

Mobile Color: From Colorimetric Estimations to Calibrated Pages

Nathan Moroney
HP



Acknowledgements

- **This presentation is based on contributions from**
 - Udi Chatow
 - Gary Dispoto
 - Melanie Gottwals
 - Wei Koh
- **Nina Bhatti lead efforts to develop earlier versions of this technology at HP Labs**
 - "Cell phones as imaging sensors", N. Bhatti, H. Baker, J. Marguier, J. Berclaz, and S. Süssstrunk, Proc. SPIE 7708, Mobile Multimedia/Image Processing, Security, and Applications (2010)
- **This research is part of the Printing & Content Delivery Lab of Hewlett-Packard Laboratories**

Outline

- **Brief Introduction**
 - Technology Context
- **Core Concept**
 - Mobile Colorimetric Estimation
- **Discussion of Components**
 - Broader System
- **Mobile Color Variable Data Printing**

Brief Introduction

- Commercial Print Automation Project
- Research includes developing intuitive and automatic color reproduction for digital commercial print



Technology Context

- **Research context for the technology**
 - Commercial print theme
 - Printing and Content Delivery Lab
 - Automation technologies
 - Color reproduction key quality consideration
- **Industrial context for the technology**
 - Mobile sensing, processing and platforms
 - Advanced image analysis and feature detection
 - Ongoing need for color accuracy and consistency
 - Increasingly powerful smart-devices

Obligatory Color is Everywhere Slide



Core Concept

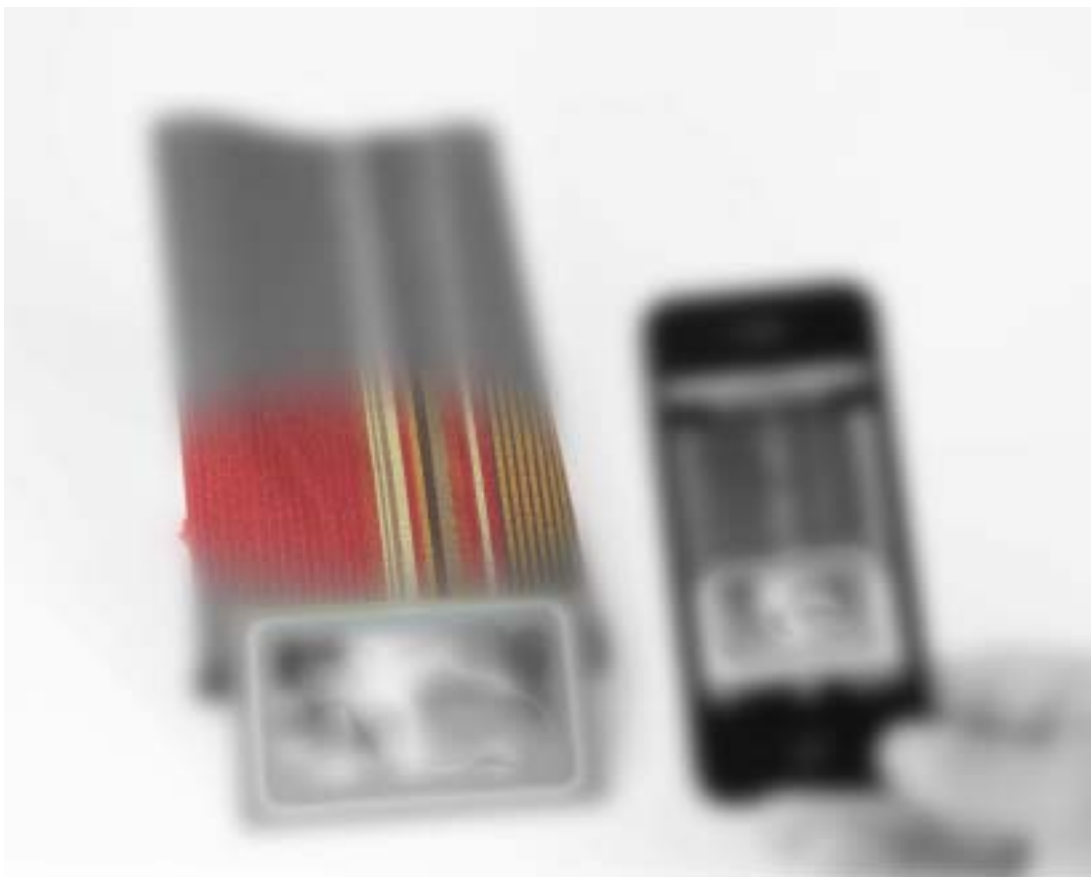
- What if we calibrated the scene instead of the camera?





