Progressive Barcode Applications

Marie Vans, Hewlett-Packard Labs, 3404 E. Harmony Rd., MS 36, Fort Collins CO 80528, USA
Steven Simske, Hewlett-Packard Labs, 3404 E. Harmony Rd., MS 36, Fort Collins CO 80528, USA
Brad Loucks, Hewlett-Packard Labs, 3404 E. Harmony Rd., MS 36, Fort Collins CO 80528, USA

Abstract

Last year, we introduced the four-dimensional Progressive Barcode which is a printed mark that does not take up more real estate as it is used to advance a workflow. Adding progressive information to a barcode allows it to change through time—supporting many different information lifecycles. It is thus a means of using the same barcode location for multiple barcodes through time. In addition, progressive barcodes can be used to support two (or more) applications or services in the same object. One of these is typically Standards-compliant and the other is typically proprietary/customized.

The key technological insight is that the progressive barcode effectively uses two planes of information. The first plane is binary, with high contrast between the two binary encoding (usually black and white) tiles in the barcode. The second is N-ary, and utilizes inks that are invisible to the binary barcode reading software. Typically this is accomplished with highly saturated colors, but it can also be incarnated with IR, UV or other "invisible" inks.

In this paper, we will demonstrate the types of applications and services that are enabled by the progressive barcode. They are most effectively deployed when there are multiple types of information payloads needed for a single object—e.g. point of sale and customer interrogation of the product. This also makes them useful in a variety of document/physical item workflows.

Introduction

Documents can be found in both physical and electronic form. The lifecycle of a single document may include movement between electronic and physical form several times. Additionally, a physical document may change as it moves along a workflow during its lifecycle. We include labels and packaging in the definition of a document, which can be used for track and trace (a form of progressive workflow); and to one-time use items, such as tickets and coupons. For these items, two-dimensional (2D) barcodes can be used for track and trace, increasingly point-of-sale, and for marking sequential steps (e.g. arrival at different locations for a package) in its workflow (e.g. from manufacturer to consumer). These 2D barcodes are thus familiar identifying objects for users. Currently, however, several barcodes (1D or 2D) are placed on the package if barcodes are to be used for more than one purpose. What is needed in many cases (particularly supply chain) is an identifying object that does not grow in size as the item moves through its workflow, but can provide multiple functionalities. The derivation of a 4D (x, y, color, time), or Progressive Barcode, enabling multiple stages in a workflow, is a function of (probability-driven) security needs and other concerns (branding, fraud prevention, robustness to damage, etc.). Building on previously published work[1, 2] we show how Progressive Barcodes can be used in specific applications.

Overview of Progressive Barcodes

As described previously [1, 2], an incremental information object (IIO) is an object that changes as a one-way function of its current state. For example, if we start with a simple binary sequence {000000000000} and then move to a next state through the replacement of four 0's by four 1's. Then, two allowable next states are {001010001100} and {100110000100}. In general, if there are N 0's left to be changed into 1's and M 1's added to the next state, then we can write $N! / [M! (N-M)!]$ different next states, where $!$ is the factorial operator. For the IIO, once a 0 has been changed into a 1, it cannot change back into a 0. Thus, each successive state can be immediately compared to a previous state to see if it is logically a part of the same workflow.

Due to the ubiquity of high-resolution mobile cameras, the Data Matrix [3] and QR 2D barcodes have been adopted for many applications [4] that tie a physical object or location to electronic workflows (as simple as connecting to a URL, or as complicated as providing a log-in screen for member services). The groundwork has been laid for higher-density color barcodes – named 3D barcodes for their 2D layout and their third, color, dimension – such as the color tile [5]. The progressive barcode uses this concept of a printed mark that does not change size (take up more "real estate") as more information is written to it, and combines it with standard barcodes and the secure color tile [5].

As described previously [2], progressive barcodes allow us to assign the statistical probability associated with any transition between two steps in a workflow based on how many bits are written and how many remain. If progression step i is defined as P_i, where the number of residual (0 bits) at the end of the workflow is N_{RB}, and the number of initial unwritten bits is N_{IU}, then the per-step governing equation is:

$$\frac{N_{IU}!}{(N_{IU}-N_{RB})!N_{RB}!} \geq \prod_i P_i \, .$$

where ! is the factorial operator. P_i may be determined from, for example, the required statistical confidence that a next step cannot be randomly guessed multiplied by the total number of progressive barcodes of the current state that will be readable in the supply chain. If the barcode is unique at step i-1, then the total number of barcodes of the current state is 1. If the progressive barcode is binary, then the number of bits in the workflow *is* N_{RB}-N_{IU}. If there are N_C colors, then the number of bits increases to $[ln(N_C)/ln(2)]*(N_{RB}-N_{IU})$. The size (height and width in tiles) of the progressive barcode to be used in the workflow can be determined from these equations along with the number of bits to write at each state.

