Intrinsic Images by Entropy Minimization

(Midway Presentation, by Yingda Chen)

Graham D. Finlayson, Mark S. Drew and Cheng Lu,

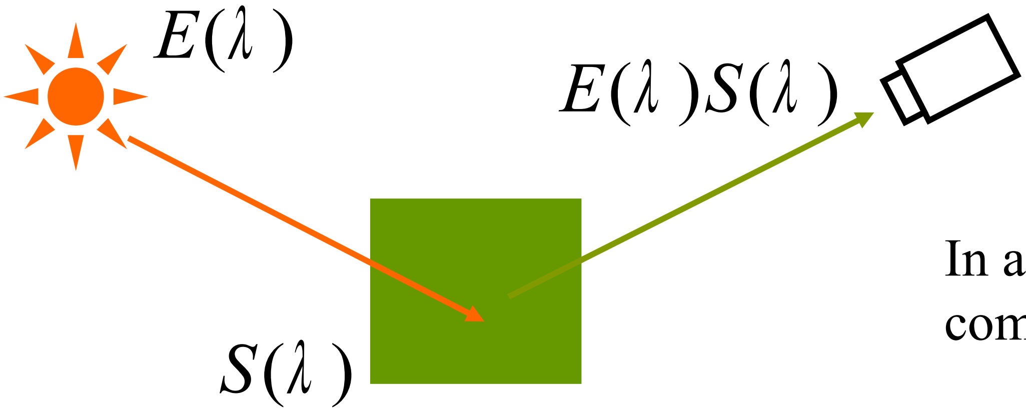
ECCV, Prague, 2004

Project Goals:

Obtain the intrinsic image by removing shadows from images :

- Without camera calibration (no knowledge about the imagery source)
- Based on one single image (instead of multiple image arrays) by entropy minimization

How an Image is Formed?



In an RGB image, the R, G , B components are obtained by:

$$r = \int R(\lambda)E(\lambda)S(\lambda)d\lambda$$

$$g = \int G(\lambda)E(\lambda)S(\lambda)d\lambda \qquad (*)$$

$$b = \int B(\lambda)E(\lambda)S(\lambda)d\lambda$$

Camera responses depend on 3 factors:

- Light (E),
- Surface (S),
- Camera sensor (R,G, B)

Planck's Law

- ## Blackbody:

A blackbody is a hypothetical object that *emits radiation at a maximum rate* for its given temperature and absorbs *all of the radiation* that strikes it.

Illumination sources such as ☀ can be well approximately as a blackbody radiator.

- ## Planck's Law [Max Planck, 1901]

Planck's Law defines the energy emission rate of a blackbody , in unit of *watts per* λ *square meter per wavelength interval*, as a function of wavelength (in meters) and temperature *T* (in degrees Kelvin).

$$P_r(\lambda) = c_1\lambda^{-5}\left(e^{\frac{c_2}{\lambda T}} - 1\right)^{-1}$$

Where $c_1 = 3.74183 \times 10^{-16}$ Wm^2 and

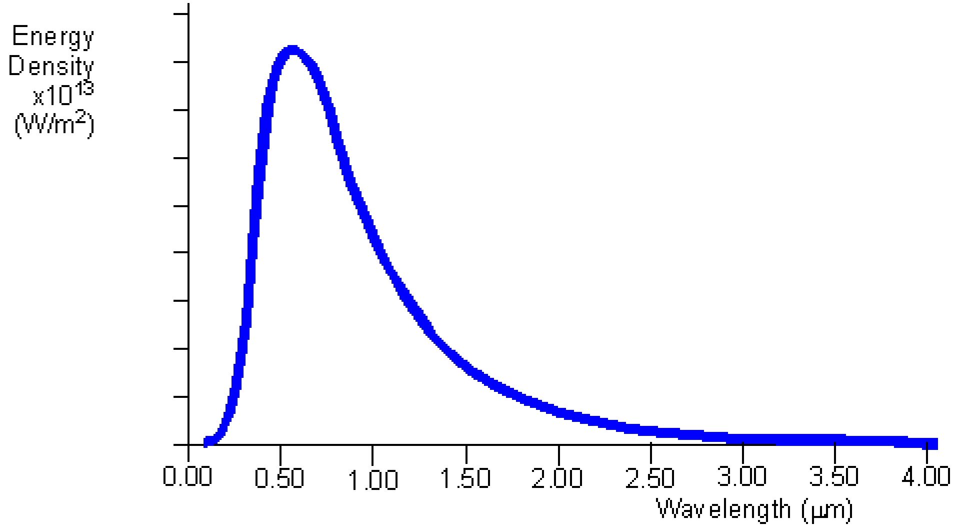
$c_1 = 1.4388 \times 10^{-2}$ mK are constants.

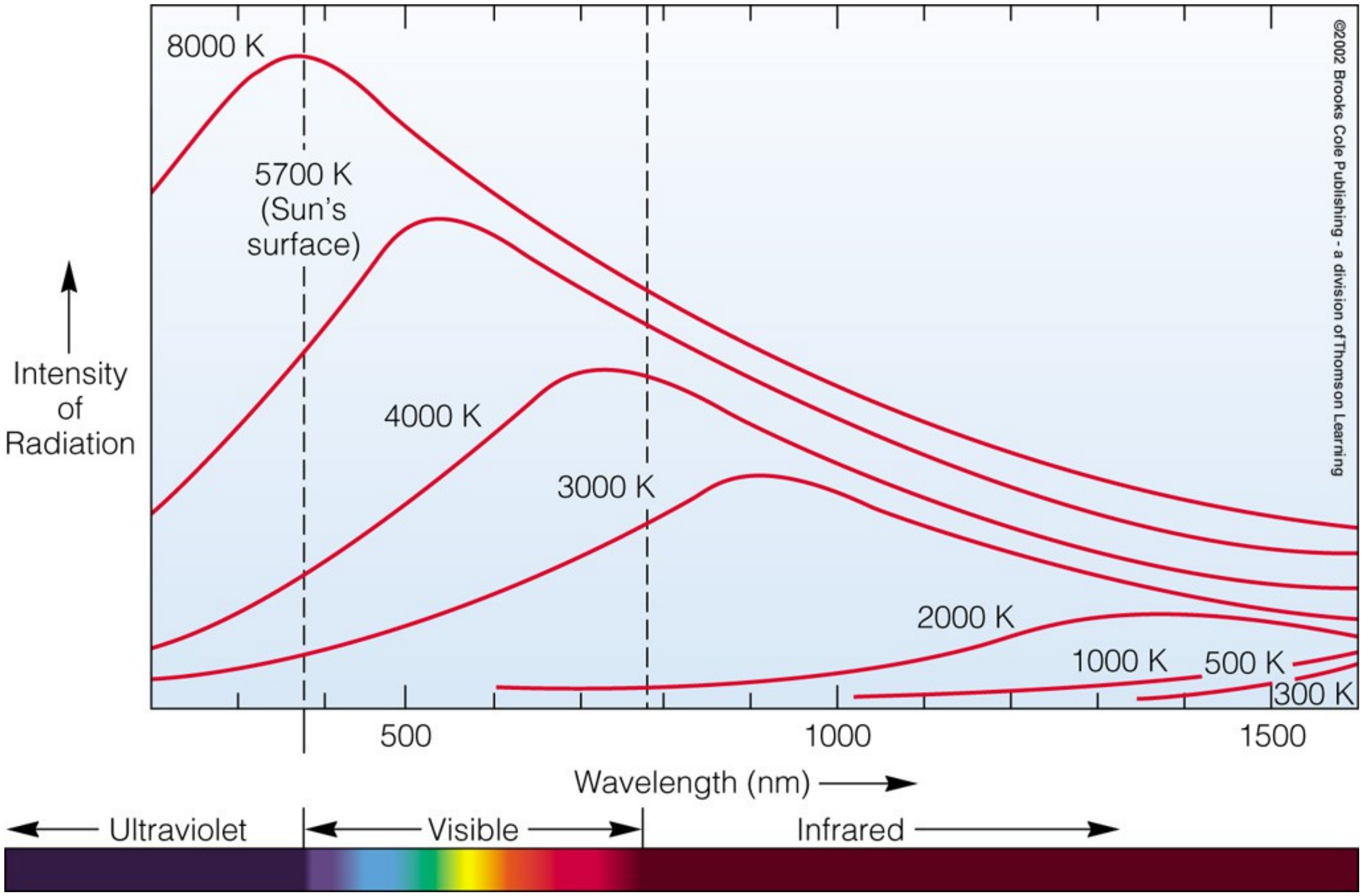
Planck's Law (cont.)

Given the intensity of the radiation I, the Planck's law gives the spectral power of the lighting source:

$$E(\lambda) = I \times P_r = I c_1 \lambda^{-5} \left(e^{\frac{c_2}{\lambda T}} - 1 \right)^{-1} \cong I c_1 \lambda^{-5} e^{-\frac{c_2}{\lambda T}}$$

The temperature of a lighting source and the wavelength together determine the relative amounts radiation being emitted (color of the illuminator).





©2002 Brooks Cole Publishing - a division of Thomson Learning

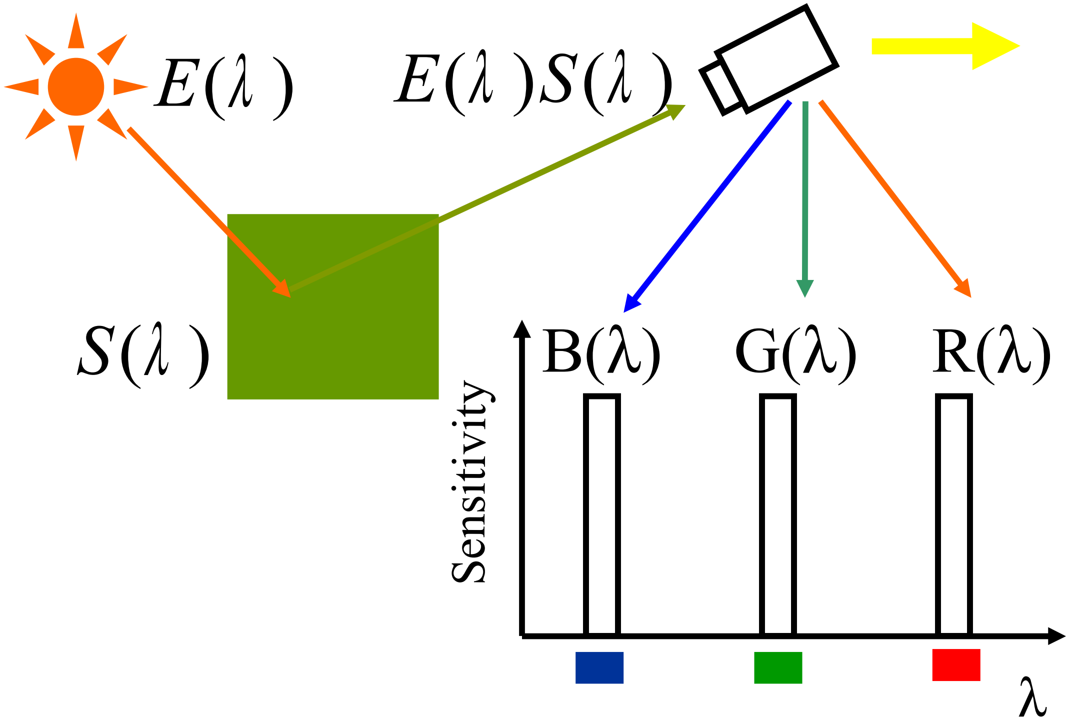
is **Blue**;

is **Red**;

