

Variable Data Void Pantographs

Steven J. Simske, Jason S. Aronoff and Margaret Sturgill, Hewlett-Packard Labs, 3404 E. Harmony Rd., MS 36, Fort Collins CO 80528, USA

Abstract

Void pantographs (VPs) have been an important part of the security printing toolkit for several decades. When precisely designed for a given printing technology, VPs provide an almost “magical” effect—they are nearly invisible in the original print and then stand out strikingly when they are copied. However, this effect comes at an expense—VPs have traditionally contained static content and so are usually fine-tuned for flexo, gravure, offset or screen printing.

In this paper, we extend variable data printing to VPs. First, we obtain optimal VP settings (background and foreground pattern—the background “disappears” when copied and the foreground “bolds” for the printer/copier pair). Next, we create a database of images for which we wish to create variable VPs (e.g. serialized numbers, individual labels, location-specific messages or images, etc.). We then apply an image filter to these images. Our most effective filters to date have been the edge-filter and the entropy-filter, both of which create a VP foreground where the information content of the image is high. For logos, using a chroma-filter is also often effective. Next, the individual VPs are written to the individual print elements (label, page, etc.) for which they are intended. The variable data message is now what “pops up” when the item is copied.

Keywords: Copy Evidence, Covert Printing, Image Analysis, VDP

Introduction

Void pantographs are used to create copy-evident backgrounds for a variety of security documents. Perhaps their best-known application is for use as backgrounds for checks, usually displaying “void” or “copy” on the reproduced image. There are a number of void pantograph approaches (see for example [1]), the most common being the use of two patterns, one of which “dilates” when copied and the other of which “erodes” when copied. In our experiments, we use a simple “dot size vs. percent black pixels” trade off to provide these two different types of patterns. It is possible to use other void pantograph constructions to achieve the same purposes. The purposes of this paper are to incorporate an image directly into a void pantograph-like representation, based on a selected attribute of the image. In so doing, the void pantograph can be used simultaneously for steganographic, copy prevention and security payload carrier roles. The main two elements of this work are (1) to explore how to embed a variable data image into a void pantograph, and (2) to use these image transformations for a form of watermarking (digital data hiding and encoding). The first of these introduces multiple filtering approaches which effectively “binarize” an image into

dilate (payload) and erode (copy-erase) parts. The second is concerned with the variable information in the payload itself.

Experiments Performed

The steps in the experiments performed here are:

- (1) Form a void pantograph from an image
- (2) Print, scan (copy) and perform segmentation on the void pantograph
- (3) Read the void pantograph information in (2.), and perform selected image-related task (as an example, a security-printing related task).

These steps are illustrated in Figure 1. We now describe these steps in more detail.

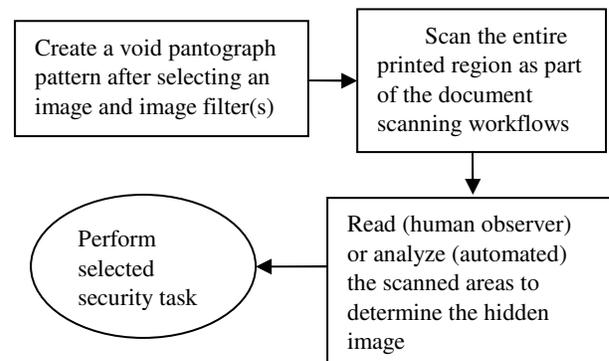


Figure 1. Outline of the steps performed

Step 1

Before producing the void pantograph, the image is filtered using one or more of the following filters:

- (a) Edge filter, such as the Canny [2] or Haralick [3] edge detectors.
- (b) Hue/chroma/color filter.
- (c) Saturation/brightness filter.
- (d) Intensity/luminance filter. This can be instantiated using a traditional luminance thresholder, such as Otsu [4] or Kittler-Illingworth [5].
- (e) Texture/specific shape filter
- (f) Local entropy/graininess filter
- (g) Sharpness/convolution filter

There are, of course, other possible filters for images. But this array suffices to address most of the variable aspects of images. These filters select areas (regions of interest) in the original image

to use for the void pantograph foreground (portion that shows after copying). Next, we describe the void pantograph so that it is clear what is done to those selected pixels.

