Geometric Methods in Passive-blind Image Forensics

Tian-Tsong Ng, Shih-Fu Chang, Jessie Hsu

Department of Electrical Engineering

Columbia University, New York, USA

Mao-Pei Tsui

Department of Mathematics

University of Toledo, Ohio, USA

Columbia University in the City of New York

dvmm DIGITAL VIDEO MULTIMEDIA LAB

Background
Passive-blind Image Forensics

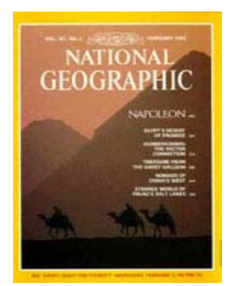
- Digital images is pliable to manipulation.
- [WSJ 89] 10% of color images published in US were altered.
- Image forensic: to find out the condition of an image without any prior information.
- Two main functions of image forensics:
  - Image Forgery Detection
  - Image Source Identification

Image Forgery Hall of Fame


LA Times '03


Fonda Speaks To Vietnam Veterans At Anti-War Rally

Internet '04



Nat. Geo. '92

Times '96

Problem I
Image Forgery Detection

- Image forgery: Photomontage, images with removed objects, retouched images, etc.
- Adobe Photoshop – 5 million registered users (2004)
- Photoshop altered images are common – 178,582 images on www.worth1000.com (2005)

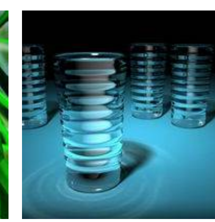
www.worth1000.com (scandal category)

Problem II
Image Source Identification

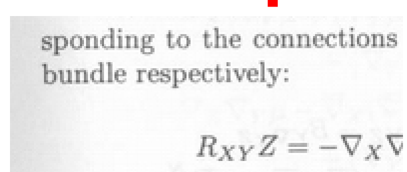
- Identify image production devices: camera, computer graphics, printer and scanner, etc.
- Identify nature of the image scene: 2D photo or 3D scene
  - A face recognition system should not be fooled when being shown a 2D face photo of someone.

From which camera?

CG Or Photo?



From which printer?



sponding to the connections
bundle respectively:

$$R_{XY}Z = -\nabla_X\nabla$$

What is a Grayscale Image?

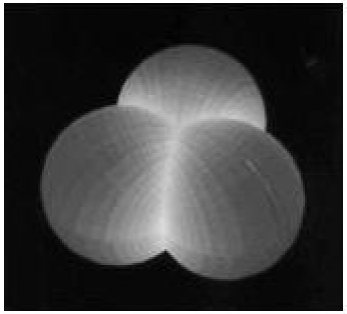
- A grid sampling of a 2-D function.

$$I : (x, y) \subset R^2 \to R$$

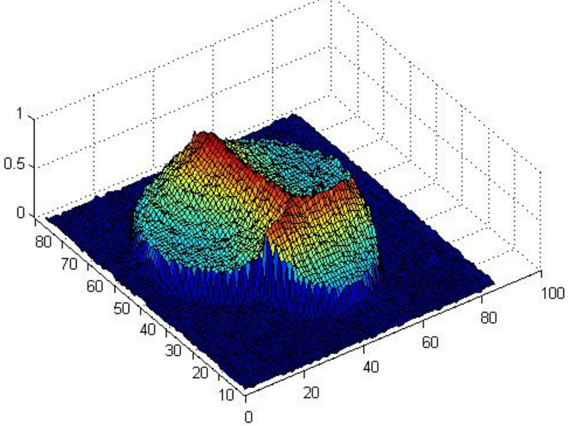
- The graph of an image is a submanifold in R^3.

$$F : (x, y) \subset R^2 \to (x, y, I(x, y)) \subset R^3$$

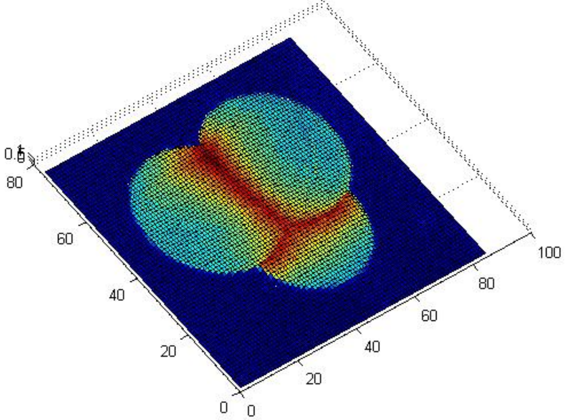
- We can use techniques in differential geometry to study images!



View 1 of I



View 2 of I



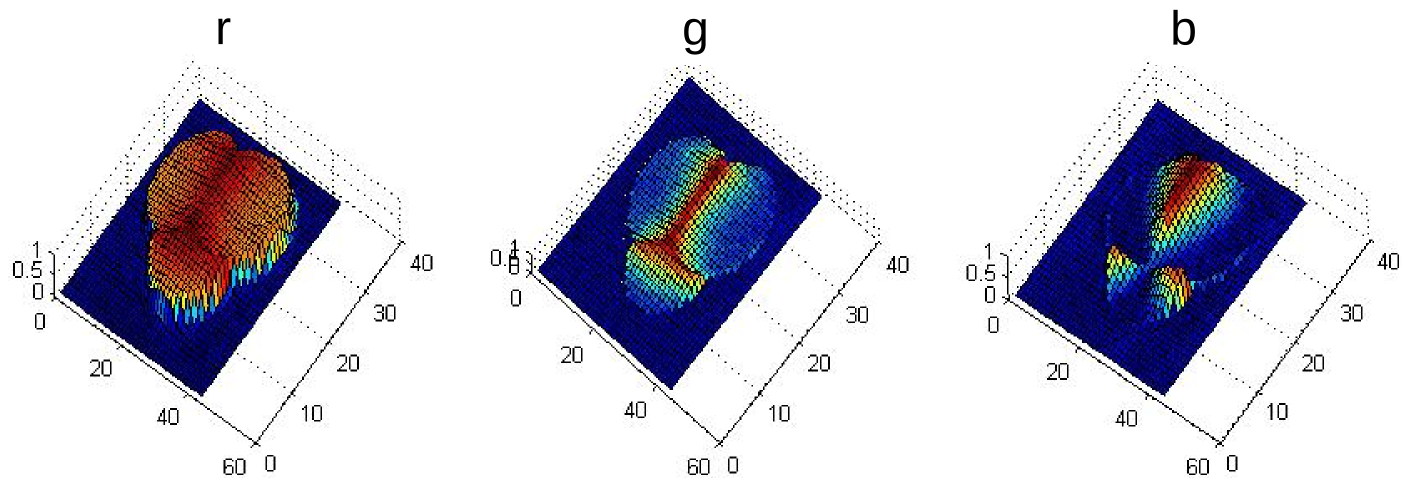
What is a Color Image?

- A 2-D vector function.

$$I : (x, y) \subset R^2 \rightarrow (r, g, b) \subset R^3$$

- The graph of an image function is a submanifold in R^5.

$$F : (x, y) \subset R^2 \rightarrow (x, y, r(x, y), g(x, y), b(x, y)) \subset R^5$$



CG Or Photo?

Part I:

Geometric Features for Distinguishing Photographic Images and Computer Graphics

Prior Work
Photo vs. CG

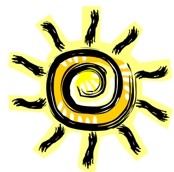
- [Ianeva et al. 03] Classifying photo and general CG (including drawing and cartoon).
  - For the purpose of improving video key-frame retrieval.
- [Lyu & Farid 05] Classifying photo and photorealistic CG.
  - Using wavelet statistics.
  - 67% detection rate (1% false alarm).
  - provides little insight into the physical differences between photo and CG.

Our Approach

- Analyze the physical differences between Photo and CG, in terms of the image generative process.
- Propose a geometry-based image description framework

Image Generative Process

- Photographic Images



Light source

(1) Complex surface model

- Subsurface scattering of human skin.
- Color dependency.

(2) Complex object geometry

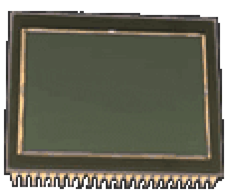
- Human skin texture follows biological system.
- Building surface formed by air erosion.

(3) Non-linear camera response function

- Not an arbitrary transform.

What is Camera Response Function (CRF)?

Scene radiance

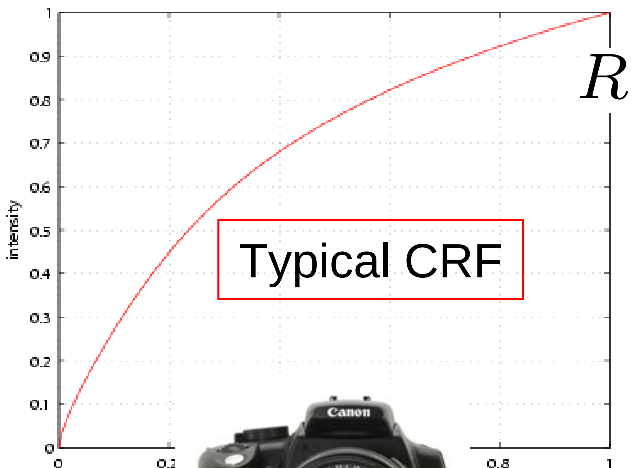


Camera sensor

$r(x, y)$

Image Irradiance

$f(\cdot)$



Typical CRF

$R(x, y) = f(r(x, y))$

Image Intensity

Every model of camera has its unique CRF.

