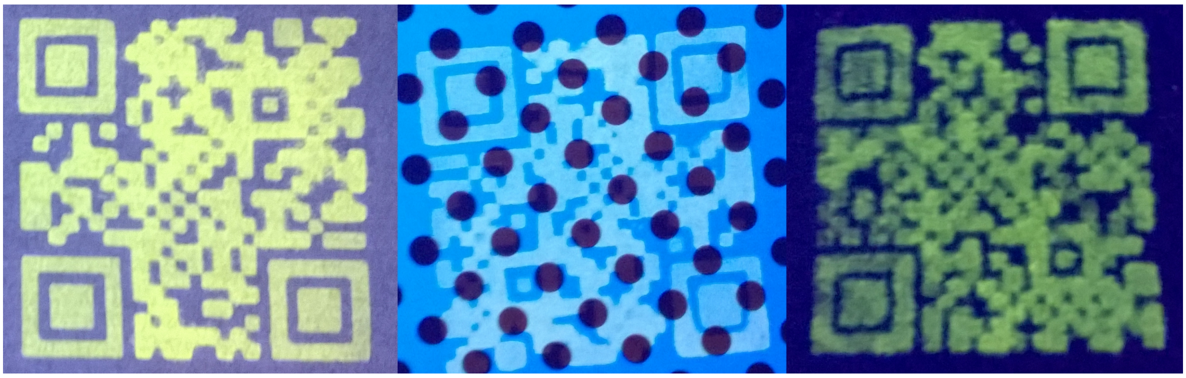

Figure 4.14: Image samples of failed verification tests; paper(left), paper with motive(center), cloth(right)

However, in order for this to be true, suitable spots on packings, on which no background motive would interfere, need to be found. Alternatively, advanced image filters could be designed, so that backgound motives will not pose an issue anymore.

4.5 Implementing a Simulation Program for IQR Overlaying

In order to solve the issue of background motives interfering with our IQR overlaying, one approach is to find suitable spots for placing our RIQR codes. For this reason, it would be desirable to have a way of telling what those spots are. In this chapter, we will design a simulation that should find those spots. The simulation will try virtually overlaying a RIQR code on different spots. If that overlay actually succeeds, that spot might be a suitable spot for overlaying the real RIQR code.

Creating a Virtual Overlay for RIQR Codes

Before we can virtually overlay RIQR codes onto articles in pictures, we first need to find a way of doing so. Our approach is to create an difference image from the differences in ARGB values of exactly the same picture with and without the overlayed RIQR code. This way, we get the RGB values change that is caused by the fluorescenting code. By then adding this difference image to an article in another picture, we should be able to get the same or at least a similar result to physically overlaying the code and then taking a picture.
For creating the virtual overlay, we made sure to first print a very fine printed RIQR code on a plain white paper sheet. Then we fix a camera facing the sheet with the code on it, while radiating the sheet with ultraviolet light. Then we take a picture and replace the sheet with a plain sheet in order to take another picture without any IQR code. After taking both pictures, we substract the picture without a code from the one with the code. The resulting picture is the difference image we require for our simulation. The picture with the code and the difference image can be seen in figure 4.15.

We have tried manually overlaying the difference image onto white paper. This was

Figure 4.15: Original picture and difference image

done by adding the ARGB values of the difference image and the ARGB values of the image of the white paper radiated with ultraviolet light. The resulting image was the picture of the radiated white paper with a fluorescenting IQR code on it. After confirming that the code could be extracted from the created image, we continued by virtually overlaying the code onto pictures of different surfaces. In figure 4.16 the results of virtually overlayed codes onto different surfaces can be seen.

The virtual overlaying technique worked for different surfaces, as we expected. When trying to use the virtual overlay for overlaying the IQR onto red fabric, we encountered

Figure 4.16: Images of virtual overlaying onto different surfaces

a problem. The virtually overlayed IQR code was not lightly blue as could be expect, but black. This is not realistic, since the fluorescence of our ink would not change this drastically because of the ground surface. The reason for this issue are the ARGB value of our difference image. When creating it, the original picture of our difference image was completely blueish, while the IQR code fluorescent in a slightly brighter color. This made the difference image contain relatively large R and G values. These caused an overflow of the R channel, which originally contained a very high value for red cloth. This set the R value to a very low one and resulted in the blackish color. We realized that we needed different virtual overlays to avoid this overflow. That is why we decided to create difference images with different dominant color channels. This way, we would be able to avoid the unintended overflow of the color channels. Figure 4.17 shows the virtual overlay with and without overflowing the R channel.

After creating virtual overlays for avoiding overflows of the different color channels

Figure 4.17: Virtual overlay with and without overflowing the R channel

and checking wether those overlays actually created readable IQR codes, we could then advance to automating this simulation method by writing a program that would process this task for us.