Void Pantograph Construction

As mentioned above, we produced void pantographs simply through the use of differential dot sizes and differential black pixel concentrations. As we printed and scanned all patterns at 600 dots/inch (dpi), we used 2x2 pixel dots (foreground) and 1x1 pixel dots (background) for the dot sizes. Then, we varied the percentage of black ink coverage (accounting for differences in ink coverage and ink spreading due to printing) by varying the density of dot placement. We initially explored a background ink coverage percentage in the range of 5%-50%. We observed that 10% ink coverage for the background was a generally useful concentration—initially printed, they have a noticeable gray appearance suitable for “disguising” a void pantograph foreground, and brightness increases (i.e. the perceptual blackness diminishes) as the dots effectively disappear when copied (printed and scanned) affording a drop-out background suitable for testing the foreground patterns. A close up of a portion of a void pantograph using these two distinct dot sizes with equal black pixel concentrations (using a “random” dot placement to avoid Moiré patterns, etc.) is shown in Figure 2.

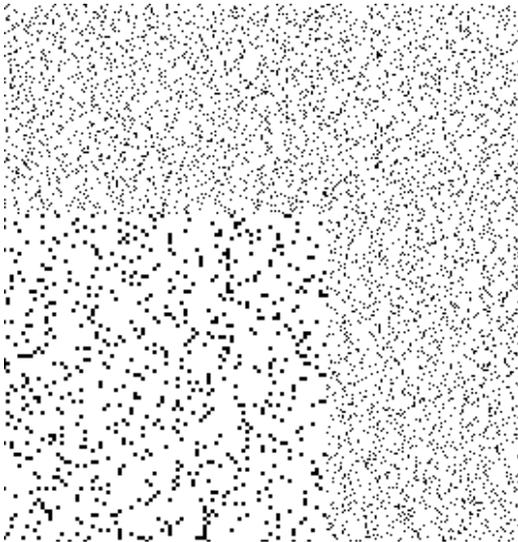


Figure 2. Same black pixel density (10%), different dot cluster size (2x2 at 600 dpi in lower left, 1x1 at 600 dpi in rest of image). Percent black pixels is 10% in each region, but the relative similarity in appearance depends on numerous factors, including scaling, printing technology used, etc.

Next, we considered different relative concentrations of foreground pixels to determine which combination of foreground and background works best with the given printer and copier/scanner. As an example of printer dependency, the dot gain after printing on inkjets tends to make the dots more similar in size than when printing using toner (e.g. Laserjets) due to ink spread into the paper fibers for the inkjet technology. Figures 3 and 4 show examples when different percentages of black are used for

the foreground (larger dots, bottom) and background (smaller dots, top).

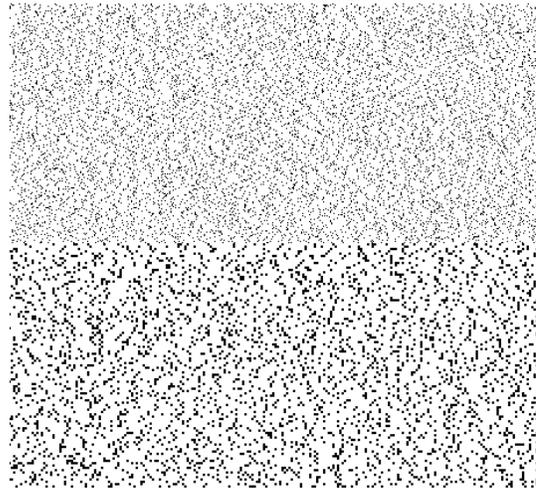


Figure 3. Different black pixel density (15% lower, 10% upper) for the same dot cluster size (2x2 at 600 dpi lower, 1x1 at 600 dpi upper) as Figure 2.

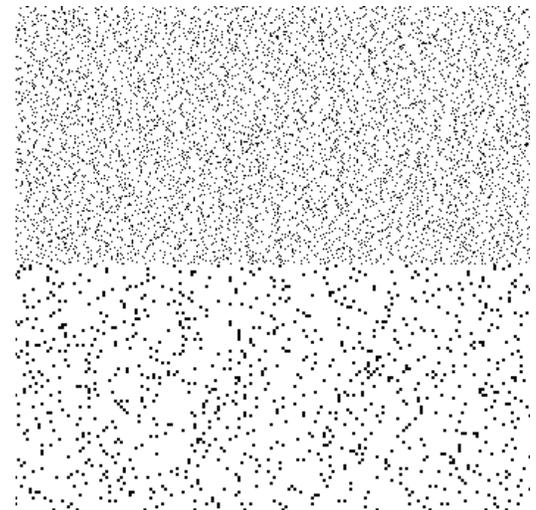


Figure 4. Different black pixel density (6.67% lower, 10% upper) for the same dot cluster size (2x2 at 600 dpi lower, 1x1 at 600 dpi upper) as Figure 2.

