Recall: Image Processing

© Yang Cao

© Chang Wen Chen

University of Science and Technology of China

Outline

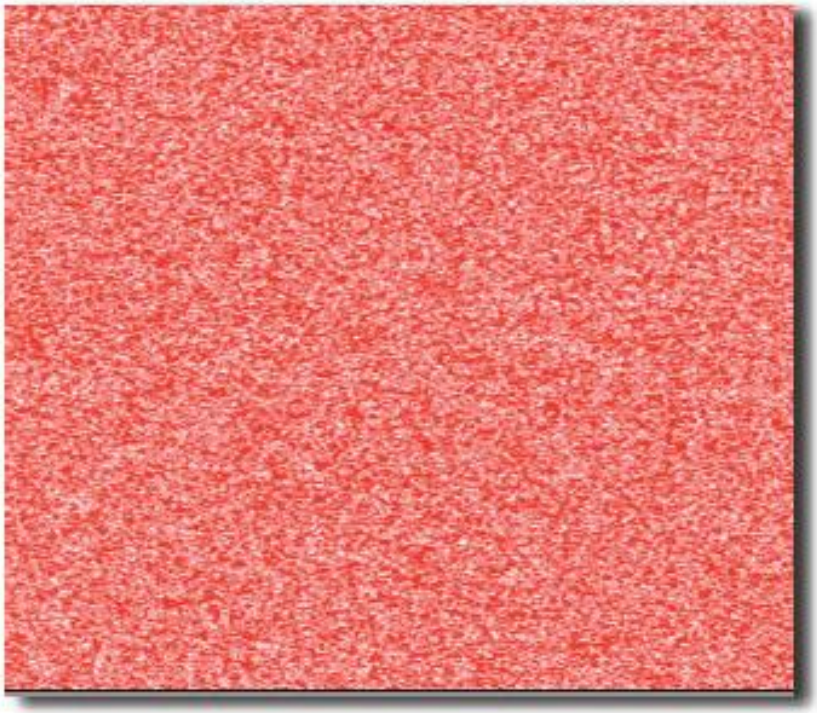
- What is an image?
- Where is an image from?
- How to process an image?

What is an image?