Image Generative Process

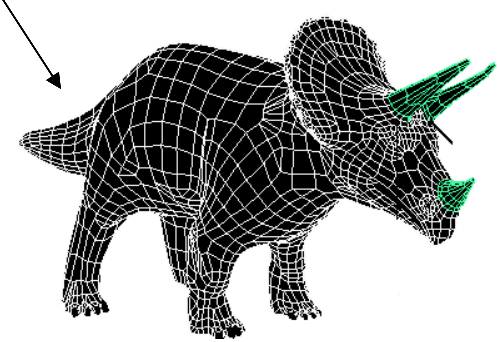
- Computer Graphics

3 Differences for Photo and CG
(1) Surface Model Difference.
(2) Object Model Difference.
(3) Acquisition Difference.

Light source

(3) Non-standard Post-processing
- Subject to the artist’s taste.
- May different from camera transform.

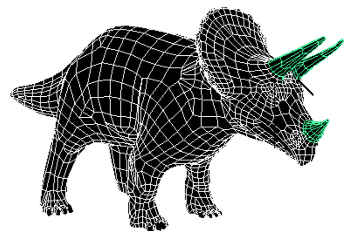
(1) Simplified surface model
- Assume color independence.



Post-processing

(2) Polygonal object geometry
- Reduced mesh resolution for computational efficiency.
- Without care, it introduces sharp structures in rendered images.

Feature Correspondences



	Acquisition Difference	Object Model Difference	Surface Model Difference
Differential Geometry	Image Gradient	Second Fundamental Form	Surface Laplacian
Fractal Geometry		Distribution of the Local Fractal Dimension	
Local Patch Statistics	Distribution of the 3x3-pixels Local patches		

Differential Geometry I
Image Gradient

- Non-linear camera transform has effects on image Gradient!

Camera Model

r
image irradiance

↓

R=f(r)
Camera Transfer Function

↓

R
image Intensity

Chain Rule

$$\frac{dr}{dx}$$

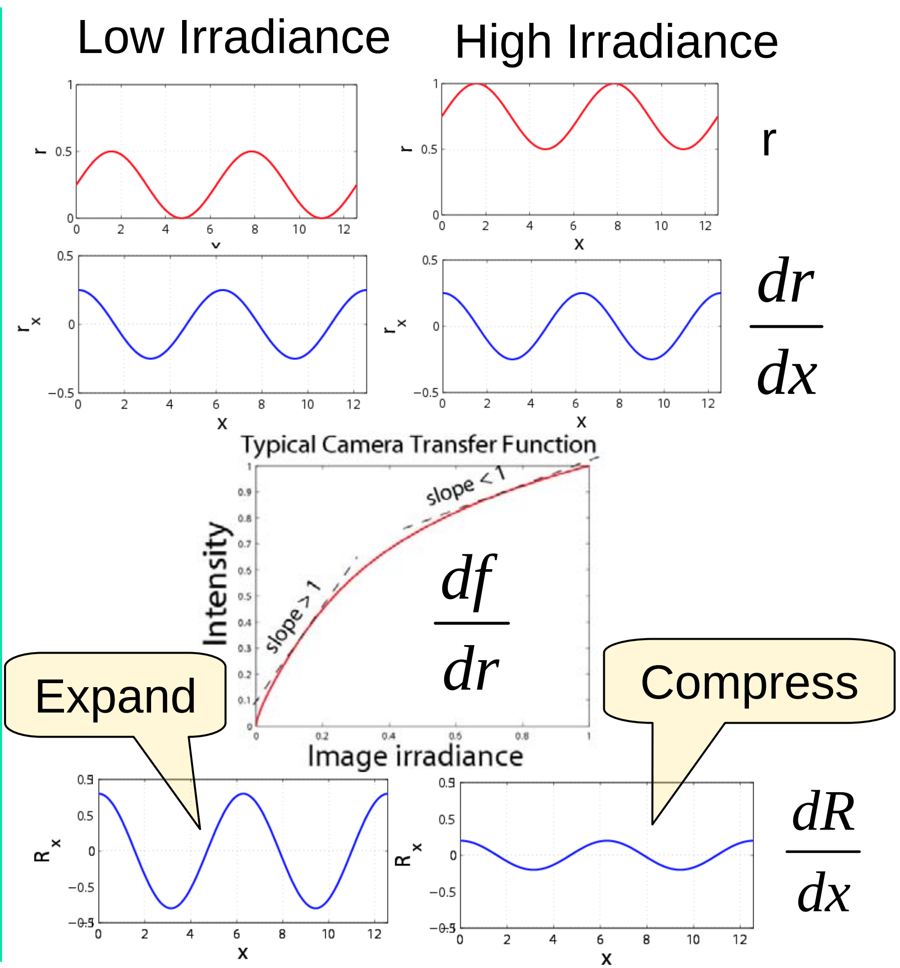
↓

$$\frac{df}{dr}$$

Slope of the curve

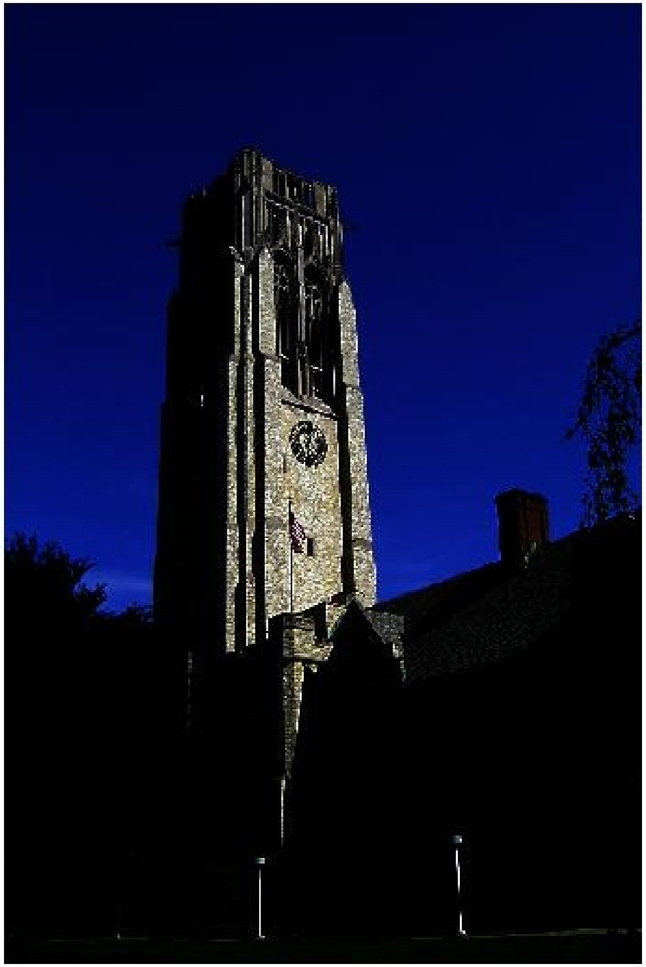
↓

$$\frac{dR}{dx} = \frac{df}{dr}\frac{dr}{dx}$$



The Visual Effect of CRF Transform

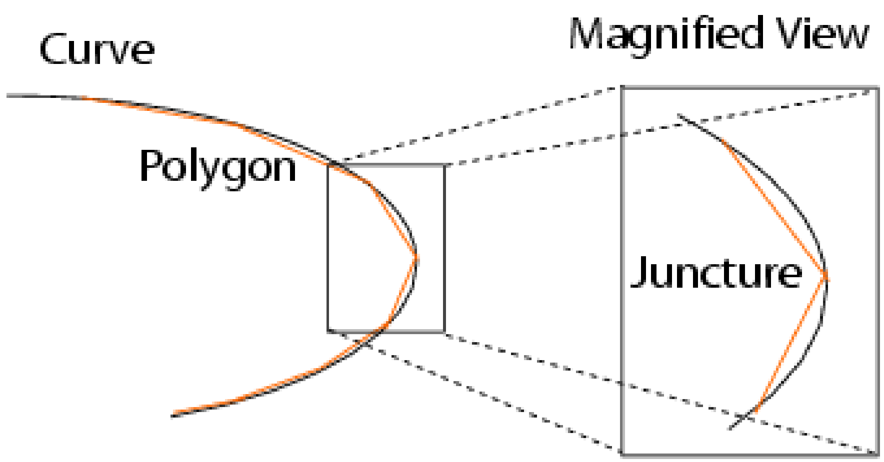
Before CRF Transform



After CRF Transform

Differential Geometry II
Second Fundamental Form

- Polygonal Model leads to sharp structures
  - At the junctures, the polygon is always sharper than the smooth curve.