Image formation for Lambertian surface

Assume idea camera sensors:

$$R(\lambda) = \delta(\lambda - \lambda_R) \quad G(\lambda) = \delta(\lambda - \lambda_G) \quad B(\lambda) = \delta(\lambda - \lambda_B)$$



$$r = \int R(\lambda)E(\lambda)S(\lambda)d\lambda$$

$$= \int \delta(\lambda - \lambda_R)E(\lambda)S(\lambda)d\lambda$$

$$= E(\lambda_R)S(\lambda_R)$$

…

$$g = E(\lambda_G)S(\lambda_G)$$

$$b = E(\lambda_B)S(\lambda_B)$$

Analysis of the components in image formation

$$\ln\left(r\right) = \ln I + \ln\left(S(\lambda_R)c_1\lambda_R^{-5}\right) - \frac{c_2}{\lambda_R T}$$

$$\ln\left(g\right) = \ln I + \ln\left(S(\lambda_G)c_1\lambda_G^{-5}\right) - \frac{c_2}{\lambda_G T}$$

$$\ln\left(b\right) = \ln I + \ln\left(S(\lambda_B)c_1\lambda_B^{-5}\right) - \frac{c_2}{\lambda_B T}$$

Need some manipulations to get rid of the illumination dependence

Depends on the surface property only

Depend on property of the illumination

How to remove shadows (illumination)?

Define a 2-D chromaticity vector V,

$$V = \begin{pmatrix} v_1 \\ v_2 \end{pmatrix} = \begin{pmatrix} \ln(r/g) \\ \ln(b/g) \end{pmatrix}$$

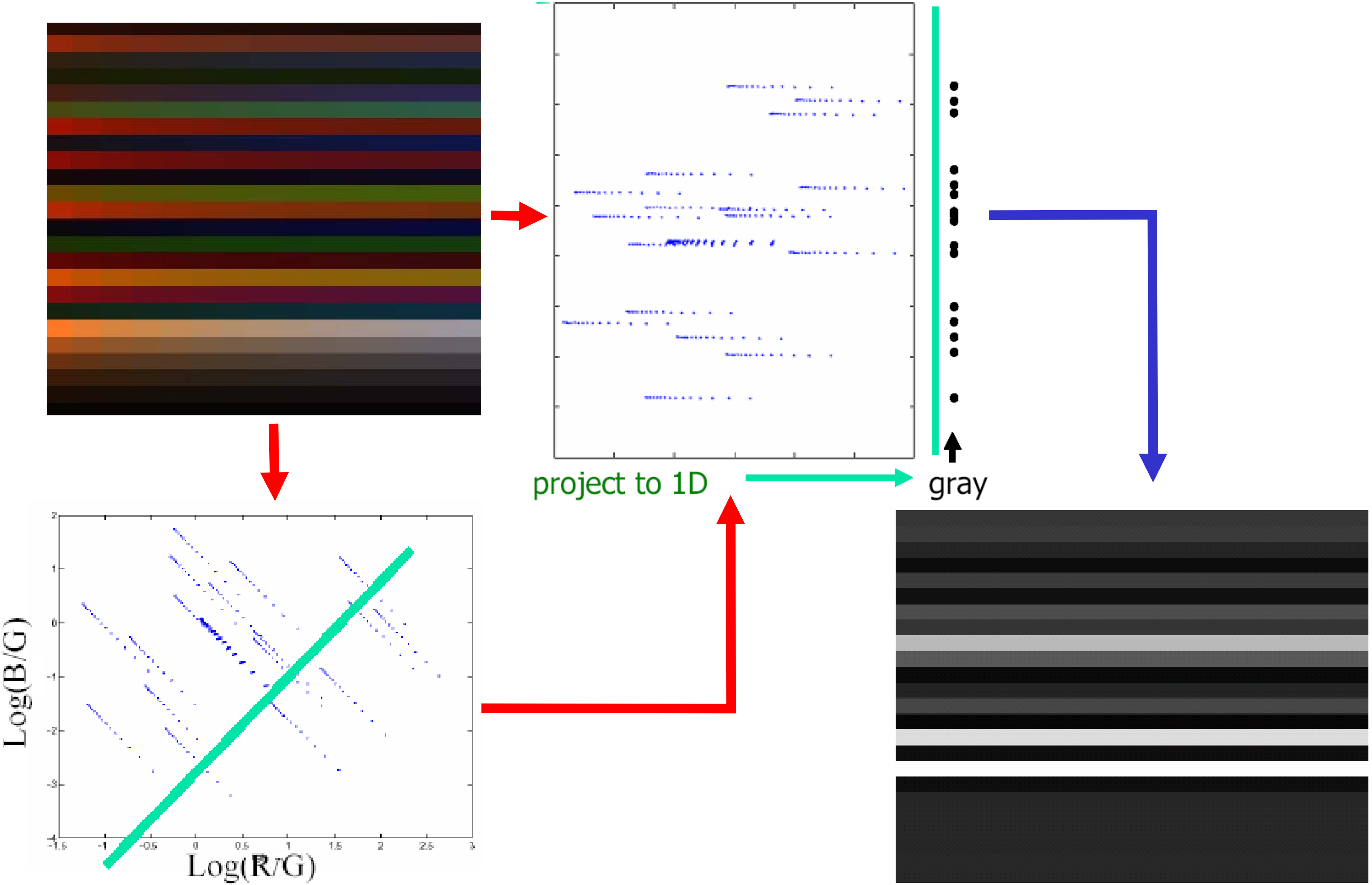
$$v_1 = \ln\left(\frac{S(\lambda_R)\lambda_R^{-5}}{S(\lambda_G)\lambda_G^{-5}}\right) - \frac{c_2}{T}\left(\frac{\lambda_R}{\lambda_G}\right)$$

$$v_2 = \ln\left(\frac{S(\lambda_B)\lambda_B^{-5}}{S(\lambda_G)\lambda_G^{-5}}\right) - \frac{c_2}{T}\left(\frac{\lambda_B}{\lambda_G}\right)$$

The 2-D vector V forms a straight line in the space of *logs of ratios*, the slope of the which is determined by T (i.e. by illumination color). Project the 2D log ratios into the direction $e^{\perp}$, the 1-D grayscale invariant image can be obtained. $e^{\perp}$ is the direction *orthogonal* to vector $[\frac{c_2}{T}\left(\frac{\lambda_R}{\lambda_G}\right), \frac{c_2}{T}\left(\frac{\lambda_B}{\lambda_G}\right)]$

Shadow which occurs when there is a change in light but *not* surface will disappear in the invariant image

How to remove shadows (illumination)? (cont.)



project to 1D

gray

How to remove shadows (illumination) ? (cont. II)

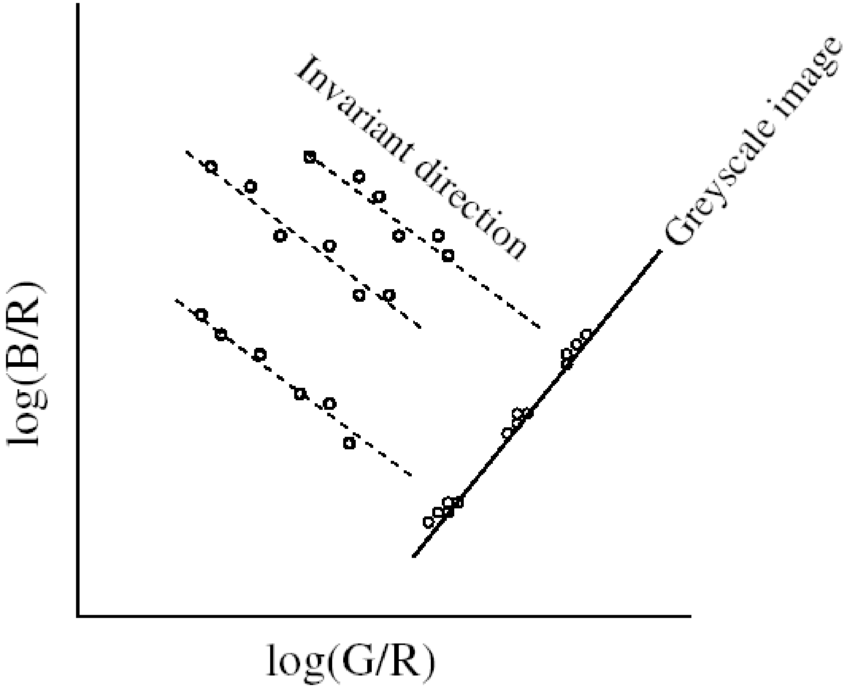
The key of the obtaining the invariant image is to determine the right projection direction. For a calibrated camera whose sensor sensitivity is known, the task is relatively easy.