Core Components

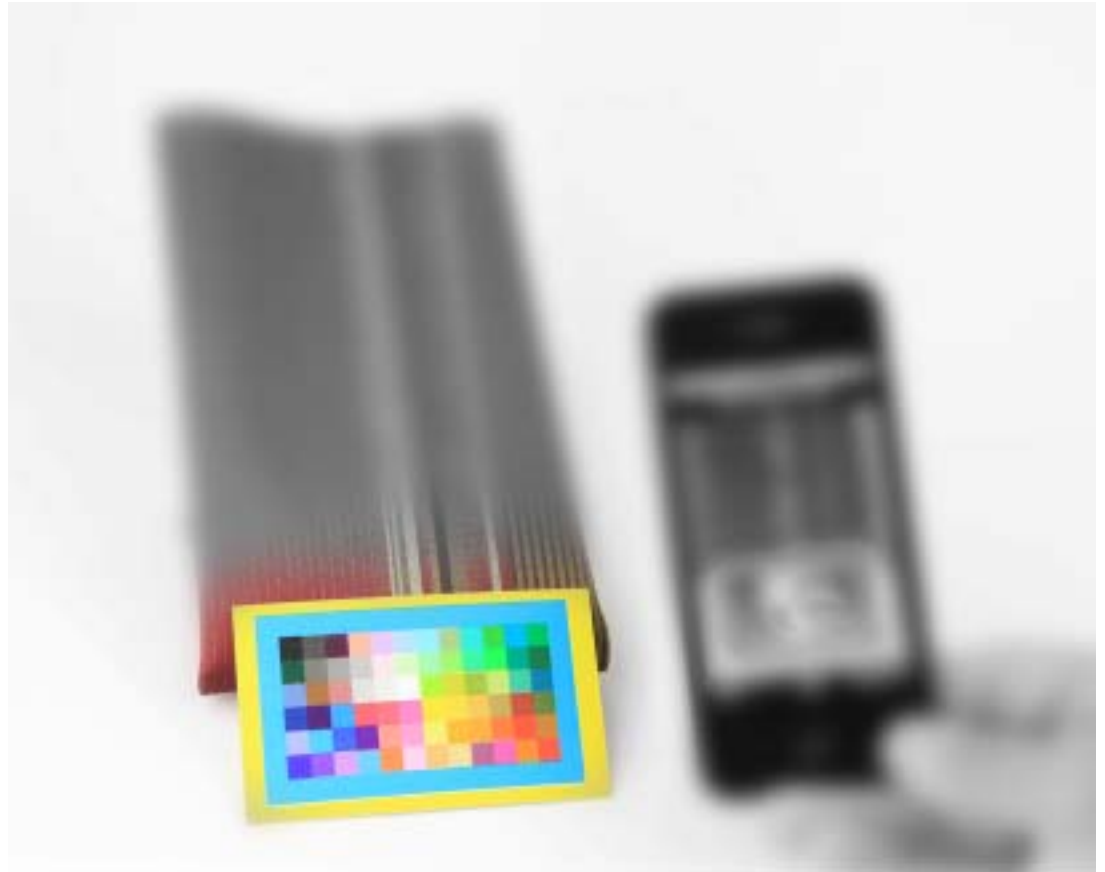
- Starting with an unknown color or colors of interest in a scene