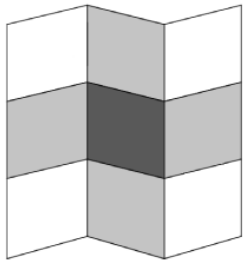
Images – Pretty Ones

Images – Not As Pretty Ones

Images – Not a set of random pixels



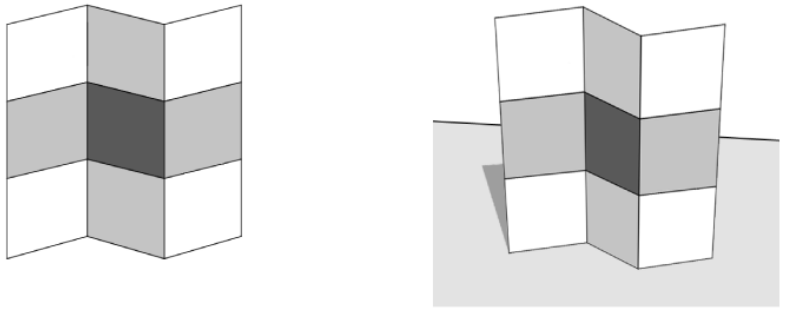
The Workshop Metaphor



(a) an image

E. Adelson and A. Pentland, "The perception of shading and reflectance," *Perception as Bayesian inference*, 1996.

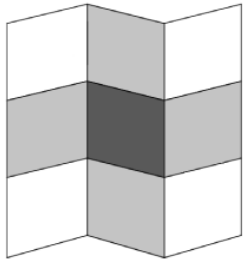
The Workshop Metaphor



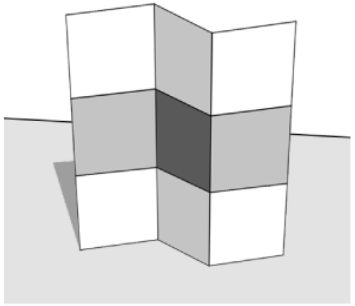
(a) an image (b) a likely explanation

E. Adelson and A. Pentland, "The perception of shading and reflectance," *Perception as Bayesian inference*, 1996.

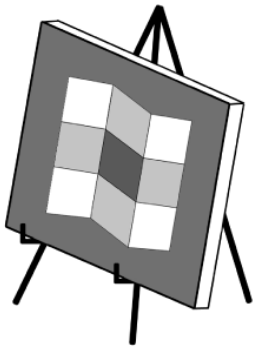
The Workshop Metaphor



(a) an image



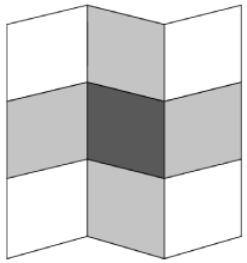
(b) a likely explanation



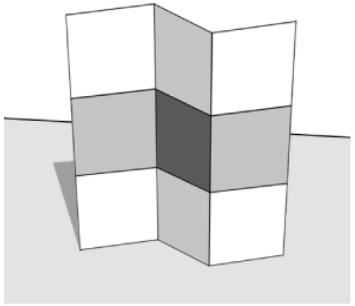
(c) painter's explanation

E. Adelson and A. Pentland, "The perception of shading and reflectance," *Perception as Bayesian inference*, 1996.

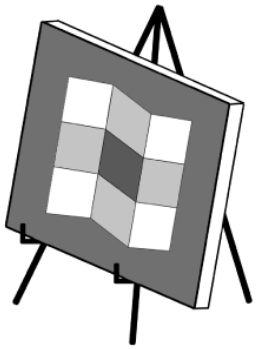
The Workshop Metaphor



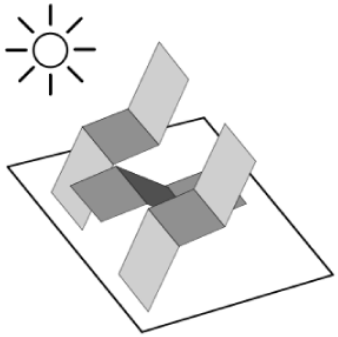
(a) an image



(b) a likely explanation



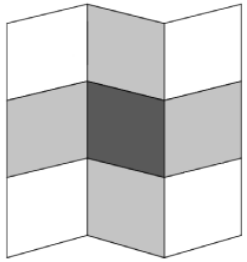
(c) painter's explanation



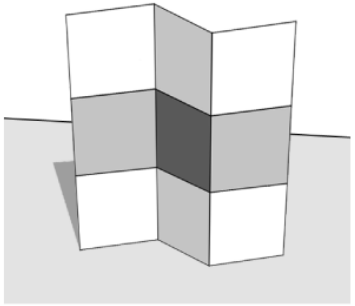
(d) sculptor's explanation

E. Adelson and A. Pentland, "The perception of shading and reflectance," *Perception as Bayesian inference*, 1996.

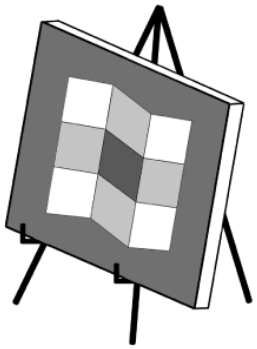
The Workshop Metaphor



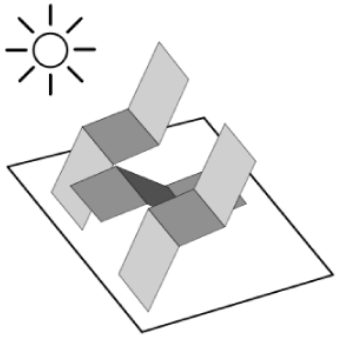
(a) an image



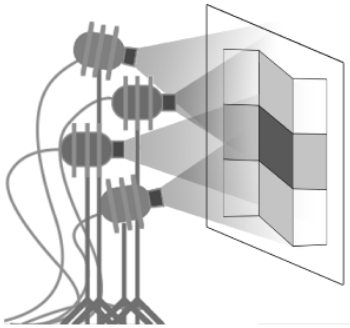
(b) a likely explanation



(c) painter's explanation