- Question: *How to determine the projection direction for images from* ***Uncalibrated*** *Cameras?*

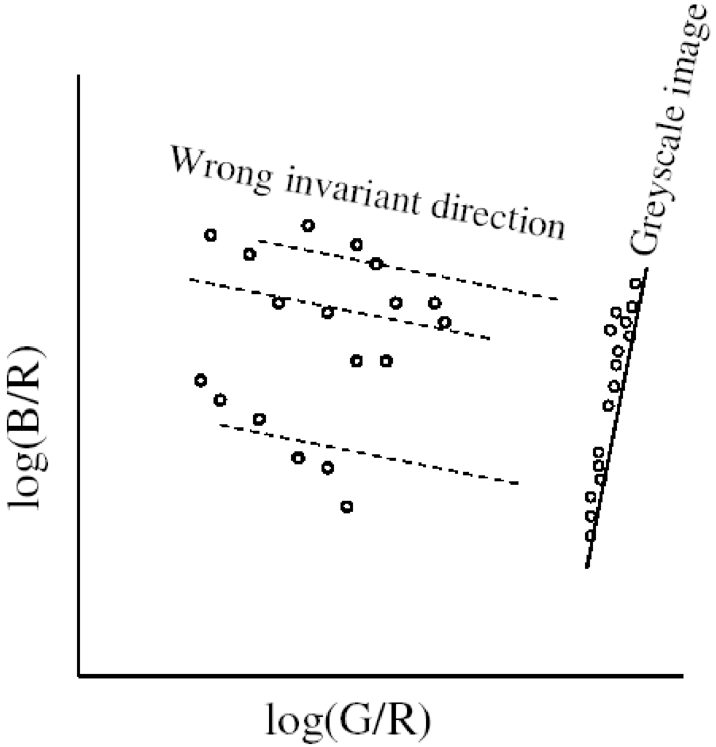
  Answer: *Problematic, but artificial calibration can still be performed by obtaining a series of image from the same camera.*

- Question: *How to determine the projection direction for images whose source is unknown?*

Entropy minimization



Correct Projection



Incorrect Projection

Entropy minimization (cont.)

Given projection angle θ , the projection result in a scalar value

$$\tau = V \times [\cos\theta , \sin\theta]^T = v_1 \cos\theta + v_2 \sin\theta$$

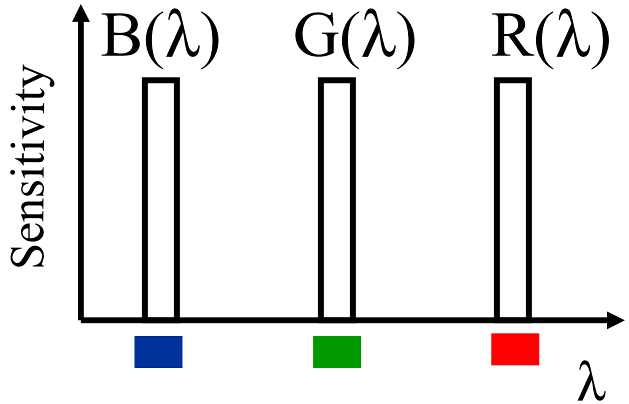
The scalar values can be encoded into a grayscale image, and the entropy be calculated as

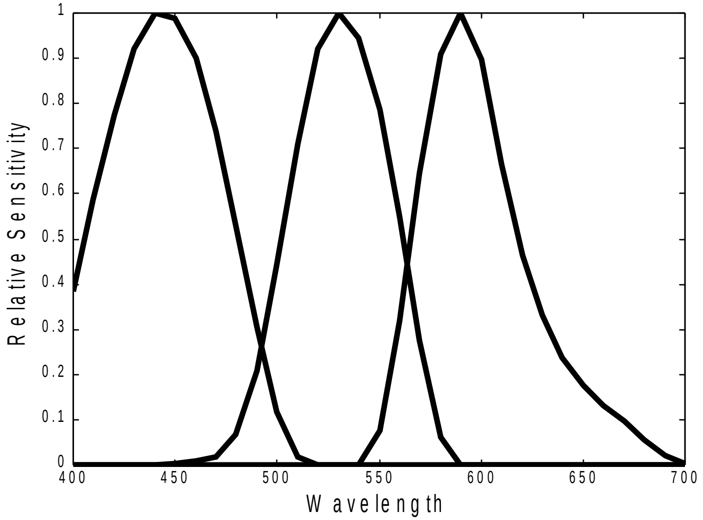
$$H = -\sum_i p(x_i) \times \log p(x_i)$$

For each $\theta = 1, 2, \cdots 180$, we can obtain an corresponding entropy. As the more “spread-out” distribution results in a larger entropy value, ***the projection direction*** θ ***that produces the minimum entropy is the correct projection direction***

Assumptions?

- Delta sensor functions of camera





This assumption is idealized, but experiments show that it performs reasonably well.

- The image must be unbiased of R,G,B

This is NOT true for many images, which can be “reddish”, “bluish” or “greenish” in color. So some more dedicated approach should be introduced to remove (or at least suppress) the potential bias.

Geometric Mean Invariant Image

Use the geometric mean as the reference color channel when taking the log ratios, so we will not favor for any particular color

$$C_{ref} = \sqrt[3]{r \times g \times b}$$

$$\ln(C_{ref}) = \ln(I) + \ln(c_1) + \frac{1}{3}\sum_{k=R,G,B} \ln(S(\lambda_k)\lambda_k^{-5}) - \frac{c_2}{T}\sum_{k=R,G,B}\frac{1}{\lambda_k}$$

$$\vec{\rho}\,' = \begin{pmatrix} \ln(r/C_{ref}) \\ \ln(g/C_{ref}) \\ \ln(b/C_{ref}) \end{pmatrix} = \begin{pmatrix} K + K_{11} - \dfrac{c_2}{T}K_{12} \\ \\ K + K_{21} - \dfrac{c_2}{T}K_{22} \\ \\ K + K_{31} - \dfrac{c_2}{T}K_{32} \end{pmatrix}$$

Where the K's are just constants

The geometric mean is unbiased, however, the invariant image is a projection from 2-D space into 1-D grayscale. The log ratios vector $\rho\,'$ is 3-D.

Geometric Mean Invariant Image (cont.)

- From 3-D to 2-D before we can get the invariant.

  A 2 x 3 U matrix can do this by $\chi \equiv U\rho'$

  The transform should satisfy:

  A straight line in the 3-D space is still straight after the transformation

- How do we find U ?

  Since $\rho_R + \rho_G + \rho_B = \ln(r \times g \times b / C_{ref}^3) = 0$

  $$\Rightarrow \vec{\rho} \cdot \vec{u} = 0, \; where \;\; \vec{u} = (1/\sqrt{3}) \cdot [1 \;\; 1 \;\; 1]^T$$

  The orthogonal matrix U satisfies

  $$U^T U = P^{\perp} = I - \vec{u}\,\vec{u}^T$$

  The rows of U are just the eigenvectors associated with the non-zero eigenvalues of the matrix $P^{\perp}$.

Geometric Mean Invariant Image (cont. II)

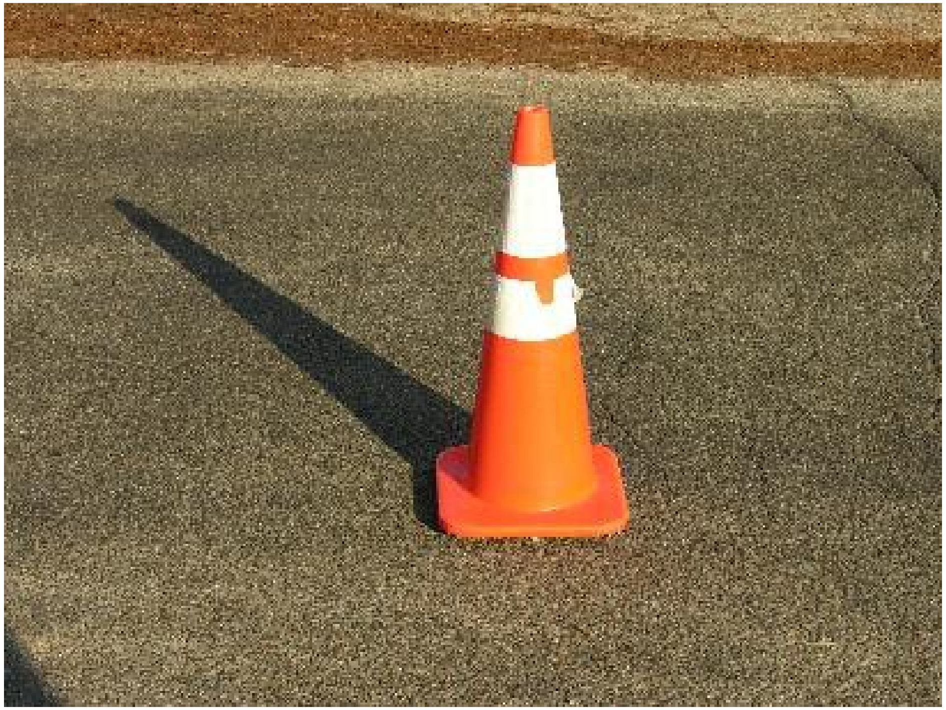
- With $\chi \equiv U\rho'$, we converted the 3-D log ratios space into 2-D, but with no color bias, the invariant image is then achieved by $\tau = \chi \times [\cos\theta, \sin\theta]^T$, θ is the correct projection angle, and we are back to the original track.

- Obtain θ by entropy minimization:
  - Decide the number of bins by Scott's rule:

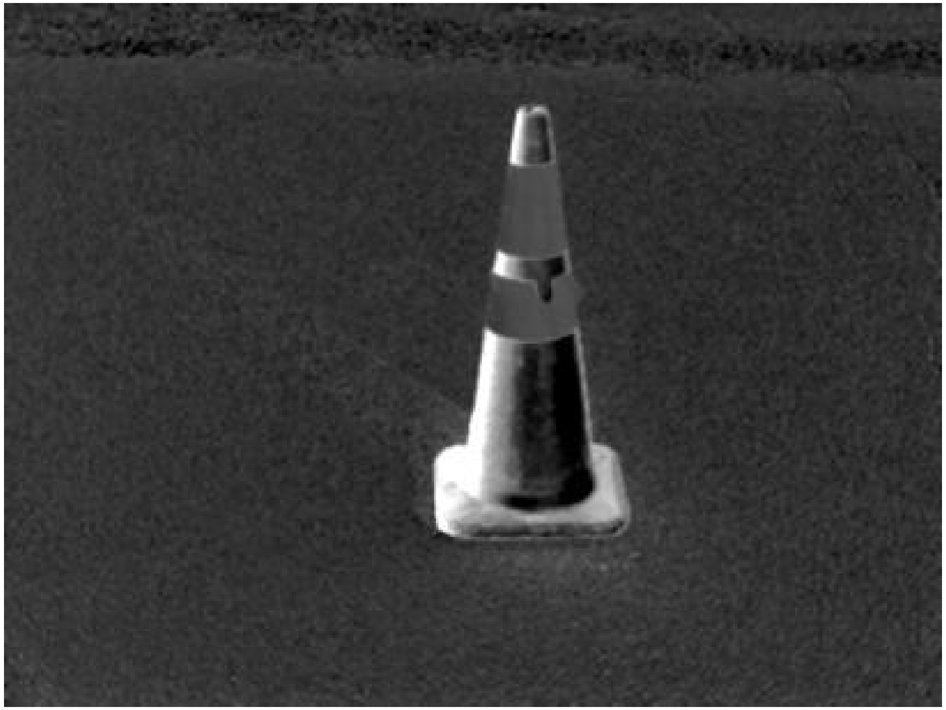
    $$\text{binNumber} = 3.5 \times \mathit{std}\,(\text{data}) \times \text{N}^{1/3}$$