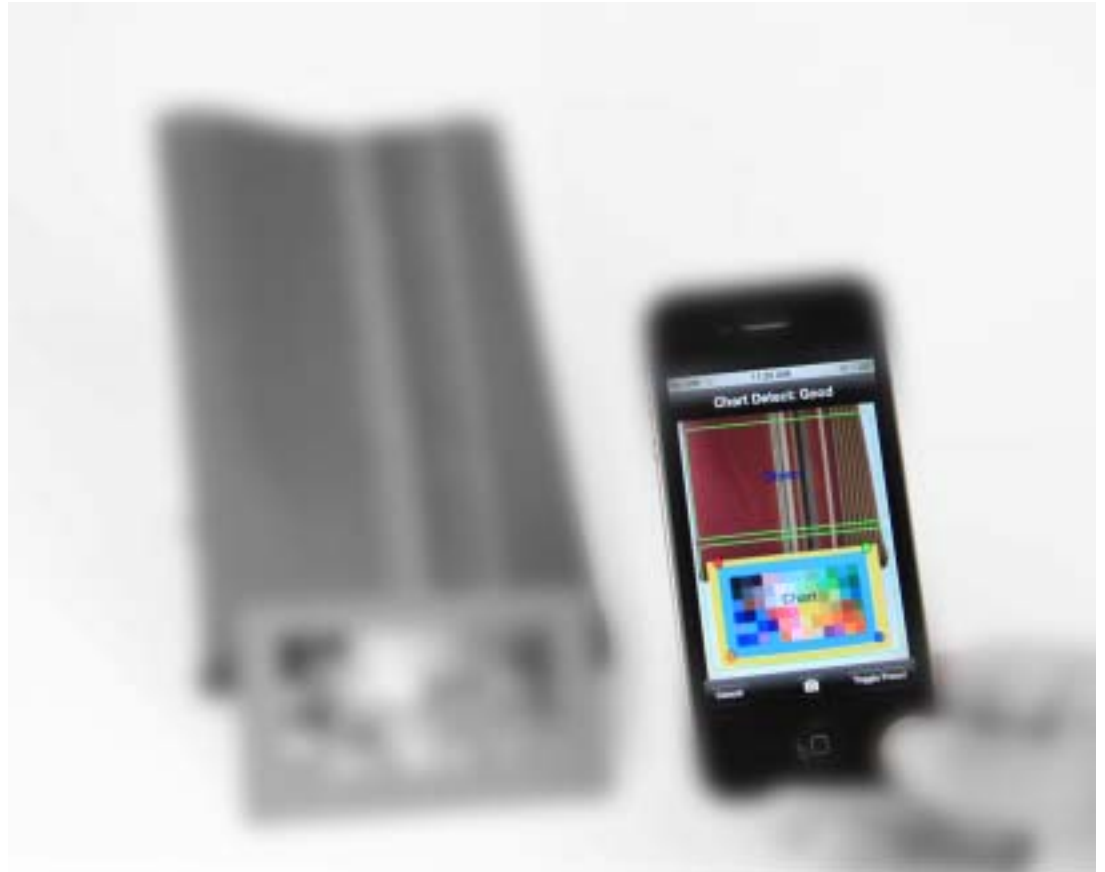
Core Components

- Introduce a carefully designed and produced color reference chart



Core Components

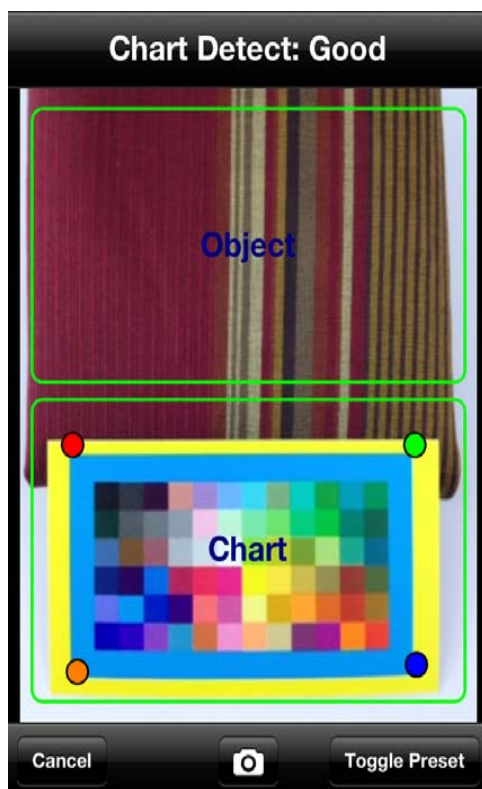
- Use a smart device to photograph the color of interest and use the chart to directly estimate the colorimetry of the color of interest



