Using a Data-bearing Frame to Capture an Extended Target

Robert Ulichney, Matthew Gaubatz, Chun-Jung Tai, Stephen Pollard, Melanie Gottwals, Ingeborg Tastl
HP Labs, Hewlett Packard, Inc., USA

Abstract

A system for the automatic detection of the entirety of an arbitrary image target is described. This is achieved by surrounding the target image with a data-bearing halftone or "StegaFrame" and by a camera-based detection system that can efficiently find the StegaFrame by means of judiciously placed detection windows. The frequency-domain-based detection scheme is fast and provides the added advantage of estimating scale and orientation of the desired content. An intuitive user interface guides the user until a successful detection is achieved. This approach is demonstrated in an application to locate and analyze a color chart for the purpose of calibrating a mobile device to be able to measure skin tone. An identifier to access print-run specific colorimetry data, along with other data for client analytics is embedded in the data-bearing border. The design of the detection system is optimized through exhaustive simulation. Experimental results confirm that the automated scheme, implemented on a mobile phone, performs as well as costly dedicated colorimetric hardware.

Introduction

There are numerous techniques which can be leveraged to flag importance of, embed data in, or associate information with a sample of graphical content. The specific problem we are concerned within this paper is the ability to identify the presence, in entirety, of an arbitrary selection of graphic content. Many analysis routines are initiated by the capture of such a desired target. Targeted inspection, forms processing and product identification are but three examples. While this result can be achieved with a process where a user makes a judgment of acceptability and manually invokes subsequent processing, our goal is to automate and therefore simplify the user-interface required to enable such an image capture operation.

Barcoding techniques and watermarking schemes offer two types of solutions with very different properties. Barcodes can be interpreted by a plethora of devices at the expense of yielding graphic design flexibility. Watermarking techniques [1][2], on the other hand, can be used, for example, to associate information with graphical with, in most cases, minor visual artifacts. For some detection and analysis tasks, however, neither approach is suitable; barcodes can be intrusive, if not disallowed from use, and some watermarking schemes (such as those based on modifying wavelet or frequency coefficients) do not provide much localization information. In addition, neither of these options may provide suitable data density given the amount of space that can be used for this purpose. Other alternatives include image matching algorithms such as the Viola-Jones object detection framework [3], SIFT-based feature detectors [4], and SURF-based feature detectors [5]; however these solutions must be designed for each specific target, they are computationally expensive, and they cannot provide accurate feedback about orientation and scale corrections.

This work discusses a different style of solution, whereby the signature of a halftone structure designed to carry data, is imposed upon a rendering of a border around the content of interest. The border serves multiple purposes. First, it draws attention to the fact that the content is of/should be of interest to a user. Second, it provides a signature of its presence strong enough to be detected automatically. Third, it can be used to represent any data critical to a given application. In this manner, it becomes possible to construct a completely generic object detection, or in this case, corralling scheme. The approach is demonstrated in an application involving a color characterization chart. When successfully captured, the image data of the color chart combined with the corresponding colorimetric data can be used to measure the color of a sample, such as skin tone. Fortified with this feature, we developed an accurate mobile color measurement scheme that can replace more costly hardware used for the same purpose.

Detecting and Corralling an Object

We have developed a system for corralling an object with a "StegaFrame", a steganographic halftone or stegatone [6] of a fixed gray frame that surrounds a target. Data is embedded by means of single-pixel shifts of halftone clusters. We exploit the close-focus ability of smartphone cameras that deliver the resolution necessary to interpret the high capacity encodings. Measurement of the effective resolution of most current smartphones as a function of distance is plotted in Figure 1. It is important to point out that this data is for video capture; full-sensor photo capture offers much higher resolutions but using it does not often result in a fluid user interface.