A smooth curve is approximated by a polygon

Unusually sharp transition

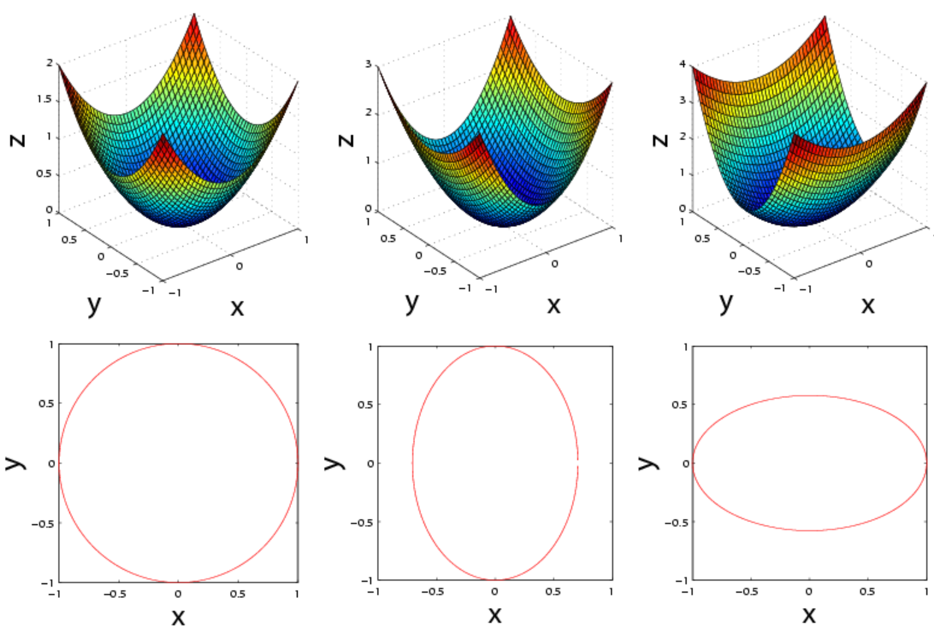
Differential Geometry II
Second Fundamental Form

- Locally, any surface can be written as a graph of a differentiable function over the tangent plane.
- The local graph can be approximated by a quadratic function.
  - The Hessian of the quadratic function is the second fundamental form.
  - The Hessian can be characterized by 2 eigenvalues
  - Large eigenvalues implies sharp structures

3D plot of elliptic Quadratic function.

Cross-section of the quadratic function at z=1.

eigenvalues



(1,1) (2,1) (3,1)

Differential Geometry III
Surface Laplacian Vectors

- Rendering of CG often assumes color independence in the object surface model (generally, not true for real-world object):
  - We capture the difference in the RGB correlation for Photo and CG using surface Laplacian vectors.
  - The vectors measures the correlation between R, G and B.

Differential Geometry III
Surface Laplacian Vectors

Graph of a RGB color image

$$F_{RGB} : (x,y) \subset R^2 \mapsto (x,y,I_R(x,y),I_G(x,y),I_B(x,y)) \subset R^5$$

Surface Laplacian Vectors
$$\begin{aligned}\triangle_g I_i &= \frac{1}{|g|}\left(\partial_x\left(\sqrt{|g|}\,(g^{xx}\partial_x I_i + g^{xy}\partial_y I_i)\right)\right) + \\ &\quad \frac{1}{|g|}\left(\partial_y\left(\sqrt{|g|}\,(g^{yx}\partial_x I_i + g^{yy}\partial_y I_i)\right)\right)\end{aligned}$$

where

$$g^{-1} = \begin{pmatrix} g^{xx} & g^{xy} \\ g^{yx} & g^{yy} \end{pmatrix} = \begin{pmatrix} 1+\sum_j(\partial_x I_j)^2 & \sum_j \partial_x I_j \partial_y I_j \\ \sum_j \partial_y I_j \partial_x I_j & 1+\sum_j(\partial_y I_j)^2 \end{pmatrix}^{-1}$$

$$|g| = 1+\sum_j |\nabla I_j|^2 + \frac{1}{2}\sum_{j,k}|\nabla I_j \times \nabla I_k|^2, j,k = R,G,B$$

Dataset

Columbia Open Dataset

- First publicly available Photo/CG dataset.
- Consists of 4 subsets, 800 images for each subset.

Personal Photo	Google Photo	Internet CG	Recaptured CG

From a few personal collections of photo

Downloaded from Google Image Search

Downloaded from the 3D artist websites

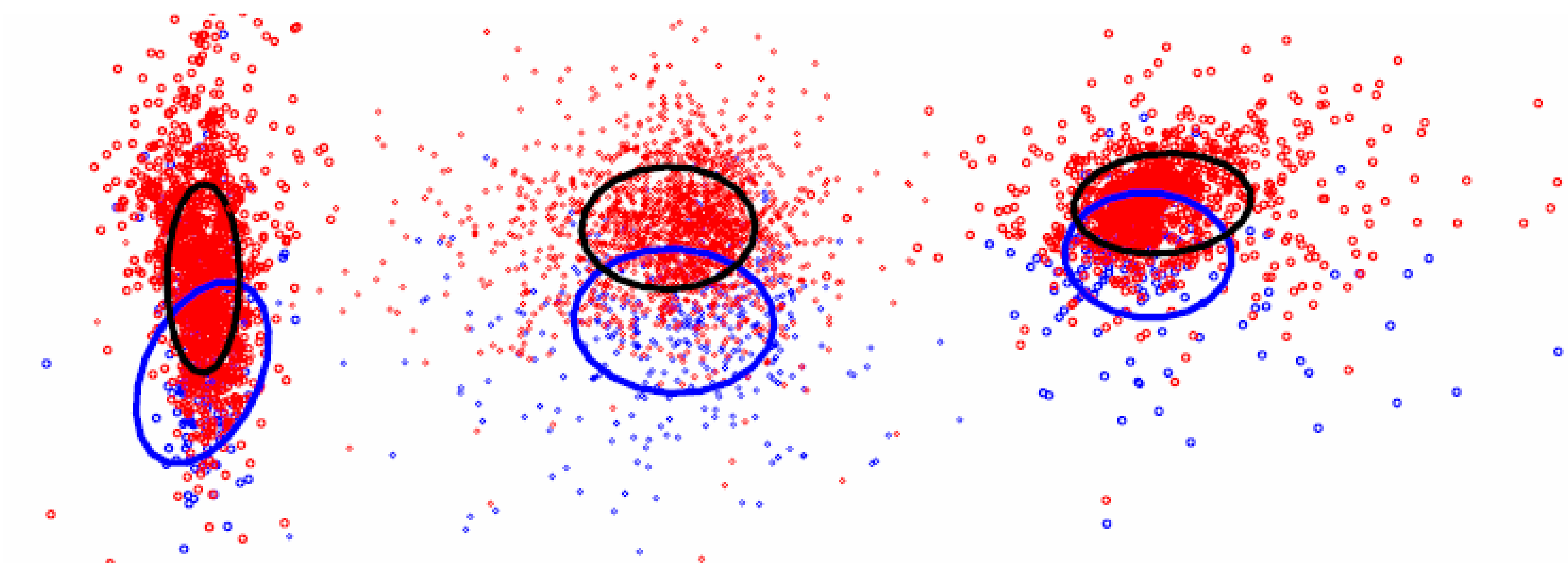
Recaptured from a LCD screen by a Canon G3 camera

Available at http://www.ee.columbia.edu/trustfoto

Feature Vector 2D Projection

- We compute the differential geometry quantities at every point of an image.
- Then, we compute the moment of the quantities.
- Below are the 2D projection of the moments.

Gradient | Second Fundamental Form | Surface Laplacian



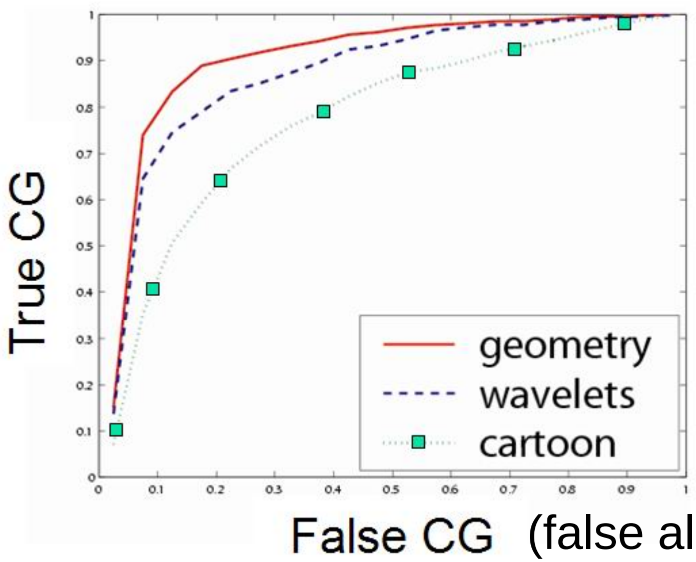
Experimental Results I
Support Vector Machine Classification

- SVM classification with radial basis function (RBF) kernel.
- Cartoon feature is the conventional feature for modeling the general computer graphics (includes cartoon or drawing)

Features	Geometry	Wavelets	Cartoon
Accuracy	83.5%	80.3%	71.0%

Receiver operating characteristic (ROC) curve

(Detection rate)



(false alarm)

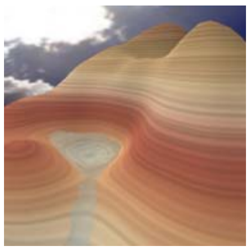
Photo
Vs
Internet CG

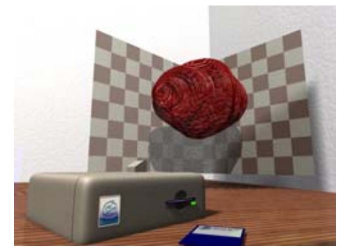
Online Demo III

Consistency with Human Judgments

Human Judgments

CG




Photo





As one of the application scenarios, the cases with disagreement may be handed to experts for further analysis.

Open issues

- Distinguishing Photo and CG at the level of the local region.
- Designing counter-measure for the Oracle attack.
  - When the attackers have access to the detector, they can modify an image until they obtains the desired output from the detector!
- Capturing global scene authenticity (e.g., consistency between lightings and shadows).