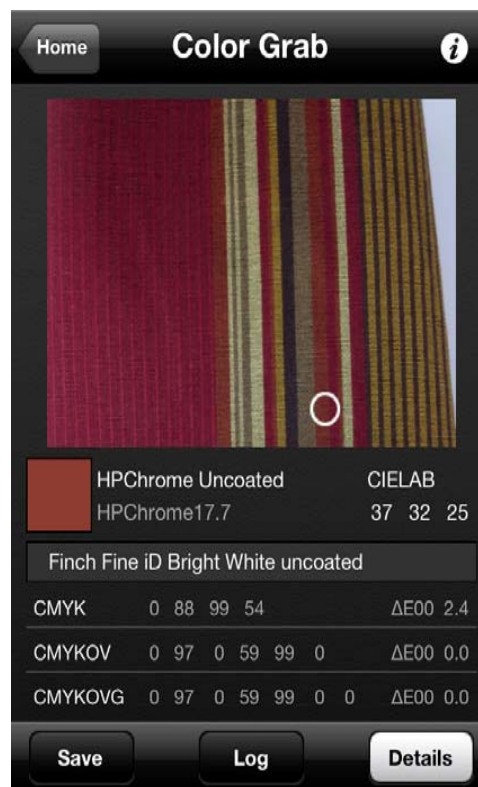


Notes

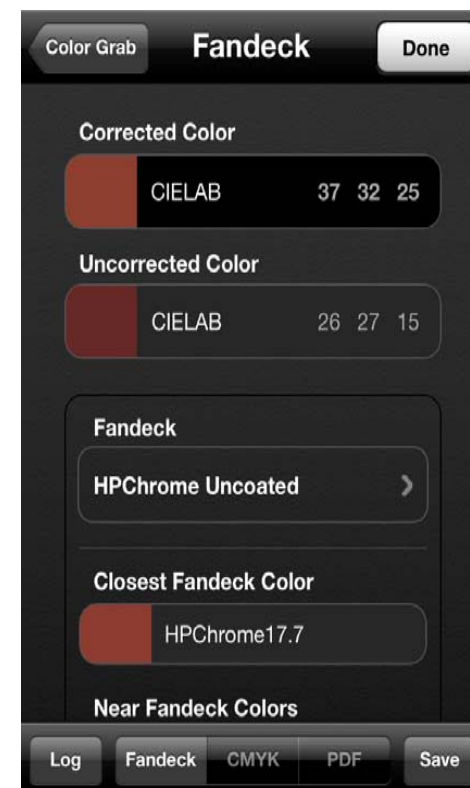
- Display is ignored
- Camera rendering is (mostly) ignored
- Photographic rendering of scene is (mostly) ignored