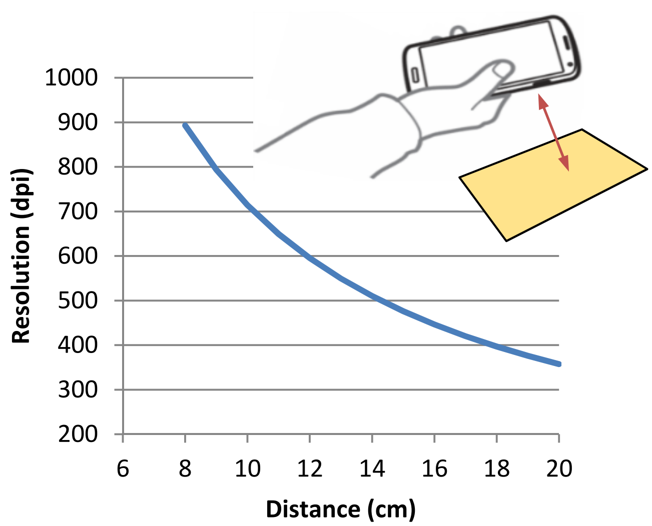

Figure 1. Smartphone video capture resolution.

An example StegaFrame, in this case used to corral color characterization patches, is shown in Figure 2, along with a depiction of how it is used for measuring the skin tone with our mobile app. 72 color patches surround a hole through which the skin sample is measured. The color chart is surrounded by a StegaFrame used for automatic detection. The 2.2 by 3.6 inch frame appears as a gray border but is comprised of 4428 halftone clusters each carrying 2 bits of data. With a redundancy factor of 4 this results in 2214 data payload bits. To account for occlusions even higher redundancy can be used.

Figure 2. The StegaFrame enclosed color calibration target and its use for capturing a skin tone sample.

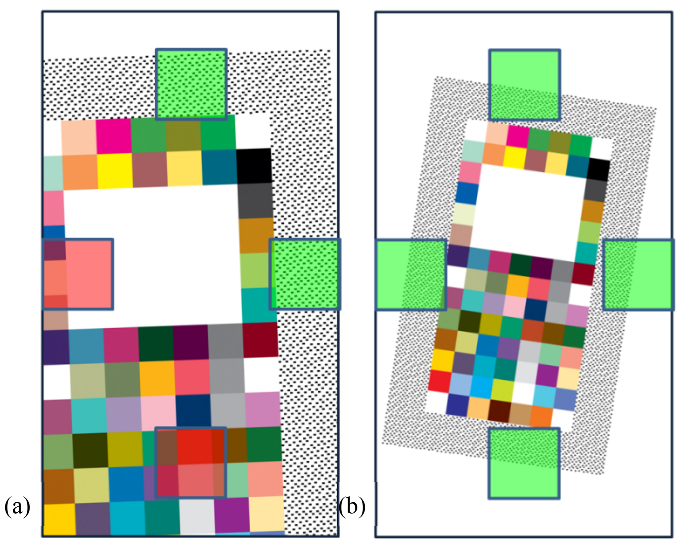

Figure 3. Smartphone video display and UI.

An example of our StegaFrame detection system UI is illustrated in Figure 3. Video is captured on an imaging device and four detection windows are positioned within the capture image to allow a range of distances at which the StegaFrame and inner chart can be captured. The detectors can, quite effectively and efficiently, identify the clustered-dot halftone pattern and determine the associated scale and orientation of the pattern within the detection window [7]. Detection windows are colored red in the overlaid display when the pattern is not detected and colored green when it is. Figure 3(a) illustrates a case where the camera is too close to the target. Two of the detection windows are red because they do not cover any part of the StegaFrame. The green windows properly detect the StegaFrame but also detect that the scale is too large (i.e., that the imaging device is too close to the target) and could report that back to the user. In Figure 3(b), the frame is detected in all four windows, the corresponding video frame is stored for processing, and the user is informed. In practice, only a subset of the correctly scaled positive detections are needed, which enables the approach to address partial occlusions.

Real-time detection is accomplished by analyzing the Discrete Fourier Transform (DFT) amplitude and searching for peaks that are characteristic of the quasi-periodic stegatone pattern. To illustrate a typical example of the frequency domain signature associated with the border, consider the right-most detection window in Figure 3(b) as detailed in Figure 4.