Figure 3 illustrates a 50% higher concentration of black pixels before printing for the larger (2x2 at 600 dpi) dots when compared to the smaller dots (1x1 at 600 dpi). Figure 4 shows the converse, where the smaller dots have a 50% higher concentration of black pixels than the larger dots. To find the optimum void pantograph densities for a given printer, the background black pixel concentration (e.g. 10% black pixels) is held constant while the foreground concentration is varied from, for example, 4% to 25% in 1% or less increments (40% to 250% as much). The combination that, after printing, looks the most similar to the human observer is the deployment candidate – meaning it is the combination of foreground and background dot concentrations to be used in subsequent steps.

Filtered Images as Void Pantograph Foregrounds

Once the void pantograph deployment foreground and background candidates are chosen, the actual image to be used is converted (scanned and transformed) into a void pantograph. The pixels (e.g. “edge pixels”) which are filtered by the image processing filter(s) described above are assigned to the void pantograph “foreground” and the rest of the image (e.g. “non-edge pixels”) are assigned to the void pantograph “background”. An example of this is shown in Figure 5, wherein salient (foreground) sections of the resulting void pantograph (lower image) were filtered using an edge detector operating on the upper image.

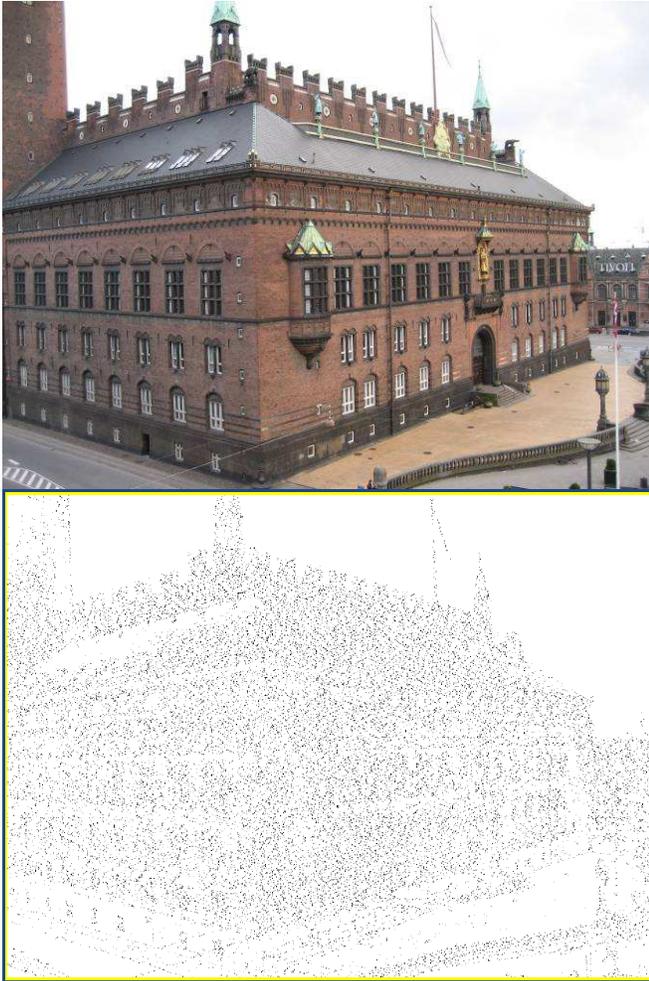


Figure 5. Original image (upper) and resulting void pantograph (lower) that “pops” out of the uniform grayscale background (shown in Figure 6) after copying.