  - The probability of the i*th* bin is $\text{P}_i = \dfrac{n_i}{N}$
  - The entropy is calculated as $\text{I} = \sum_i \left(-\text{P}_i \times \log_2(\text{P}_i) \right)$

Midway Results: invariant image

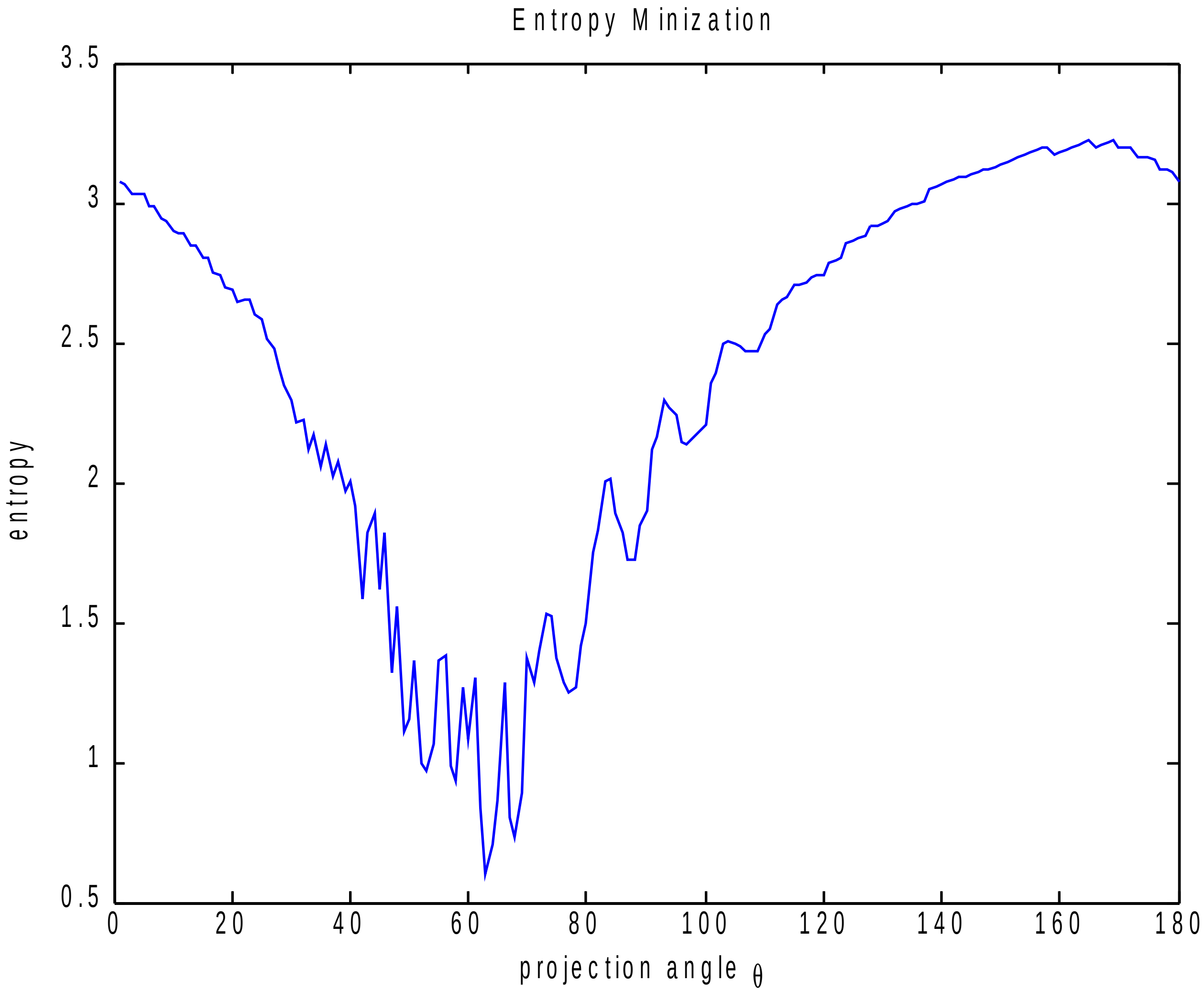


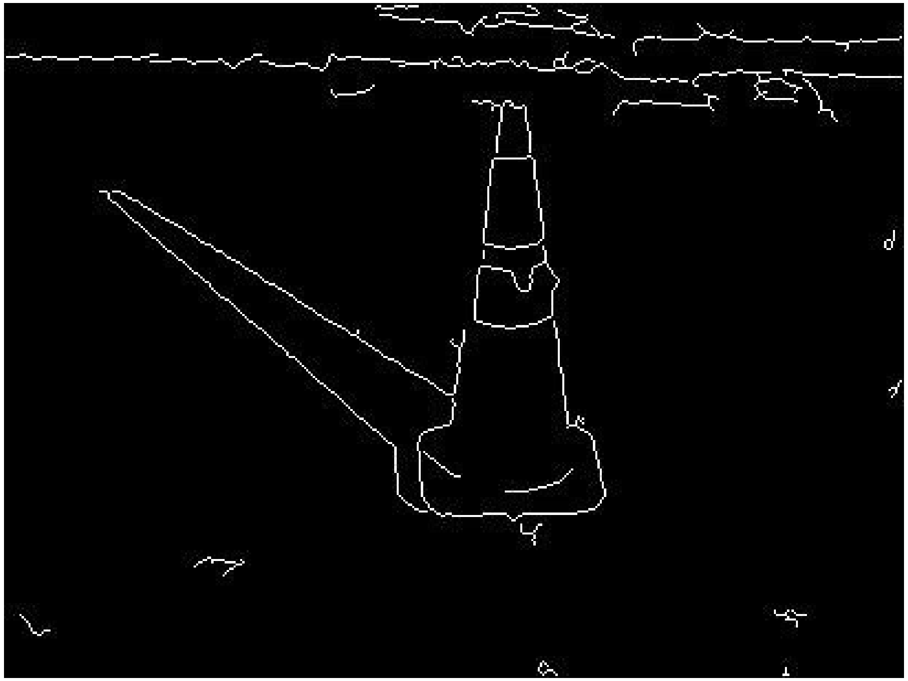
Original Image



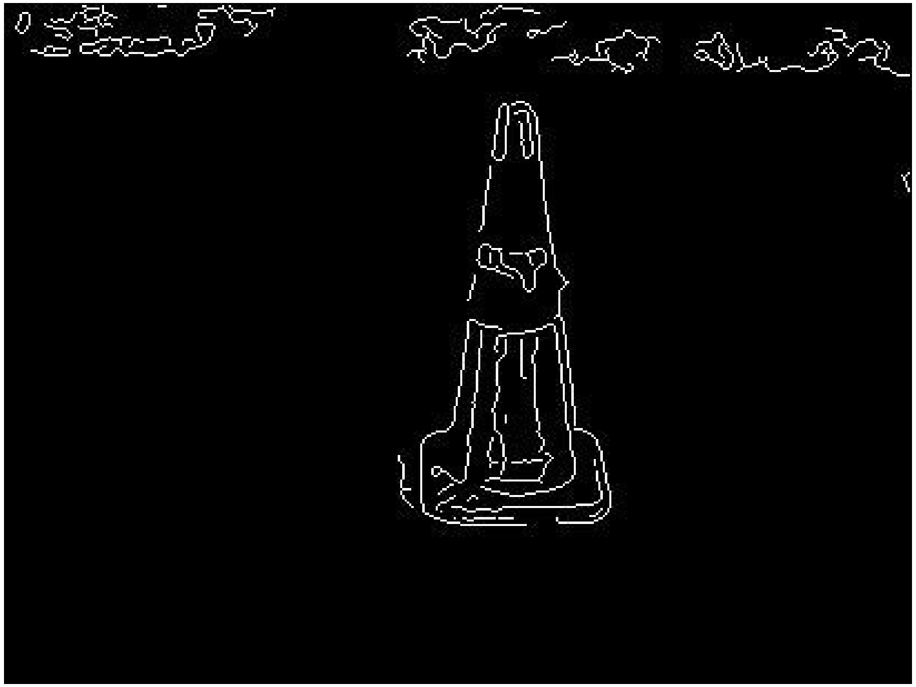
Invariant Image

Entropy Minimization (Camera: Nikon CoolPix8700)

