Quantification of colorimetric sensor response from digital images

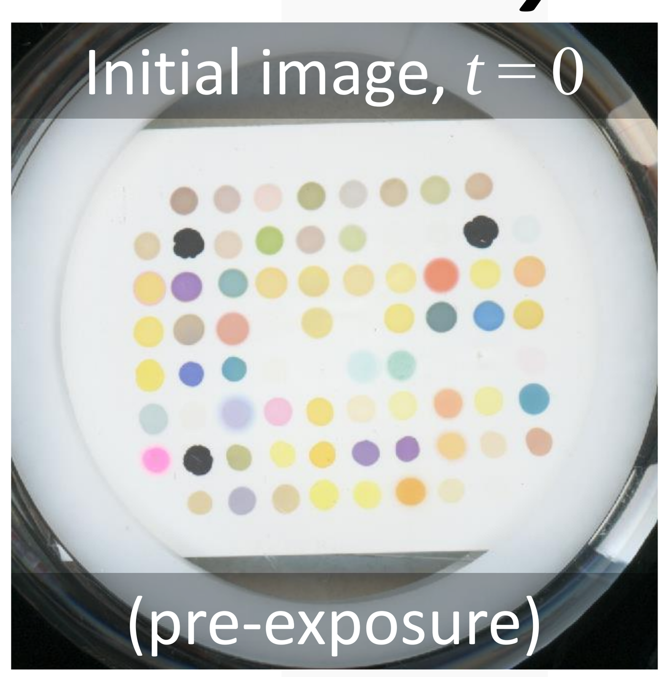
Charles Davidson[1], Aleksandr Miklos[2], Ronald Miles[3], Melissa Hulet[4], Barry Williams[4], Melissa Dixon[3], Avishai Ben-David[1], Sung Lim[5], Paul Rhodes[5], Alan Samuels[3], Peter Emanuel[3], Calvin Chue[3]

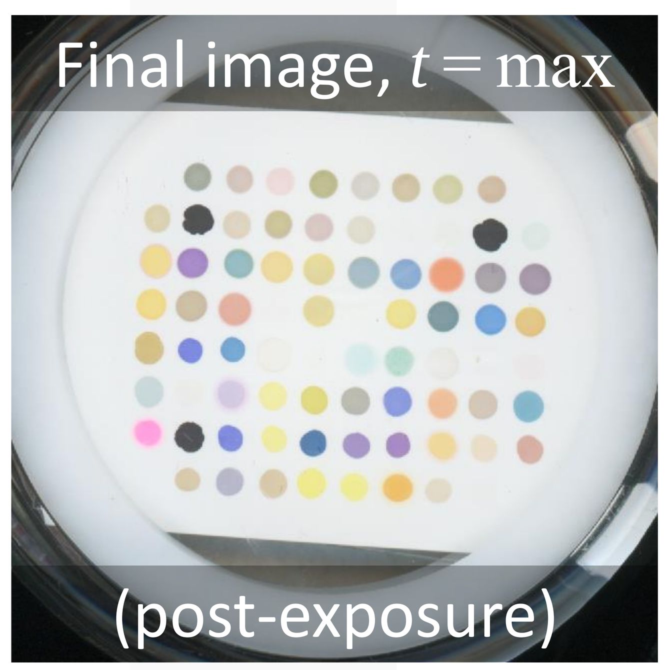
[1]STC, [2]Excet, [3]ECBC, [4]Leidos, [5]iSense

Introduction

Cheap and disposable, **colorimetric sensor arrays (CSAs)** can identify a wide range of samples, and thus are of interest to the warfighter in many contexts. CSAs consist of multiple chemical assays ("**spots**") which change color in response to different chemical moieties or properties. A repeatable ***pattern* of color changes** upon exposure to a particular sample serves as the basis for identification. The utility of CSAs depends upon the critical first step of **reading and interpreting** the color changes captured by digital imagery.

Can *you* do it…?





Which spots changed?
How did they change?
What colors, how strong?
Did you find them all?

If you are **shown a line-up** of before & after images, **will you be able to pick the right one**?

…Maybe not.

