Forensic Markings for Progressive Barcodes

Steven Simske, Marie Vans, Stephen Pollard & Guy Adams

Keywords: barcode, data matrix, workflow, inference, qr code

Abstract

During the lifecycle of a single document, it may have to move between electronic and physical forms several times. A physical document may progress along several steps in a workflow during its lifecycle. Information-laden printed features such as bar codes can be used to move the physical item (document, package, envelope, carton, etc.) from node to node in its workflow (e.g. from manufacturer to consumer). These information-laden features are termed identifying objects. Current approaches use sequential barcodes for this purpose, which can use significant. “real estate” on the physical item. What is needed is an identifying object that does not grow in size as the item moves through its workflow. We have previously described such an incremental identifying object (IIO), familiarly designated a **progressive barcode**, as barcodes are one of the most logical instantiations and provide a means to tie the alteration of the IIO necessary to move one step along in the workflow to the physical attributes of the printed barcode itself. Specifically, we designate this as the **forensic marking-IIO, or IIOfm**. The approach outlined provides the means to tie a physical item to a workflow at each stage, even if the document is printed multiple times during its lifecycle.

Introduction

Recently, there has been a huge increase in the adoption of 2D barcodes for enterprise and consumer applications. The Data Matrix 2D[2] barcode has become a primary carrier of supply chain information, most notably for track and trace. The QR (Quick Response)[3] 2D barcode has spread from Japan to the rest of the world, and is a standard means of connecting a barcode to a URL. We have previously described the *Progressive Barcode*[7] as a method for increasing information capacity without increasing the footprint of the barcode. This 4D barcode (color + 2D + time) is a means of using the same barcode location for multiple barcodes through time. This supports many enterprise workflows, including document lifecycles. In this paper, we consider a method for securely linking a physical item to a workflow stage. This approach creates a direct relationship between forensics/authentication and the security of the

HP Labs

incremental step and provides authentication linked to mass serialization. Additionally, the approach is difficult to re-engineer by a fraudulent agent.

Progressive Barcodes

A general outline of a progressive barcode (or incremental identifying object) is given in Figure 1. 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 progress 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.

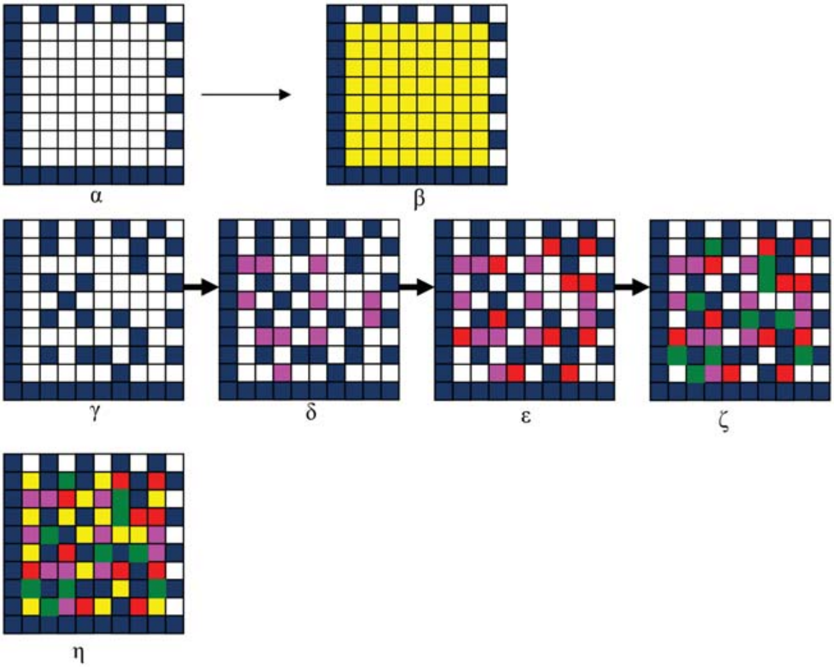

Figure 1[6]***.*** *Illustration of new data being added to an IIO as it progresses through the workflow. Upper row. The upper leftmost image (**α**) represents a 2D barcode with only the non-payload indicia indicated. The non-payload indicia (NPI) are the perimeter pixels on all four sides. Two sides (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. Middle row. Here, the initial IIO is pre-filled with, in this case, 16 data bits (still in indigo color), as shown in the leftmost image (**γ**). Next, three workflow stages result in magenta (**δ**), red (**ε**), and green (**ζ**) pixels being infilled. Finally, in the lower image (**η**) the residual pixels (which do not get written to, even in the last step of the workflow, to ensure there is "entropy" in the IIO between successive steps in the workflow) are shown in yellow.*

While progressive barcodes provide a secure means to transition from node to node in a workflow, the means to encode individual nodes proposed to date do not incorporate the physical attributes of the current printed barcode. The shape distortion encoding difference (SDED) approach solves this issue.