Figure 5 illustrates the successful creation of a variable data void pantograph from an image. The upper image is the original image, and the lower image is the appearance of the void pantograph after its settings are optimized for the given printer/copier combination. The printed pixels correspond to areas selected by a nearest-neighbor edge filter based on a 3x3 pixel region surrounding each pixel. The output pixel $P_o(x,y)$ is determined from the sum of $|P(x+1,y+1)-(P(x,y))| + |P(x+1,y-1)-(P(x,y))| + |P(x-1,y+1)-(P(x,y))| + |P(x-1,y-1)-(P(x,y))|$ divided by 4.

We then use the Otsu threshold [4] on the histogram of $P_o(x,y)$ and the pixels above the threshold are added to the void pantograph foreground. The original void pantograph (foreground + background) after printing is shown in Figure 6. As another example, Figure 7 shows another image and the resulting void pantograph.

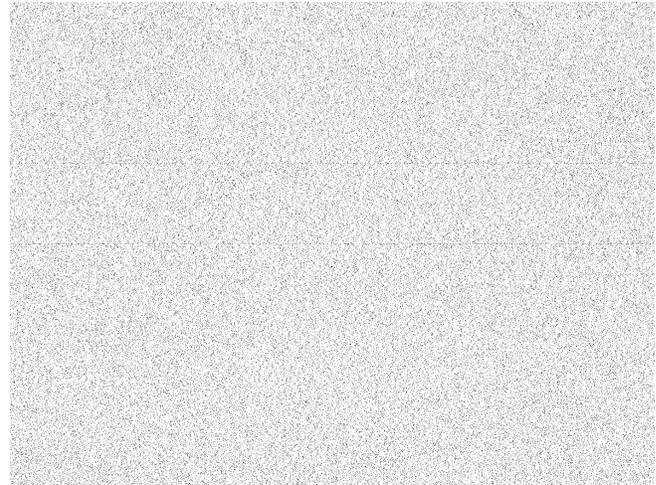


Figure 6. Original image (upper) and resulting void pantograph (lower) that “pops” out of the uniform grayscale background after copying. This void pantograph had its settings optimized for a Laserjet, and thus its appearance will vary depending on the device on which you are reading this paper, and if printed should show somewhat prominently on most (esp. non-Laserjet) printers.

Step 2

After scanning/copying, the void pantograph becomes visible (Figures 5 and 7) to the human eye. In addition to manual identification of the void pantograph by a human viewer, a void pantograph can be identified with existing segmentation software. The “information” in the void pantograph can be segmented (aggregated into usable regions of interest) as follows:

- Threshold the image (this “binarizes” it, leaving the ink areas black and the non-ink areas white)—this results in an image like that shown in the lower images of Figures 5 and 7.
- Perform erosion of the resulting connected components, as needed. This completely erases the small dots and shrinks, without erasing, the large dots. Generally, 1-pixel boundary erosion suffices for this.
- Perform dilation of the remaining connected components. This returns the larger dots to their original size (but does not reconstitute the small dots, since they have been erased).
- Form regions of interest (connected components) from the remaining dots, if needed for Step 3. Here, run length smearing is used to cluster the dots left over into their original associated shapes or forms [6].

Step 3

The regions of interest from Step 2 are then analyzed in one or more of the following ways:

- Human evaluation. Does the void pantograph image (VPI) look good? Does it, for example, match the look/feel of the

original image (or “OI”)? Are there areas (perhaps intentionally) left out?