Figure 4. Detail of the pattern captured in the right-most detection window in Figure 3(b).

The DFT magnitude of that detection window is shown in Figure 5. Even though only part of the detection window includes StegaFrame samples, the DFT is dominated by the energy from this halftone pattern as seen by the four peaks. To accentuate the presence of these peaks we mask the DC area that can be seen as the central disc. Also, the effect of windowing imparts strong harmonics on both axes, so we mask those areas as well. Figure 6 illustrates how detected peak positions to measure the distances H and V from which the horizontal and vertical scale are found, along with the rotation angle θ. The peaks in Figure 5 reveal that there are 3.43 horizontal samples/cluster, 3.41 vertical samples/cluster, and that the target is rotated by 8.49 degrees. Thus, the DFT of the contents of each window can be used to quickly estimated if the StegaFrame is present, along with an accurate determination of both scale and orientation angle. The combined results from all detection windows is used to establish if the target is successfully captured.

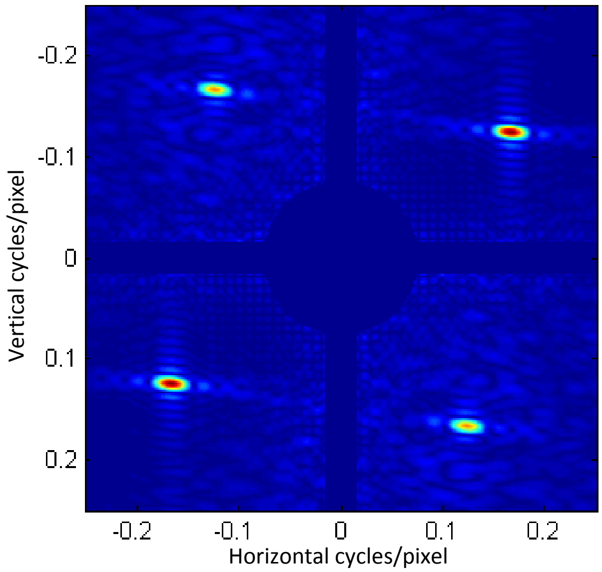

Figure 5. DFT of the rightmost detection window in Figure 3(b).

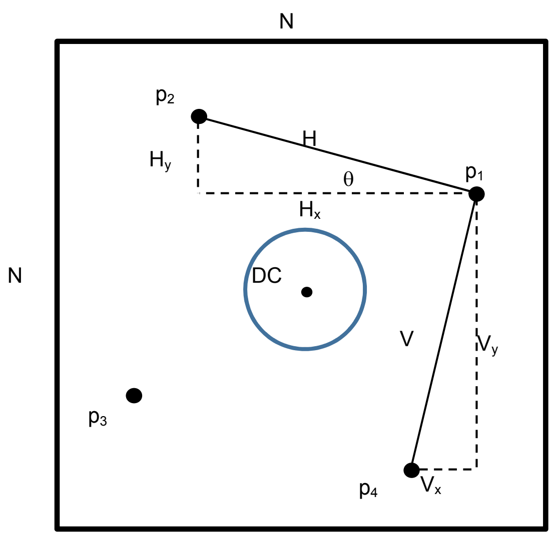

Figure 6. DFT peak locations.

Exhaustive Simulation

Unlike lab measurement devices that can be carefully positioned and fixed in a location relative to a target to be captured, portable imaging systems may be hand-held and positioned by users in non-advantageous lighting conditions. The variability of capturing conditions must be accounted for in order to ensure both a fluid user experience and high quality results.