Any number and combination of colors may be used to create progressive barcodes. However, for demonstration purposes we show 6 color barcodes utilizing the pure printing colors: Cyan,

©2013; Society for Imaging Science and Technology

Magenta, and Yellow. These can be later overwritten or overprinted to create three additional colors, Red, Blue, and Green. Figure 1 demonstrates the concept of color progression. Each cell in the barcode starts out in a particular color state and can only progress accordingly. For example, if a cell is currently magenta, '*M*', the next allowable state for the cell can be either blue, '*B*', or '*R*', red. It may not progress to green. Once a cell has reached the Black stage, it can no longer progress. Figure 1 illustrates this progression.

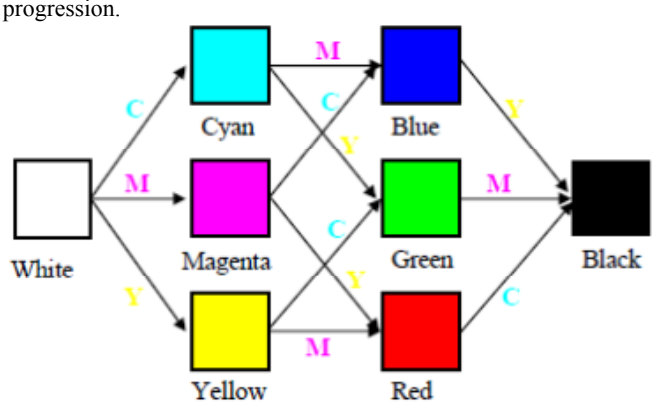

***Figure 1**. The basic lifecycle of a color tile, where the colors White, Cyan, Magenta, Yellow, Blue, Green, Red and Black are shorthanded as W, C, M, Y, B, G, R and K, respectively. Note that the tile may be written to three times in its lifecycle, and contain one of 8 states (3 bits). At the first stage, the W tiles can be overprinted with C, M or Y to create a C, M, or Y colored tiled, respectively (pretty obvious in hindsight—such is the nature of subtractive printing). At the second stage, the C tile can be overprinted with M or Y to create a B or G tile, respectively; the M tile can be overprinted with C or Y to create a B or R tile, respectively; and the Y tile can be overprinted with C or M to create a G or R tile, respectively. The (CMY) characters associated with the arrows indicate the additionally printed color required for the progression.*

For the purposes of determining the absolute data content of a color tile, we consider each color tile to be independent. We define the tile to be **N**-ary, where **N**=the number of colors allowed at each tile. Thus, there are $\log_2(n)/\log_2(2) = \log_2(n)$ bits at any stage. If **n**=2, as for 2D DataMatrix, QR, Aztec and similar binary 2D barcodes, then there is 1 bit per tile. For a color tile with six colors {RGBCMY}, there are 2.585 bits/tile. For seven colors, {RGBCMYK} or {RGBCMYW}, there are 2.807 bits/tile. Finally, if there are eight colors allowed {RGBCMYWK}, there are obviously 3.0 bits/tile exactly. This means that a color tile barcode that is *X* data tiles wide and *Y* data tiles high contains exactly $XY\log_2(n)$ bits.

Adding color progression allows us to use the "static" data encoded within the black and white modules for standard purposes such as point of sale, serial numbers, and product information while allowing a "separate channel" for encoding additional, workflow-related information that changes over the course of the workflow. For the static data, the off-the-shelf reader reads the "black-as-black" modules and the "rest-as-white", regardless of the color that may appear in the "formerly" white modules. Obviously, we need to make sure the colors don't interfere with the decoding of static messages. It is important to understand how saturation of colors affects the performance of standard barcode readers. Note that for color progression in this instance, colors are not allowed to progress all the way to black. Instead, the progression terminates at red, green or blue.