Canon
Rebel XT

Canon
Powershot
G3

Nikon D70

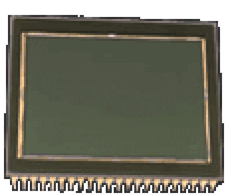
Part II:

Camera Response Function Estimation from a single-channel Image Using Differential Invariants

Prior Work in Image Forgery Detection – by Camera Authentic Characteristics

- What is Camera Response Function (CRF)?
- CRF can be used for:
  - Identifying the model of the camera for an image.
  - Detecting photomontage by identifying image fragments from different models of camera.

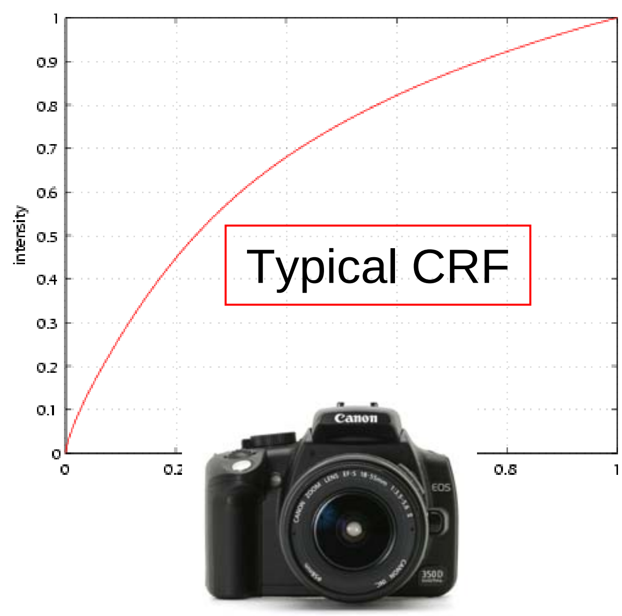
Scene radiance



Camera sensor

$$r(x,y)$$

Image Irradiance



$$R(x,y) = f(r(x,y))$$

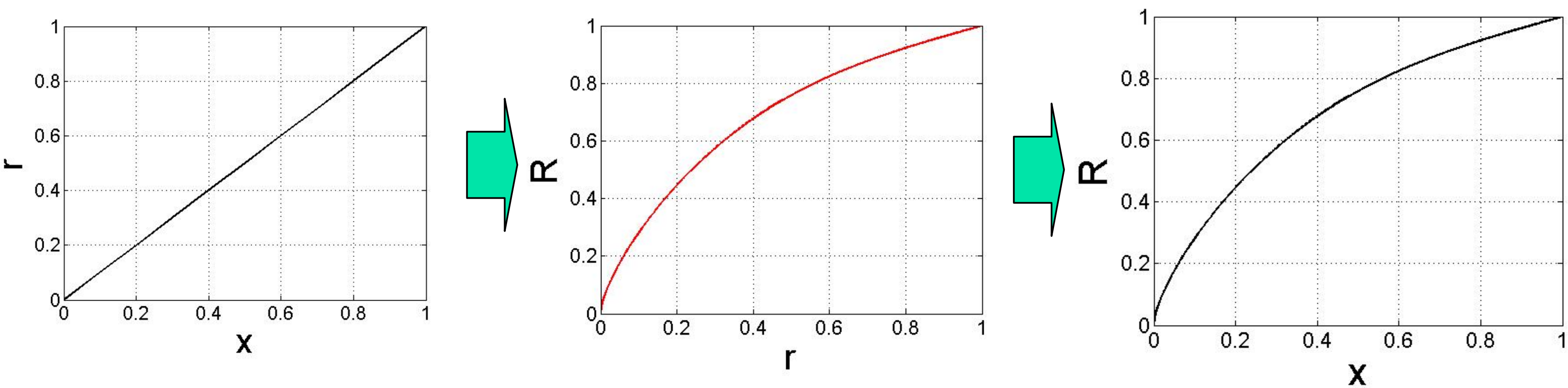
Image Intensity

Our Approach

- Our work proposes a CRF estimation method for a single image based on a set of differential invariants.
- Prior work uses either more than one image or the entire RGB image (3-color channels) to recover the CRF.
- We tested on a single-color-channel images!

Main Idea behind Our Approach

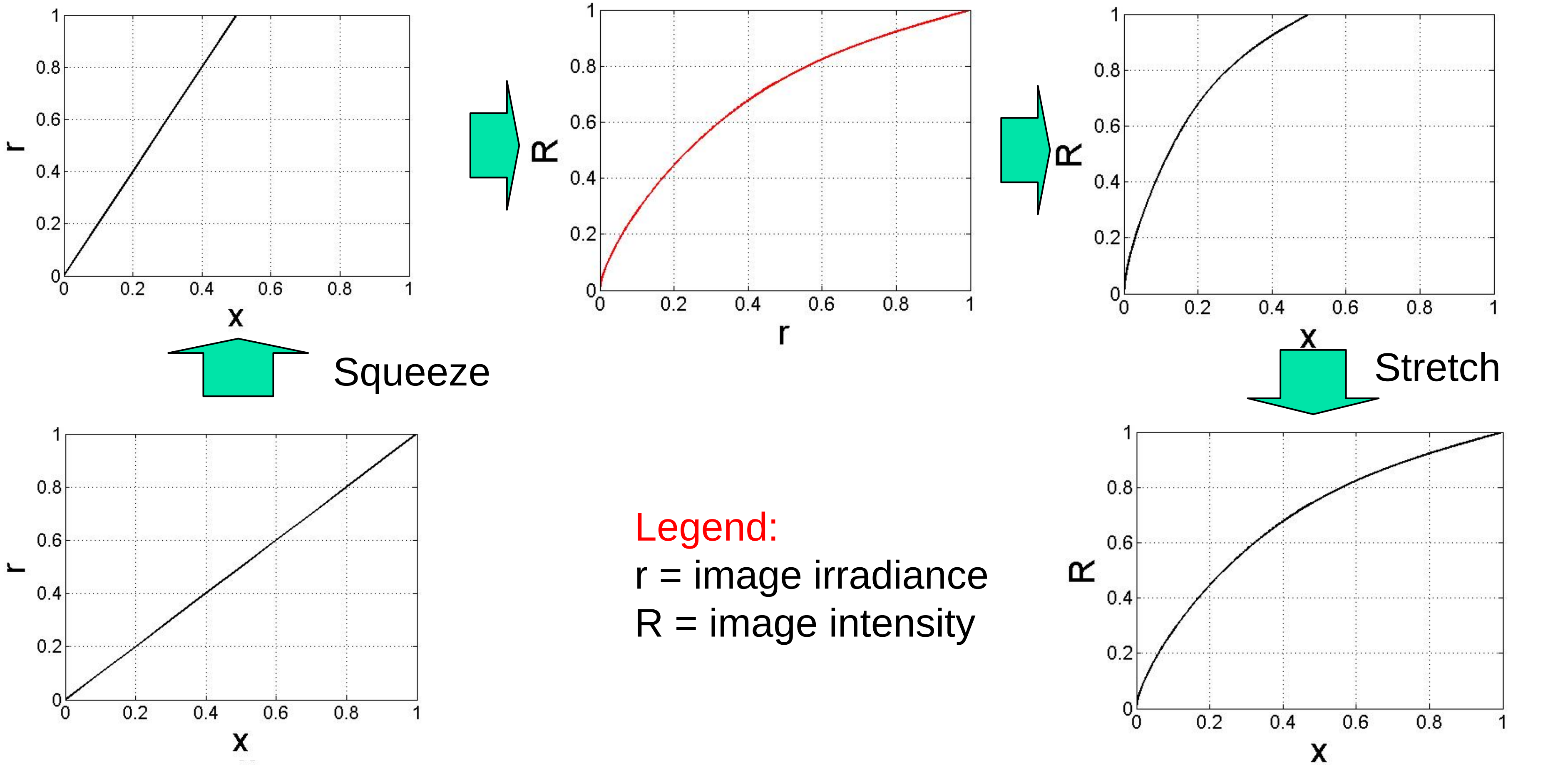
- 45 degree linear image irradiance



Legend:
r = image irradiance – unknown to users
R = image intensity – output of a camera