Detection performance is measured in terms of correct or full detections (true positives), incorrect or partial detections (false positives), missed detections (false negatives) and cases of no detection when the object is not fully present (true negatives). Examples of each of these four states are illustrated in Figure 7. A true positive results when the whole of our corralled target is captured in the video frame and detection is indicated; a false positive occurs when the system indicates a detection and the whole of the target is not captured. False negatives events occur when the whole of the target is captured but detection is not indicated. True negatives are simply a no detection event when all or any part of the target is outside the video frame. The goal herein is to produce the largest number of true positives, with the least amount of false positives and false negatives.

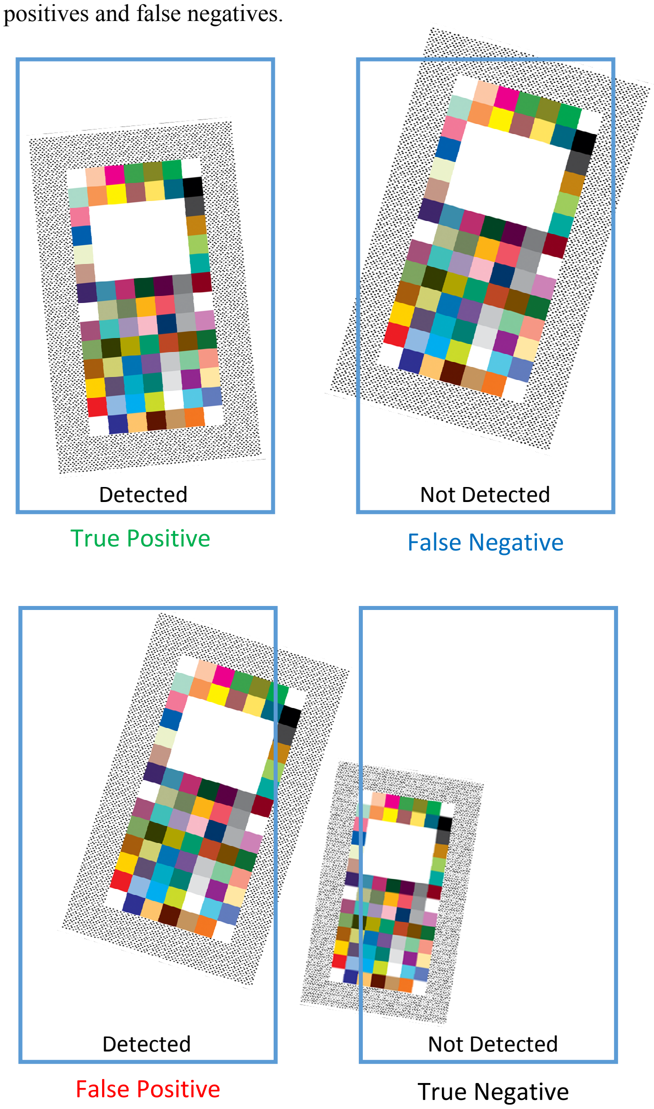

Figure 7. Examples of the four possible detection states.

Results vary considerably with design choices that include the number of detection windows, their size, and position. Testing can be tedious because of the large number of target positions that includes scale (distance from the camera), orientation (rotation angle), and translation. To remedy this we built a software simulator that exhaustively tests all translation positions for each scale and orientation and reports the total number of true positives, false positives, true negatives, and false negatives.

We define scale to be the ratio of the width of the StegaFrame to the width of the video frame. Both the target and the video are nominally oriented in portrait mode. For each discrete scale and angle of orientation we step through all translations in raster order as shown in Figure 8. Test parameters include the resolution of the video, the horizontal and vertical translation step size, the range and step size of the scale and orientation. We also set the percentage of the StegaFrame under a window that is required to trigger a detection based on experimentation, and the shape and size of the corralled object (the array of color tiles in this case). As for the number of detection windows, we tested the configurations shown in Figure 9 of 4, 6, and 8 windows with parameters being their size and margin from the edge of the video frame.