Shape Distortion Encoding Difference

Suppose we divide the idealized perimeter of the IIO in Figure 1 (β) into 10 line segment elements along each of the 4 sides (one element for each of 40 exposed "sides" of one of the 36 edge modules of the 100 module example). Then, we compute a sum squared error (SSE) of the residual, of some image related metric *p*, for each of the 40 elements (which is actually akin to a variance metric). The overall SSE of the deterrent (or progressive barcode in this case), designated SSE_{Det}, is defined in simplest form as:

$$SSE_{Det} = \sum_{elems} \sum_{p(i) \in element} (p(i) - \mu_{element})^2 \quad \textit{Equation 1}$$

Where *p(i)* is the orthogonal displacement with respect to the deterrent model of each point on the perimeter for a particular element and $\mu_{element}$ is the mean of such over the whole of that element. Each edge element therefore provides a "piecewise sum-squared error". The sum in Equation 1 is divided by 40*n* (where *n* is the number of samples of *p* per element) to determine the atomic unit of encoding, and then a 40-position string, P, is created for the deterrent (by dividing the SSE of each element by the atomic unit and rounding).

The shape distortion encoding difference, or SDED, for comparing any two deterrents, is then defined as:

$$SDED = \sum_{j \in elems} | P_1(j) - P_2(j) | \quad \textit{Equation 2}$$

This can be considered a form of modified Hamming Distance[1] where the expected value of P(*) is 1 at each element.

For example, a SDED:

Original image P: {3001001004010100020012301240050020401200}

implies the original image had significant variability at positions 1, 11, 24, 28, 31 and 36 since the encoding is for 3, 4, 3, 4, 5 and 4, respectively, at those positions. Areas of such high variability are shown in Figure 2, where yellow bars are used to "underscore" them.

When this same IIO is imaged later, some variability in the encoding is likely, so that the following may be recorded:

Second image P: {0210100000500110003001130115006002030120}

The modified Hamming Distance between them is 12. The Hamming Distance (HD) between either of these and the completely random image:

Random image P:{1111111111111111111111111111111111111111}
happens to be 44 in each case. In general, HD for the same image captured at two different times and/or with different imaging devices will be < N (the length of the encoding), while those between unrelated images will be >= N The key is that there is a threshold between matching images and non-matching images that is difficult to replicate during the printing, and difficult to tamper with (without the tampering being obvious; for example, by creating quality grading failure, visible anomalies, etc.).

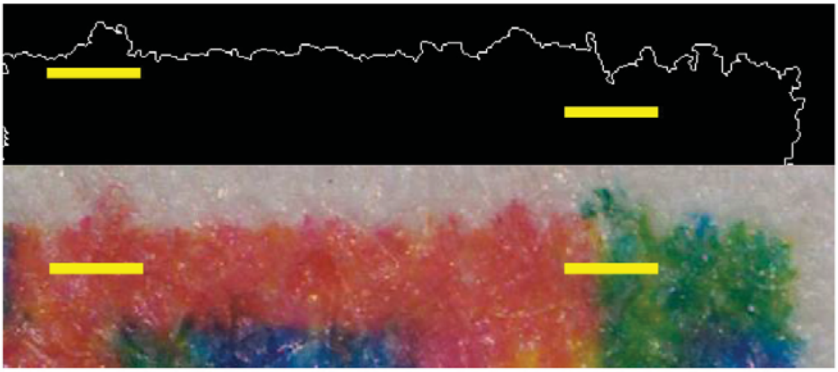

Figure 2*[5]*. *Areas of high variability along the edge of, in this case, a color IIO. The upper image shows the high-resolution perimeter of the IIO, while the lower image shows the original color IIO. In each case, the yellow bars underscore areas of high variability, that may be encoded with a "3" or higher in a piecewise encoding of the IIO suitable for computing the SDED later.*

SDED Applications

The workflow for the forensic markings being used for the IIO is therefore as follows:

1. Capture previous IIO while capturing the document and compute its P string, e.g. {03001001004010100020012301240050020401 20}.
2. Expand the IIO into a binary form. e.g. the above becomes (Huffman codes would be used in practice): {0111001001001111010100011001111110111111100111110011011110 1110}.
3. Use this string (together with any padding, deleting, scrambling, encryption, or other encoding techniques needed for the specific workflow and IIO type) to determine the new information for the IIO.
4. Archive the previous image.
5. Print the new document with the altered IIO (and other altered workflow data) included. Note that this new document has a cleaned-up IIO and a new forensic boundary, which is not reflected in the IIO (yet) in the scenario depicted.