1. Capture Object and Chart



2. Select Region of Interest



3. View Corrected Color



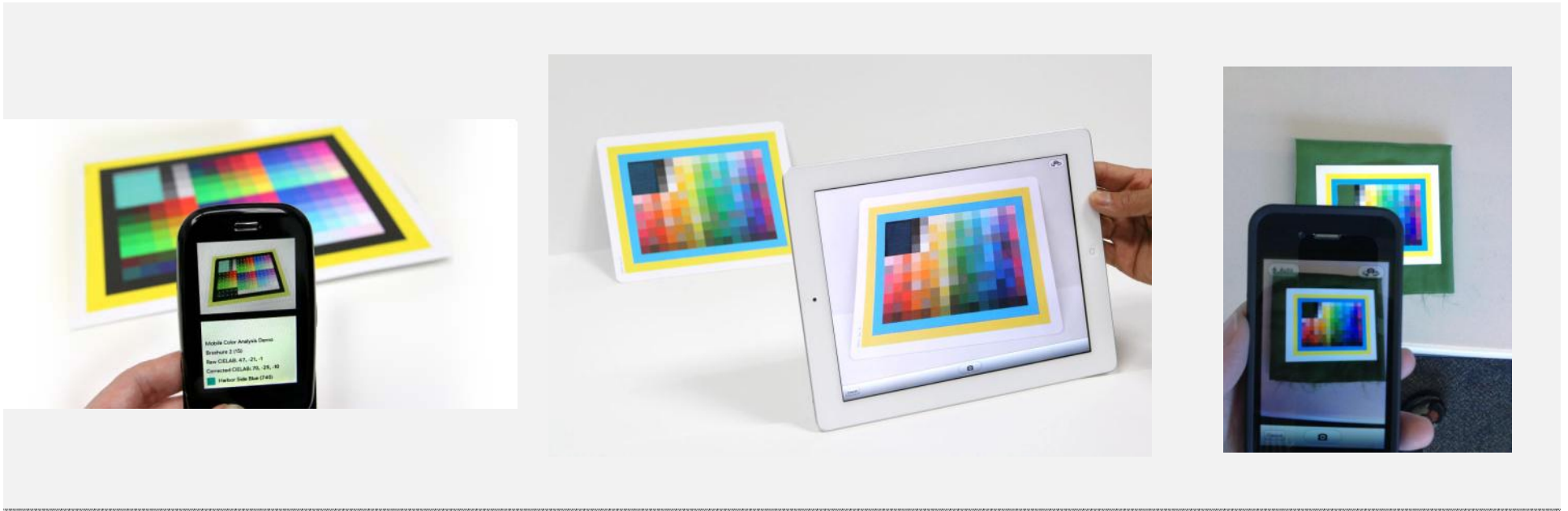
Color Chart



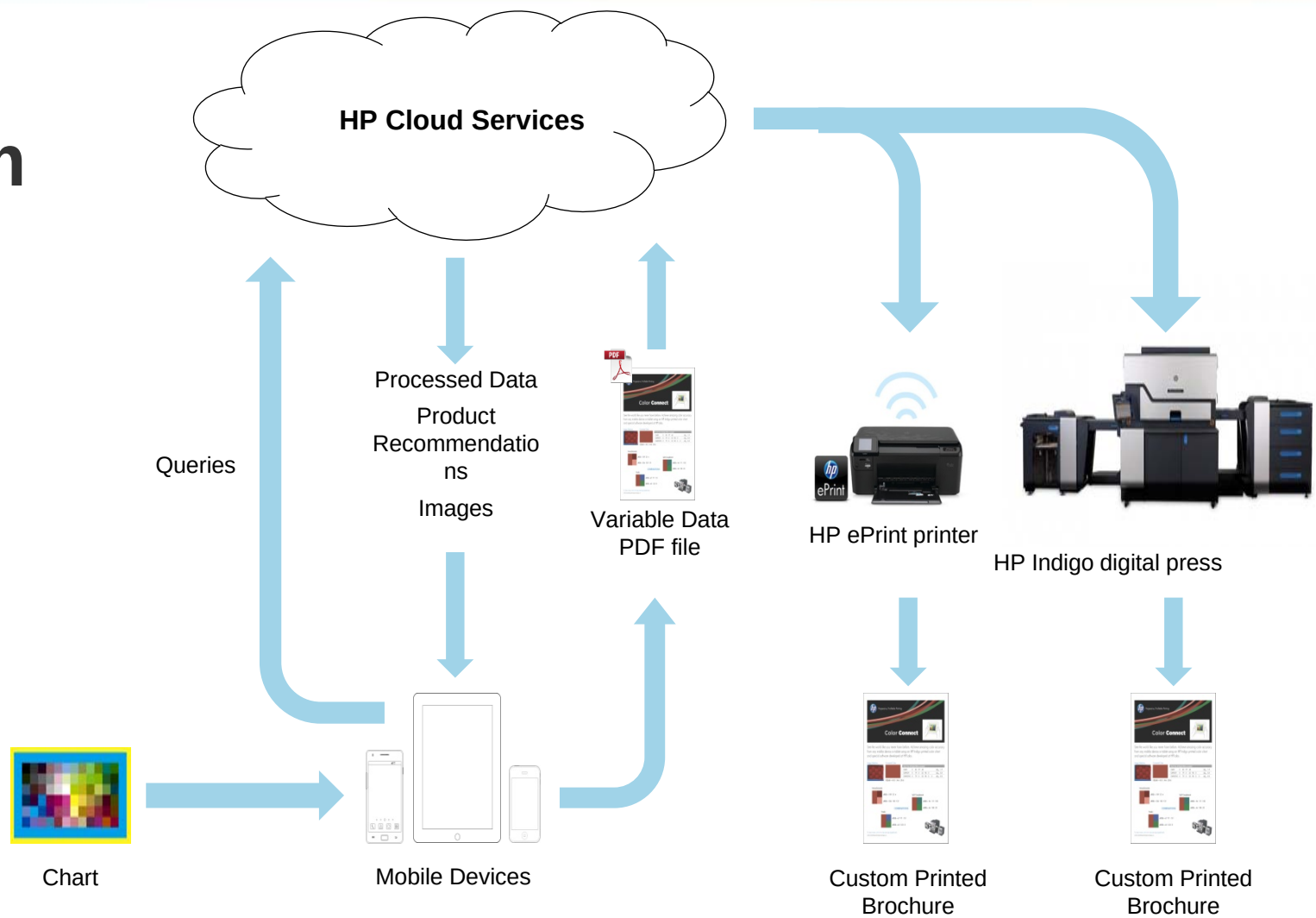
- Custom 7 color indigo printed
- Business card sized
- Color fiducials for video-rate machine recognition and sampling rectification
- Void-filling color sampling
 - CIC20 paper
 - “Analysis of Irregular Sampling for Local Interpolation”, Kok-Wei Koh, Nathan Moroney, and Melanie Gottwals
- Laboratory color measurements of the chart relative to a given white point included in the app for dynamic interpolation of unknown color

Smart Devices

- **Core algorithms platform independent**
 - Android, iOS, WebOS, Matlab, HP Cloud Services (*nix)