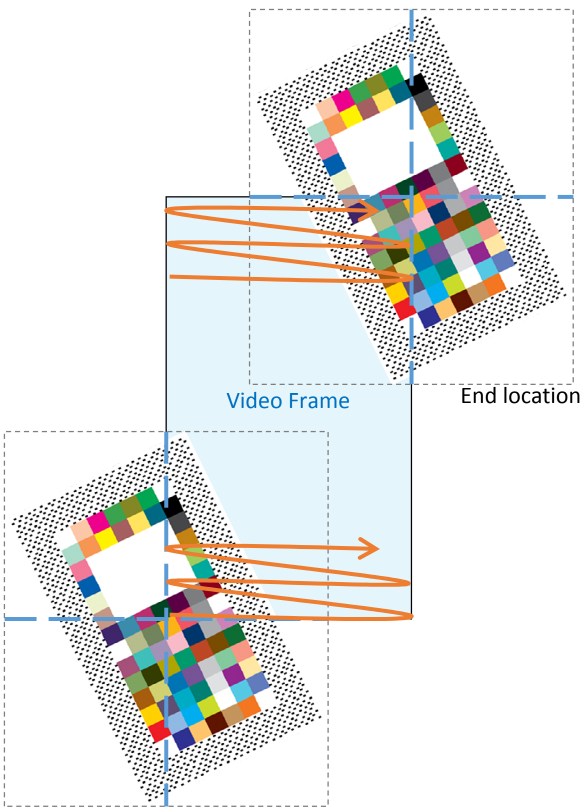

Figure 8. Stepping through all translations in raster order.

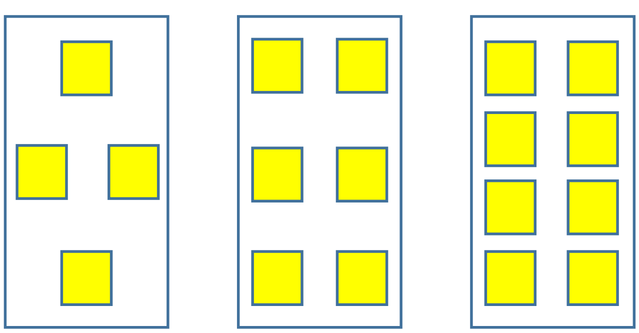

Figure 9. Placement of 4, 6, and 8 detection windows.

For a given choice of system design parameters, we visualized the results as shown in Figure 10. In this case we tested 6 detection windows of size 220x220 for a video resolution of 960x540 pixels with a margin of 7% separation form the edge of the video frame. True negatives have the most number of occurrences but are not nearly as important to the overall system performance as the other results. For this reason we do not report true negative counts. We tested scales ranging from 0.7 to 1.1 in increments of 0.05, and orientation angles from 0 to 45 degrees in increments of 5 degrees. At each scale and angle the total number or non-true negatives from our exhaustive testing is depicted as the size of a small pie chart. Each pie chart shows the relative number of true positives (green), false positives (red), and false negatives (blue). The chart in Figure 10 also reveals what combinations of scale and orientation work well, and what combinations do not. A priority is the avoidance of false positives.

It is important to be able to filter out candidate images that are likely to produce an incorrect detection result. This problem is easy to address do because the Fourier-domain-based detector estimates the scale and orientation. The diagonal line in Figure 10 is one way to segment the scale-angle space into valid and rejected regions. All points above and to the right of the line are rejected because of the high percent of false positives (red).

The collective results for the design tested in Figure 10 (6 detection windows of size 220x220) is the sum of all the true positives, false negatives and false positives minus those cases in the rejected region. These sums are part of the plots shown in Figure 11 which show the collective results for several such experiments over a range of detection window sizes and number of windows. The total number of true positives from Figure 10 are indicated by the arrow. That particular design choice of 6 windows and 220x220 window size turns out to be a good combination of high true positives, and low false positives and false negatives.