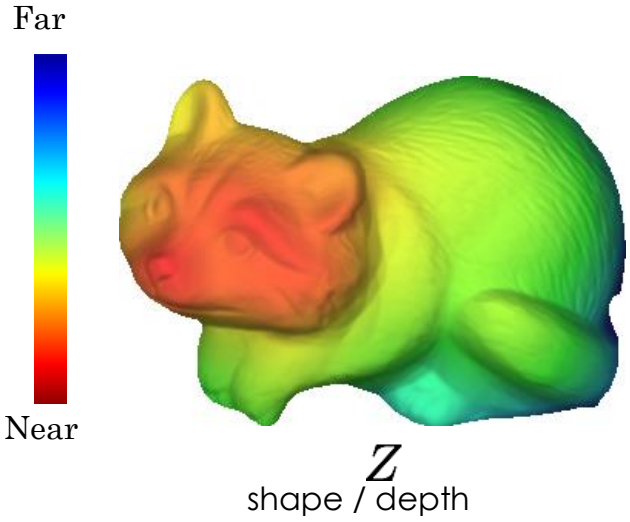
(d) sculptor's explanation



(e) gaffer's explanation

E. Adelson and A. Pentland, "The perception of shading and reflectance," *Perception as Bayesian inference*, 1996.

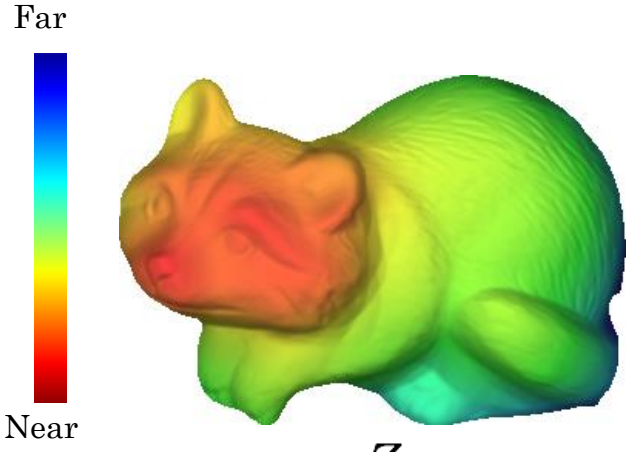
Forward Optics



Z

shape / depth

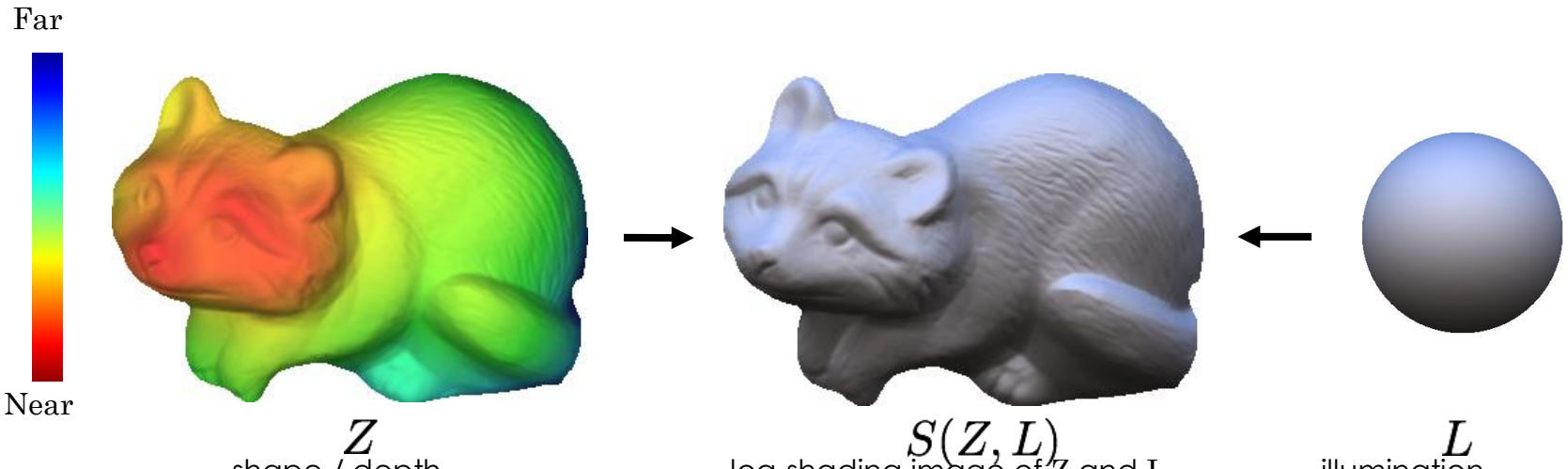
Forward Optics



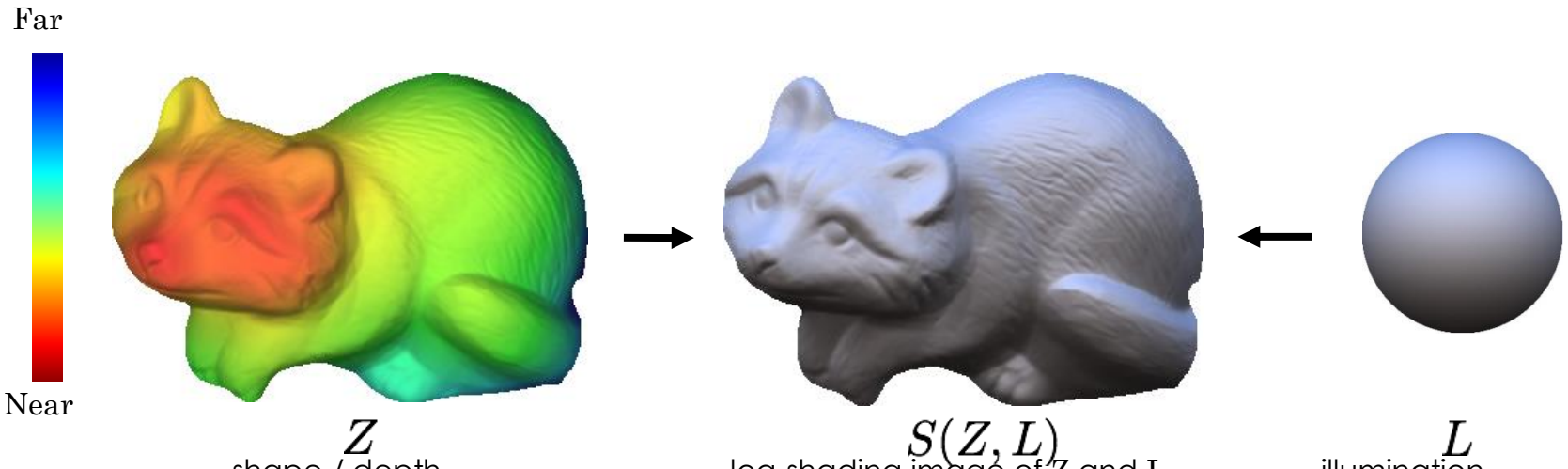
Z
shape / depth

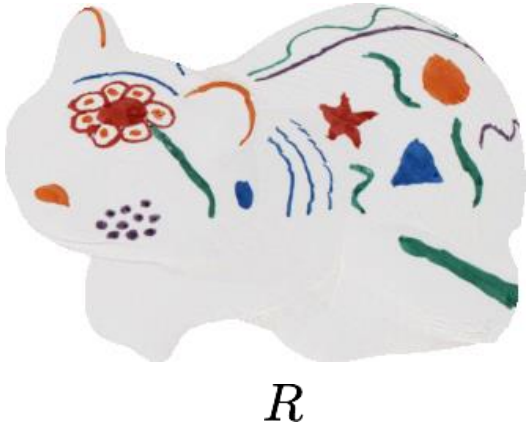
L
illumination

Forward Optics



Forward Optics

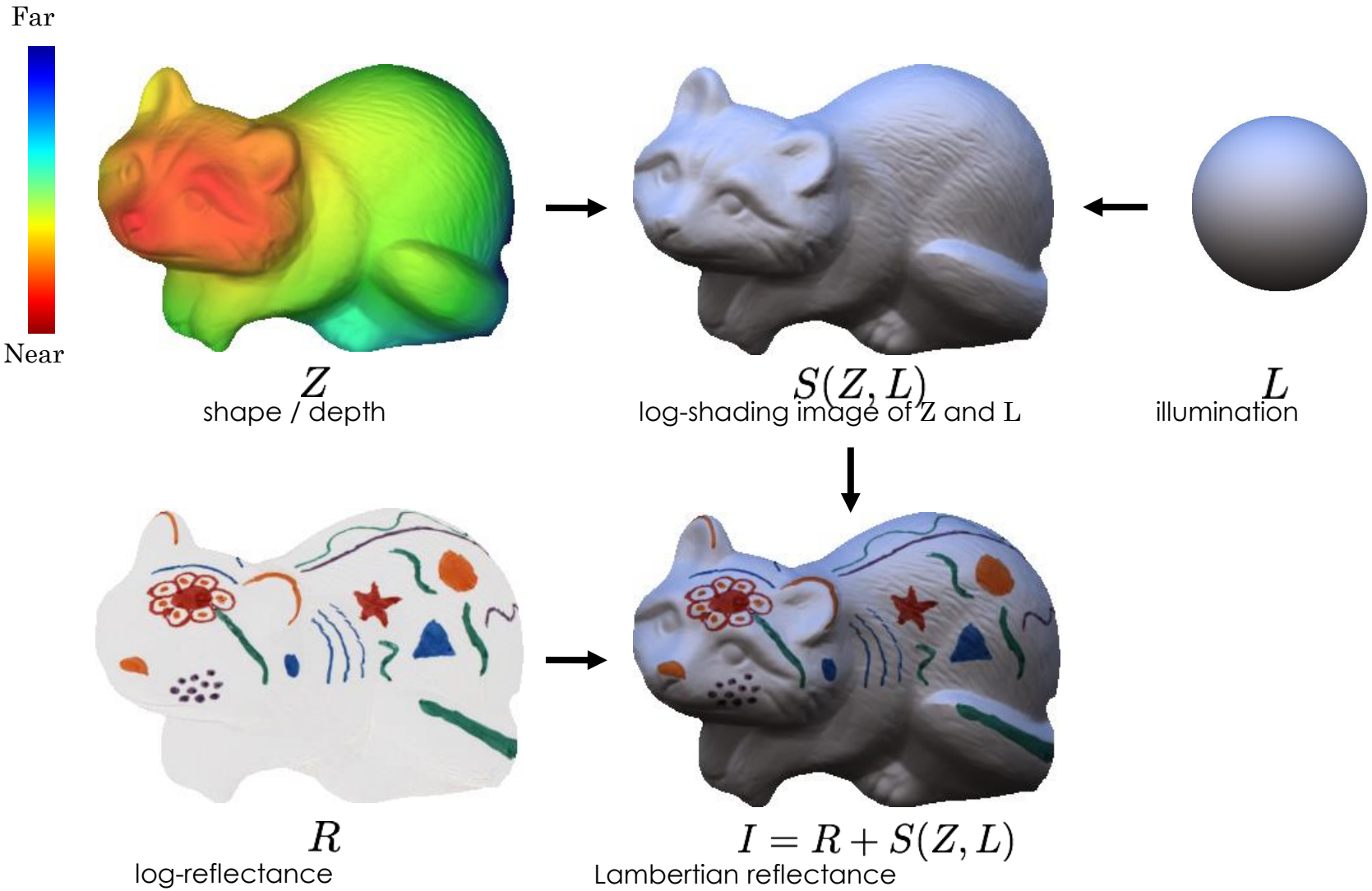




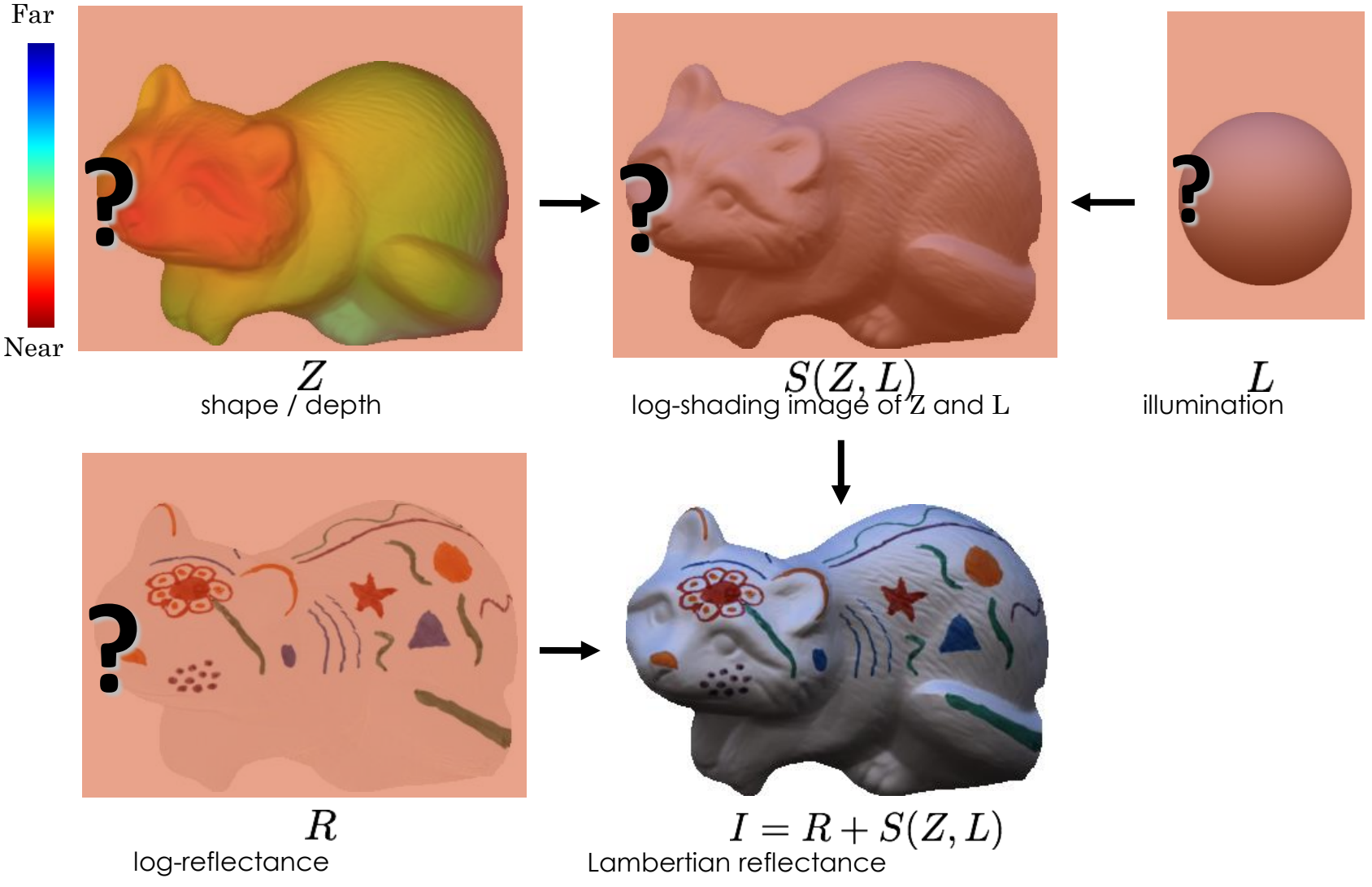
R

log-reflectance

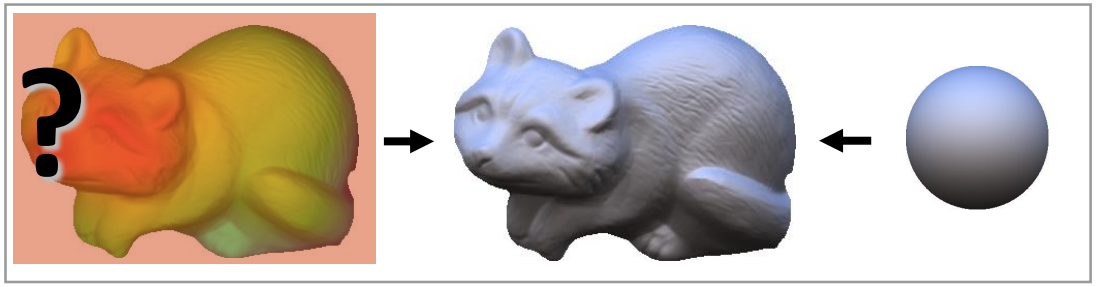
Forward Optics



Our problem


Far
Near
?
?
?
Z
shape / depth
$S(Z, L)$
log-shading image of Z and L
L
illumination
R
log-reflectance
$I = R + S(Z, L)$
Lambertian reflectance

Existing Work: Shape from Shading



Basic Assumption: illumination and albedo are known.

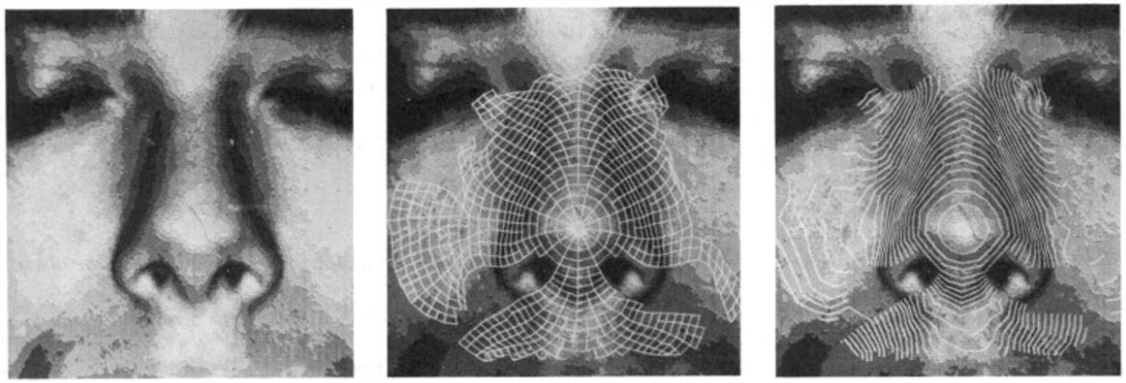
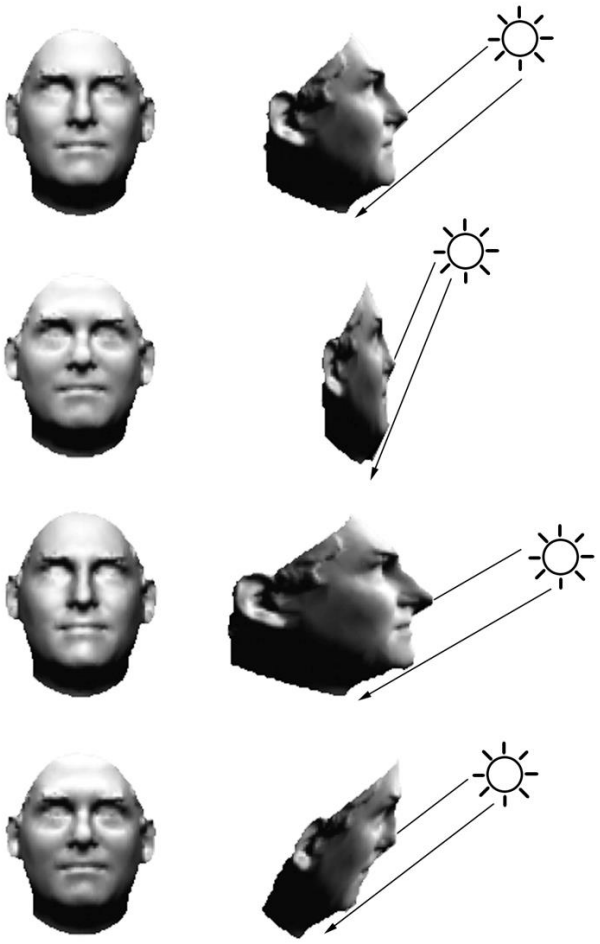


Figure 11-7. The shape-from-shading method is applied here to the recovery of the shape of a nose. The first picture shows the (crudely quantized) gray-level image available to the program. The second picture shows the base characteristics superimposed, while the third shows a contour map computed from the elevations found along the characteristic curves.

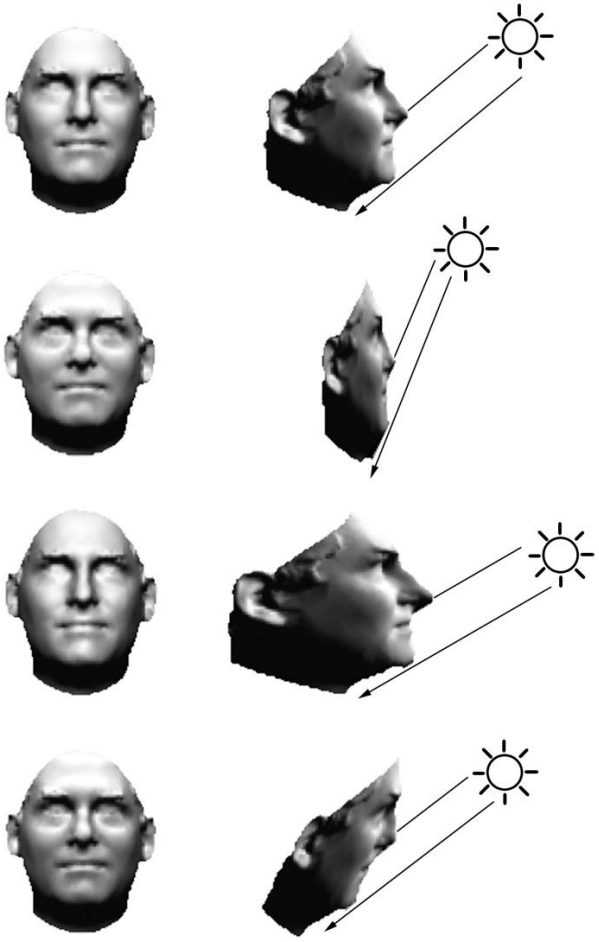
B. K. P. Horn. Shape from shading: A method for obtaining the shape of a smooth opaque object from one view. Technical report, MIT, 1970.