We experimented with maximum saturation levels for each of the pure CMYRGB colors. Adding color to the white modules in Data Matrix barcodes so that they remain readable with standard devices allows us to progress a workflow using a wide variety of colors and simultaneously carry static identifying data through each step. For each color, we found the saturation level above which all barcodes remain readable. Experiments showed that the colors CMYR but not B and G could be read at full saturation (i.e. for R, the channels values can be set to {255, 0, 0} but for green the minimum value of the other two channels must stay above 104 ({104, 255, 104}). We had similar results with blue. Tests were performed at 1200dpi (dots per inch), 600dpi, and 300dpi with little differences found and 1200 dpi performing slightly worse due to halftone affects. We used an InData Systems LDS-4600 reader with 405 nm w.p.e. LEDs as its light source for these experiments.

Figure 2 contains an example barcode which is readable with standard off-the-shelf Data Matrix reading software. This simple example demonstrates the power of this approach. We will focus here on two applications: point-of-sale (POS) customer focused applications and typical workflows, such tracking a package from manufacturer through multiple distribution points to an end customer.

***Figure 2**. An example progressive barcode using a standard Data Matrix. The black and rest-as-white modules will read using standard Data Matrix software.*

Applications of Progressive Barcodes

Several applications can be created to work in conjunction with progressive barcodes for POS and tracking applications. Obviously, for the special non-standard applications, information must be encoded as colored and or specialty inks in modules normally read as white by standard applications. While the standard readers read the white-as-white modules and will ignore the color, the non-standard applications read the "white as N-ary" modules to decode the data.

Workflows

GS1 (Global Standards One) [6] is a non-profit organization that develops standards for world-wide track and trace applications. Progressive barcodes provide a solution for GS1 product workflows in which multiple barcodes are used through-out the supply chain. Figure 3 demonstrates how this might work. The original GS1-compliant barcode contains a product number that identifies the product and remains static as it progresses along its workflow (indicated using the right pointing arrows). Concurrently, the information encoded in the colors changes at each step. The data contained within the white-as-N-ary modules can be used for data normally encoded in separate barcodes

attached to the package. It is easy to see the real estate savings in this example, as the footprint for the barcode remains static.

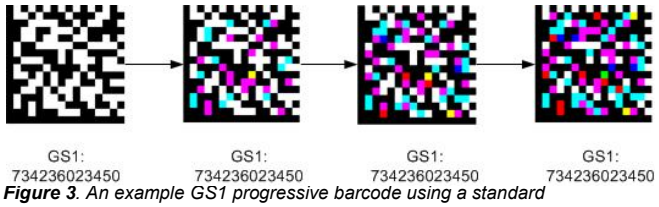

Figure 3*. An example GS1 progressive barcode using a standard GS1compliant barcode. The black and rest-as-white modules will read a GS1 product code using standard Data Matrix software, while the rest-as N-ary modules contain progressive information.*

In another example, progressive information can be used for inference applications. In Track and Trace lingo, inference is the relationship between an item and the items related to it. This includes items contained by it (smaller) and containing it (larger) along with the requisite links "forward" (to a larger item) and "backward" (to a smaller item(s)). In this case, the serialized numbers for items packaged within a larger container can be reasonably inferred from the enclosing container without having to physically open the container to check each individual item. As seen in Figure 4, reading the white-as-white would give the GS1 product code for each of the individual units packed into cartons which are placed into cases which are, in turn, placed onto a pallet. Information in the white-as-N-ary modules could be used at each level to infer the correct, perhaps mass-serialized code for the unit in which it is contained. Additionally, each level could be digitally signed (DS) for added security.

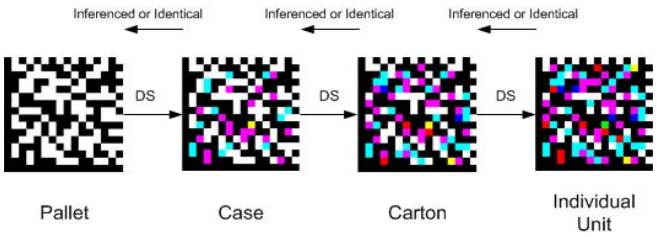

Figure 4. An example GS1 progressive barcodes using a standard GS1compliant barcode with digital signatures and inference application.