Application for Skin Tone Measurement

To process the stored video frame we first align the StegaFrame, then recover the embedded data. For our application we store an index to the print run that generated the color charts. As the colorimetry of the characterization charts vary between print service providers as well as from one print run to the next, the colorimetry of the charts is spectrophotometrically measured and stored for each print run and associated with the index in the StegaFrame for that run. In addition to this, data useful for subsequent analytics can be stored, such as name and location of store where the chart was being provided. The corners of the chart are used to align and measure each color patch. This information is used together with the stored colorimetry data to correct the raw camera data captured inside the hole of the chart and finally provide a skin tone measurement.

Color measurements are traditionally performed with devices that range in cost from several hundred to tens of thousands of dollars [8]. Without the benefit of a dedicated color sensor, color estimation is quite inaccurate due to a number of factors, most notably camera pipelines that are optimized for pleasing and not accurate images as well as broad variation of illumination environments [9]. The use of mobile phones for skin tone measurement was pioneered in 2008 [10]. In this experiment, skin tone measurements were performed for 12 people of different ethnicities using our mobile app to be compared with a 3rd party dedicated skin tone measurement device called Capsure from X-Rite. The values were compared against an averaged ground truth (5 measurements taken with a tele-spectrophotometer) per person.

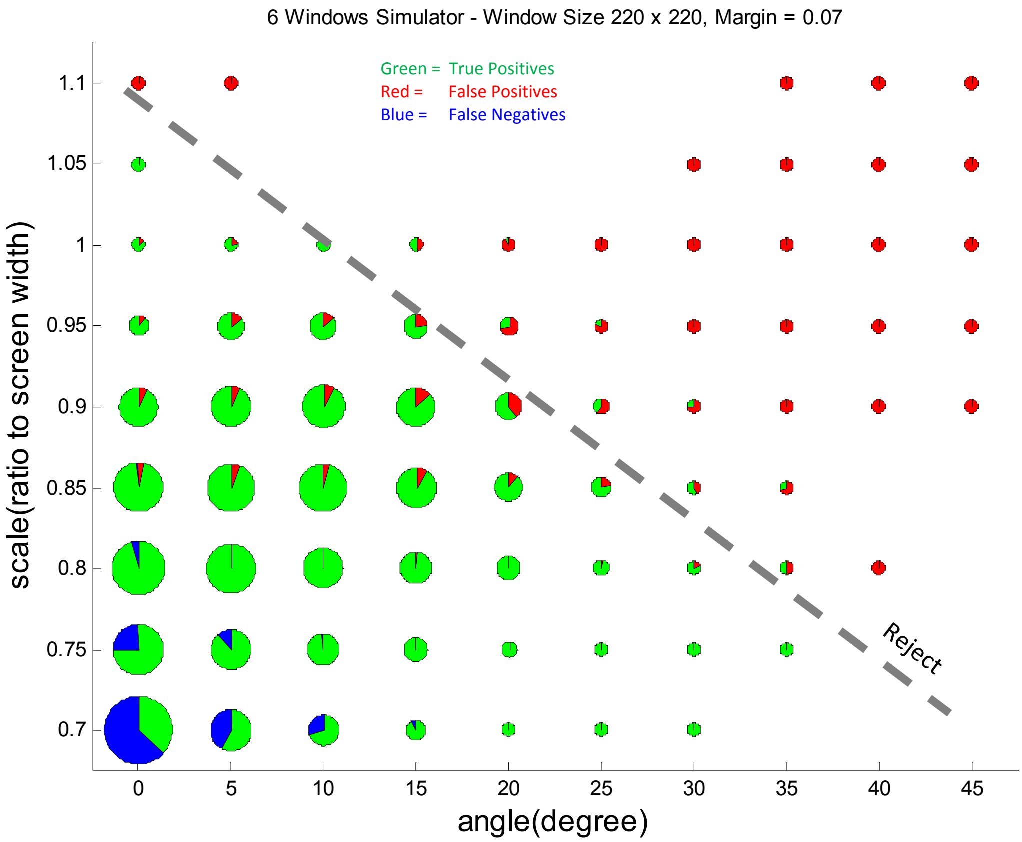

Figure 10. Results for one particular number of windows (6) and size (220x220).