Existing Work: Shape from Shading

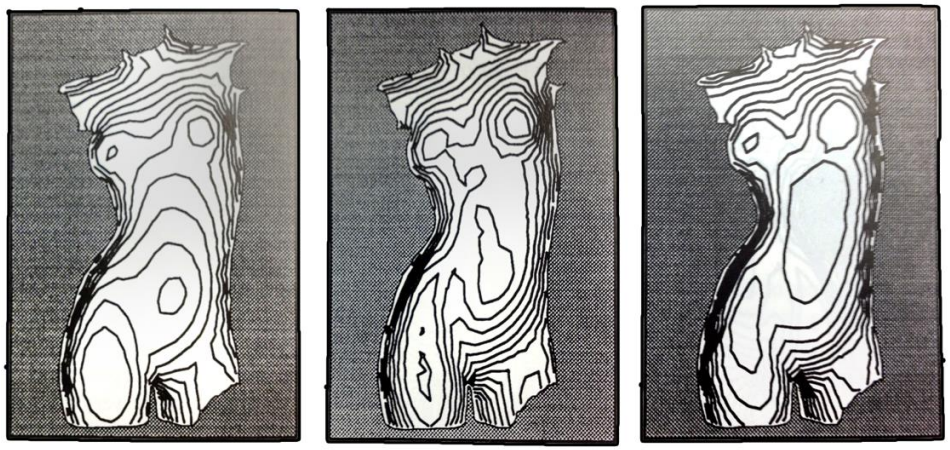


P. Belhumeur, D. Kriegman, and A. Yuille.
The Bas-Relief Ambiguity. *IJCV*, 1999.

Existing Work: Shape from Shading

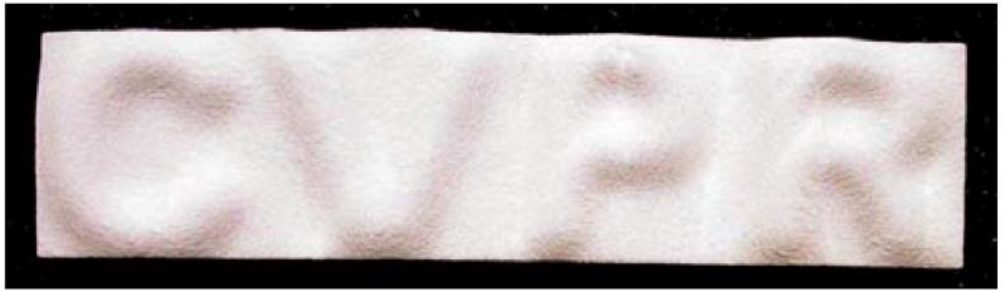


P. Belhumeur, D. Kriegman, and A. Yuille. The Bas-Relief Ambiguity. *IJCV*, 1999.



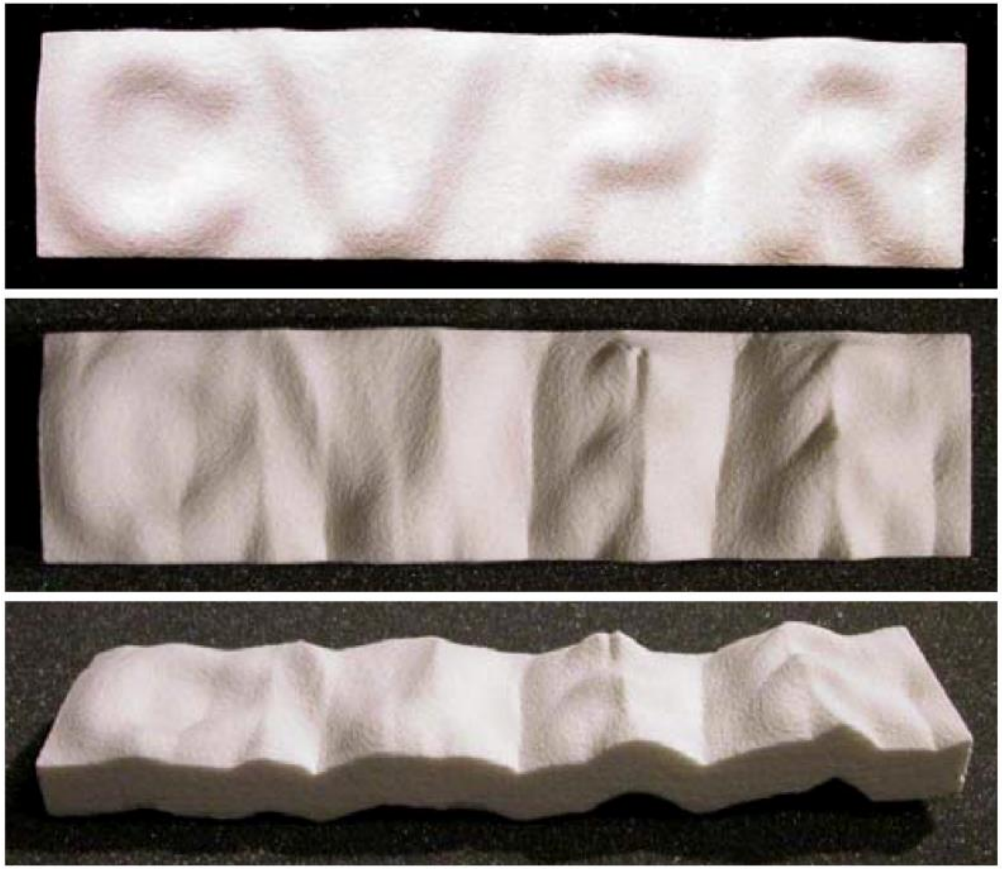
J. Koenderink, A. van Doorn, C. Christou, and J. Lappin. Shape constancy in pictorial relief. *Perception*, 1996.

Existing Work: Shape from Shading



Ecker & Jepson, Polynomial Shape from Shading, *CVPR* 2010

Existing Work: Shape from Shading

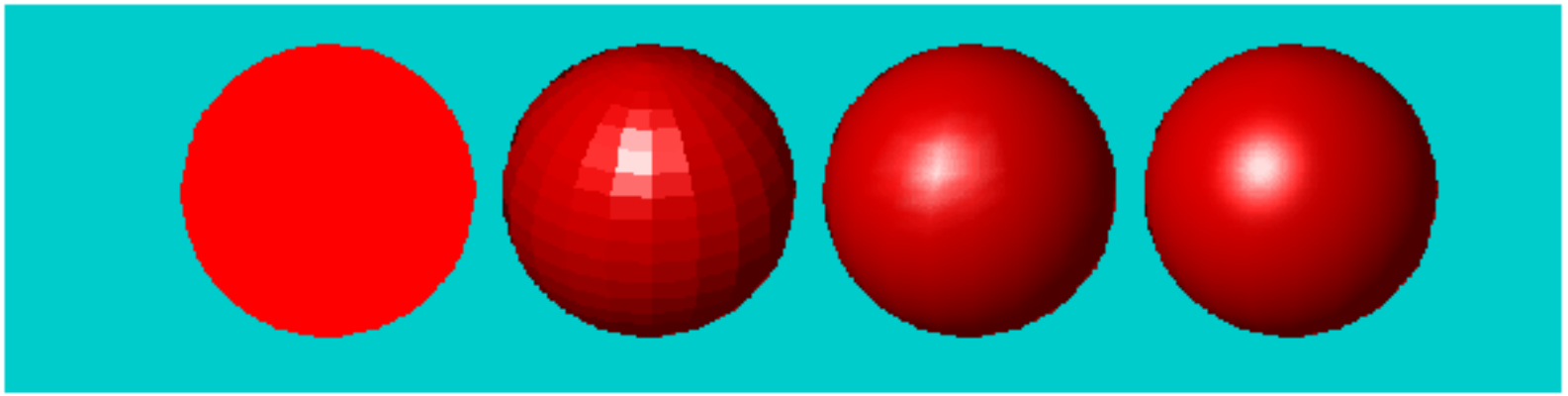


Ecker & Jepson, Polynomial Shape from Shading, *CVPR* 2010

Shape From Shading

What is Shading?

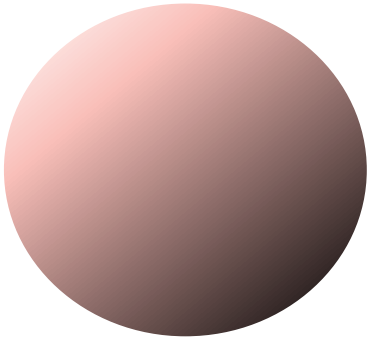
- **Variable levels of darkness**
- **A visual cue for the actual 3D shape**
- **A latent relation between intensity and shape**



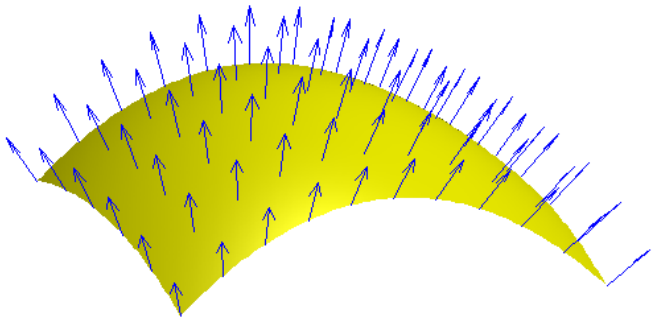
Shape From Shading

What is the problem of Shading From Shading?