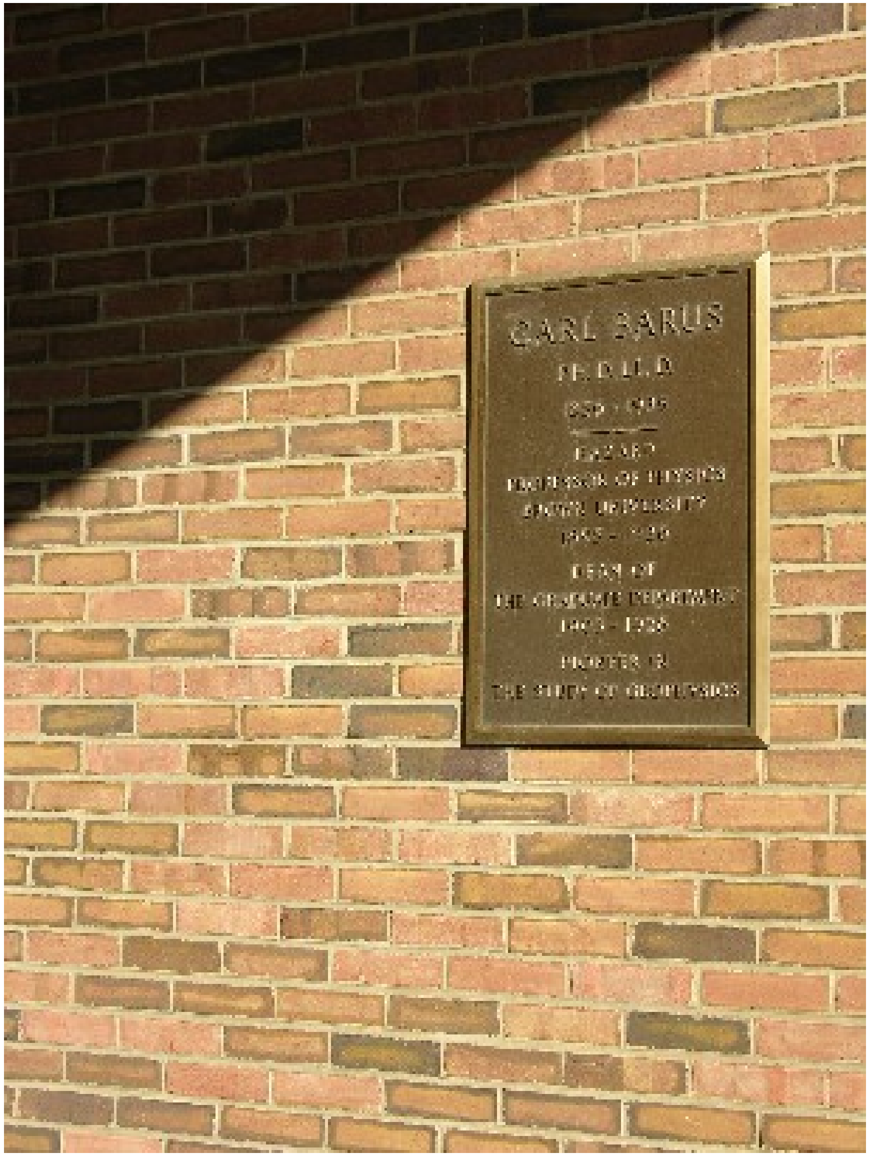




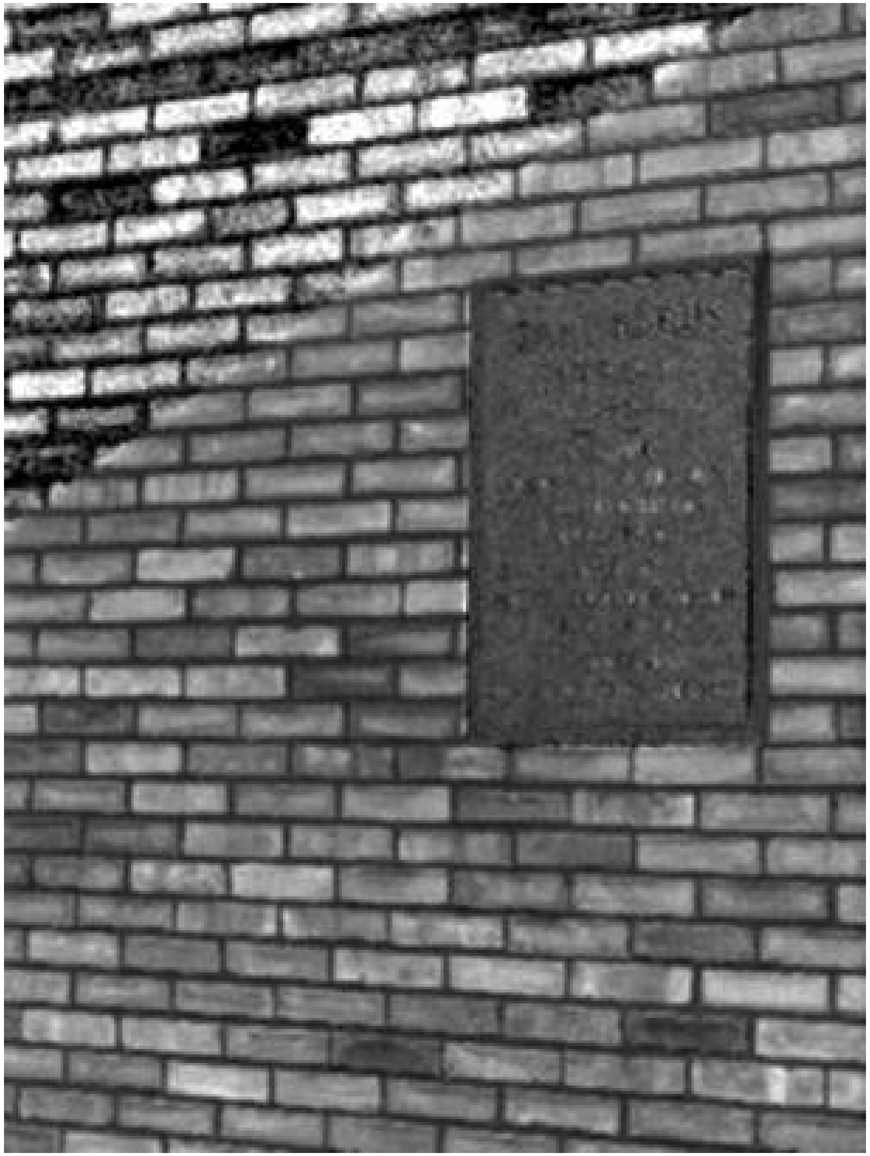
Edge in Original Image



Edge in Invariant Image

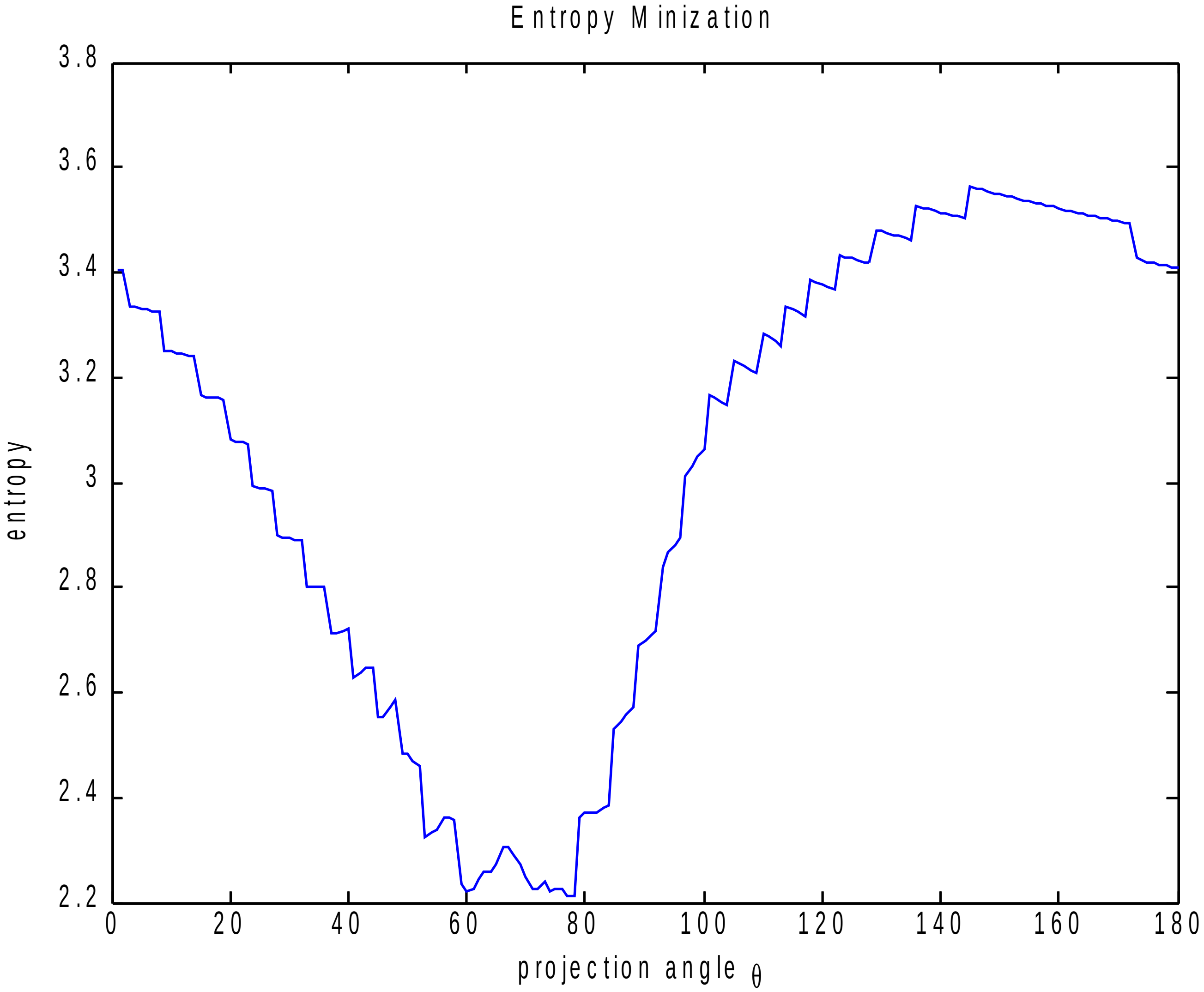