Development of a Simulation Program

In this chapter, we will develop a program, which will be able to automatically run the described simulation method for finding suitable spots for overlaying IQR codes. The automated simulation process can be divided into two parts: first the virtual overlaying itself and second the process of reading the overlayed code. Since the latter is handled in exactly the same way as the reading of an actually printed code, we will only discuss the virtual overlaying in detail.
In order to find suitable spots for overlaying the IQR code, we want to try placing the overlay onto different possible spots on the target surface. One way of doing so is to try every possible position of x and y coordinates of the surface's picture. Doing this would result in two downsides. The first would be a relatively huge computation time that would be required for all the differnt positions, while only few would be interesting, since for small differences in position many resulting images deliver similar results. The other downside is the fact that in case of a surface that does not fill up the whole picture, we might place the overlay in the background and not the surface. For these reasons, we decided to only run the simulation for some positions with a certain distance to each other. The distance is the horizontal or vertical distance between adjacent positions and can be specified in pixels. This way, we can reach a trade-off between the computation time required and the probability to find a certain spot on a surface.
There is the possibility of placing an overlay in the background in case that a surface does not fill up the whole picture. Avoiding this would require recognizing the actual surface that we want to place our IQR code on. Manually sorting out the images of no interest after the simulation only consumes little time. Since ultimately each surface should only need one simulation in total, the additional time spent manually is rather small, while introducing object recognition into the program would probably involve major efforts. The program itself shall assist in overlaying IQR codes and proving the concept of IQR overlaying. For this reason, we will take in account the additional effort of manually filtering the resulting images over the effort of introducing object recognition.
A flowchart of the program that we developed under these circumstances can be seen in figure 4.18.

Our program starts by first taking a picture of the surface we want to base the simulation on. Obviously the surface must be placed into the blackbox where the surface is radiated with ultraviolet light. The ultraviolet light is needed to create the same surrounding as the surrounding back from the creation of the difference image. For each position that is considered interesting, we run a seperate simulation. Interesting positions are those with the specified distance to adjacent positions starting from the top left of the picture. Each seperate simulation first chooses the next position to consider. It then adds the values of the corresponding color channels of the original and the difference image. This means the values of the color channels of the pixel at the considered position are increased by the top left pixel of the difference image. The pixel to the right gets its color channels increased by the values of the pixel to the right in the difference image and so on. This way, we get the picture of the surface with the IQR code, since we add the fluorescenting IQR code to the picture. Then we run the IQR code extraction on the image, just as in the reading software. If this is successfull, we consider the position to be suitable for an IQR code to be placed there. The position is

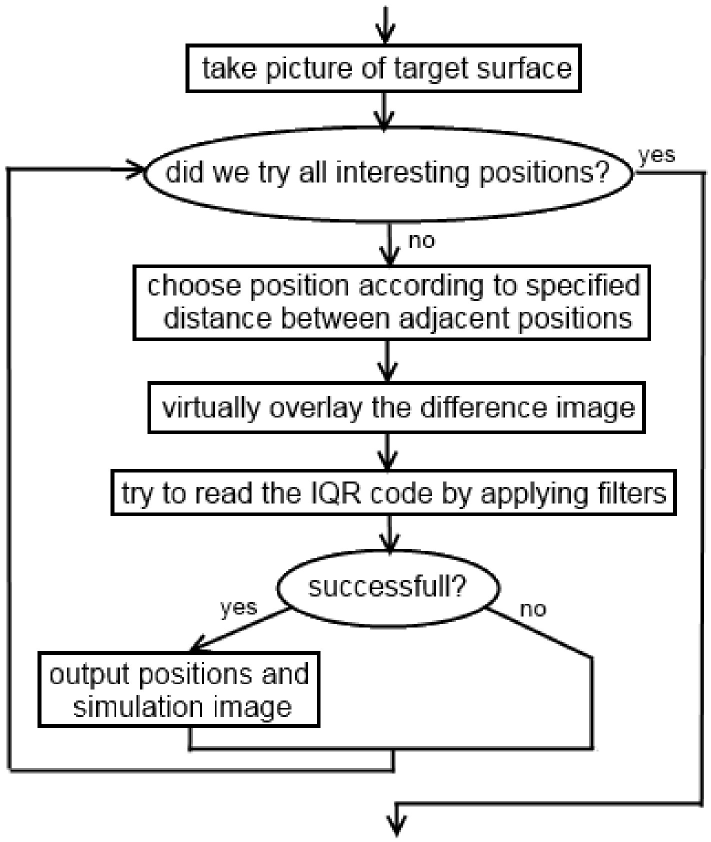

Figure 4.18: Flowchart of the simulation program for IQR overlaying

then output and the simulated image is stored. When the simulations on all positions are finished the program terminates.

4.6 Verification of the Simulation

After designing a simulation method and implementing it, we want to also research whether the simulation is really applicable. To achieve this, in this chapter we will compare the results of our simulation with the results of reading printed IQR codes. The simulation, as well as the reading are performed exactly the way that we described before. The results we received from experimenting with different surfaces is shown in figure 4.19.