- Given a grayscale image
  - And albedo
  - And light source direction



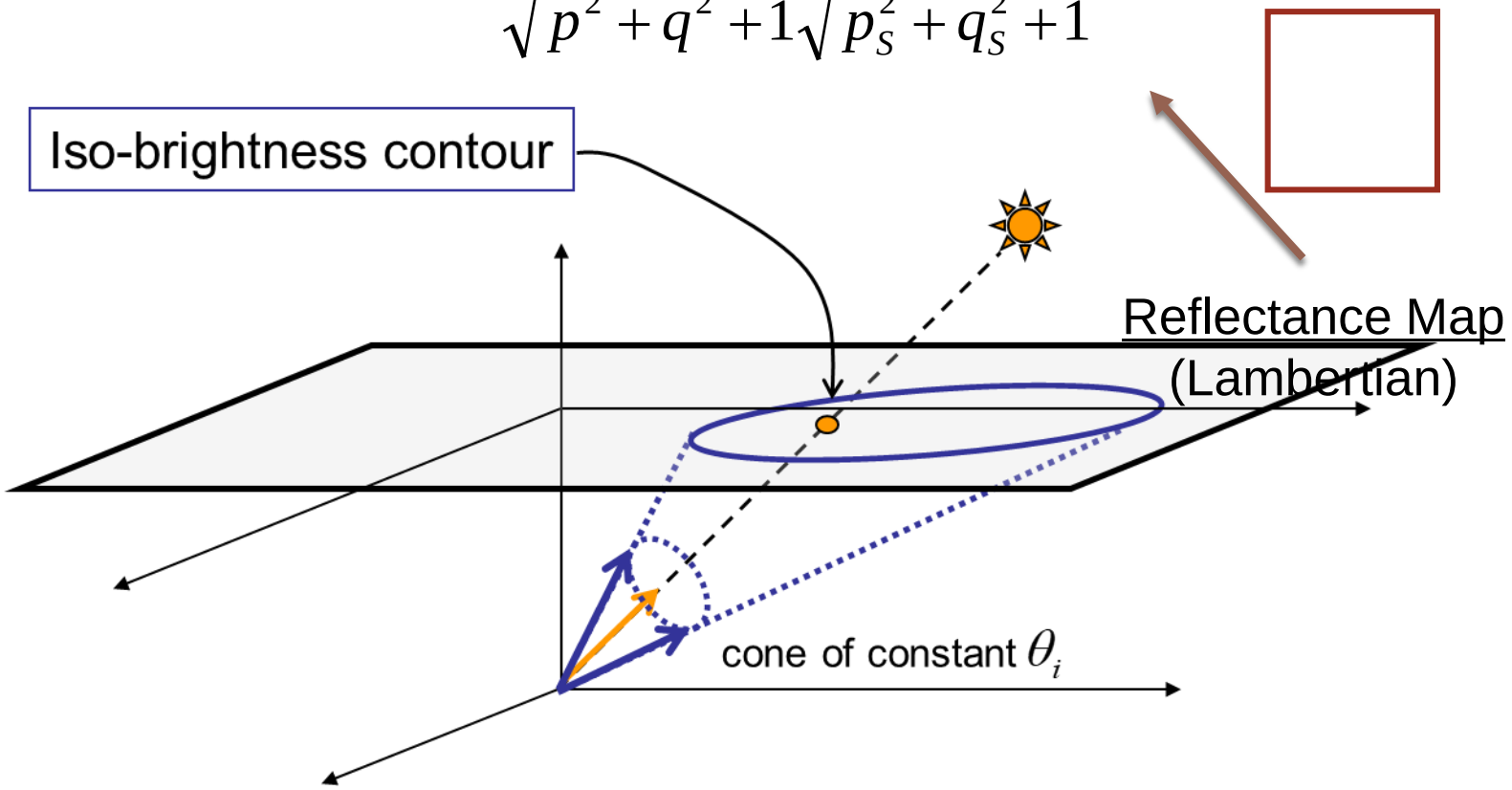
- Reconstruct scene geometry
  - Can be modeled by surface normal



Shape From Shading

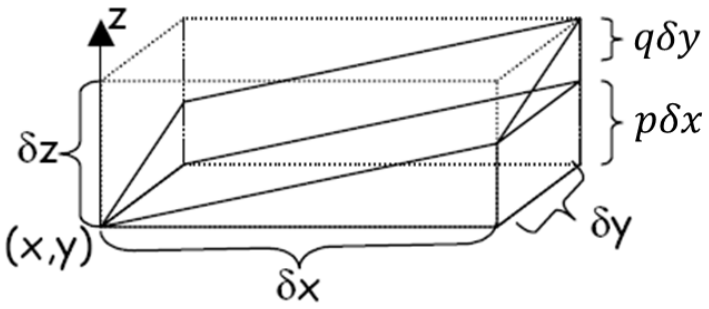
Reflectance Map

$$I = \cos\theta_i = \mathbf{n}\cdot\mathbf{s} = \frac{(pp_s + qq_s + 1)}{\sqrt{p^2+q^2+1}\sqrt{p_S^2+q_S^2+1}} = R(p,q)$$



Shape From Shading

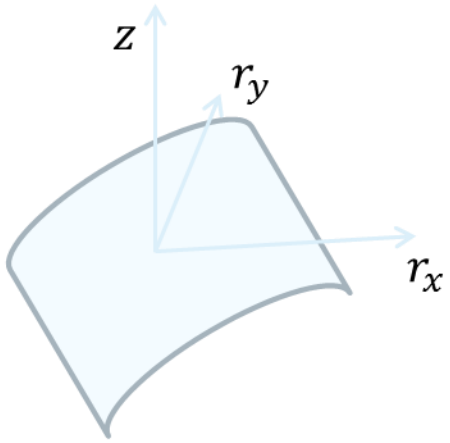
- **A smooth surface has a tangent plane at every point**
  - **Mark** $p = \frac{\partial z}{\partial x}$; $q = \frac{\partial z}{\partial y}$
- **Parametrize surface orientation by first partial derivatives of** z



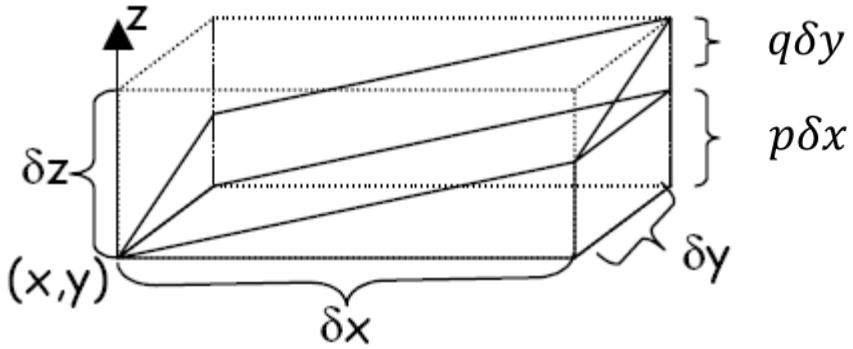
Shape From Shading

Surface normal

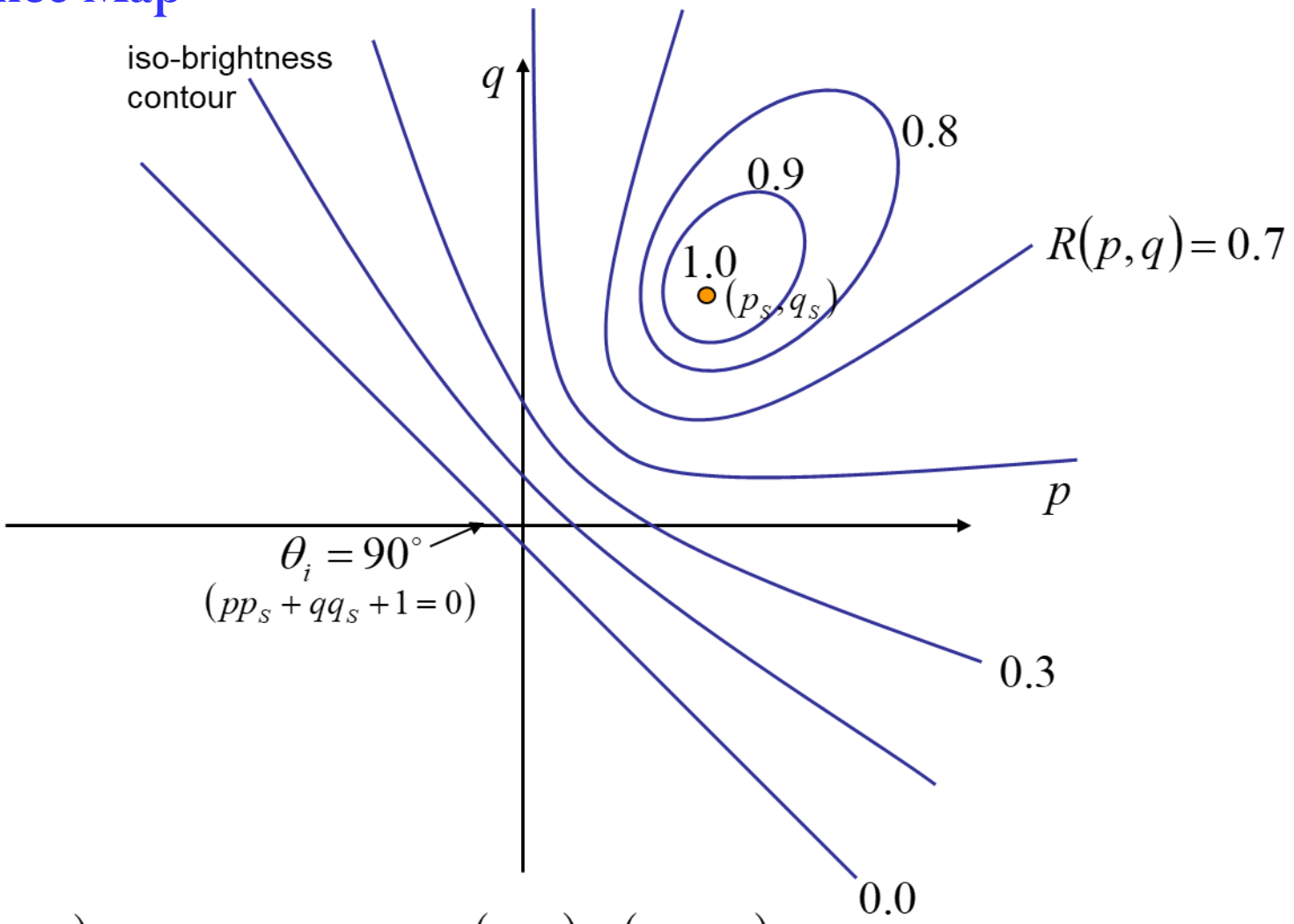
- $r_x = (p, 0, 1)$, $r_y = (0, q, 1)$
- $n = r_x \times r_y = (p, q, -1)$
- Normalize $\hat{n} = \frac{n}{\|n\|} = \frac{(p,q,-1)}{\sqrt{p^2+q^2+1}}$