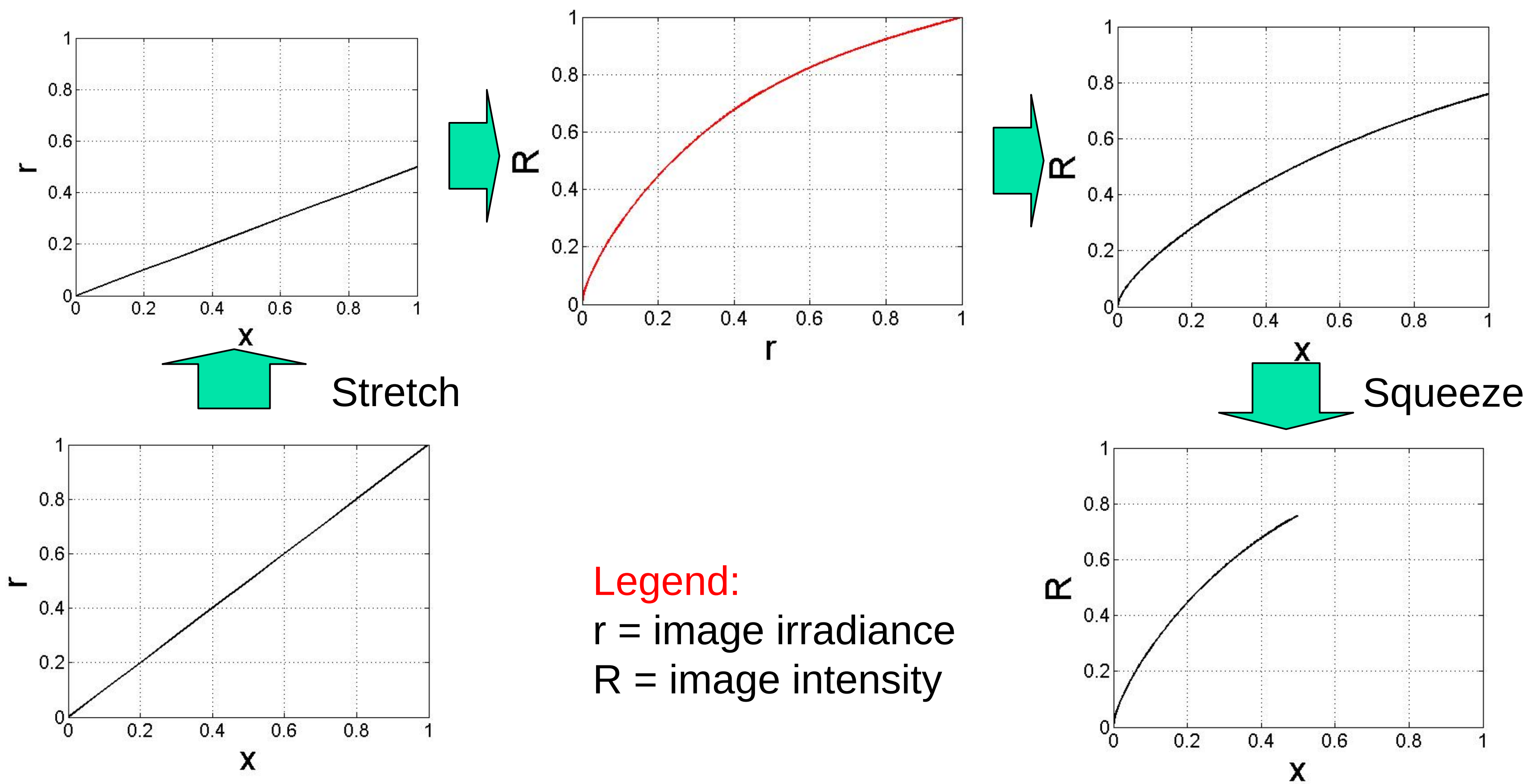
Main Idea behind Our Approach

- squeezed linear image irradiance



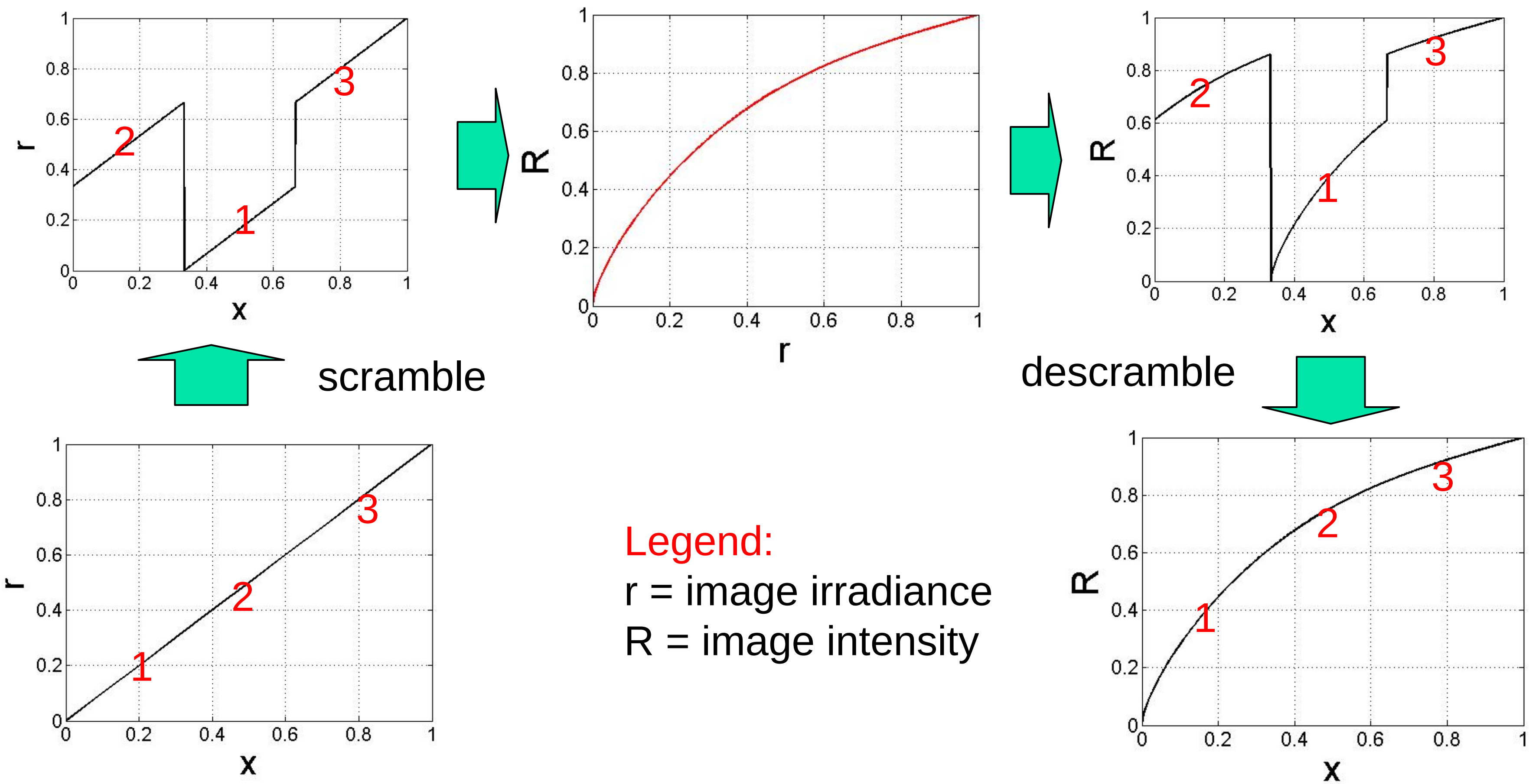
Main Idea behind Our Approach

- stretched linear image irradiance



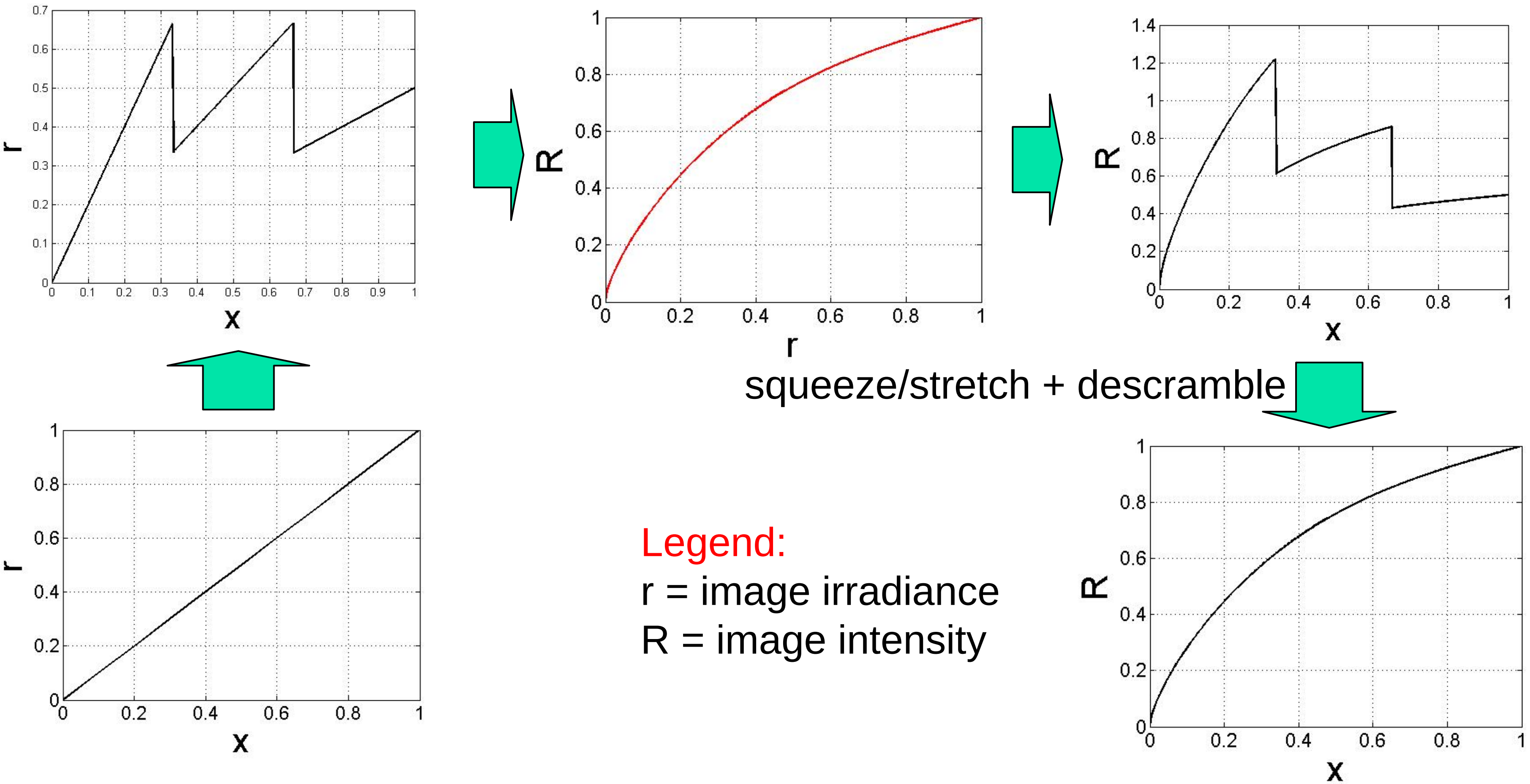
Main Idea behind Our Approach

- scrambled linear image irradiance



Main Idea behind Our Approach

- Squeezed/stretch + scrambled linear image irradiance



Legend:
r = image irradiance
R = image intensity

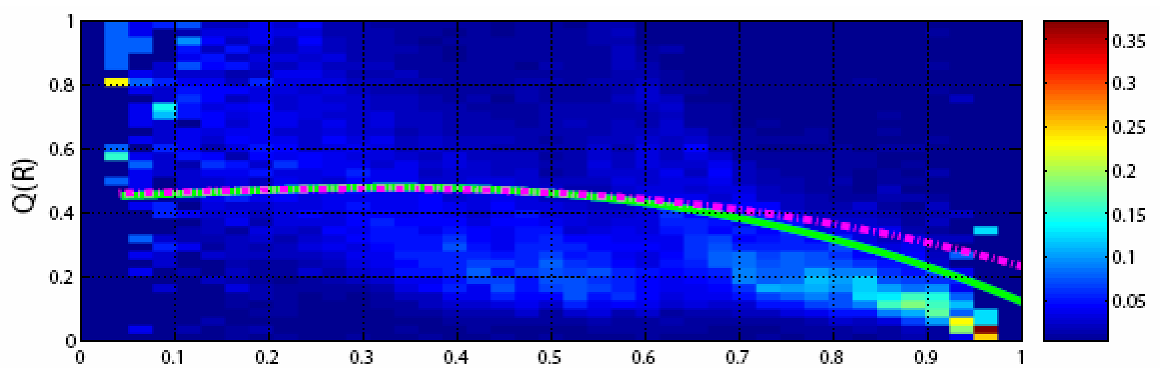
2 Main steps in Our Approach for estimating CRF

- 1) squeezing and stretching:
  - Obtain a equivalence class for the action of squeezing and stretching by a set of geometry invariants.

Linear Geometry Invariants

$$\frac{R_{xx}}{R_x^2} = \frac{R_{yy}}{R_y^2} = \frac{R_{xy}}{R_x R_y} = \frac{f''(f^{-1}(R))}{(f'(f^{-1}(R)))^2}$$

- 2) Descrambling
  - Fit a parametric CRF curve to the computed geometry invariants.



Legend:
r = image irradiance
R = image intensity

First-order Differential Invariants

$$R = f(r), r = f^{-1}(R)$$

Assume locally planar image irradiance $r = ax + by + c$

$$R_x = f'(r)r_x, R_y = f'(r)r_y$$

Depends on CRF

Depends on the irradiance geometry

$$(R_x, R_y) = f'(r)(r_x, r_y)$$

$$(R_{xx}, R_{xy}, R_{yy}) = f''(r)(r_x^2, r_x r_y, r_y^2)$$

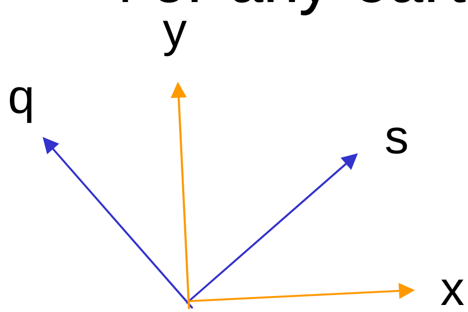
Linear Geometry Invariants

Independent of Geometry $\dfrac{R_{xx}}{R_x^2} = \dfrac{R_{yy}}{R_y^2} = \dfrac{R_{xy}}{R_x R_y} = \dfrac{f''(r)}{f'(r)^2} = \dfrac{f''(f^{-1}(R))}{(f'(f^{-1}(R)))^2}$

Note: $\dfrac{R_{xxx}}{R_x^3}$ is also an invariant. The above involves the lowest order derivatives.

Rotational Symmetry for Linear Geometry Invariants