It can be seen that for paper in general the results between the normal reading and the simulation do not differ by much. In the cases where we were not able to successfully read the IQR code, it can be argued that this was due to bad printing quality. As for the paper with the dot motive, it might be hard to actually get a readable result by manually stamping codes. After running the simulation, we tried to stamp the code onto the spots that were suggested by the simulation. However, we believe that utmost precision is needed to get a good print in exactly the right spot with manual stamping. Being able to use a special printer for this task might allow printing well readable

Material	Color	Motive	Simulation successfull?	Reading successfull?
paper	white	none	yes	yes
paper	gray	none	yes	yes
paper	green	none	yes	yes
paper	blue	none	yes	yes
paper	light green	none	yes	yes
paper	yellow	none	yes	yes
paper	orange	none	yes	yes
paper	brown	none	yes	no
paper	black	none	yes	no
paper	gold	none	yes	yes
paper	pink	none	yes	yes
paper	sand	none	yes	yes
paper	red	none	yes	yes
paper	purple	none	yes	no
paper	white with red dots	none	yes	no
paper	green with white dots	none	no	no
cardboard	gray	none	yes	yes
cardboard	brown	none	yes	yes
cloth	white	none	yes	yes
cloth	red	none	yes	no
cloth	green	none	no	no
cloth	blue	none	no	no

Figure 4.19: Results of the reading and the simulation

codes, enabling successfull reading. For cardboard, the simulation seems to be close to the real printing for the tests we did. Lastly, comparing the results for cloth we found out the simulation delivers a slightly more positive result. Stamping onto cloth posed a difficulty, since we could not easily get good quality prints. We believe that the gap between simulation and reality can be narrowed by using the right printing techniques.

5 Results

In this chapter we will summarize the results of this work, as well as the insight we have gained through them. In chapter 5.1 we will cover the process of choosing an appropriate pair of ink and ultraviolet light for IQR overlaying. Chapter 5.2 and 5.3 cover the printing and reading methods that were used during the course of this work. Definite instance-recognition and its effects are covered in chapter 5.4. Chapter 5.5 covers the factors influencing the reading process. In chapter 5.6 we will cover the evaluation of IQR overlaying. Lastly chapter 5.7 will cover the issues with complex surface patterns interfering with IQR overlaying.

5.1 Appropriate Pairs of UV Ink and Ultraviolet Light

In order to find an ink that performs well with IQR overlaying, we have tested different inks. The tested inks can be seen figure 5.1.

The ink that we judged to be the most appropriate was *UV-TextJet*. The *UV-TextJet* ink stands out, because of its bright fluorescence when radiated with ultraviolet light. Another important advantage is that the dried ink is only hardly noticable. The performance of *UV-B* and *UV-BA*, as well as the stamping inks' performance was not sufficient in terms of fluorescence. The *eurolite UV varnish* radiates brightly when radiated, but lacks invisibility when not radiated with ultraviolet light. Therefore, the varnish is no appropriate choice for printing invisible codes.

As for the light source required for reading IQR codes, we discovered that most inks respond differently, when radiated with ultraviolet light of different wavelength. *UV-TextJet* excels at fluorescenting, when radiated with the light from an UV tube, while *UV-BA* performs better, when radiated with light from UV LEDs. Most of the inks that perform poorly with LEDs, improve when used with UV tubes. UV tubes evenly send out a wide specturm of UV waves, for which reason it is recommended to use them

Ink name	Producer	Performance
UV-BA	I.P Printing	hardly readable
UV-B	I.P Printing	hardly readable
UV TexJet	I.P Printing	well readable
uv aktive Stempelfarbe rot	uv-elements	hardly readable
uv aktive Stempelfarbe gelb	uv-elements	hardly readable
uv aktive Stempelfarbe blau	uv-elements	hardly readable
UV-aktive Leuchtfarbe (UV varnish)	eurolite	well readable

Figure 5.1: List of the UV inks that were tested during research

as the light source, especially in cases in which it is unknown to which wavelength the used ink reacts best. We furthermore recommend to refrain from using UV LEDs, since they distribute the light unevenly, which makes reading of codes harder.

5.2 Printing Methods

In order to acquire well readable codes, we have used different approaches to printing the IQR codes. We have found out that stamping using inkpads is not a possible method, since inkpads filter the fluorescenting particles from the ink. Inkjet printing with regular home printers is also not possible for the same reason. Sponges or sponge-like objects in general turn out to often filter the particles, so ink coming into contact with them should be avoided. The printer that showed us this behaviour is the Hewlett Packard PaintJet XL. On the other hand, when manually applying the ink to the stamp, stamping of codes does succeed. The particles inside the ink are preserved. However, since the retail scenario requires unique codes and thus a new stamp for each code, stamping would be too inefficient to be of any use in practice. We still recommend stamping as a printing method for prototypes, of which only a small set of different codes need to be printed. When implemented in retail, IQR overlaying requires a way of printing diverse codes without overhead of efforts. Printers that can efficiently print UV ink while maintaining high quality are needed.