Figure 3. *Default workflow for the IIO_{fm} and a workflow kiosk, such as an all-in-one.*

This approach resets the forensic information for the IIO boundary (or other portion of the IIO—the boundary is simply the easiest to illustrate here) each time there is a new document. The IIO can be changed with access to the previous document or the captured image by decoding the new information and comparing the P to the one from the captured image. If the computed difference, SDED, is less than the threshold, there is a match.

We also may wish to combine barcodes directly with the current forensics. In this situation, the reading device collects the forensics at the same time as the current image of the IIO, and so the incremental elements written to the IIO reflect the forensics of the original IIO, which "never change", because there is only ever one physical item printed to—such as a unique label, packaging, special document, etc. Here, encryption/signing of the incremental data is generally required. Note that the image registration issue here is non-trivial, although exactness is generally not a requirement for accurate barcode reading.

The above approach could also be used on a workflow kiosk such as an all-in-one or copier if a workflow-enhancing appliance or a dual path print is available. In this case, the base IIO and the rest of the document is printed, scanned, and the forensic descriptor collected. The next path is used to write the incremental portions of the IIO_{fm} so that the IIO_{fm} reflects the current forensics. This is a more user-involved, "2 pass" process, but is doable with the existing imaging technologies. The workflow kiosk requires special software/firmware to make this happen. Registration of the image during the second print here is facilitated by the page location/framing of the platen. The short steps to this approach are:

i. Print document with IIO updated to previous stage
ii. Capture IIO and compute its forensics, e.g. the sequence P
iii. Convert P into the additional IIO modules as described above. Note that all conversions, even in color, are possible here, since we can, for example, convert magenta to red by overprinting with yellow, etc.
iv. Run the page through the workflow kiosk (e.g. laserjet printer) once more and register and overwrite the IIO to make it the current IIO_{fm}. Registration here is aided by the pagefind/placement information coming from the scanning platen.

Conclusions & Future Work

We have shown how the SDED approach can be used to generate secure IIOs for document workflows. A series of experiments previously described[4] have been performed between false and valid images using a very high resolution scanning device. For this, the number of segments *N* and the atomic unit of coding was varied. The results showed that by using the SDED measure on between 50 and 400 samples of the string *P* (as described above) the best forensic security (at 200 samples) had a probability of false detection less than 10^{-9}. Encoding the position string P into the payload modules of the barcode would then provide authentication and forensic capabilities on the fly.

References

1. Hamming, Richard W. (1950), "Error detecting and error correcting codes", Bell System Technical Journal 29 (2): 147–160, doi:10.1002/j.1538-7305.1950.tb00463.x

2. International Standards Organization (ISO). 2006 A. 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.

3. International Standards Organization (ISO). 2006 B. International Standard ISO/IEC 18004:2006(E), Second edition 2006-09-15. Information technology - Automatic identification and data capture techniques - QR Code 2005 bar code symbology specification, 114 pp.

4. Pollard, S.B., Simske, S.J., and Adams, G.B.(2010). Model based print signature profile extraction for forensic analysis of individual text glyphs. Information Forensics and Security (WIFS), 2010 IEEE International Workshop on, Seattle, WA, December 12-15.pp.1-6.

5. Simske, S., Sturgill, M., Adams, G., & Everest, P.(2010). Document Imaging Security and Forensics Ecosystem Considerations. Proceedings of the DocEng '10 Proceedings of the 10th ACM symposium on Document engineering, Manchester, United Kingdom — September 21 – 24. pp. 41-50.

6. Simske, S., Vans, M., & Loucks, B.(2013). Incremental Information Objects and Progressive Barcodes. Journal of Imaging Science and Technology, 57(3), pp. 30405-1-30405-9(9).

7. Vans, M., Simske, S., & Loucks, B.(). Progressive Barcodes. NIP28: International Conference on Digital Printing Technologies and Digital Fabrication 2012, Quebec, Canada; September 2012, p. 368-370.