$$p = \frac{\partial z}{\partial x} \; ; q = \frac{\partial z}{\partial y}$$



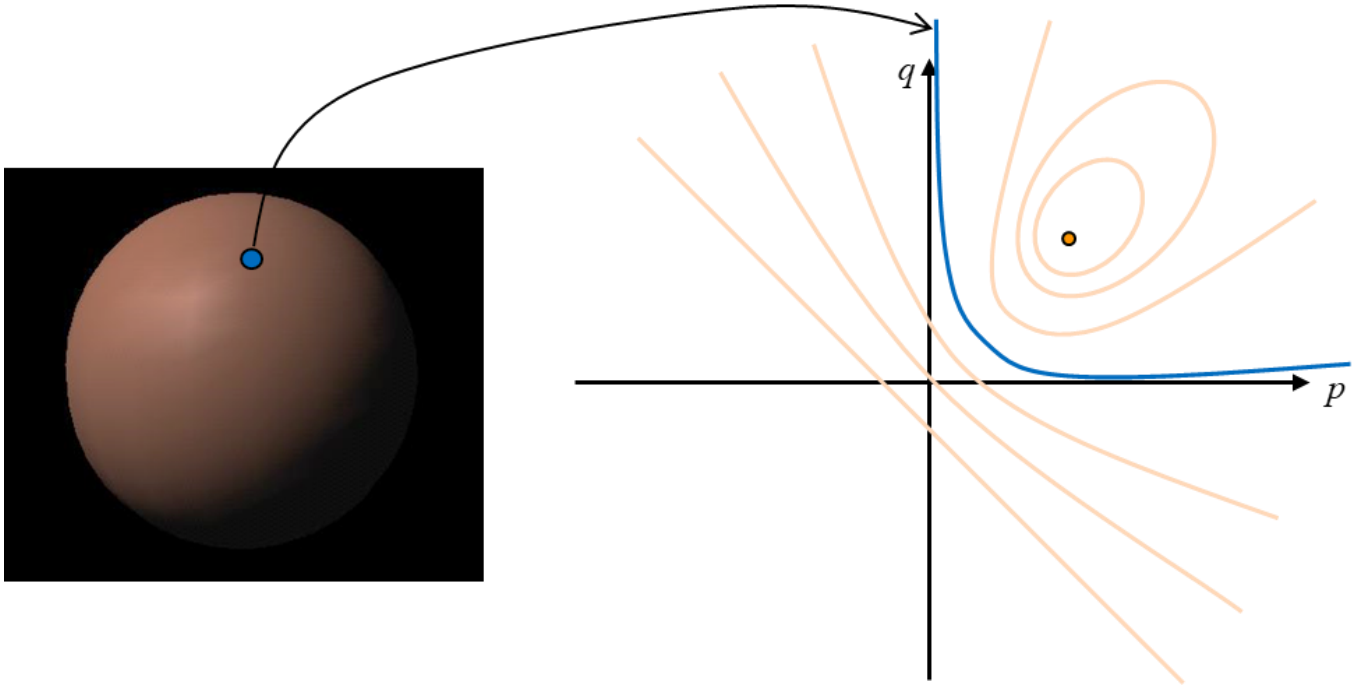
Reflectance Map



Note: $R(p,q)$ is maximum when $(p,q) = (p_s, q_s)$

Reflectance Map

Brightness is considered as a function of surface orientation

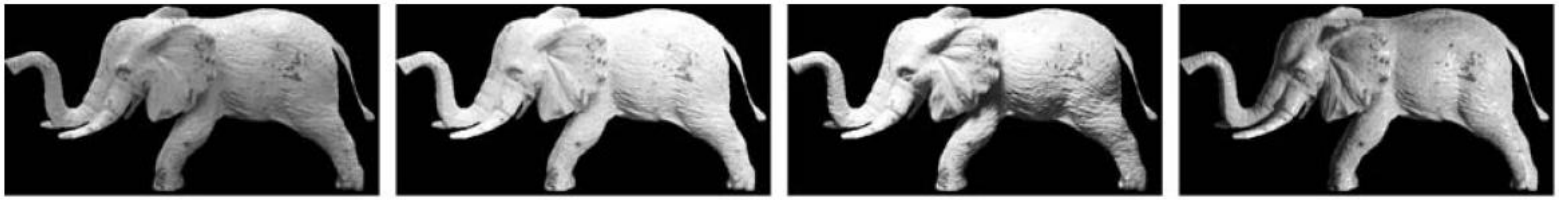


$$R(p,q) = \frac{1 + p_s p + q_s q}{\sqrt{1 + p^2 + q^2}\sqrt{1 + {p_s}^2 + {q_s}^2}}$$

Two variables p and q with one equation
How to do?

Shape From Shading

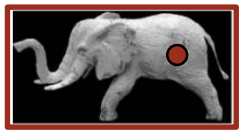
- Use more images
  - Photometric stereo

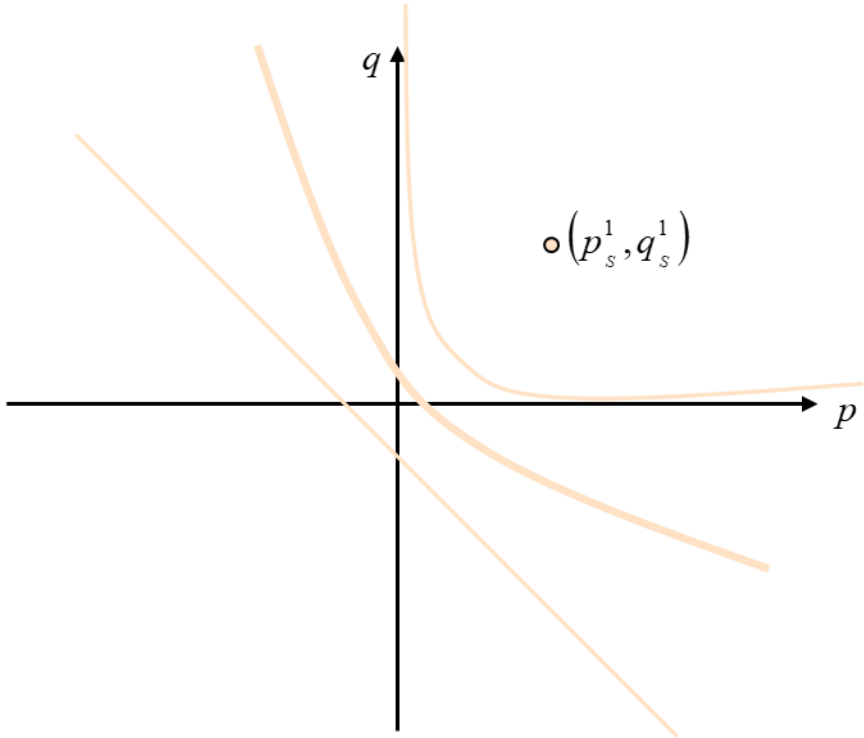


- Shape from shading
  - Introduce constraints
  - Solve locally
  - Linearize problem

Shape From Shading

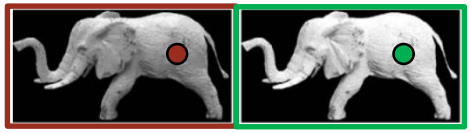
- Take several pictures of same object under same viewpoint with different lighting

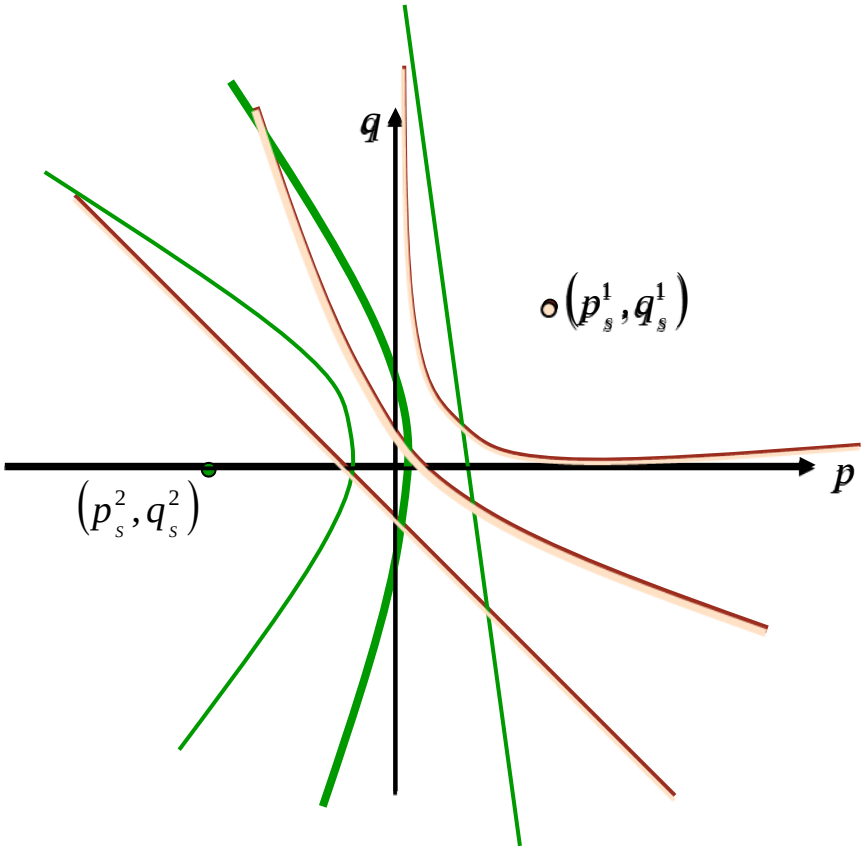




Shape From Shading

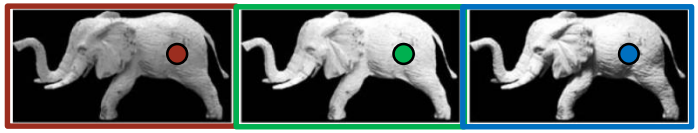
- Take several pictures of same object under same viewpoint with different lighting