- For any Cartesian coordinate frame with basis s and q:



$$\frac{R_{ss}}{R_s^2} = \frac{R_{qq}}{R_q^2} = \frac{R_{sq}}{R_s R_q} = \frac{f''(r)}{(f'(r))^2}$$

With respect to the original x-y coordinate, the derivative operators is given by:

$$D_s = \cos(\theta)\frac{\partial}{\partial x} + \sin(\theta)\frac{\partial}{\partial y}$$

$$D_q = -\sin(\theta)\frac{\partial}{\partial x} + \cos(\theta)\frac{\partial}{\partial y}$$

$$D_{ss} = D_s^2, D_{qq} = D_q^2, D_{sq} = D_s D_q$$

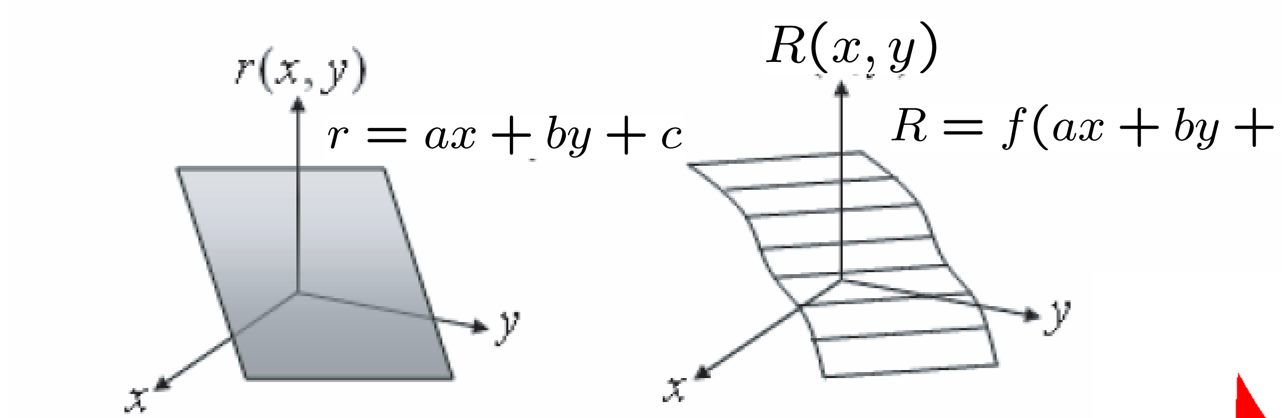
Two Issues

- How do we know the locations where the image irradiance is locally planar?
  - Note: We do not have access to the image irradiance.

- How to compute derivatives on an image?
  - Note: Digital images are discrete in space and magnitude.

Locally Planar Point Selection

- Two properties of the transformed locally planar points $R = f(ax + by + c)$
  - Locally linear isophote.
  - Symmetric on the sides of the gradient.



Isophote Curvature

$$\kappa = \frac{2R_{xy}R_xR_y - R_{xx}R_y^2 - R_{yy}R_x^2}{(R_x^2 + R_y^2)^{\frac{3}{2}}}$$

Locally Planar Point Selection

- A natural constraint is obtained from the equality of the linear geometry invariants:

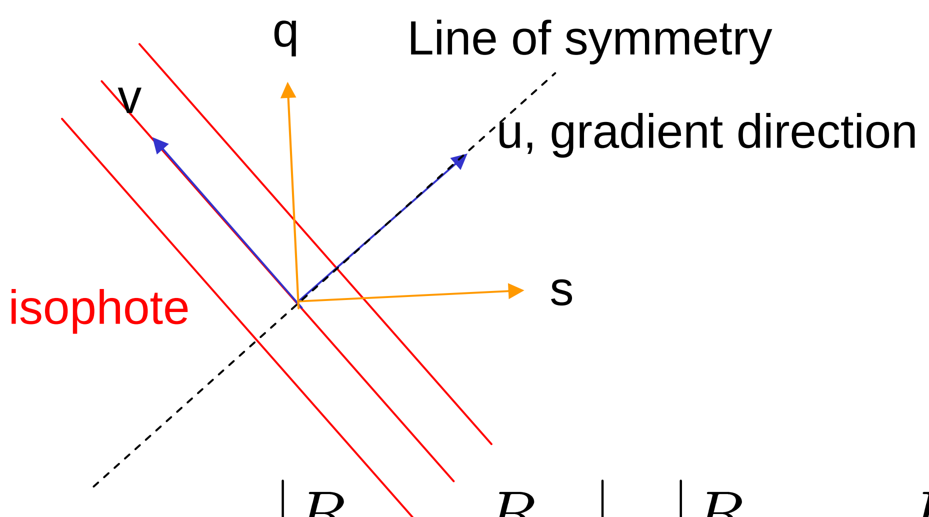
$$\frac{R_{xx}}{R_x^2} = \frac{R_{yy}}{R_y^2} = \frac{R_{xy}}{R_x R_y}$$

$$C(R) = \left|\frac{R_{xx}}{R_x^2} - \frac{R_{yy}}{R_y^2}\right| + \left|\frac{R_{xx}}{R_x^2} - \frac{R_{xy}}{R_x R_y}\right| + \left|\frac{R_{yy}}{R_y^2} - \frac{R_{xy}}{R_x R_y}\right|$$

- To see that C(R) enforces the symmetry over the gradient, let's consider C(R) in a gauge coordinate.
  - Note: the rotational symmetry property also holds for C(R).