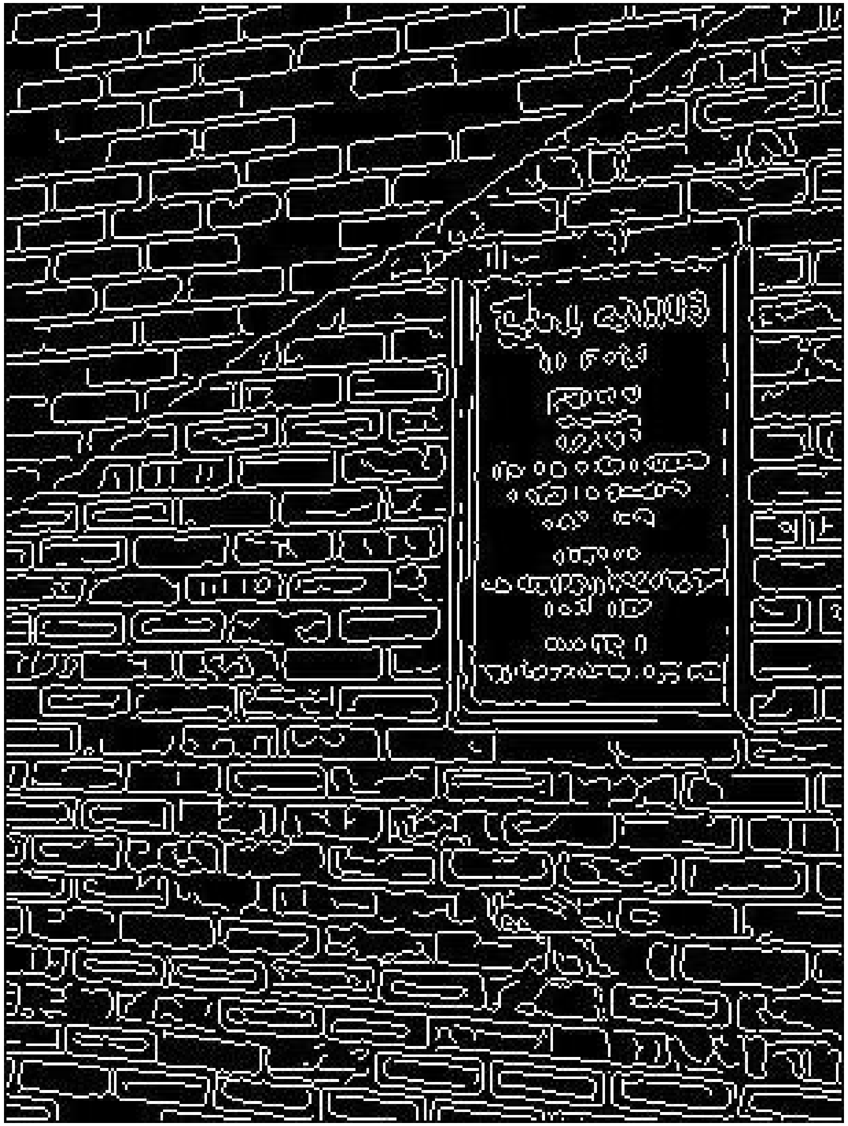
Original Image



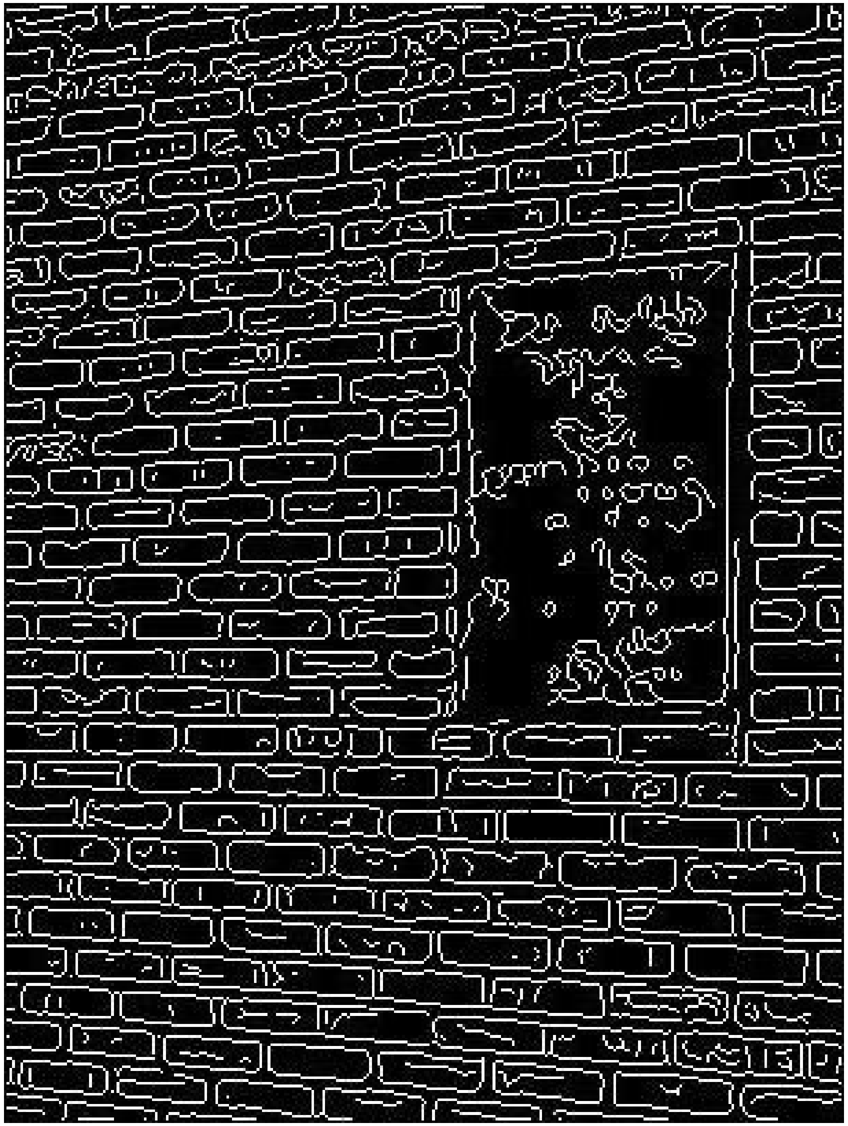
Invariant Image

Entropy Minimization (Camera: Nikon CoolPix8700)