Figure 5 illustrates a workflow using both a GS1 compliant barcode with color-overlaid inference in a situation where branching occurs. The bubble prior to "Step A" demonstrates that the initial GS1 code itself can be derived from several parts prior to the start of the current workflow. At step A, we have the initial GS1 code which identifies a particular product. For illustrative purposes, we could say this was a single dose of a highly regulated prescription drug, created using the ingredients indicated by the arrows flowing into the bubble in the step prior to Step A. At Step A.B1, we may automatically create a package containing a single month's supply, in this case 30. From this point, the workflow diverges. The divergence could occur as a result of different dosages packaged. For example, at A.B1.C1, we could create cartons with 12 packages of medication (a total of 360 doses) in a single carton while at A.B1.C2 only 6 packages of medication (a total of 180 doses) are packed into a carton. Finally at steps A.B1.C1.D1-A.B1.C1.D3 different numbers of cartons are packed onto single pallets. One thing to note is that inference is possible since each step not only identifies the previous step, but all steps prior to that. The main component, the single dose medication, is statically identified by the GS1 identifier encoded into the black and white-as-N-ary modules. Inference is enabled and quantities are encoded using color in the white-as-N-ary modules. Additional information can be encoded in the white-as-N-ary modules in color, such as provenance and workflow step-specific information. A digital signature can be added to increase security.

Point-of-Sale

Point-of-Sale (POS) is another opportunity for which progressive barcodes seem ready-made. Many standard barcodes (typically black/white) contain static data which is read by scanners at checkout. The data encoded may contain product numbers and/or sale price, but can also be connected to applications for inventory and other purposes. These barcodes are created for the use of the retail outlet and are not necessarily interesting for the customer. While some products now carry secondary barcodes for customer applications, these usually take up more space on the product packaging. Using the white-as-n-ary modules with color, additional customer applications can be added to the same barcode used by retailer. Figure 6 illustrates this concept. Applications that issue rewards for buying the product, gaming and incentivization for loyal customers, location-based, and URLs for product website and further customer interactions are but a few examples. In general, any additional, interrogatable by reading the white-as-N-ary modules could be added to the barcode without requiring more real estate on the package.

Conclusions

We have shown that the 4D progressive barcode can be used to carry multiple layers of information, each customized to a particular application. Standard barcode readers decode one plane of information using the black and white read as white only and ignoring any color. Applications that recognize colors decode the white as N-ary and constitute another plane of information invisible to standard readers. The examples shown herein are but a few of the multifarious ways in which a great deal of information can be packed into a small footprint.

References

[1] S J. Simske, A.M. Vans and B. Loucks. "Incremental Information Objects and Progressive Barcodes", Proc NIP28, 28:375-377(2012).

[2] A.M. Vans, S J. Simske, and B .Loucks. "Progressive Barcodes", Proc NIP28, 28:368-370(2012).

[3] International Standard ISO/IEC 16022:2006(E), Second edition 2006-09-15, "Information technology – Automatic identification and data capture techniques – Data Matrix bar code symbology specification," 142 pp., 2006.

[4] http://www.redlaser.com, last accessed 2 February 2012.

[5] Simske SJ, Sturgill M, Aronoff JS: Effect of copying and restoration on color barcode payload density. Proc. ACM DocEng 2009 127-130, 2009.

[6] GS1. EPCglobal Homepage. http://www.epcglobalinc.org/home (accessed February, 2012).

©2013; Society for Imaging Science and Technology

Author Biography

Marie Vans is currently a Research Scientist with Hewlett-Packard Labs in Fort Collins, Colorado. Her main interests are security printing and imaging for document workflows, statistical language processing, and other approaches to document understanding. She holds a Ph.D. in Computer Science from Colorado State University.