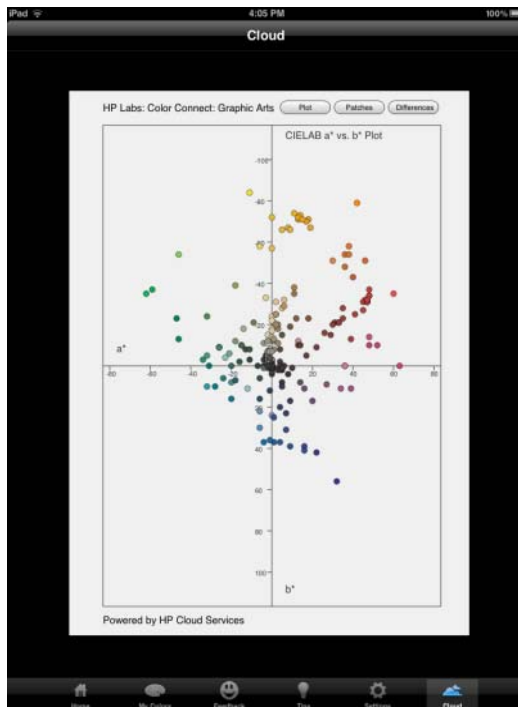


Larger System

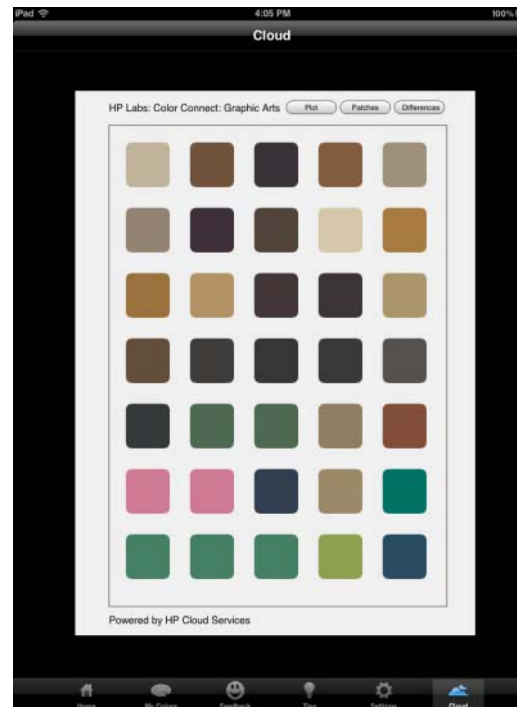




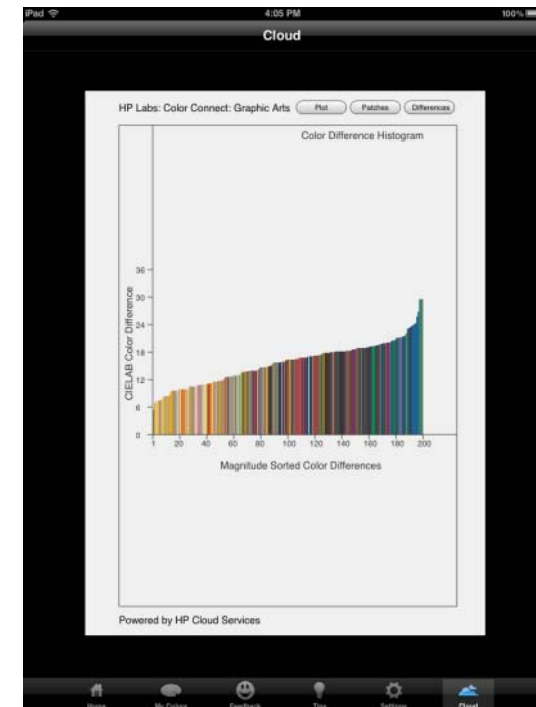
Cloud Data Logging



View CIELAB plot of corrected colors



View last 35 color patches



View color difference histogram

(as an aside, HTML5 useful for wide range of visualizations)



Dynamic PDF(s)

- Dynamically create custom PDF file for sharing with customers or printing directly to HP ePrint printer

hp Progressive, Profitable Printing

Color Connect

See the world like you never have before. Achieve amazing color accuracy from any mobile device or tablet using an HP Indigo printed color chart and special software developed at HP Labs.

Image Selection **Corrected Color**

Finish: Fine ID Bright White uncoated								
CMYK	0	88	99	54	ΔE_{94} 2.4			
CMYKCV	0	97	0	59	99	0	ΔE_{94} 0.0	
CMYKCVG	0	97	0	59	99	0	0	ΔE_{94} 0.0

CIEAB = 36.8 32.5 24.5

Monochromatic
sRGB = 99 20 6
sRGB = 226 147 133

Split Complement
sRGB = 42 103 93
sRGB = 41 117 19

COMBINATIONS
Triadic
sRGB = 60 84 135
sRGB = 58 111 0

To learn more, visit www.hp.com/go/graphicarts
© 2012 Hewlett-Packard Development Company, L.P.

hp Progressive, Profitable Printing

Color Connect

See the world like you never have before. Achieve amazing color accuracy from any mobile device or tablet using an HP Indigo printed color chart and special software developed at HP Labs.