5.3 Reading IQR Codes

Given a pair of an IQR code printed with UV ink and a compatible source of ultraviolet light, it can be tried to read the IQR code. Common QR code scanners like smartphone applications cannot read IQR codes, because of the different appearance of the codes. An adapted way of scanning is needed. It is possible to process the picture of the IQR code with image filters between taking a photo of the code and reading the code from it. We developed a software that takes a picture of the radiated IQR code and then uses image filters to convert the image of the blue code to an image of a black code on white background. After the processing, we are able to read the code with a normal reader. The required processing for successful reading depends on the ambient light and the surface from which we read. There are many possible ways of filtering the image of the IQR code in order to get a well readable code, while some are more or less efficient for different ambient lights or surfaces. We found out that with a small set of different processings, it is possible to read the codes from most of the tested surfaces.

5.4 Definite Instance-Recognition

In our implementation of RIQR overlaying, we included definite instance-recognition as a key feature. Using definite instance-recognition, it is possible to create an advanced inventory, in which each dataset can be matched to one article. It turns out that this

Factor	Effect
material	influences printing quality
surface color	influences contrast between code and background
surface pattern	interferes with filtering the image
code size	small size requires finer reading
ambient light	influences brightness of code and surface

Figure 5.2: Effects of different surface factors

feature does not only extend the inventory, but also plays an important role for the automized reading process at the checkout. As we are trying to speed up the scanning at the checkout, we implemented the simultaneous scanning of multiple codes at a time, while articles move through the blackbox at the checkout. This normally requires keeping track of the codes that move through the blackbox, since we do not want to scan the same article twice. The reading method we implemented does not prevent such duplicate scans. Since definite instance-recognition makes each code unique, duplicates are easily recognizable and can be ignored. We thus highly recommend to use some form of definite instance-recognition, when implementing IQR overlaying in scenarios, where duplicates are to be avoided.

5.5 Influencing Factors for Reading

While researching the readability of IQR codes, we found a set of factors that influence a code's readability. The factors alongside their effect can be seen in figure 5.2.

The material of a surface is mainly a printing matter. Depending on the material, the quality of the print increases or decreases. Unwanted effects like blurring of the printed code can be caused by coated surfaces for example. This can decrease the readability, so the materials should always be taken into consideration, when printing. The surface color influences the contrast between the fluorescenting code and the background. Depending on the color, the contrast can be high or low. The lower the contrast the more precise is the filter sequence that will be needed for a successful extraction of the code. The surface pattern behaves similar to the surface color. When a pattern is present, there are many different contrasts between the code and the different parts of the background pattern. In cases where these contrasts differ by much it is harder to extract the code. Therefore, we recommend to choose spots on surfaces, on which the pattern is relatively simple. The size of a code does influence the reading of the extracted code rather then the extraction. Assuming that a code can be successfully extracted, the size of a code determines how fine the reader must work for the QR code to be recognized. With different ambient light, the brightness of the code and the background changes. Similar to the surface color, this influences the contrast between the code and the background, which in turn might require more precise filters for successfully extracting the QR code.

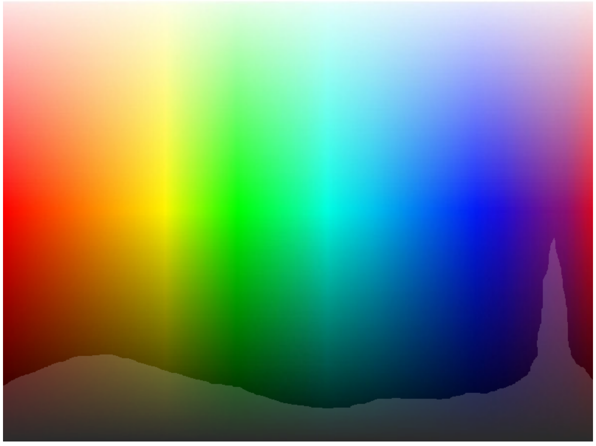

Figure 5.3: Surface colors that allow successful reading of IQR codes

Material	Result
paper	mostly positive results
cardboard	mostly positive results
cloth	negative results due to printing quality
paper with dot pattern	negative result due to lacking IQR extraction

Figure 5.4: Results of testing different surface materials

5.6 Evaluation of Readability

As the surface color below the printed code turned out to be important factor for the readability of an IQR code, we researched IQR codes on surfaces of different colors. Figure 5.3 estimates which surface colors allow successful IQR overlaying, based on the experiments performed in this work. The greyed out colors are those which where unreadable.