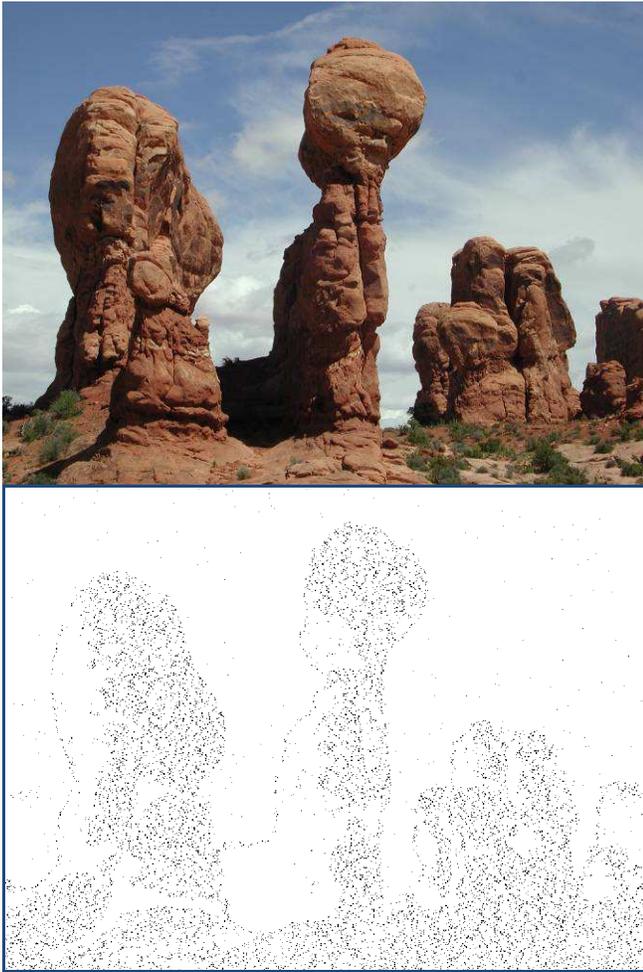


Figure 7. Original image (upper) and resulting void pantograph (lower) that “pops” out of the uniform grayscale background (shown in Figure 6) after copying.

b. Machine evaluation. The machine can directly analyze the resulting void pantograph. Simple examples are where the void pantograph contains specific text, shapes, or glyphs. If, for example, specific text is included in the OI, then OCR (optical character recognition) of the VPI is used for interpretation. If a specific shape is included in the OI, shape analysis of the VPI is used for interpretation. Any other type of pattern recognition suitable for the embedded pattern can be used at this stage, including glyph analysis. These patterns can be scanned, “emerge” from the void pantograph when it is copied, and then be directly analyzed using the appropriate inspection and/or authentication algorithm(s). Sample glyphs that can be included and readily translated (with varying payload densities, dependent on the particular void pantograph instantiation, printer, scanner and

substrate) are 1D, 2D and 3D (multi-gray level) bar codes, graphical alphanumeric (code sets where different glyphs represent different alphanumeric characters), and curvature-based glyphs (e.g. guilloches).

c. Comparing the VPI to the original image for authentication (if a secure connection to a back-end server is available) or for detection of (intentional) changes in the VPI consistent with mass serialization, data hiding, etc.

Other security-related information can be encoded directly in the VPI. Watermarks, copy detection patterns and line/halftoning/Moiré patterns can be scaled and represented in the VPI. Many other workflows can be enabled, since the approach outlined allows any printed information to be hidden in plain sight (assuming the void pantograph is of sufficient size).

Discussion and Conclusions

We have shown how variable data images can be readily incorporated into void pantographs. These pantographs form an effective covert-to-overt bridge for security printing. Importantly, the approaches outlined can be tuned to any printer used to create the void pantographs. Any image can be “filtered” to produce a void pantograph. Our examples show a city scene and nature image, but the approach is just as readily applicable to logos, graphics and text, allowing a machine-readable “payload” to be included as described in Steps 2 and 3. This extends the utility of void pantographs to automated, machine vision applications such as optical character recognition and shape recognition.

References

- [1] Arcis Digital Security, <http://arcisdigitalsecurity.com/>, last accessed 24 June 2011.
- [2] J. Canny, “A computational approach to edge detection”, *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol 8, pages 679-714 (1986).
- [3] R. Haralick, “Digital step edges from zero crossing of second directional derivatives”, *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 6(1):58–68 (1984).
- [4] N. Otsu, “A threshold selection method from gray-level histograms,” *IEEE Trans. Syst. Man Cybern.* Vol. SMC-9, No. 1, 62-66 (1979).
- [5] J. Kittler and J. Illingworth, “Minimum error thresholding. *Pattern Recognition*,” Vol. 19, No. 1, 41-47 (1986).
- [6] F.M. Wahl, K.Y. Wong, and R.G. Casey, “Block segmentation and text extraction in mixed/image documents,” *Computer Vision Graphics and Image Processing*, Vol. 2, pp.375-390 (1982).

Author Biography

Steve Simske is an HP Fellow and the Director and Chief Technologist of the Document Lifecycle & Security Printing & Imaging portfolio in Hewlett-Packard Labs. Steve is currently on the IS&T Board. He is also an IS&T Fellow and a member of the World Economic Forum’s Global Agenda Council on Illicit Trade. He holds more than 40 US patents and has more than 250 peer-reviewed publications. He holds advanced degrees in Biomedical, Electrical and Aerospace Engineering.