Image Selection **Corrected Color**

Finish: Fine ID Bright White uncoated								
CMYK	100	35	0	76	ΔE_{94} 1.2			
CMYKCV	99	0	0	84	0	25	ΔE_{94} 0.6	
CMYKCVG	100	0	0	82	0	28	5	ΔE_{94} 0.0

CIEAB = 31.1 -6.7 -17.0

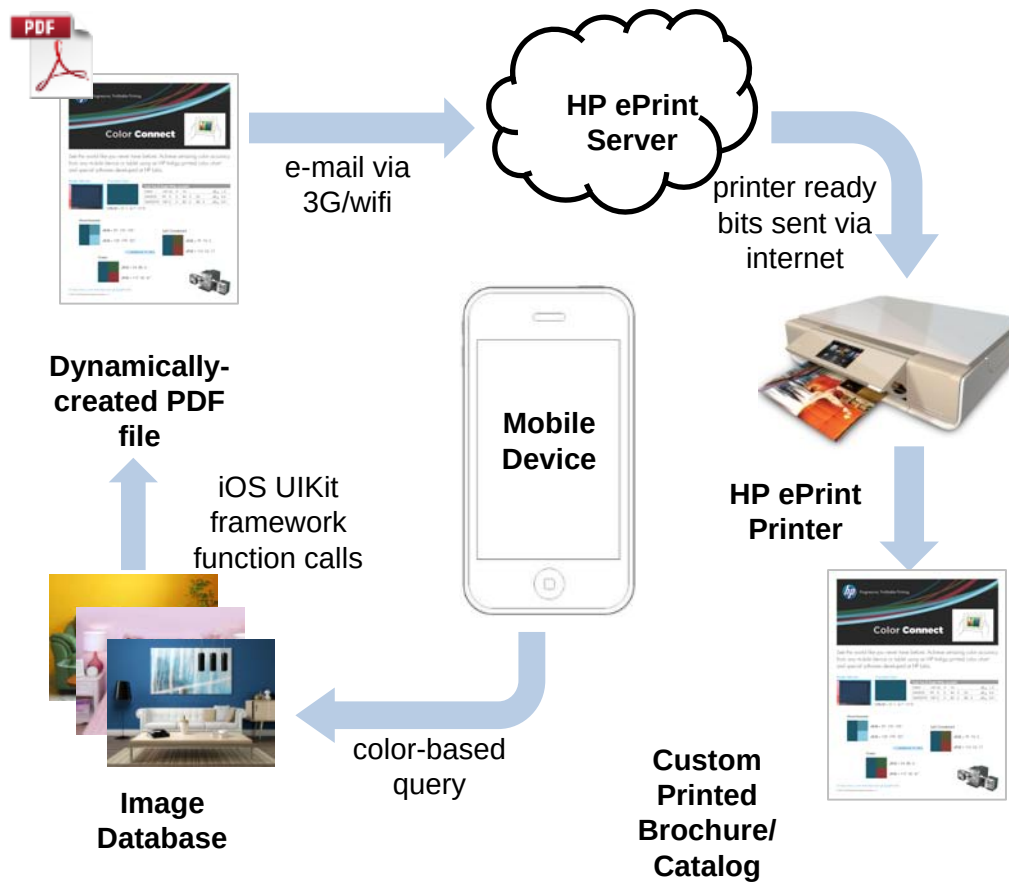
Monochromatic
sRGB = 97 137 159
sRGB = 159 199 221

Split Complement
sRGB = 79 74 2
sRGB = 115 52 17

COMBINATIONS
Triadic
sRGB = 54 88 0
sRGB = 117 45 47

To learn more, visit www.hp.com/go/graphicarts
© 2012 Hewlett-Packard Development Company, L.P.