We have also tested different surface materials for IQR overlaying. Figure 5.4 shows the materials and the evaluation of our experiment.

Overlaying codes onto cardboard delivered mostly positive results. Due to the similarity of cardboard to paper these results could be expected. Printing on cloth resulted in not well readable codes. In most of these cases, the code blurred after being applied to the cloth. Having a way of high-quality printing might be recommendable for cloth. When overlaying codes onto paper with a dot pattern, no codes could be successfully read. Due to the significantly differing contrasts between the code and the dots, and between the code and the background, it was not possible to successfully extract the code. Therefore, it might not be possible to read codes from surfaces with patterns, when only using a single filtering sequence.

We would like to mention that the presented results are based on our experiments of trying to read stamped IQR codes with the developed software. Using appropriate printers will likely give a more positive result, since failures due to printing quality will be avoided.

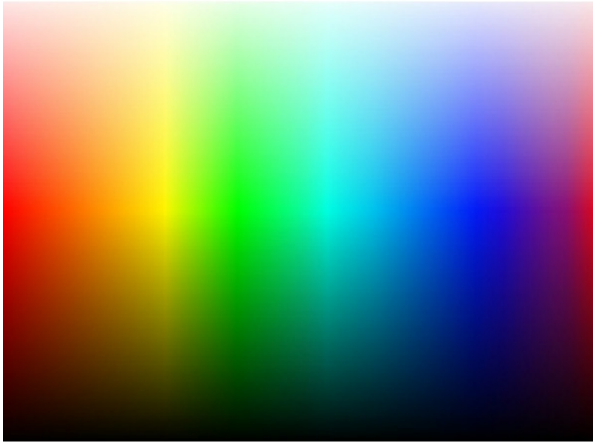

Figure 5.5: Surface colors that allow successful reading of IQR codes according to the simulation

5.7 Approaching the Surface Pattern

The surface pattern turns out to be an influencing factor when reading IQR codes from a surface. Surfaces without any special patterns were found to cause basically no issues in most cases, while complex surface patterns cause problems when filtering the image, not allowing us to extract any readable code.

With complex surface patterns interfering with IQR reading, we have developed a method of virtually overlaying IQR codes in order to avoid conflicts with such surface patterns. Using the method, several possible IQR placements on a surface can be checked, whether the code is actually readable without printing a single code. On spots, where the surface pattern is more simple, it is easier to successfully extract a code. We have tested this for the same surfaces as in chapter 5.6. The result can be seen in figure 5.5. The virtual overlaying delivered slightly better results. This is based on the absence of any actual printing, as well as on the fact that virtually overlaying allows running a higher amount of tests in almost no time. Therefore, the simulation can deliver an image of how the code could look like, if it was printed with good quality in several spots. The chance of finding a spot, where the overlaying succeeds is likely to be higher this way.

6 Conclusion

In this work we have developed IQR overlaying, a method of invisible information overlaying, and RIQR overlaying, a version of IQR overlaying meant for the use in retail. The method can be used to print invisible QR codes using UV fluorescent inks. The codes can be read by turning them visible using ultraviolet light. In retail, this allows to cover the surface of articles with big, well readable codes in order to aid automation of retail processes with the purpose of enhancement.
We have suggested concepts for the checkout and the stock-take process. By analyzing these concepts, we found out that it might be possible to partially automize the checkout by introduction of RIQR overlaying. There is evidence that the current stock-take process is not directly affected by RIQR overlaying. While there are potential improvements that are related to the inventory, there are no noteworthy changes in the process itself. However, in a simplified stock-take scenario, it might be possible to aid the workers in the registration of the available articles.
Aside from the theoretical concept for the functionality and application of RIQR overlaying, we have implemented hardware and a library for prototypical use. IQR overlaying is not bound to the use in retail. By adjustment of the code content and the library accordingly, it is possible to adapt IQR overlaying to other environments.

6.1 Reached Goals

In the introduction, we have established five goals for this bachelor thesis. We now want to talk about to what extent these goals have been accomplished:

- **The design of the RIQR overlaying method.**
  We have designed a concept for RIQR overlaying, which includes the printing and reading process, as well as concepts for the application of RIQR overlaying in retail. According to the concept, we have chosen an appropriate pair of ink and light sources to ensure the readability of overlayed QR codes. The printing process has been implemented using three different approaches. Stamping of codes using ink pads and printing of codes using inkjet printers turned out as a failure. Stamping of codes by spraying ink onto the stamp succeeded, however, it is merely a prototype solution. An appropriate printing technique has to be developed for IQR overlaying to be applicable. We have succeeded in reading overlayed RIQR codes, by photographing fluorescenting codes and processing the image, turning them into black-on-white codes. By application of QR readers to the image we are able to extract the code content. The concept introduces some changes to the processes that should be enhanced. We extend the checkout by a blackbox, which automatically scans articles. This way we should be able to save time and effort. The stock-take is changed by using a reading device to scan each

article, instead of manually counting. In theory, we should be able to minimize the error potential, but since we use a simplified setup for the stock-take scenario, this changed process might not be appliable in every setup.