A Gauge Coordinate

- Define axis u to be in the gradient direction.



$$s = \frac{1}{\sqrt{2}}(u - v)$$

$$q = \frac{1}{\sqrt{2}}(u + v)$$

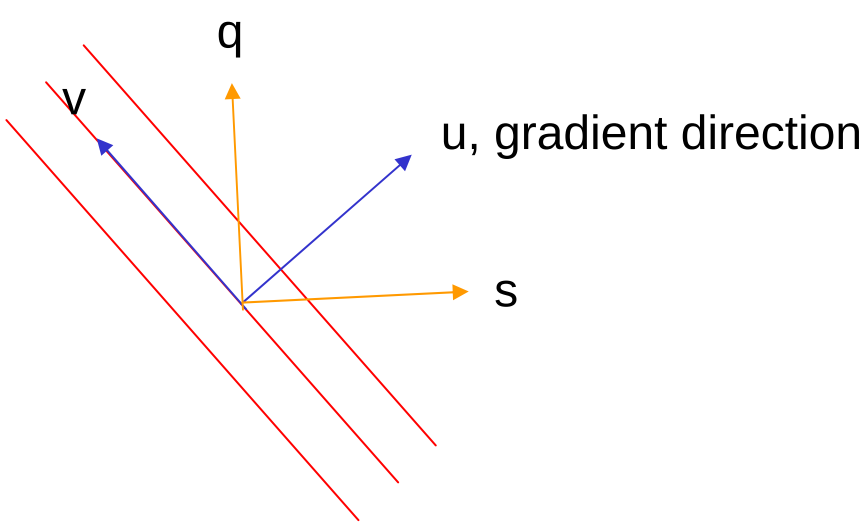
$$C(R) = \left|\frac{R_{ss}}{R_s^2} - \frac{R_{qq}}{R_q^2}\right| + \left|\frac{R_{ss}}{R_s^2} - \frac{R_{sq}}{R_s R_q}\right| + \left|\frac{R_{qq}}{R_q^2} - \frac{R_{sq}}{R_s R_q}\right|$$

When the derivatives on q and s direction are the same! $\Rightarrow$ $C(R) = 0$

Note: C(R) only enforces the symmetry on gradient up to 2nd order derivatives.

A Gauge Coordinate

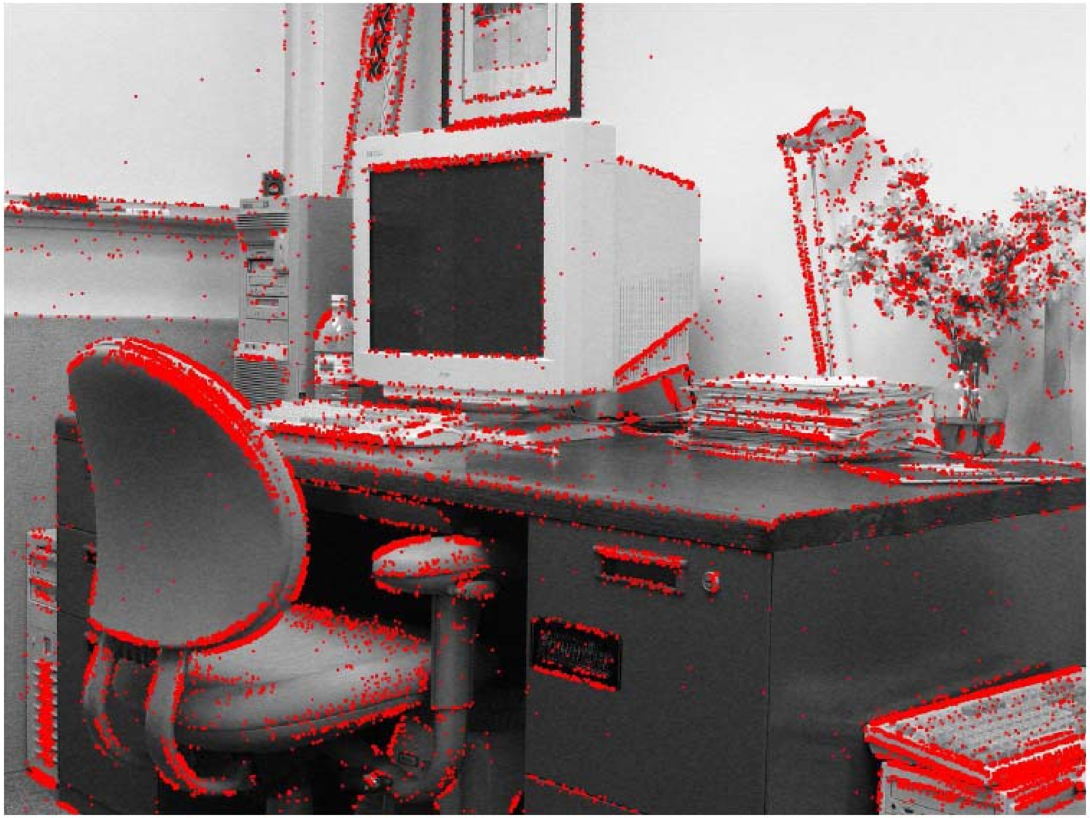
- The gauge coordinate is also a coordinate frame for reliable estimation of C(R) and the geometry invariants.
  - Due to the singularity for geometry invariants in the v (isophote) direction.



$$\frac{R_{vv}}{R_v^2} = \frac{0}{0}$$

Examples of the Selected Points

$$C(R) = \left|\frac{R_{ss}}{R_s^2} - \frac{R_{qq}}{R_q^2}\right| + \left|\frac{R_{ss}}{R_s^2} - \frac{R_{sq}}{R_s R_q}\right| + \left|\frac{R_{qq}}{R_q^2} - \frac{R_{sq}}{R_s R_q}\right| < \epsilon$$



Computing Derivatives on an Image

- Prior work in derivative kernel
  - [Canny 83] Optimized for localization of image edge.
  - [Haralick 84, Vieville et al. 92, Meer et al. 92] Unbiased or polynomial preserving derivative kernel, meant for computing differential quantities.
  - [Koenderink 84] Gaussian scale-space derivative kernel.
  - [Simoncelli et al. 94] Steerable directive kernel.
  - [Farid et al. 04] Separable kernel optimized for directional derivative.

Computing Derivatives on an Image

- To be robust to image noise, an image is often convolved with a Gaussian kernel before computing derivatives. What's wrong with that?
  - Gaussian convolution systematically (biased) alters the shape of an image.

$$g(x,\sigma) * x^2 = x^2 + \sigma^2$$

$$g(x,\sigma) * x^3 = x^3 + 3\sigma^2 x$$

- [Vieville et al. 92, Meer et al. 92] With the property of preserving polynomial and Gaussian noise, derivative can be computed by fitting a polynomial function.

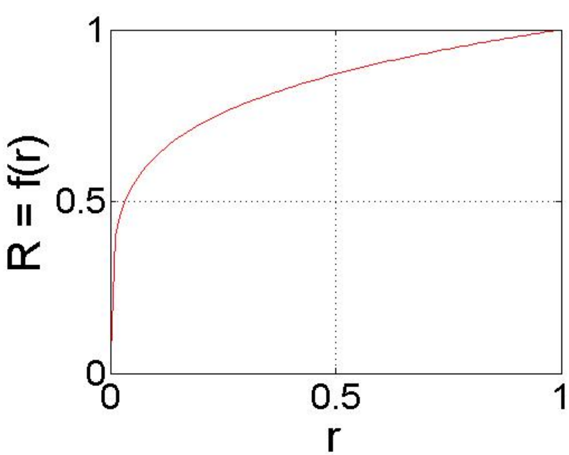
$$R(x,y) = R_o + R_x x + R_y y + \frac{R_{xx}}{2}x^2 + \frac{R_{yy}}{2}y^2 + R_{xy}xy + \dots$$

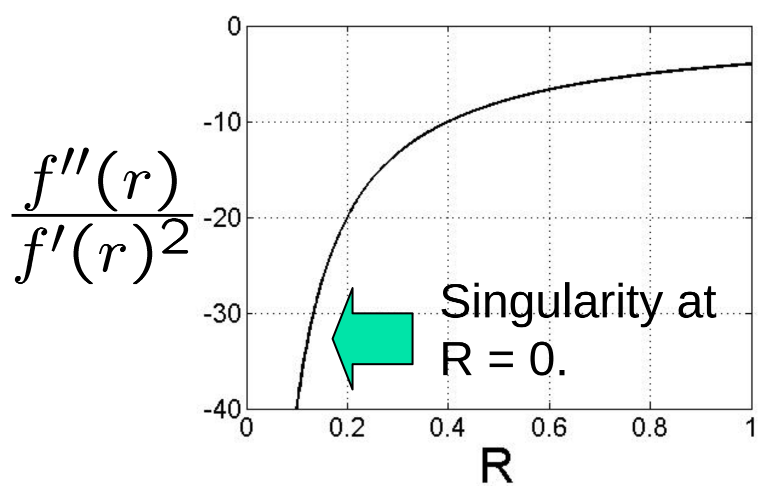
The Space for Curve Fitting

- After computing the linear invariant, we estimate the CRF by curve fitting.
- The space for the linear geometry invariants is difficult to handle – due to a singularity.