ePrint Printing Process





Mobile Color VDP: Code Fragment 1

```
UIGraphicsBeginPDFContextToFile(appFile, CGRectZero, nil);
UIGraphicsBeginPDFPage();

context = UIGraphicsGetCurrentContext();

CGFloat bannerStartX = 27.0;
CGFloat bannerStartY = 27.0;
CGFloat bannerWidth = 558.0;
CGFloat bannerHeight = 230.0;

// programatically draw Color Connect banner at top of page
CGRect bannerRect = CGRectMake(bannerStartX, bannerStartY,
                               bannerWidth, bannerHeight);
CGContextSetRGBFillColor(context, 0.0, 0.0, 0.0, 1.0);
UIRectFill(bannerRect);

// draw blue HP circular logo
NSString* imageFilename = [[NSBundle mainBundle]
    pathForResource:@"HPCircularLogoBlue_430x430" ofType:@"png"];
UIImage* currentImage = [[UIImage alloc]
    initWithContentsOfFile:imageFilename];
CGRect imageRect = CGRectMake(47, 47, 48, 48);
[currentImage drawInRect:imageRect];
```



Mobile Color VDP: Code Fragment 2

```
// draw hp indigo digital press at bottom of page
NSString* hpIndigoName;
int randomInt = arc4random()% 7;
NSLog(@"randomInt = %d", randomInt);

switch(randomInt){
    case 0: hpIndigoName = [[NSBundle mainBundle]
                           pathForResource:@"Indigo7000forPDF" ofType:@"png"];
        break;
    case 1: hpIndigoName = [[NSBundle mainBundle]
                           pathForResource:@"Indigo7000forPDF2" ofType:@"png"];
        break;
    case 2: hpIndigoName = [[NSBundle mainBundle]
                           pathForResource:@"Indigo7500forPDF" ofType:@"png"];
        break;
    case 3: hpIndigoName = [[NSBundle mainBundle]
                           pathForResource:@"IndigoWS6000forPDF" ofType:@"png"];
        break;
    case 4: hpIndigoName = [[NSBundle mainBundle]
                           pathForResource:@"IndigoWS6000forPDF2" ofType:@"png"];
        break;
    case 5: hpIndigoName = [[NSBundle mainBundle]
                           pathForResource:@"IndigoWS6600forPDF" ofType:@"png"];
        break;
    case 6: hpIndigoName = [[NSBundle mainBundle]
                           pathForResource:@"IndigoWS6600forPDF2" ofType:@"png"];
        break;
    default: hpIndigoName = [[NSBundle mainBundle]
                           pathForResource:@"Indigo7000forPDF" ofType:@"png"];
}
}
```



Mobile Color VDP: Code Fragment 3

```
// SPLIT COMPLEMENT
```

```
startX = 350.674;  
startY = 554.958;
```

```
currentString = [NSString stringWithFormat:@"Split Complement"];  
[currentString drawAtPoint:CGPointMake(startX, startY-21.7) withFont:labelFont];
```

```
imageRect = CGRectMake(startX, startY, 36, 72);  
CGContextSetRGBFillColor(context, r/255.0, g/255.0, b/255.0, 1.0);  
UIRectFill(imageRect);
```

```
imageRect = CGRectMake(startX+36.1, startY, 36, 36);  
CGContextSetRGBFillColor(context,  
    splitComplementScheme[0]/255.0,  
    splitComplementScheme[1]/255.0,  
    splitComplementScheme[2]/255.0,  
    1.0);  
UIRectFill(imageRect);
```

```
imageRect = CGRectMake(startX+36.1, startY+36.1, 36, 36);  
CGContextSetRGBFillColor(context,  
    splitComplementScheme[3]/255.0,  
    splitComplementScheme[4]/255.0,  
    splitComplementScheme[5]/255.0,  
    1.0);  
UIRectFill(imageRect);
```





Mobile Color VDP: Code Fragment 4

```
// CORRECTED COLOR
currentString= [NSString stringWithFormat:@"Corrected Color"];
[currentString drawAtPoint:CGPointMake(159.3, 360) withFont:labelFont];

CGRect correctedRect = CGRectMake(159.3, 378, 108, 72);
CGContextSetRGBFillColor(context, r/255.0, g/255.0, b/255.0, 1.0);
UIRectFill(correctedRect);

CGContextSetCMYKFillColor(context, 0.0, 0.0, 0.0, 1.0, 1.0);

currentString = [NSString
    stringWithFormat:@"CIELAB = %.1f %.1f %.1f", _L, _a, _b];
[currentString drawAtPoint:CGPointMake(159.3, 455)
    withFont:labelFont];
```

Mobile color measurement drives
the dynamic Variable Data Print on
the device...





Summary

- Mobile image capture with carefully designed chart and corresponding app enable colorimetric estimates of unknown scene colors
- Broader system includes cloud color analytics for processing and analyzing all the measurements
- Using web-enabled printers, can use the mobile colorimetric estimates to drive mobile variable data print or dynamic PDF pages
- Questions?