Robust Content-Dependent Photometric Projector Compensation

Presentation for Procams 2006 workshop on projector-camera systems

Mark Ashdown, Takahiro Okabe, Imari Sato, Yoichi Sato
University of Tokyo, and National Institute of Informatics, Japan
Motivation

- Cheap and portable projectors will be used in non-ideal situations
  - Surface reflectance makes patterns
  - Ambient light reduces contrast
- We want to be robust to difficult projection situations
Goal

- Compensate for irregularities in the projection system
- Achieve contrast and saturation as close as possible to that in the ideal condition
Overview

- Characterizing the projection system
- Fit the image to the display
  - Image fitting
  - Framework
  - Our implementation
- Results
- Future work
Characterization

- Assumptions
  - Environment is static
  - Surface is Lambertian
  - Three additive primaries
- Linearize projector response
Characterization

- Get radiometric model that defines a per-pixel mapping RGB to XYZ

\[
\begin{bmatrix}
X \\
Y \\
Z
\end{bmatrix} = \begin{bmatrix}
\bullet & \bullet & \bullet & \bullet & \bullet \\
\bullet & \bullet & \bullet & \bullet & \bullet \\
\end{bmatrix} \begin{bmatrix}
R \\
G \\
B \\
1
\end{bmatrix}
\]

Surface - uncompensated result - compensated result

Grossberg, Peri, Nayar, and Belhumer, *Making One Object Look Like Another: Controlling Appearance Using a Projector-Camera System*, CVPR 2004
Characterization

- RGB input is limited to unit cube
- This corresponds to a gamut in XYZ space
- Gamut is different at every display pixel
Image Fitting

- Consider radiometric model and image content
- Fit image to spatially-varying gamut
- Balance competing properties
Content Dependence

- Fitting is tailored to the radiometric model and the original image
Robust Content-Dependent Photometric Compensation
Mark Ashdown <www.mark.ashdown.name>

Spatial Variation

- Fit image to spatially-varying gamut
Properties

- During image fitting we will balance four properties:
  - Gamut
  - Extent
  - Uniformity
  - Deviation
Framework

- Desired appearance
- Chrominance fitting
- Range calculation
- Luminance fitting
- Compensation image
Desired Appearance

- First we must know the appearance under ideal conditions
  - Original image in sRGB
  - Convert to L*u*v*

![sRGB](image1)

![L*u*v*](image2)

![u*v*](image3)
Chrominance Fitting

- Compute a transformation to fit the image chrominance to the spatially-varying gamut
- Find a transformation with three parameters, $s$, $a$, $b$, by minimizing $E$

\[
\begin{bmatrix}
    u_1 \\
    v_1
\end{bmatrix} = s \begin{bmatrix}
    u_0 \\
    v_0
\end{bmatrix} + \begin{bmatrix}
    a \\
    b
\end{bmatrix}
\]

\[
E = c_1 (1-s)^2 + c_2 (a^2 + b^2) + \frac{1}{n} \sum_{\text{pixels}} \sum_{\text{lines}} e^{c_3 (r \cdot m + l_c)}
\]

- extent
- deviation
- gamut
Range Calculation

- Calculate a range $[G^l, G^h]$ within which $(u_1, v_1)$ can be produced
  - Approximate the gamut with 12 triangles
  - Linearly interpolate $L$
Luminance Fitting

- Fit the image luminance to the previously calculated range
  - Linearly interpolate $L_0$ between two spatially-varying values

\[ L_1 = F^l + (F^h - F^l) L_0 \]
Luminance Fitting

- Optimize to get $F_l$ and $F_h$

$$L_1 = F_l + (F_h - F_l) L_0$$

$$e = \sum_i \sum_j \left( s_{i,j} + d_1 r_{i,j} + d_2 t_{i,j} + d_3 w_{i,j} \right)$$

$$s = (F_x^l)^2 + (F_y^l)^2 + (F_x^h)^2 + (F_h^h)^2$$

$$r = \begin{cases} (L_1 - G_l)^2 & \text{if } L_1 < G_l \\ (L_1 - G_h)^2 & \text{if } L_1 > G_h \\ 0 & \text{otherwise} \end{cases}$$

$$t = (F_l)^2 + (F_h - 1)^2$$

$$w = e^{d_4(F_l - F_h)}$$

- Uniformity
- Gamut
- Extent & deviation
Compensation Image

- Calculate the projector RGB value that will produce the fitted result
  - Convert from L*u*v* to RGB
  - Clip to RGB cube

\[
\begin{bmatrix}
X \\
Y \\
Z
\end{bmatrix} =
\begin{bmatrix}
\cdot & \cdot & \cdot & \cdot & \cdot \\
\cdot & \cdot & \cdot & \cdot & \cdot \\
\cdot & \cdot & \cdot & \cdot & \cdot \\
\cdot & \cdot & \cdot & \cdot & \cdot \\
\cdot & \cdot & \cdot & \cdot & \cdot \\
\cdot & \cdot & \cdot & \cdot & \cdot \\
\cdot & \cdot & \cdot & \cdot & \cdot \\
\cdot & \cdot & \cdot & \cdot & \cdot \\
R & G & B & 1
\end{bmatrix}
\]
Results

- Compensating for varying surface colour
Results

- Compensating for large variation in surface brightness

Uncompensated result

Compensation image

Compensated result
Results

- Balancing the various aims

Surface

Compensation image

Uncompensated result

Compensated result
Future Work

- **Speed**
  - Alternative method for optimization
  - Use coarser luminance fitting
- **Video**
  - Allow image to vary over time
  - Apply temporal uniformity constraint