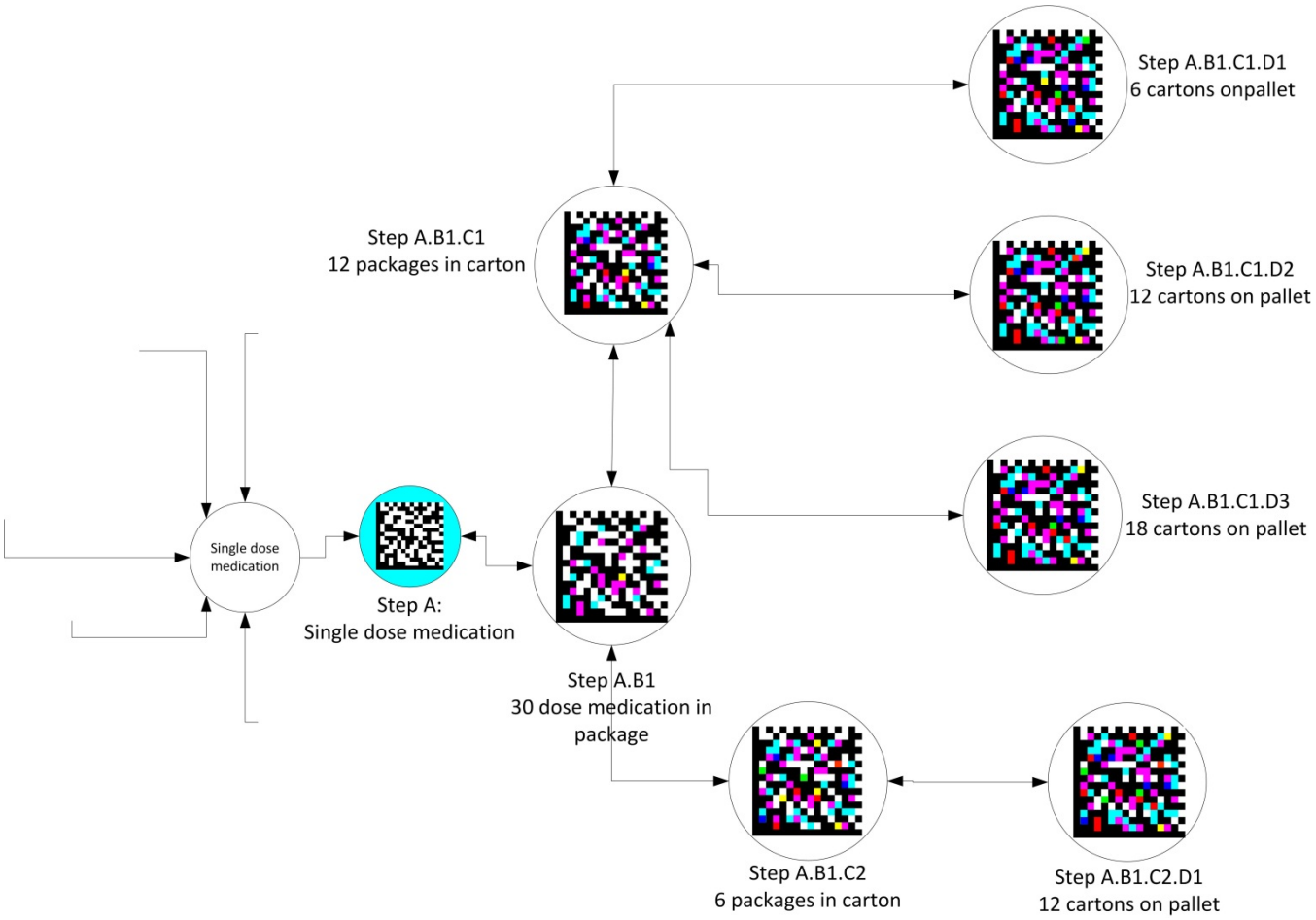

Figure 5. *An example GS1 workflow with branches using GS1compliant barcode with digital signatures and inference application.*

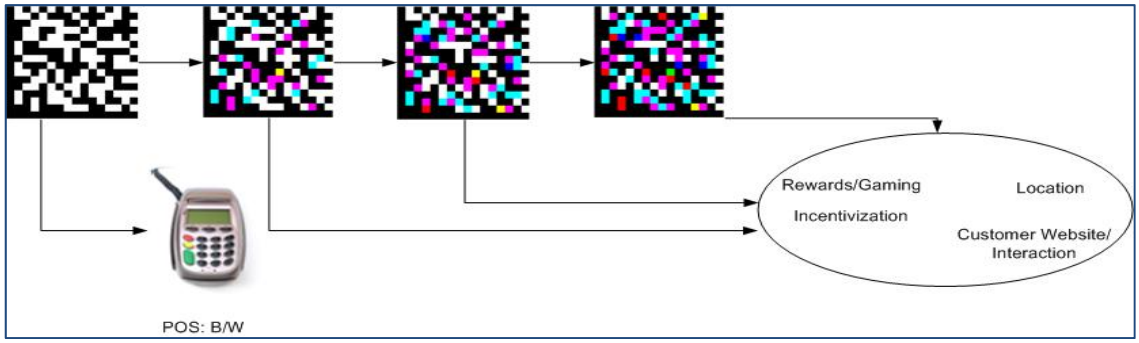

Figure 6*. POS system using standard off-the-shelf retail checkout software to read product information/sale price. Progressive barcodes with additional layers for customer applications.*