- **The verification of the RIQR overlaying method.**
  Using the implemented hardware and the library, we have tested whether RIQR overlaying actually works. We have found five important, influencing factors. The surface material influences how well the ink is printable onto a surface and how well it keeps in shape. The surface color and the surface pattern, as well as the ambient light influence how and how precise the processing of a photographed IQR code has to be performed. The code size has to be appropriate for the QR reader to succeed in reading the code. We have run a set of tests for the surface material, the surface color and the surface pattern.
  Surface material testing has been performed by printing IQR codes onto different types of materials. We have found out that paper or paperlike materials are relatively easy to print on, while clothlike materials or coated surfaces pose issues. Blurring or dispersing of the printed codes are the result. When testing the surface color, we have overlayed RIQR codes onto surfaces of colors all over the color spectrum. We have performed the reading process and evaluated, whether a code was readable. For most colors, we succeeded in reading the overlayed codes. However, for black and purple surfaces reading the overlayed codes failed. We only have done few tests for complex surface patternss. After overlaying IQR codes onto the surfaces we performed the reading process as with the surface colors. None of the codes could be read. We believe that our filter sequences are not yet precise enough to be able to read IQR codes from surfaces with complex patterns.

- **Implementing definite instance-recognition.**
  We have implemented definite instance-recognition by adding the EAN of an article, a timestamp of when the code was printed and a globally unique printer ID to the RIQR code content. RIQR codes may contain further useful information like for example in our case, the expiration date of the article. However, this additional data is optional. Depending on the use-case, there may be other data that is added to the code's content. Under the assumption that a printer can only print one code at a time, the EAN, the timestamp and the printer ID yield a key that is unique to each article. With this key, we are able to distinguish between any two articles.
  Automation of the checkout can lead to multiple codes being inside the blackbox at the same time. While articles move through the blackbox, it is necessary to keep track of which articles have already been scanned and which not, in order to avoid duplicate scanning. We have solved this issue by using definite instance-recognition. Since each RIQR code is unique, it is easy to recognize duplicates.

- **Providing software and equipment that can read RIQR codes.**
  After designing IQR overlaying, we have built a reading device and developed a library that automatically reads IQR codes which the reading device detects. The

reading device is based on the blackbox from the checkout concept. We have added UV tubes to the interior as a ultraviolet light source for radiating IQR codes. The photographing is done by a camera that is located at the ceiling of the blackbox. The picture is sent to a computer, where the processing of the image is done. The library we developed is able to take a photo and then apply filtersequences to the image in order to extract a readable code. Because of changing ambient light or different surface colors, a set of filtersequences with different filters or parameters has to be implemented. After the processing, a QR reader is used to read the code that was extracted. The code content can then be processed by a checkout software for example.

- **Providing a program coordinates the overlaying process.**
  When testing IQR overlaying on different surfaces, we found out that often it is not possible to read a code due to complex surface patterns interfering with the code extraction. We have worked around this, by trying to avoid printing IQR codes onto spots, where it is impossible for us to read the code. By virtually overlaying a code onto several spots of a surface's picture and trying to read the codes, it is possible to find spots, where the overlaying is likely to succeed. We have automized this process of virtually overlaying codes by extending the provided library with virtual overlaying and using it in a tool we developed. The tool takes a photo of a surface and starts to virtually overlay some IQR code onto several spots of the surface. Now it tries to read each code. If it succeeds, we can guess that it might be possible to overlay a code on that spot, so it can be read successfully. The hardware that we have used for testing this tool is the blackbox. We have also run the simulation on all test surfaces we used for verification. Virtual IQR overlaying delivers a more positive result than normal IQR overlaying, however it aids in eliminating spots, where IQR overlaying is unlikely to succeed.

6.2 Future Work

In this section we will present possible future work that could have this work as a basis. The aim of these future works would be further improving IQR overlaying, as well as finding other applications for RIQR overlaying.

Solutions for Issues During Printing and Reading

During our research, we encountered issues when printing IQR codes onto certain surfaces, as well as when reading printed codes under certain conditions. In the end, we are still not able to print various codes with the ink still fluorescenting, when using printers. Therefore, it could be sensible to experiment with different printer types and printers in order to find printers that are able to apply the ink, while the fluorescence is still given. This way, different codes could be printed in a short time, without having to create stamps for each code as we did in this work. Also, we are not able to read IQR codes from any surface, yet. Especially surfaces that have patterns with high contrast pose a serious issue for our filter sequences. Not being able to extract the RIQR code from the surface, we are not able to read the code. Testing could deliver different filter sequences that might succeed in obtaining a readable version of a code. While this approach is the same as ours, it would be reasonable to try different reading devices instead of a normal camera. Adding a no-UV filter to the camera might enable us to only detect the UV rays and thus to completely ignore the surface that we have printed onto.