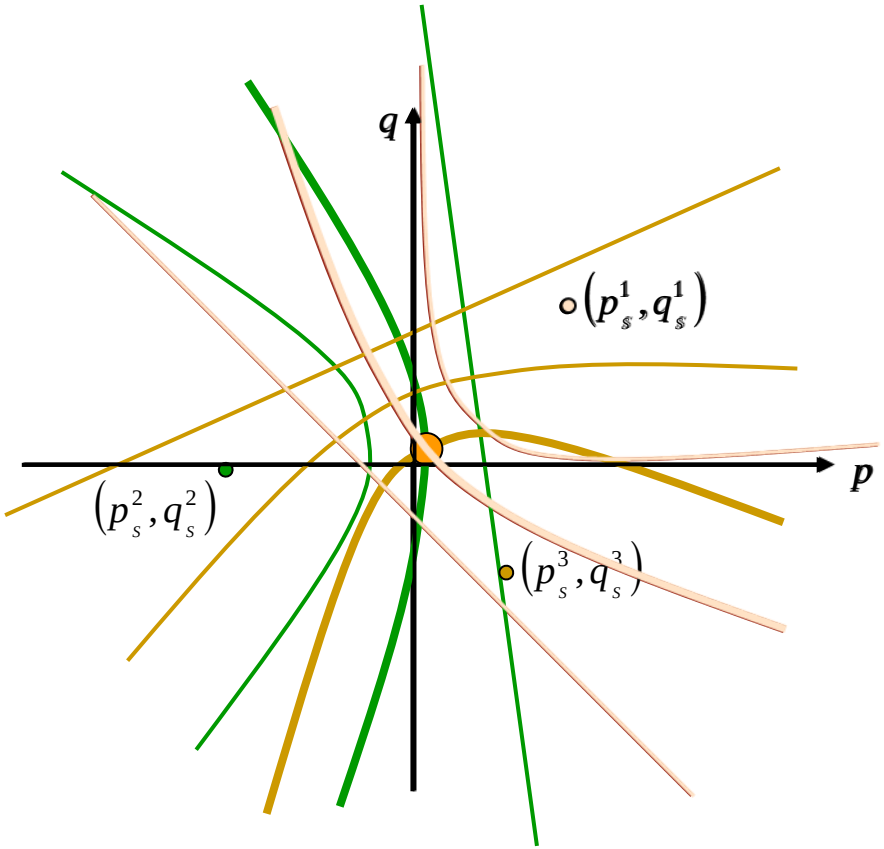




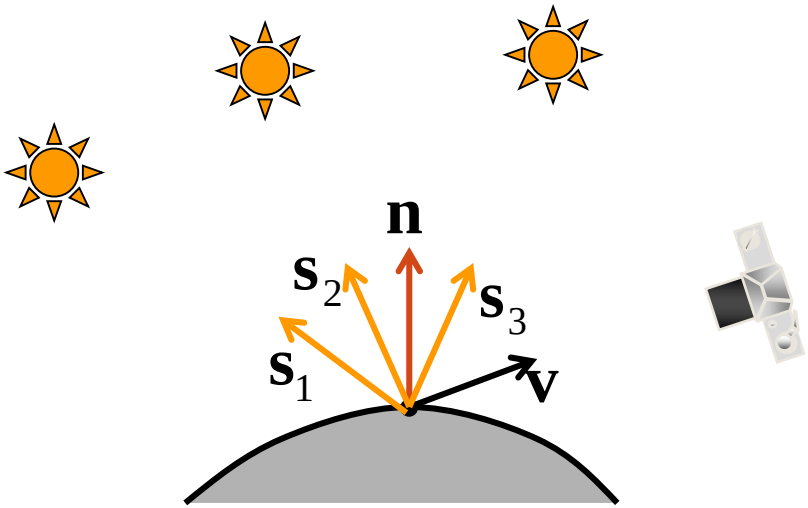
Shape From Shading

- Take several pictures of same object under same viewpoint with different lighting





Photometric Stereo



Lambertian case:

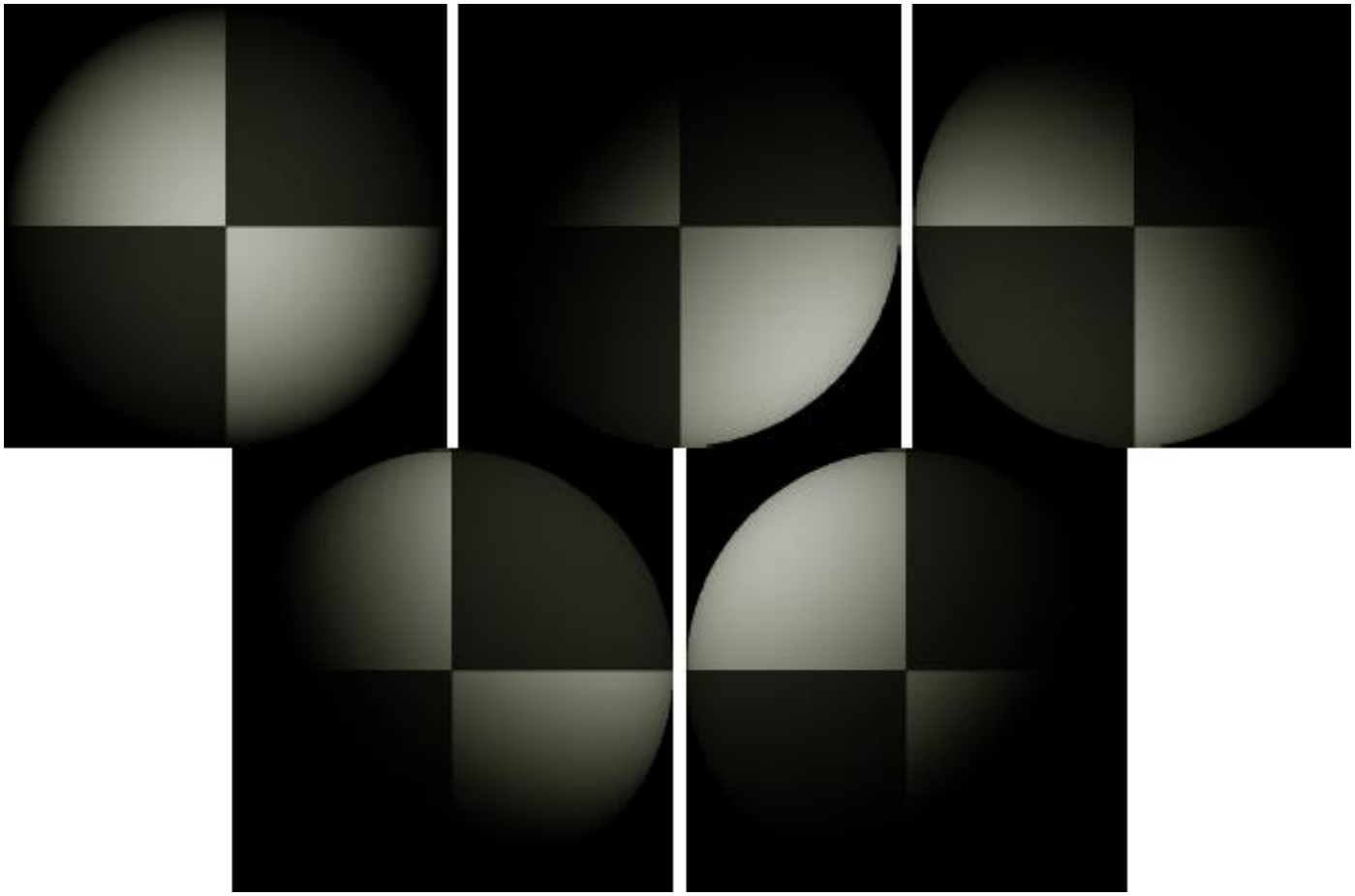
$$I = \frac{\rho}{\pi} kc \cos\theta_i = \rho \mathbf{n} \cdot \mathbf{s} \quad \left(\frac{kc}{\pi} = 1 \right)$$

Image irradiance:

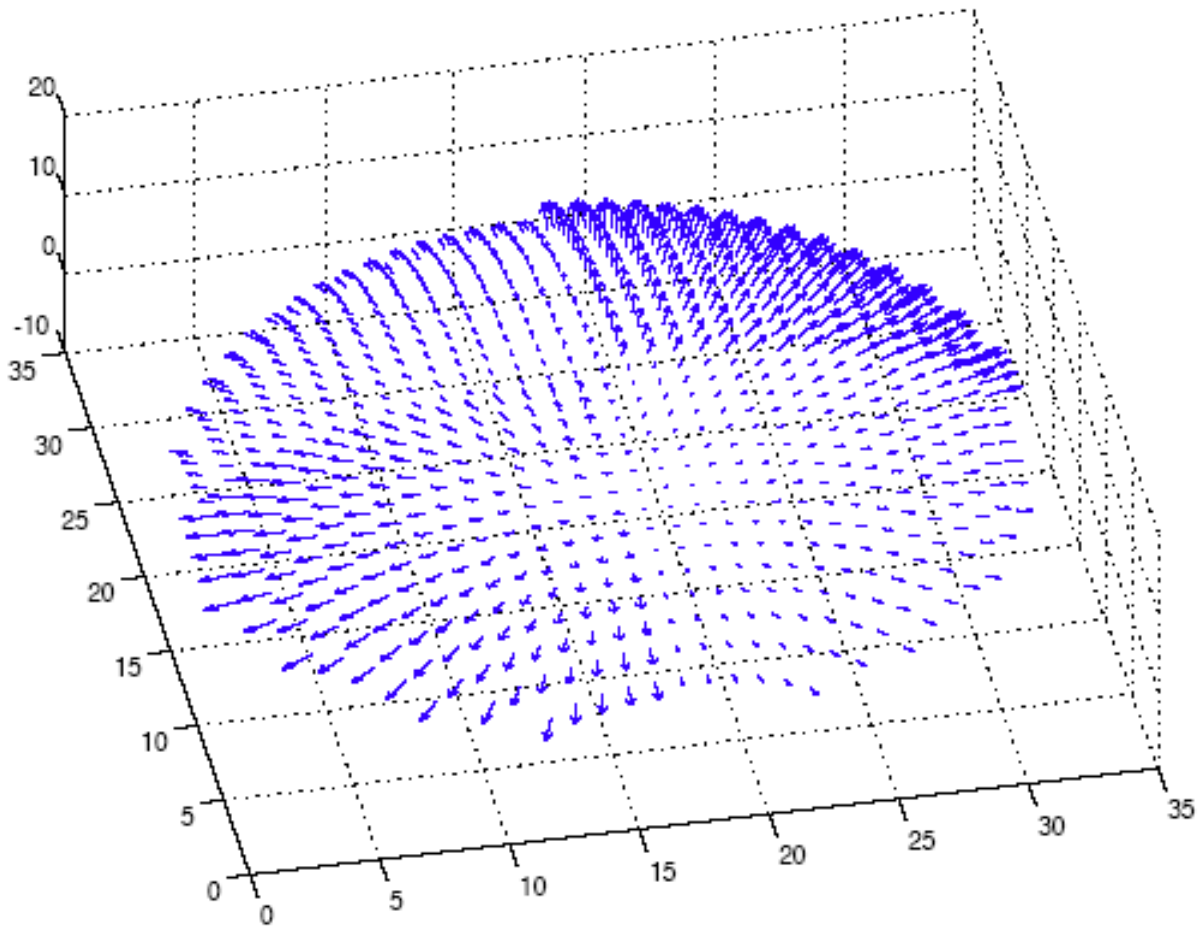
$$I_1 = \rho \mathbf{n} \cdot \mathbf{s}_1$$
$$I_2 = \rho \mathbf{n} \cdot \mathbf{s}_2$$
$$I_3 = \rho \mathbf{n} \cdot \mathbf{s}_3$$