$$f(r) = r^{\gamma}$$
$$\gamma = 0.2$$

$$\frac{f''(r)}{f'(r)^2} = \frac{\gamma - 1}{\gamma} R^{-1}$$





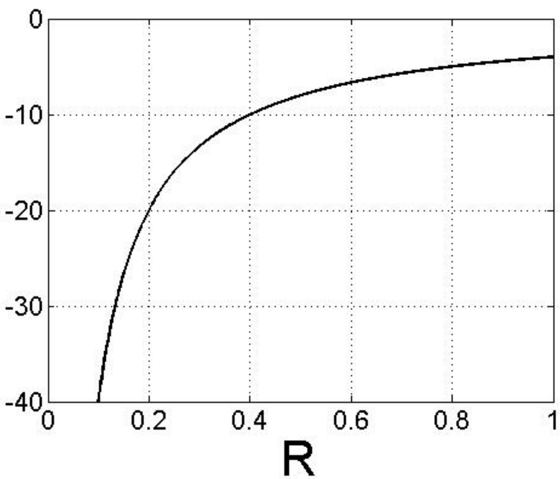
Q space

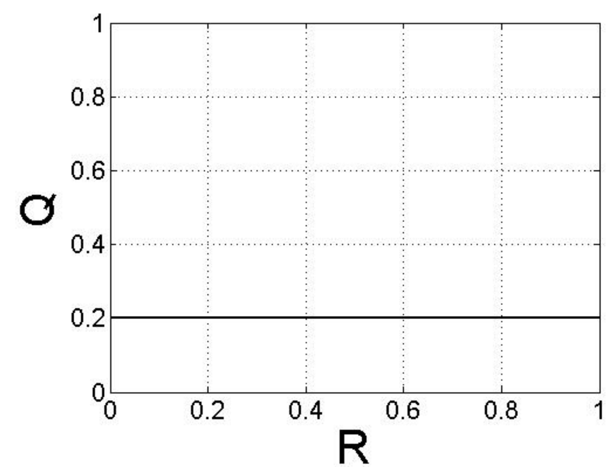
- We take advantage of the relationship between a gamma curve and geometry invariants, we define Q(R).

$$R = f(r) = r^{\gamma}$$
$$r = f^{-1}(R)$$
$$\gamma = 0.2$$

$$Q(R) = \frac{1}{1 - \frac{f''(r)}{f'(r)^2} f(r)} = \gamma$$

$$\frac{f''(r)}{f'(r)^2}$$





A New Parametric CRF Model

- Analytic CRF Model
  - [Mann et al. 95] Gamma curve
  - [Mitsunaga et al. 99] Polynomial model
- Empirical Model
  - [Grossberg et al. 03] PCA model

$$f(r) = r^{\sum_{i=0}^{n} c_i r^i}$$

A generalization of the gamma curve

	Number of model parameters			
Model	1	2	3	4
polynomial exponent	5.18	2.34	1.16	0.60
EMOR	4.00	1.73	0.63	0.25
polynomial	7.37	3.29	1.71	1.06

Fitting Parametric Curve in Q-R Space

- Our experiments are performed only on the linear exponent CRF model.
- We can extend the higher-order CRF model.
  - Problem: the parametric form is more complex.
- By fitting the Q(R) in the Q-R space, we can estimate the parameters of the CRF curve.

$$f(r) = r^{c_0 + c_1 r}$$

$$Q(R) = \frac{1}{1 - \frac{f''(r)}{f'(r)^2} f(r)} = \frac{(c_1 r \ln(r) + c_1 r + c_0)^2}{c_0 - c_1 r}$$

Algorithm

- 1) Select locally linear points.
- 2) Compute Q for the locally linear points.
- 3) Fit the parametric curve in Q-R space, with a sequence of two-step iterations.
  - Given (c0, c1), find r.
  - Given r, find the optimal (c0, c1).

$$(c_0^*, c_1^*) = \arg \min_{(c_0, c_1)} \sum_{j=0}^{J} \sum_{k=0}^{K} W(j,k) \left| Q_j - \frac{(c_1 r_k \ln(r_k) + c_1 r_k + c_0)^2}{c_0 - c_1 r_k} \right|^2$$

where: W = weight – given by the 2D Q-R histogram

Experiments

- 3 models of camera
  - Canon Powershot G3, Canon Rebel XT, Nikon D70
- 2 Sets of images acquired
  - Natural or textured objects with irregular edges: Trees, plants, ...
  - Man-made objects with straight edges: books, computers, ...
- Ground-truth CRF curves
  - The average of the color checker results and the curve obtained from multiple exposure method [Debevec et al. 97, Mitsunaga et al. 99].
  - Verified with RAW image data.
- Evaluation Metric: Root Mean Square Error (RMSE)

$$\sqrt{\frac{1}{N}\sum_{i=1}^{N}\left(f(r_i)-f_O(r_i)\right)^2}$$

Images

- Textured or natural objects with irregular edges



- Man-made objects with straight edges

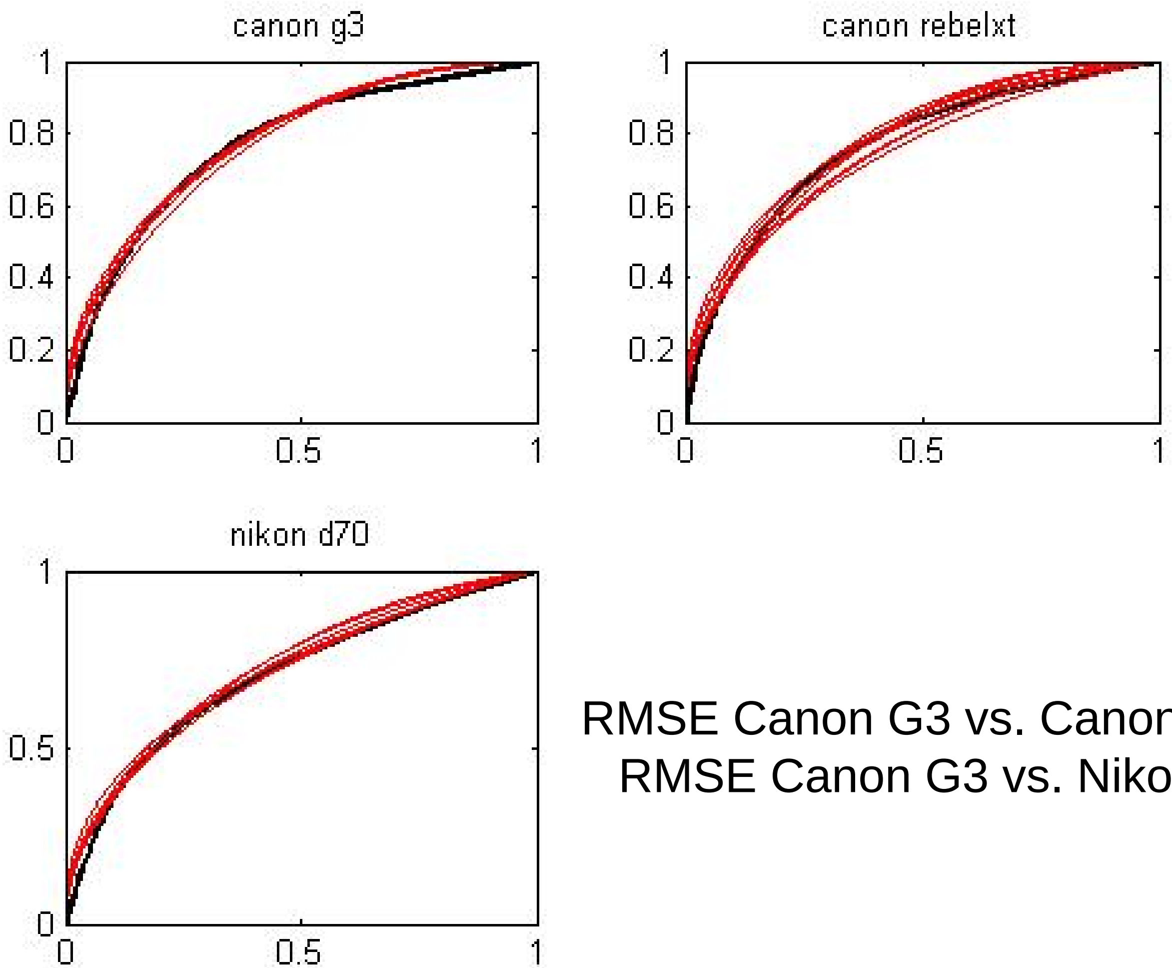


Experiments and Observations

- Task
  - Given a RGB color channel image, we estimate a curve for each of the single-color channel images.
- The algorithm performs well on the man-made object image set with straight edges.
  - Average RMSE = 0.028
- The estimation is unstable for natural object images with irregular edges.
  - Average RMSE = 0.060

Reference → RMSE Canon G3 vs. Canon rebel XT = 0.0146
RMSE Canon G3 vs. Nikon D70s = 0.0697

Estimated Curve for R-channel



RMSE Canon G3 vs. Canon rebel XT = 0.0146
RMSE Canon G3 vs. Nikon D70s = 0.0697

CRF from Motion Blur

- An interesting observation:
  - the algorithm seems to perform well on motion blurred images, even though it is an image of a plant!

Open Issues

- Can we improve the algorithm so that it can work well on all/most types of image?
- Can we extend the algorithm to the set of the higher-order geometry invariants?
  - Accurate estimation of the higher-order derivative is difficult.
- Can we apply machine learning to improve the stability and robustness of the algorithm?
- Given the limitation, how the algorithm is useful for passive-blind image forensics?

Thank you!

Dataset and Project Website: http://www.ee.columbia.edu/trustfoto