Research of Influencing Factors

During our research we identified material, surface color, surface pattern, code size and ambient light as the influencing factors for IQR overlaying. While we mainly focused on the material and the surface color, the other factors are still of importance to the performance of IQR overlaying. The surface pattern and the ambient light relate to how well a code is extractable using our filter sequences, while it depends on the size of a successfully extracted code, whether it is readable. Knowing how exactly these influences occur could be a key information in improving the reading process. This could be researched by performing a set of experiments for each factor. This is similar to what we have done for the surface color. By repeating the same experiment with only one factor changed insight about the influence of that factor can be gained. For example, by repeatedly reading the same code from the same surface with different ambient light, it would be possible to determine which ambient lights makes the extraction of codes the easiest. Using the determined ambient lights should then help to deliver better results, when reading IQR codes from surfaces.

Enhancement Measurement

Due to a lack of time, it was not possible for us to implement RIQR overlaying in a retail environment. Thus, it was not possible for us to actually implement the concepts that we suggested with the purpose of enhancing retailing processes. We also were not able to actually measure any enhancement. Nonetheless, the required manpower for performing the retail processes, as well the the required time for their execution could be measured with the right means. To achieve this, some sort of supermarket would have to be accessed. All the articles in the supermarket could then be equipped with a RIQR code and half the checkouts could be extended by the blackbox from the concept. Whenever a new customer is served at any checkout time measurement could be started. The blackbox would then assist the cashier in scanning the articles at the extended checkouts while the cashiers at the normal checkouts would scan everything manually. After the customer has payed the time would stop. Comparing both checkout times for similar or equal lists of bought articles could then be evaluated. This way, it would be possible to find out if RIQR overlaying in retail would actually improve the checkout. The stocktaking process can be measured accordingly. When beginning the stocktake, the timer will start and after the last article has been registered and stops after the last. Then, the resulting inventories are compared to the actually available articles. This way, it is also possible to check whether IQR overlaying helps in creating inventories with less mistakes.

Modification of the IQR Overlaying

IQR overlaying is a method that is not necessarily bound to be implemented the way we implemented it during this work. In general, any method that overlays invisible QR codes is a form of IQR overlaying. For example the use of UV fluorescent inks or definite instance-recognition are not essential to the method. Therefore, it might be of interest to implement different versions of IQR overlaying with the same or maybe a different purpose in mind. Different invisible fluorescent inks like infrared inks might increase or decrease readability when printing QR codes onto surfaces. Implementing other versions of IQR overlaying might provide an IQR overlaying method that performs better than our implementation. It could even be possible to implement IQR overlaying in ways that are completely resistant to influences that caused issues for us during this work. Finding these implementations could greatly improve the performance of IQR overlaying and also widen the spectrum of possible applications.

IQR Overlaying in Other Scenarios

In this work we approached the design of RIQR overlaying with the improvement of retail processes in mind. While RIQR overlaying is meant to be used in retail, there might be plenty of other environments, where IQR overlaying could be applied when aiming for enhancement. Even if RIQR overlaying would turn out meaningless for retail, the general IQR overlaying could still be used for other processes. Examples for scenarios, where the application of IQR overlaying might be of interest are for example libraries, warehouses or any kind of place where great amounts of objects have to be identified and organized. Warehouses on the one hand might not profit from the invisibility of the codes in many cases, depending on the wares that are stored. IQR overlaying would be inappropriate in this case or similar cases. Libraries on the other hand might greatly profit from invisibility. No unsightly matters of identification would have to be added to the books. Many customers might approve of this because to them, books might also be a matter of entertainment that should look good. Furthermore, it could be argued that the books value decreases less if the change of adding some tag is invisible. After finding a scenario for application, IQR overlaying should be adapted to the special needs of the environment. As we did in our work, one should think about the content of the code and about appropriate reading devices for the environment. Using the blackbox from retail, for example, is not likely to be a good solution in a library, where a librarian might have to open each book to scan the code that could be printed on the very first page. If possible, once the IQR overlaying is specialized, measurement of improvements or effects of IQR overlaying on the environment could be performed and it could be evaluated whether the introduction of the invisible codes would be appropriate.

Acknowledgements

I would now like to thank the people that supported me during the course of this work. First, I would like to thank Christian Bürckert for giving me the chance to write this Bachelorthesis at Prof. Dr. Krüger's chair and for supervising me to the full extent of his capabilities. I also would like to thank Dennis Königsmark, who assisted me in spell checking. Next, I would like to thank Prof. Dr. Krüger and the DFKI GmbH who provided the required resources that were used during the research. Last but not least, I would like to thank my parents Ullrich Barth and Bettina Barth for all the support up to now, be it mental or financial. I could not have made it this far without them.