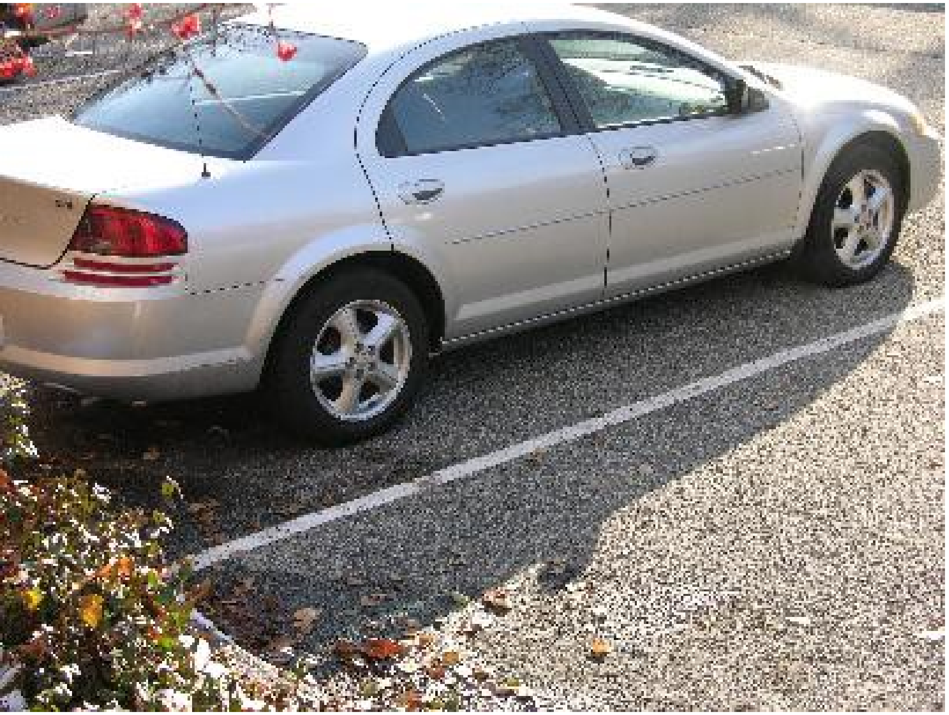




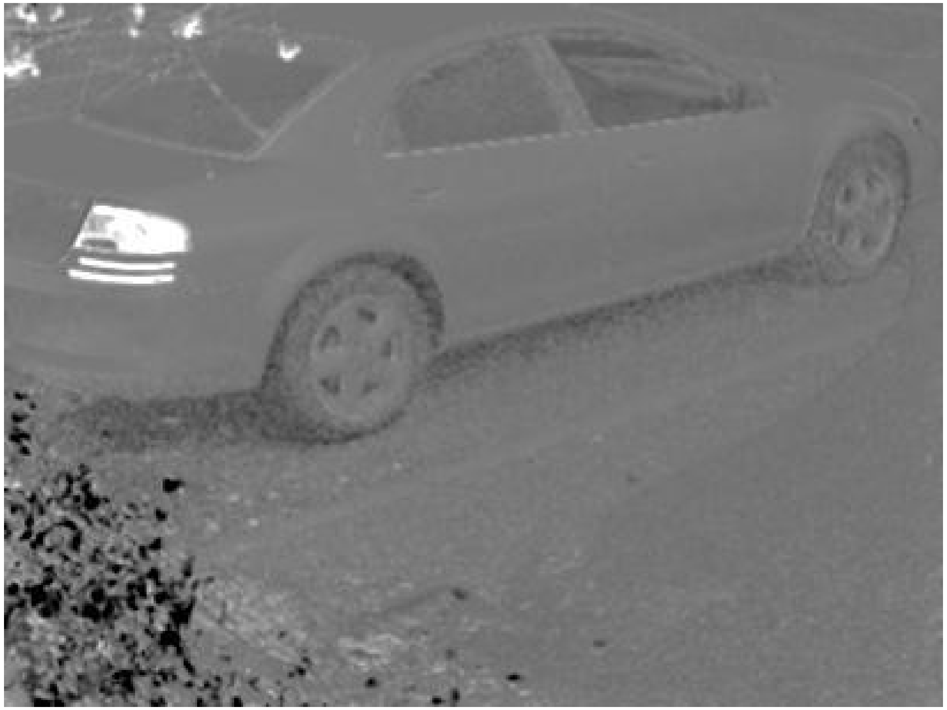
Edge in Original Image



Edge in Invariant Image

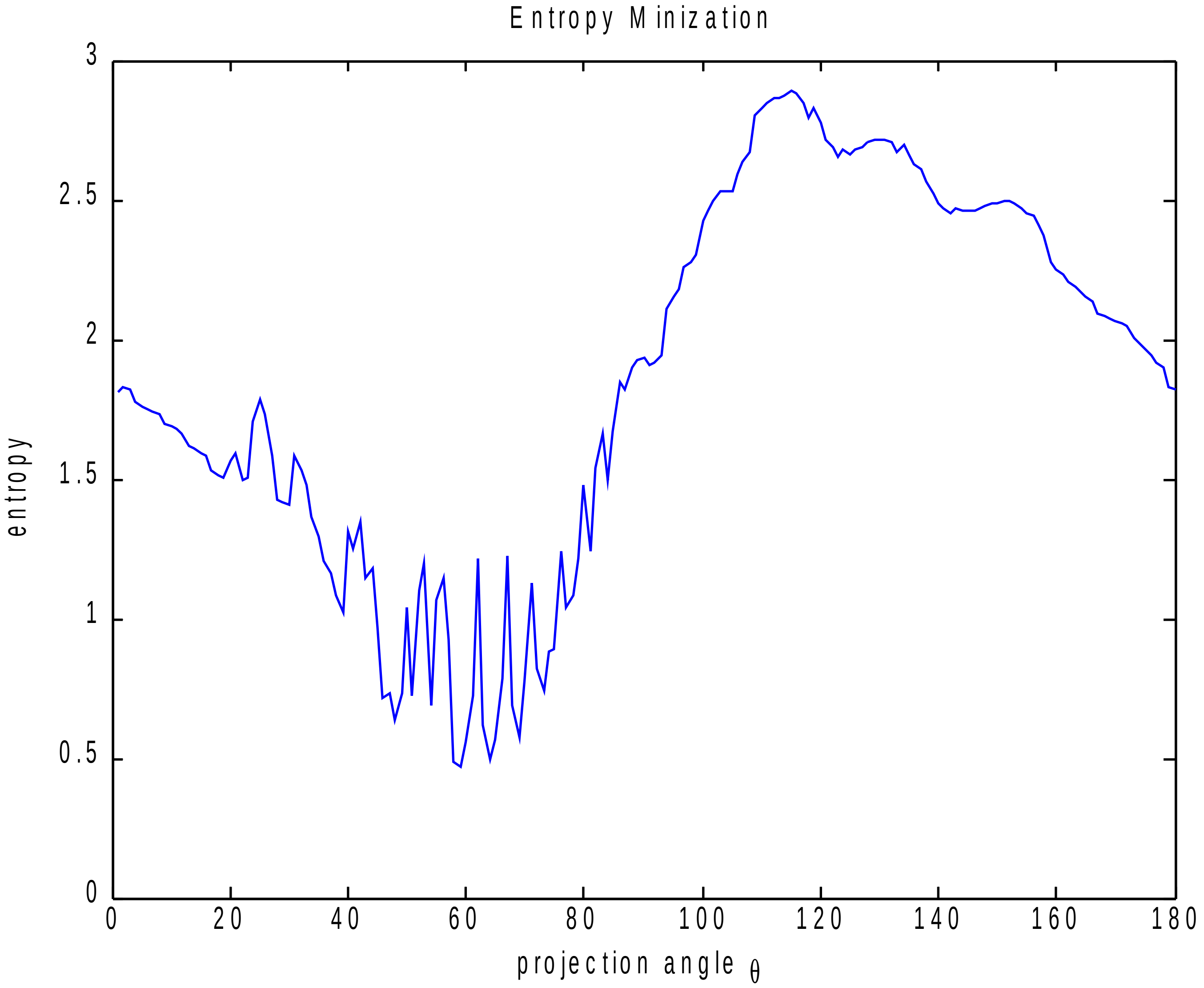


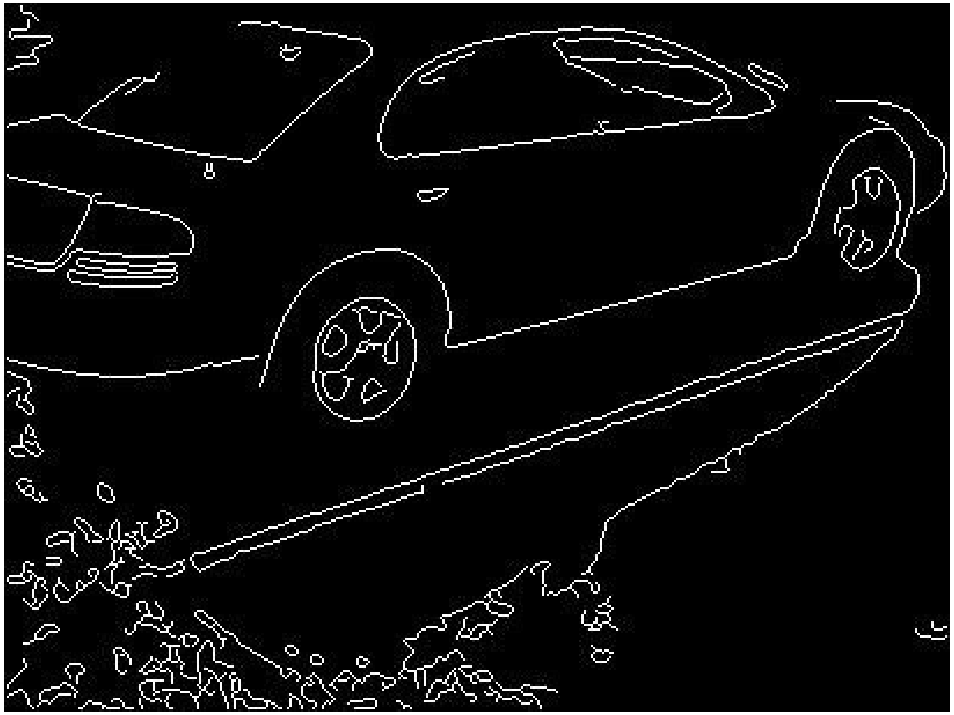
Original Image



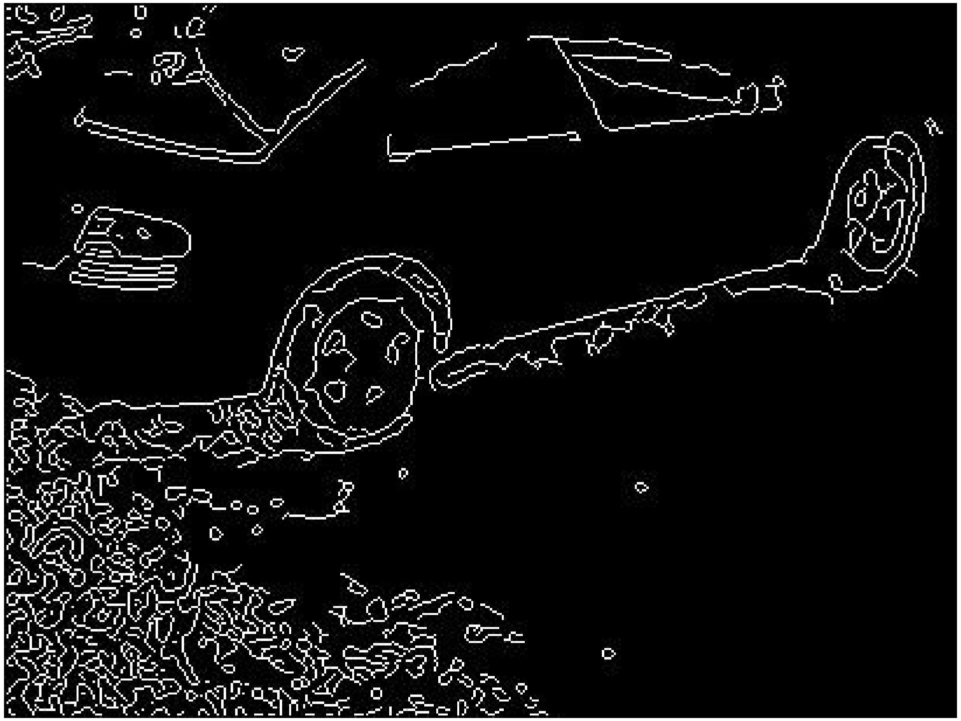
Invariant Image

Entropy Minimization (Camera: Nikon CoolPix8700)