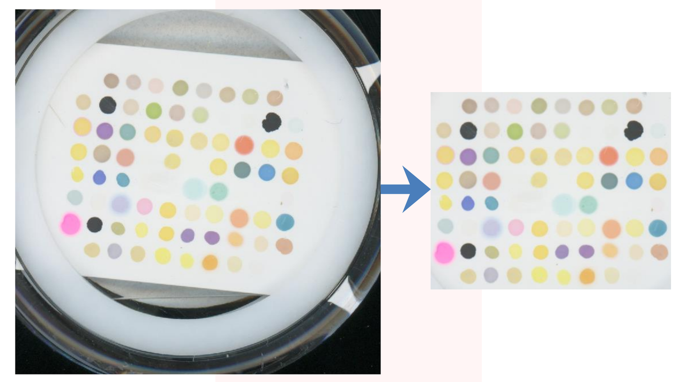
It seems evident that we need an automated procedure. However, we will still need to overcome the following factors:

- uncertain CSA placement/orientation in image;
- unknown image resolution;
- non-uniformity in the layout/sizes of spots;
- presence of scratches, dust, or damaged spots;
- changes in illumination.

Here we detail an adaptive, automatic, multi-step procedure to address these issues and quantify color change of colorimetric sensor arrays.

Step 1: Alignment

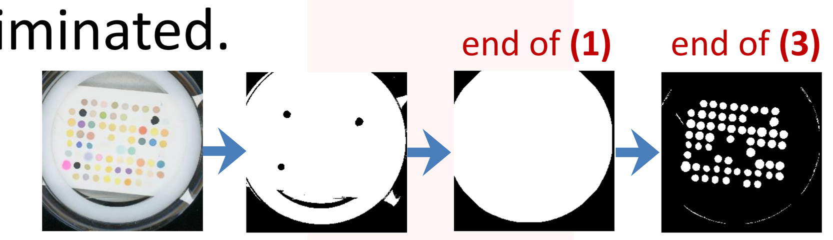
Given an image of a CSA containing 76 spots in a loose 8-by-10 grid, produce an aligned and cropped image.



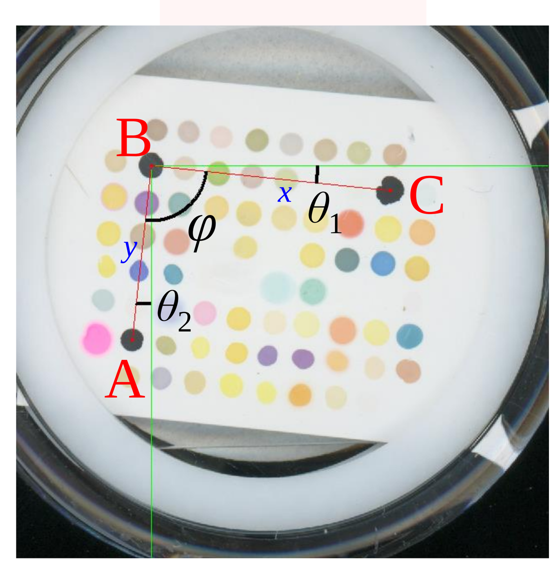
(1) Identify ticket area: threshold image to identify bright objects; single biggest object is assumed to be the CSA ticket.

(2) Compute typical spot size from the estimated ticket area. Estimates may be poor, but loose bounds on spot size, other parameters determined.

(3) Identify potential spots by threshold and/or gradient segmentation. Objects too big or too small are eliminated.



(4) Identify & analyze control spots: Angle between edges connecting control spot centroids ($\varphi = \angle ABC$, see figure below) should be near 90°, aspect ratio and edge lengths should match expectations. Rotation necessary for alignment estimated by θ_1, θ_2.

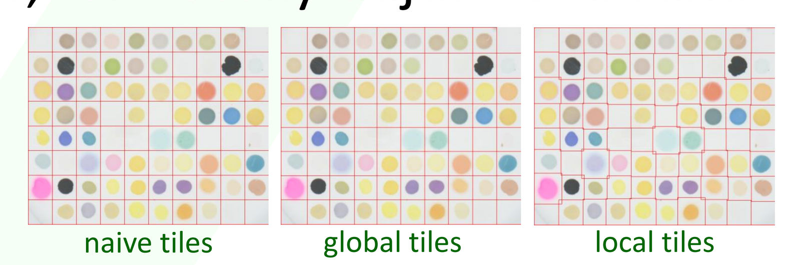


Step 2: Spot Detection

Given an aligned and cropped image, produce a map of spot boundaries. Since spots are arranged loosely in a grid, the approach is to tile the image such that each tile contains a spot and process each spot separately.

