

Progressive Barcodes

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

Barcodes are a familiar data-carrying mark printed onto physical items such as packaging, labels, magazines, etc. Many consumers, retailers, and distributors are routinely reading them with their mobile devices. We introduce 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.

Introduction

Documents exist, by necessity, in both physical and electronic form. During the lifecycle of a single document, it may have to move between its electronic and physical form several times. In addition, a physical document may progress along several steps in a workflow during its lifecycle. We extend the concept of “document” to labels and packaging, 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, information-laden 2D barcodes are typically 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; however, several barcodes (1D or 2D) are currently placed on the package if barcodes are to be used for more than one purpose. What is needed in many cases (particularly for the supply chain) is an identifying object that does not grow in size as the item moves through its workflow, and 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.).

Progressive Barcodes

In order to utilize the concept of a printed mark that does not change size (take up more “real estate”) as more workflow information is written to it, a printed mark that can be read and written to incrementally during various stages in its lifecycle is needed. Such a mark, based on a combination of standard barcodes and the secure color tile [2], is the progressive barcode.

The process of writing incremental information to the barcode is given by these steps:

- A. Scan (or capture with an imager such as a camera) the current image of the progressive barcode.
- B. Perform segmentation of the barcode (extract it from the overall image; determine the content of the barcode). This is a two-stage process. First, the barcode itself is identified in the image and segmented from the entire image. Next, the barcode itself is interpreted (decoded).

This latter step is performed either in conjunction with calibration (next step, C) or after it.

- C. Calibrate (orient, de-warp, etc.) the barcode as needed. This step uses the non-payload indicia associated with the barcode to determine its orientation. For example, Figure 1 shows the non-payload indicia as the upper leftmost image (α).
- D. Determine the location of candidate areas to write incremental information. In Figure 1, these areas are payload indicia (yellow in image β). We can use any of a variety of algorithms (scrambling, random/nonce, one-time-pad, encryption, data replication, etc.) to determine where to place the new elements in the progressively written color barcode.
- E. Determine the amount of security information to write. This can be based on the relative security needs of the given stage, the amount of available elements in the barcode to write to, and/or any minimum absolute security requirements. The number of bits to write at each stage in a (pre-defined) workflow are in general known beforehand, and help to define the size of the payload indicia section of the barcode. In Figure 1, we use 20 bits of the 192 bit payload to “set” the entropy at the start of the workflow, and write another 20 bits at each of the three successive stages shown.
- F. Determine how (location, what information, etc.) to write the security information. Here is where the algorithm for incorporating the current stage data is deployed (scrambling, random, encryption of the user’s public key+nonce, etc.). Note that it is easy to simply replicate the data in this stage in the associated electronic workflow (e.g. cloud network for the workflows), so that random/nonce methods are perfectly valid. If an independent workflow that has no backward visibility in the workflow node-node transmission is desired—e.g. in cases where the distribution network organization is proprietary—each stage can be an intelligent hash of the previous barcode state. Alternatively, a full onion-shell track and trace model can be deployed. These options are outlined Figure 2. The magenta workflow (boxes in middle of the image) refer back to the registry for the progressive barcode information at each stage, while the lower (green) workflow only refers back to the previous state (consistent with a track and trace “onion” model).
- G. Write the information to the progressive barcode. Each element may be written to/overwritten three times per the allowable progression shown in Figure 3.
- H. Print and store as necessary. Importantly, the same document formerly containing the previous-stage barcode now has the current-stage barcode written in its place. However, in the case of Figure 1, the previous-stage barcode will still be “provable” on the new barcode since there will be no new “open elements” on the later-

stage barcode (one-way progression only). Since other barcodes will have their open elements in different locations, this helps prevent casual counterfeiting since the odds of two progressive barcodes having non-conflicting elements at any stage of the workflow is very small.

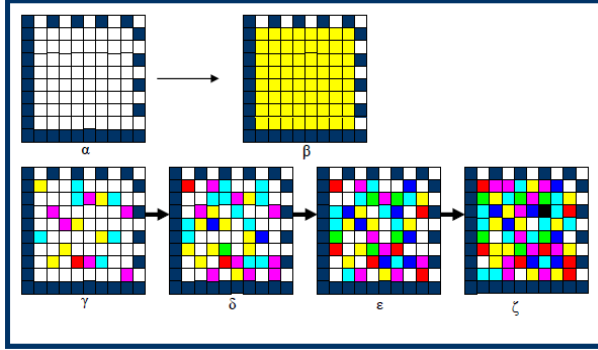


Figure 1 Adding data to barcode

Figure 1 illustrates new data being added to a progressive barcode as it progresses through the workflow. The upper leftmost image (α) represents a barcode with only the non-payload indicia indicated. The non-payload indicia (NPI) are the perimeter pixels on all four sides. Two sides (left, bottom) are solid and two sides (right, top) are alternating light/dark to provide calibration. The yellow pixels shown on the upper row, center, image (β) are the data pixels which can be written to as part of the incremental writing process. In the lower row, the initial barcode is pre-filled with, in this case, 16 data bits as shown in the leftmost image (γ). Next, the three workflow stages from barcode, γ , result in incremental writing of pixels to the barcode (δ , ϵ , and ζ).