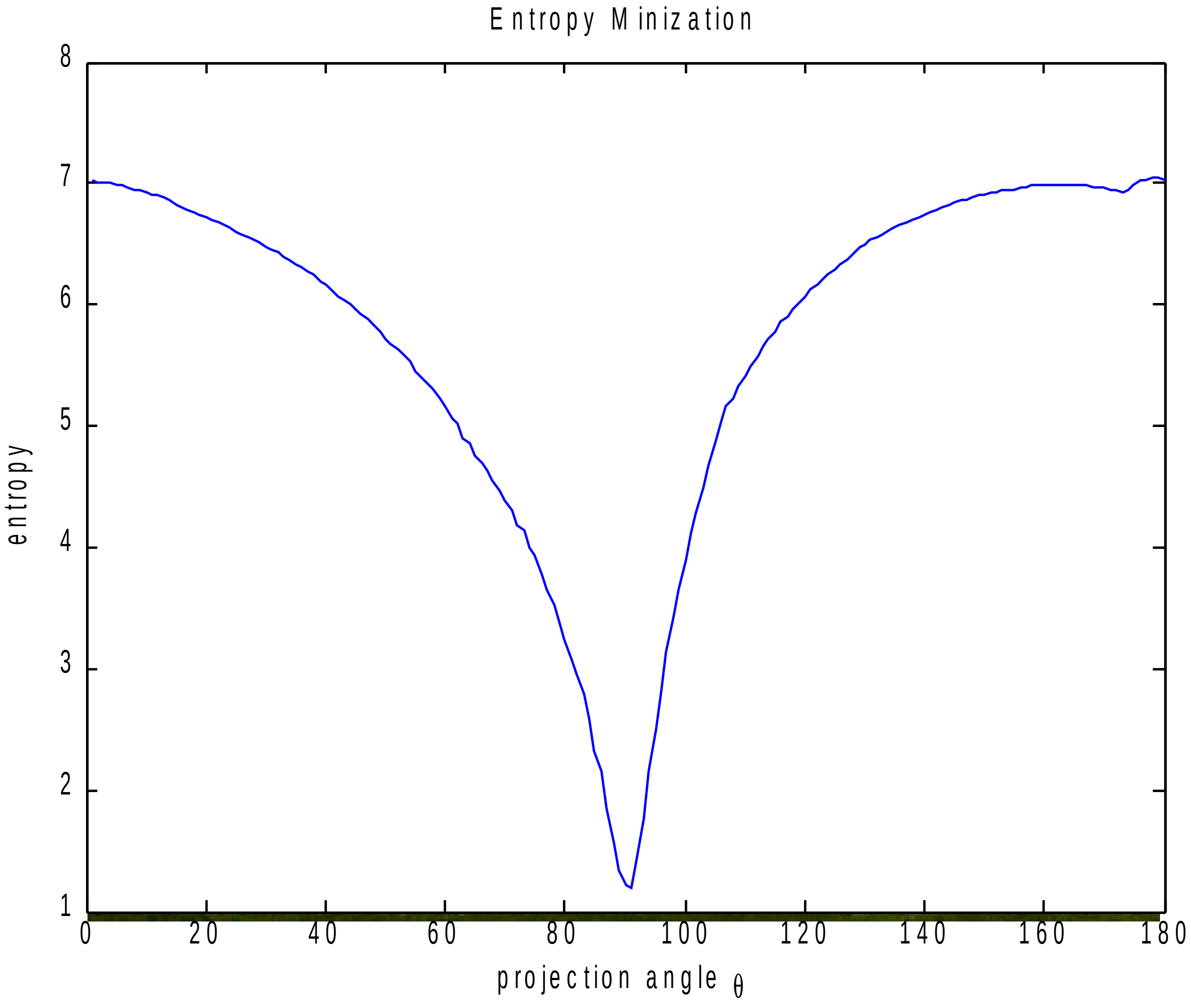
Edge in Original Image



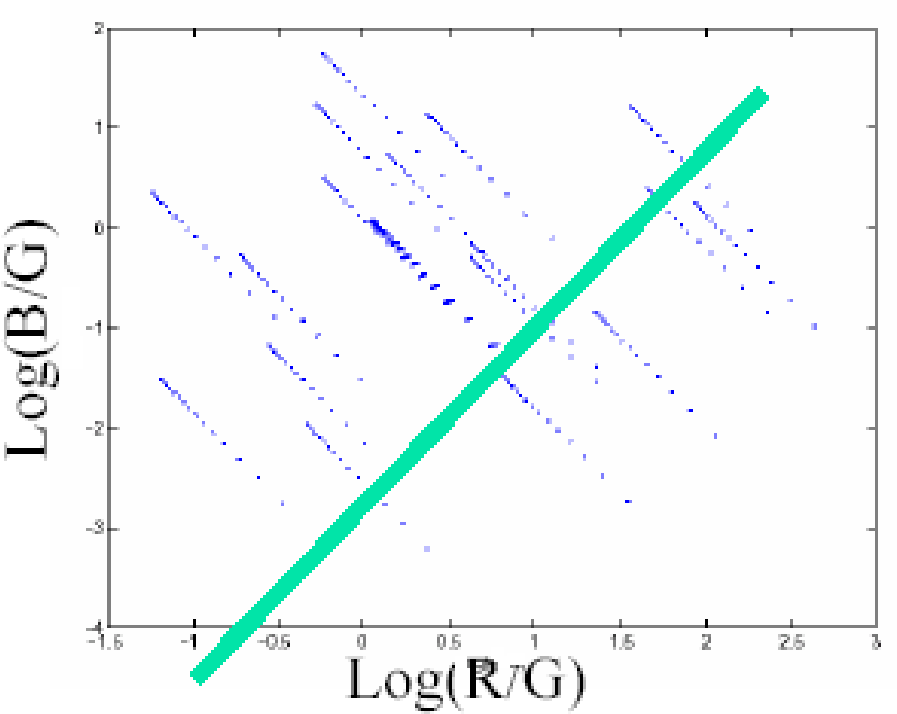
Edge in Invariant Image

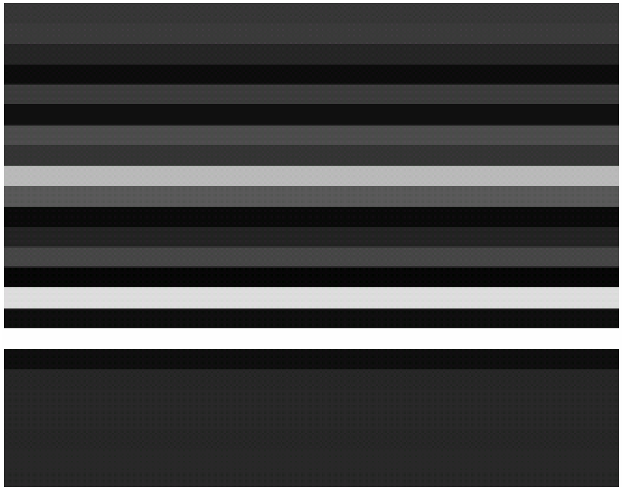
Entropy Minimization

(Camera: HP-912, better camera sensors?) [from author's website]



Is the projected 1-D data really “colorless”?





We can recover 2-D chromaticity along the line

Invariant chromaticity Image

- Recall that the grayscale image $\tau = \chi \times e^{\perp} = \chi \times [\cos\theta, \sin\theta]^{T}$

  let $\chi_{\theta} = P_{\theta}\chi$, where $P_{\theta} = e^{\perp}\left(e^{\perp}\right)^{T}$, the 3-D log ratio is recovered by $\mathring{\rho} = U^{T}\chi_{\theta}$

- Invariant chromaticity image:

$$\tilde{r} = \exp(\mathring{\rho}) / \sum_{i} \exp(\mathring{\rho}_{i})$$

$\tilde{r}$ is the invariant chromaticity image that presents the color information inherent in the 1-D projection yet absent in the grayscale invariant image τ

Determine the shadow edges

- Comparing the edges in the original image and the invariant image (Mean Shift?)

  Shadow Edge = Edge in Original but NOT in Invariant

- Thicken the identified shadow edges using morphological operato

- Let $\rho(x, y)$ be one of the RGB channels, and $\rho'(x, y)$ be one of the log of RGB channels. {Recall that $\rho_k = E(\lambda_k)S(\lambda_k)$, $k = r, g, b$ }, difference of $\rho'(x, y)$ reflects a change in either illumination or surface, the change in reflectance can be revealed by the operation:

$$S(x, y) = \begin{cases} 0 & if \left\|\nabla \rho'(x, y)\right\| > thre1 \ \& \left\|\nabla g(x, y)\right\| < thre2 \\ \nabla \rho'(x, y) & otherwise \end{cases}$$

Reintegrate

- Solve the Poisson Equation $\nabla^{2} q'(x,y) \amalg \nabla \times S(x,y)$ under Neumann boundary condition ($\nabla q'(x,y) \equiv 0$ on the boundary), where $q'(x,y)$ being the log of the *shadow-free image we wish to recover*.

How to solve the Poisson Equation?

In Fourier Domain, rewrite the equation as :

$$\Im\left[\nabla^{2} q'(x,y)\right] = -(u^{2}+v^{2})Q(u,v) = \Im\left[\nabla \times S(x,y)\right]$$

Solve for $Q(u,v)$,and the IFFT($Q(u,v)$) gives the $q'(x,y)$ up to an unknown constant.

- Map the maximum of each of the R,G,B channel to 1 and factor the unknown constant.

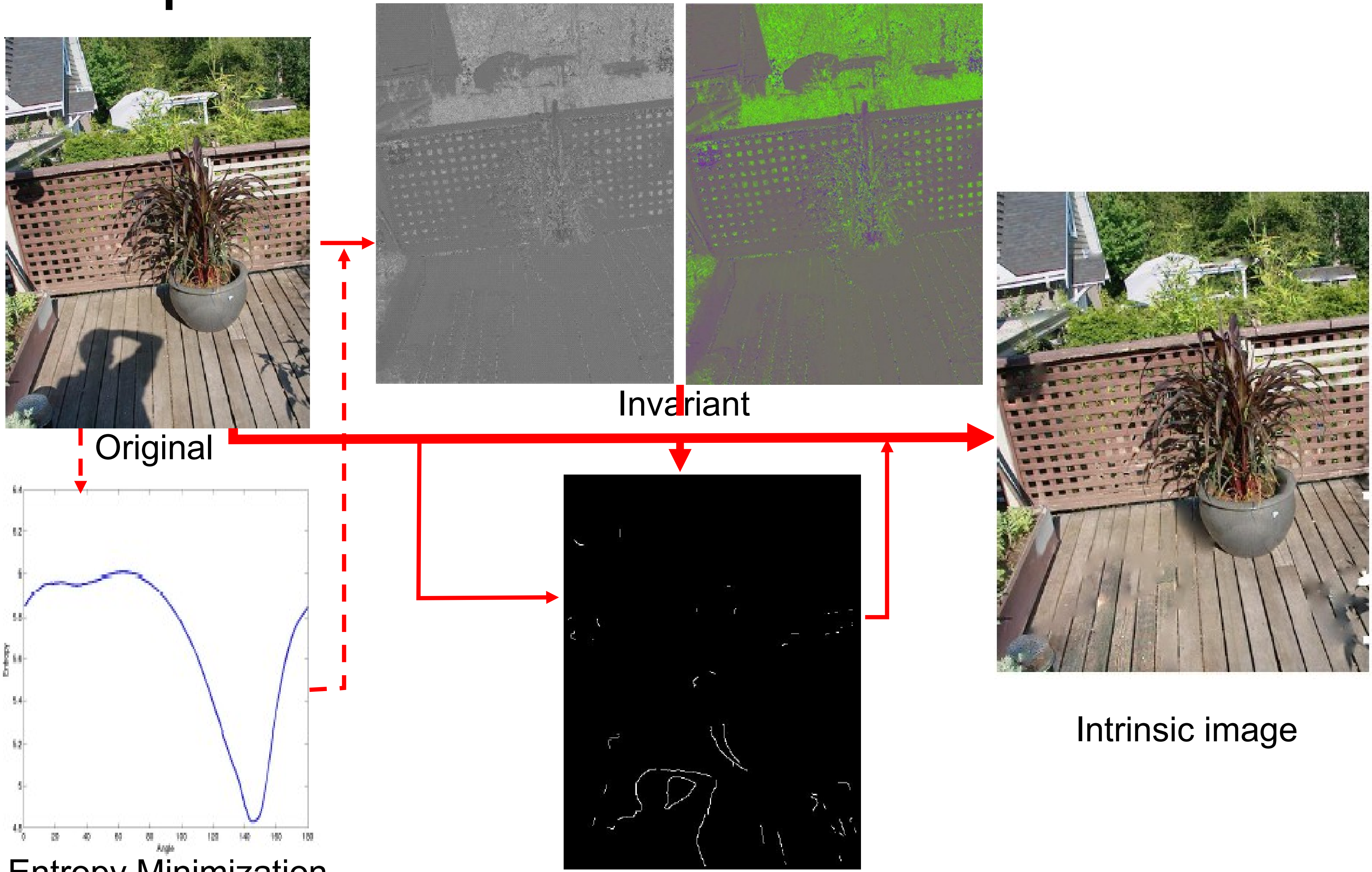
Plan

- Take some qualified pictures
- Preprocessing (Smoothing, Data refinement etc.)
- Geometric Mean: to exclude bias for a certain color channel
- Entropy Minimization to obtain the best projection angle
- Project to get 1D grayscale invariant
- Recover part of the color information by chromaticity invariant image
- Determine shadow edge (edge thickening, clear noisy edges, etc.)
- Remove shadows in the gradient of the original image
- Solve Poisson PDE in Fourier domain (perform reintegration)
- RGB intrinsic image.

Dreams: Improvement and Compare to other Algorithms

← I am somewhere here

Expected Results :



Original

Invariant

Entropy Minimization

Shadow Edge

Intrinsic image

Reference

- “*Intrinsic Images by Entropy Minimization*”, Graham D. Finlayson, Mark S. Drew and Cheng Lu, ECCV, Prague, 2004;
- “*Removing Shadows from Images*”, Graham D. Finlayson, Steven D. Hordley and Mark S. Drew; Removing shadows from images. In European Conference on Computer Vision, pages 4:823–836, 2002;
- “*Color constancy at a pixel* ”, G.D. Finlayson and S.D. Hordley, Journal of Optical Society of America. A, 18(2):253–264, Feb. 2001.
- “*On the Law of Distribution of Energy in the Normal Spectrum*”, Max Planck, *Annalen der Physik,* vol. 4, p. 553 ff (1901)
- **“***Recovery of Chromaticity Image Free from Shadows via Illumination Invariance***”**, Mark S. Drew, Graham D. Finlayson and Steven D. Hordley, ICCV’03 Workshop on Color and Photometric Methods in Computer Vision, Nice, France, pp. 32-39
- “*Deriving intrinsic images from image sequences*”, Yair Weiss, ICCV ’01, Volume II pages: 68-75.

Thanks!