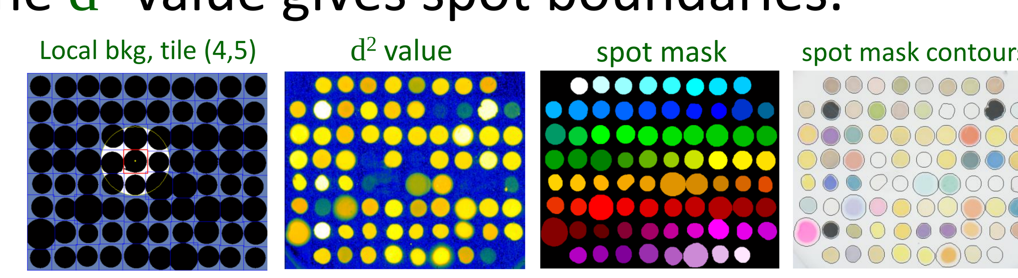
(1) Re-estimate typical spot-size, identify high-contrast spots, i.e., repeat **(2)** and **(3)** from left. Due to alignment step, spot parameters can be determined more precisely.

(2) Determine tile boundaries: naively from the known layout (i.e., 8-by-10 grid); globally optimized row and column breaks that minimize intersection with high-contrast spots; or locally-adjusted bounds for each tile.



(3) Determine probable background areas: These are areas far enough away from tile centers and previously identified high-contrast spots that they are likely non-spot areas.

(4) Local anomaly detection: mean RGB vector ($\boldsymbol{\mu}_0$) and covariance matrix ($\mathbf{C}_0$) of nearby background areas used to compute the squared Mahalanobis distance score, $d^2 = (\mathbf{x} - \boldsymbol{\mu}_0)^T \mathbf{C}_0^{-1} (\mathbf{x} - \boldsymbol{\mu}_0)$ for each pixel in the tile. d^2 measures dissimilarity to the empty areas of the ticket; segmentation of the d^2 value gives spot boundaries.



Step 3: Color Change

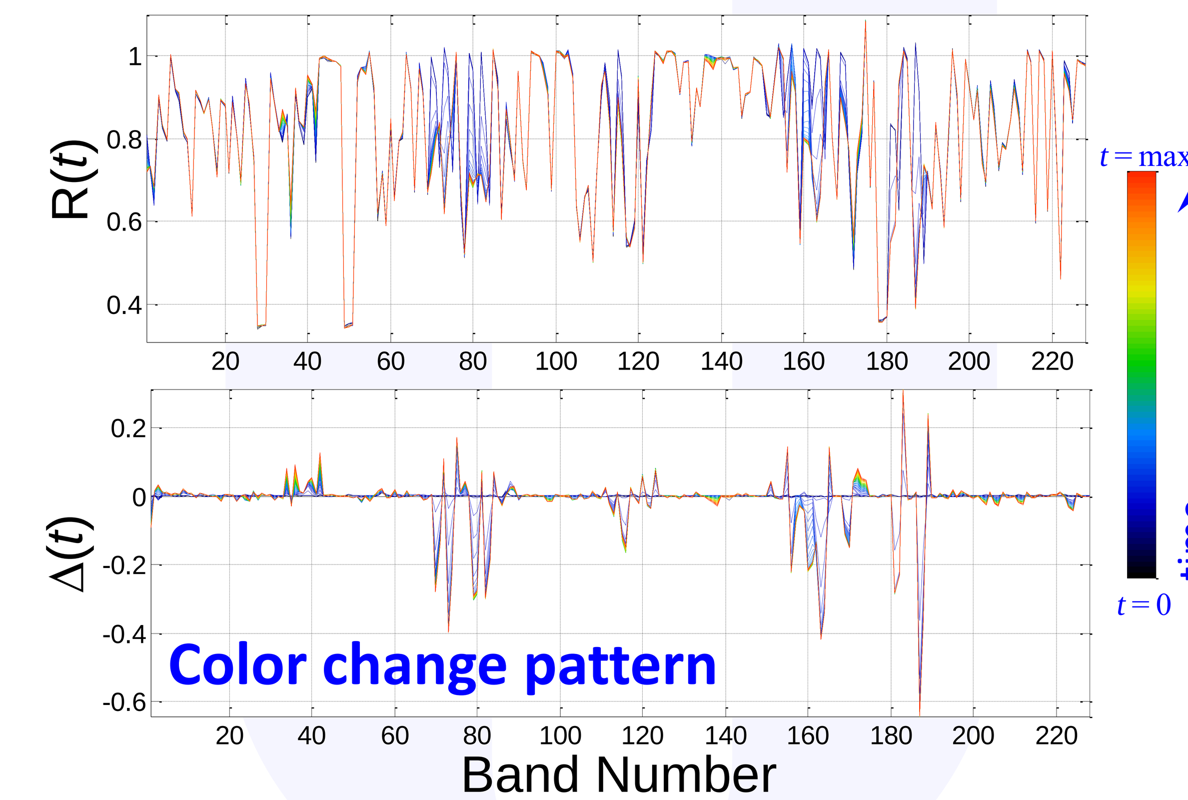
(1) Optionally erode away an outer margin or "guard area" so that only the central spot portion is considered.