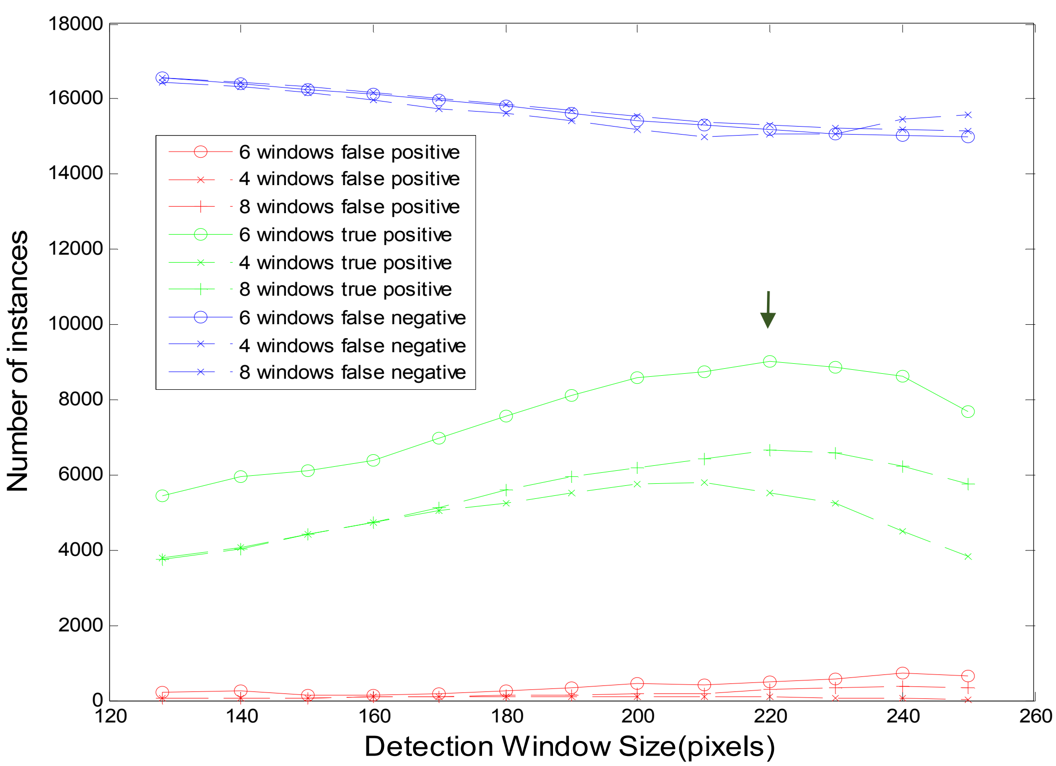

Figure 11. Collective results for a range of design choices.

The results depicted in Figure 12 show that our color technology has a comparable accuracy to X-Rite's dedicated skin color measurement device. The red curve shows the much larger color errors resulting from using raw smartphone camera RGB values.

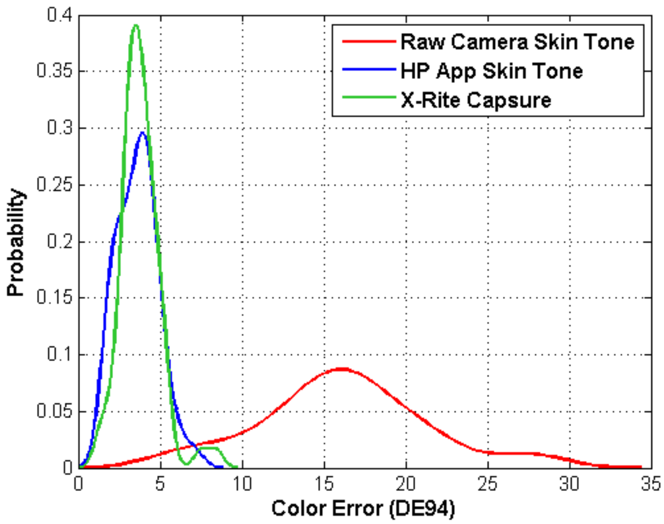

Figure 12. Measurement error comparison.