Saarbrücken, 5th October 2015

Marvin Barth

Bibliography

[1] Manal Abd-ElAziz Abd-ElZaher. Different types of inks having certain medicolegal importance: Deciphering the faded and physically erased handwriting. Egyptian Journal of Forensic Sciences, 4(2):39–44, June 2014.

[2] Judith D. Auslander. Composition For Invisible Ink Responsive to Infrared Light. United States Patent, 1997.

[3] Judith D. Auslander and William Berson. Bar Codes Using Luminescent Invisible Inks. United States Patent, 1996.

[4] William Berson and Judith D. Auslander. Bar Code Scanner for Reading a Lower Layer Luminescent Invisible Ink that is Printed Below a Upper Layer Luminescent Invisible Ink. United States Patent, 1996.

[5] David Chaum, Richard Carback, Jeremy Clark, Aleksander Essex, Stefan Popoveniuc, Ronald L. Rivest, Peter Y. A. Ryan, Emily Shen, and Alan T. Sherman. Scantegrity II: End-to-End Verifiability for Optical Scan Election Systems using Invisible Ink Confirmation Codes. Webpaper.

[6] Abbas Cheddad, Joan Condell, Kevin Curran, and Paul Mc Kevitt. Digital image steganography: Survey and analysis of current methods. Signal Processing, 90(3):727–752, March 2010.

[7] Binh Duong, Helin Liu, Liyuan Ma, and Ming Su. Covert thermal barcodes based on phase change nanoparticles. Scientific reports, 4:5170, January 2014.

[8] Nan-Wei Gong, Steve Hodges, and Joseph a. Paradiso. Leveraging Conductive Inkjet Technology to Build a Scalable and Versatile Surface for Ubiquitous Sensing. Proceedings of the 13th international conference on Ubiquitous computing - UbiComp '11, page 45, 2011.

[9] Antonio Grillo, Alessandro Lentini, Marco Querini, and Giuseppe F. Italiano. High Capacity Colored Two Dimensional codes. Computer Science and Information Technology (IMCSIT), Proceedings of the 2010 International Multiconference, pages 709–716.

[10] F. Hartung and M. Kutter. Multimedia watermarking techniques. Proceedings of the IEEE, 87(7):1079–1107, July 1999.

[11] William A. Houle. Light Sensitive Invisible Ink Compositions and Methods For Using The Same. United States Patent, 2000.

[12] Duncan Johnston, Chris Barnardo, and Chris Fryer. Passive multiplexing of printed electroluminescent displays. Journal of the Society for Information Display, 13(6):487–491, 2005.

[13] Yoshihiro Kawahara, Steve Hodges, Benjamin S. Cook, Cheng Zhang, and Gregory D. Abowd. Instant Inkjet Circuits: Lab-based Inkjet Printing to Support Rapid Prototyping of UbiComp Devices. Proceedings of the ACM International Joint Conference on Pervasive and Ubiquitous Computing, pages 363–372, 2013.

[14] Yoshihiro Kawahara, Steve Hodges, Nan Wei Gong, Simon Olberding, and Jurgen Steimle. Building functional prototypes using conductive inkjet printing. IEEE Pervasive Computing, 13(3):30–38, 2014.

[15] Antonio Krüger, Lübomira Spassova, and Ralf Jung. Innovative Retail Laboratory — Investigating Future Shopping Technologies Innovative Retail Laboratory — Erforschung von Technologien für das Einkaufen von morgen. it - Information Technology, 52(2):114–118, 2010.

[16] Mohamed Lazzounim and Mohamed Yousaf. Infrared Printing Ink and Method Of Making Same. United States Patent, 1995.

[17] Jeevan M Meruga, William M Cross, P Stanley May, QuocAnh Luu, Grant a Crawford, and Jon J Kellar. Security printing of covert quick response codes using upconverting nanoparticle inks. Nanotechnology, 23:395201, 2012.

[18] A Sankara Narayanan. QR Codes and Security Solutions. International Journal of Computer Science and Telecommunications, 3(7):1–4, 2012.

[19] Simon Olberding and Michael Wessely. PrintScreen: Fabricating Highly Customizable Thin - film Touch - Displays. Proceedings of the 27th annual ACM symposium on User interface software and technology, 2014.

[20] Suppat Rungraungsilp, Mahasak Ketcham, Virutt Kosolvijak, and Sartid Vongpradhip. Data Hiding Method for QR Code Based on Watermark by compare DCT with DFT Domain. International Conference on Computer and Communication Technologies, pages 144–148, 2012.

[21] Nadine Walz. QR - Codes, Schnittstelle Online/Print. Merzakademie.

[22] Karl D.D. Willis, Eric Brockmeyer, Scott E Hudson, and Ivan Poupyrev. Printed Optics: 3D Printing of Embedded Optical Elements for Interactive Devices. Proceedings of the 25th annual ACM symposium on User interface software and technology, 2012.