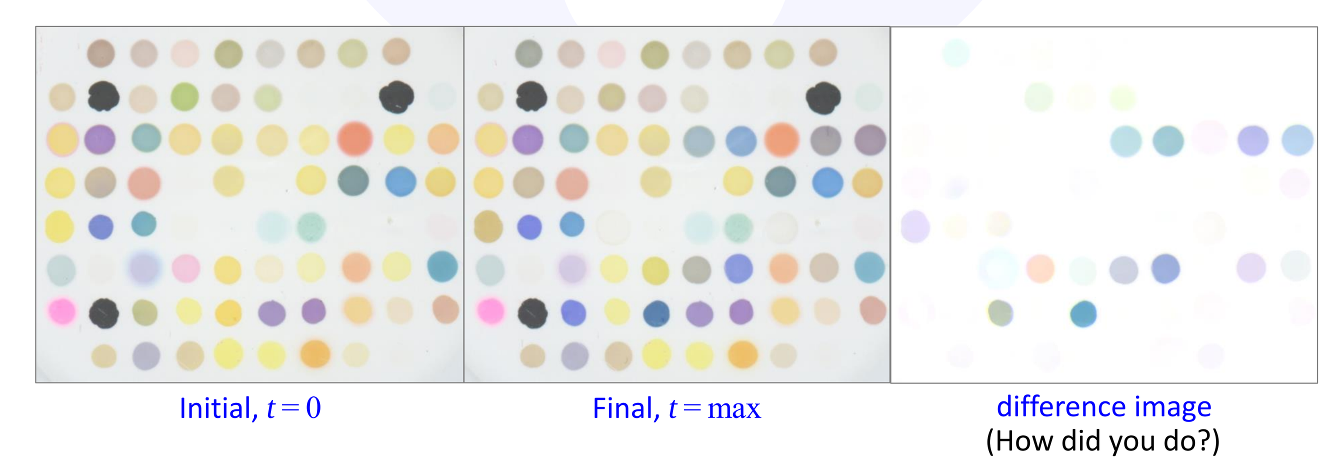
(2) Compute statistics for each spot, such as mean RGB vector, $\boldsymbol{\mu}_1$.

(3) Basic illumination correction. White-balance each spot with respect to the white ticket areas, $\boldsymbol{\rho} = \boldsymbol{\mu}_1/\boldsymbol{\mu}_0$. White-balancing converts to relative reflectance and normalizes for global changes such as source intensity. For N spots on the array there are N reflectance vectors $\boldsymbol{\rho}_j$, $j = 1, 2, \ldots, N$.

(4) Stack individual spot responses into a single vector characterizing the array: $\boldsymbol{\rho}_j \rightarrow \mathbf{R}$, a $3{\times}N$ vector. For vapor CSAs $\mathbf{R}$ is length 76×3 = 228.

(5) Compute difference to response collected pre-exposure: $\boldsymbol{\Delta}(t) = \mathbf{R}(t) - \mathbf{R}(t{=}0)$





TECHNOLOGY DRIVEN. WARFIGHTER FOCUSED.