For this particular application, the Stegaframe is used to ensure that all the individual color patches as well as the skin tone sample are properly captured and that the location of the corners of the rectangle enclosing all the color patches is communicated back to the skin tone estimation algorithm. PSP and the print run specific identifiers are encoded in the Stegaframe associated with the color chart and recovered automatically. Based on that the colorimetric information corresponding to charts in the specific print run can be retrieved and used to correct the camera raw data obtained for the skin tone captured within the hole of the color chart and enabling an accurate skin tone estimation. The variability of the colorimetric output from one PSP to the next and from one print run to the next is simply too high for this application to operate without suitable calibration. Thus an average data set captured at one time and then being reused for future print runs would significantly degrade the accuracy of the skin tone estimation. Thus, being able to retrieve a data set that is associated with a specified print run is essential.

Concluding Remarks

A StegaFrame detection system was analyzed via exhaustive simulation, and implemented on both iOS and Android mobile devices. Processing speeds allow for the computing for desired signatures in all detection windows at a rate close to 20 video frames/sec to deliver real-time results. To exploit the efficiency of the FFT algorithm we zero-pad the data from our detection windows to fill a power-of-2-size array. The StegaFrame is an efficient solution to easily detect the whole of an extended target with a mobile device without unsightly fiducials, while doubling as a means for high capacity data embedding.

References

[1] P. Korus, J. Bialas and A. Dziech, "Towards Practical Self-Embedding for JPEG-Compressed Digital Images*," IEEE Transactions on Multimedia*, Vol. 17, No. 2, 2015.

[2] Q. Liu and J. Ying, "Grayscale Image Digital Watermarking Based On Wavelet Analysis," *IEEE EEESYM*, 618-621, 2012.

[3] P. Viola and M. Jones, "Robust Real-time Object Detection," *ICJV*, 2001.

[4] D. G. Lowe, "Object Recognition from Local Scale-Invariant Features," *Proceedings of the International Conference on Computer Vision 2*, 1999.

[5] H. Bay, A. Ess, T. Tuytelaars, L. Van Gool "SURF: Speeded Up Robust Features", *Computer Vision and Image Understanding (CVIU)*, Vol. 110, No. 3, 2008.

[6] R. Ulichney, M. Gaubatz, and S. Simske, "Encoding Information in Clustered-Dot Halftones", *IS&T NIP26 (26th Int. Conf. on Digital Printing Technologies*), Austin, TX, 602-605, Sep 2010.

[7] R. Ulichney, S. Pollard, M. Gaubatz, "Fast Mobile Stegatone Detection using the Frequency Domain", *IS&T NIP30 (30th Int. Conf. on Digital Printing Technologies)*, Philadelphia, PA, Sep 7-11, 2014.

[8] P. Green, L. MacDonald, *Color Engineering – Achieving Device Independent Color*, Chapter 2 – "Instruments and Methods for Color Measurements", by D. Rich, and Chapter 9 – "Color Processing for Digital Photography", by J. Holm, I. Tastl, L.Hanlon and P. Hubel, Wiley 2002.

[9] R.W.G. Hunt and M.R. Pointer, *Measuring Colour – Fourth Edition*, Wiley 2011.

[10] J. Jain, et al., "Color match: an imaging based mobile cosmetics advisory service", *Proceedings of the 10th International Conference on Human Computer Interaction with Mobile Devices and Services*, pp. 331-334, 2008.

Author Biography

Robert Ulichney is a Distinguished Technologist with HP Labs focusing on systems for high capacity data embedding, and structures for variable density 3D printing. He received a Ph.D. from MIT in electrical engineering and computer science. Before joining HP he was with Digital Equipment Corp for several years then with Compaq's Cambridge Research Lab where he led a number of research projects on image and video implementations for both hard copy and display products.