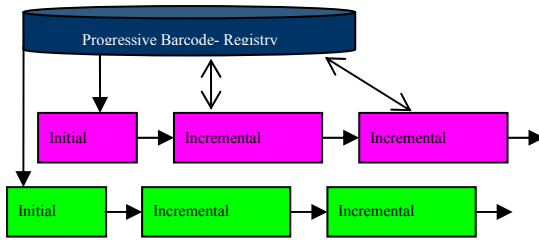


Figure 2: Registry-based (magenta boxes) or independent (green boxes) incremental workflows.

While any number and combination of colors may be used to create progressive barcodes, for purposes of illustration we show 6 color barcodes utilizing the pure printing colors: Cyan, Magenta, and Yellow. These can be later overwritten or overprinted to create three additional colors, Red, Blue, and Green. Figure 3 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.

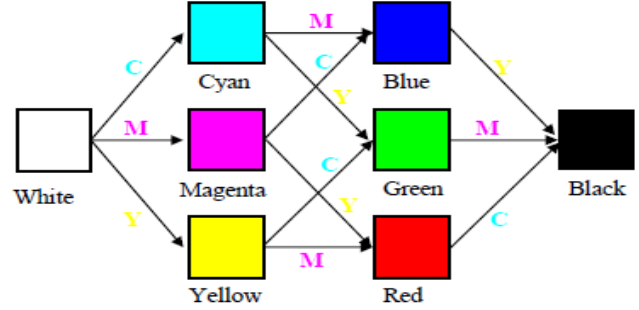


Figure 3: Allowable color state changes for 6 color Progressive Barcodes.

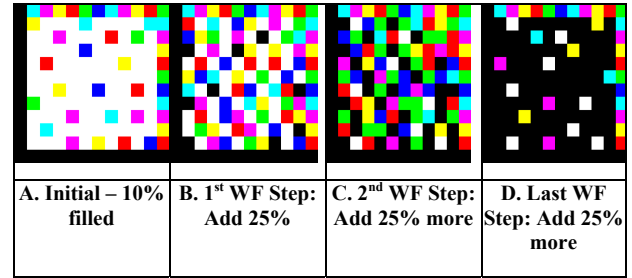


Figure 4. Four Stage Progressive Barcode Workflow Progression in which 25% of the available color transitions are utilized at each step. Final barcode contains 10% unused states.

Figure 4 illustrates a simplistic 4-step workflow in which the initial barcode, Figure 4A, progresses from 10% filled to 10% unfilled cells (Figure 4D) at the end of the workflow. Each workflow step involves the addition of color transitions representing 25% of the remaining 80% available states in the workflow.

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 $\lceil \ln(N_C)/\ln(2) \rceil * (N_{RB}-N_{IU})$. From these equations, we can readily determine the size (height and width in tiles) of the progressive barcode to be used in the workflow along with the number of bits to write at each state.

Conclusions

We have described a technology that can be used in place of traditional static barcodes. We currently have an application that can generate and recognize 6-color progressive barcodes (see user interface screenshot, Figure 5). The user interface takes a symbol – for example, an 18x18 matrix; a size (in pixels) for the x and y dimensions of each individual cell (module); a color scheme; and a directory in which initial data is stored. The application displays each barcode generated in the workflow according to the total number of workflow steps. Currently, we are using Serialized Global Trade Identification Number (SGTIN-96) compliant barcodes [4] to populate the initial barcodes and demonstrate the utility of adding information to this widely used standard for track and trace of products world-wide.

We have shown how color progressive barcodes allow incremental information to be added to the mark as it progresses through a workflow without taking up additional real estate. We have taken the idea of a tile deterrent [1, 2] and extended it for application in the document workflow ecosystem. We have also shown that whether a workflow is registry-based or stand-alone, progressive barcodes provide a pre-designated statistical security at

the workflow-stage level and a compact method for tracking progressive data.

References

- [1] S. J. Simske, S., G. Adams, J.S. Aronoff, M.M. Sturgill., and A.M. Vans. “Staggered and Dual-Channel Barcodes”, Proc. NIP27, 27:682-685(2011).
- [2] S.J. Simske, J.S. Aronoff, M.M. Sturgill and J.C. Villa, “Spectral pre-compensation and security deterrent authentication,” Proc. NIP24, 24:792-795 (2008).
- [3] R. Villán, S. Voloshynovskiy, O. Koval and T. Pun, “Multilevel 2D bar codes: towards high capacity storage modules for multimedia security and management,” IEEE Transactions on Information Forensics and Security, 1(4):405-420 (2006).
- [4] GS1. EPCglobal Homepage. <http://www.epcglobalinc.org/home> (accessed February, 2012).

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 Screen shot of the user interface for the creation of Progressive Barcodes