- We can write this in matrix form:

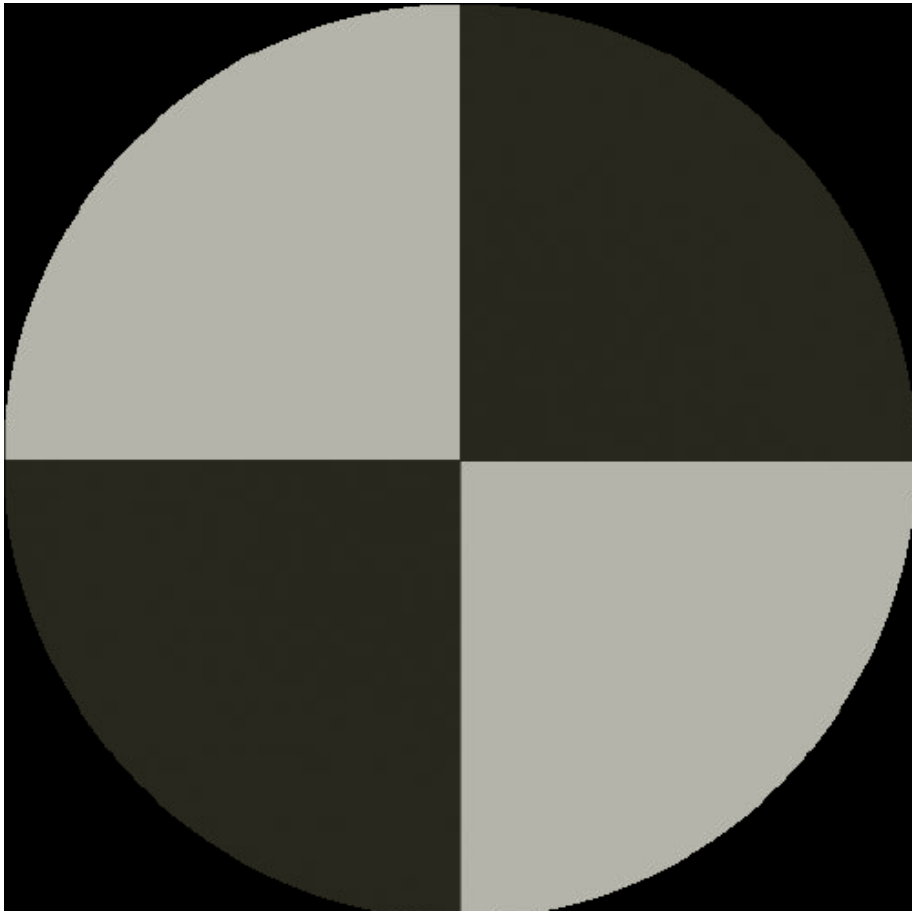
$$\begin{bmatrix} I_1 \\ I_2 \\ I_2 \end{bmatrix} = \rho \begin{bmatrix} \mathbf{s}_1^T \\ \mathbf{s}_2^T \\ \mathbf{s}_3^T \end{bmatrix} \mathbf{n}$$



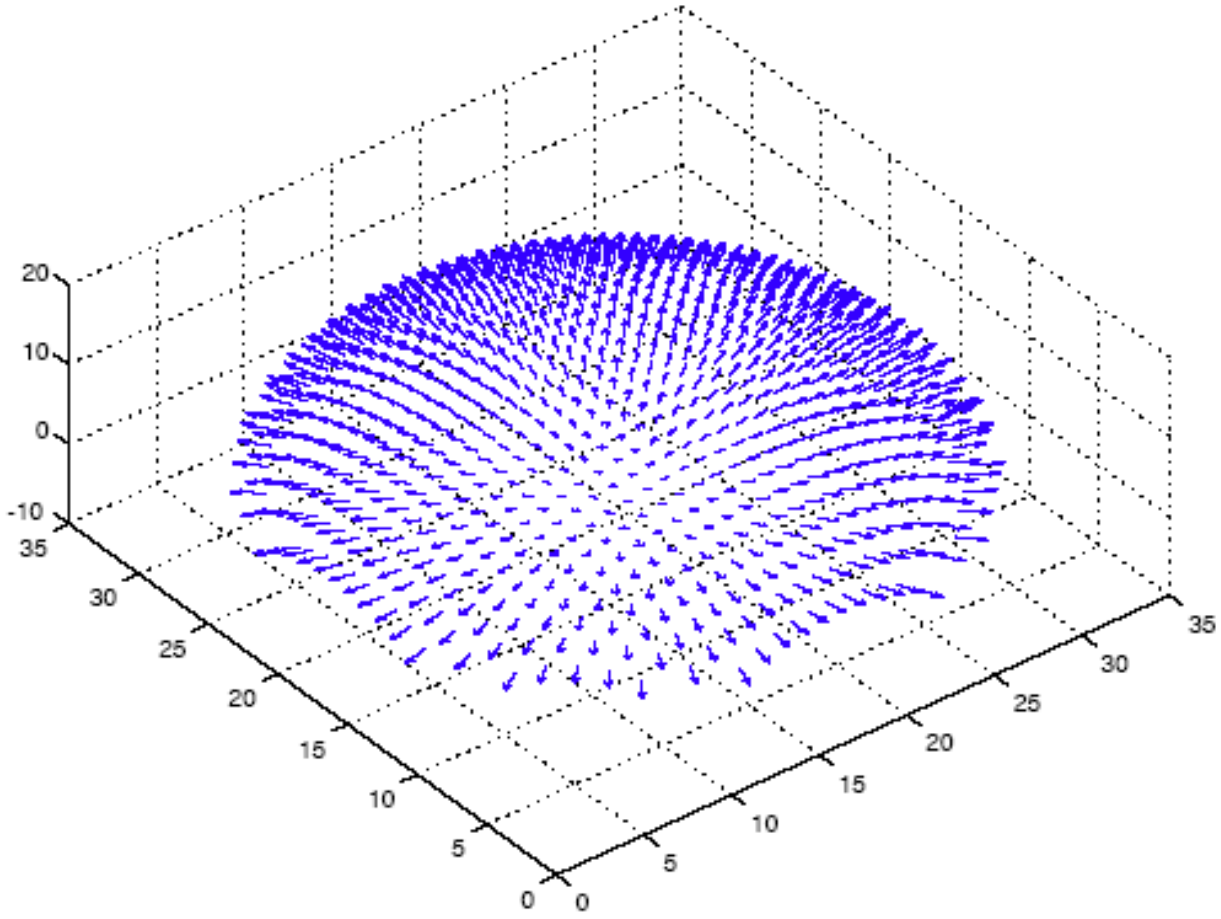
改变光源所获得的同一个球的五幅图像



$$\boldsymbol{g}(x, y)$$



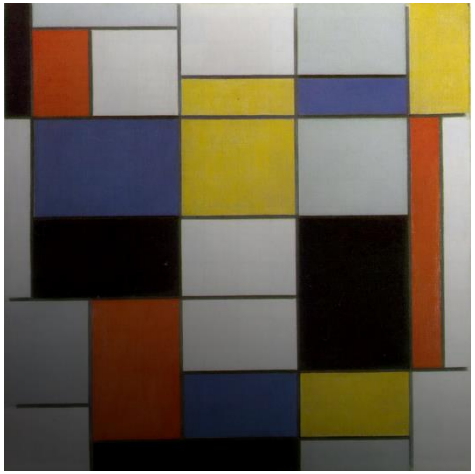
$$\rho(x, y)$$



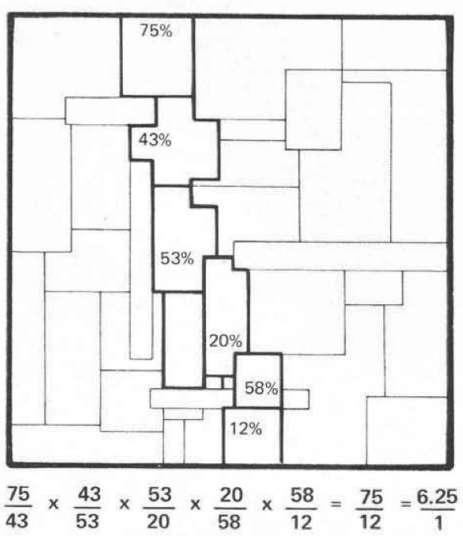
$$n(x, y)$$

Existing Work: Intrinsic Images

Basic assumption: Shape and illumination are ignored, shading varies slowly, reflectance varies quickly.

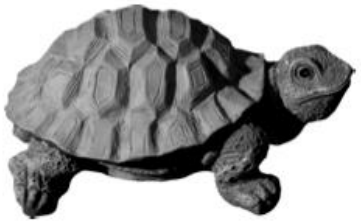


Piet Mondrian, Composition A. Oil on Canvas, 1920.



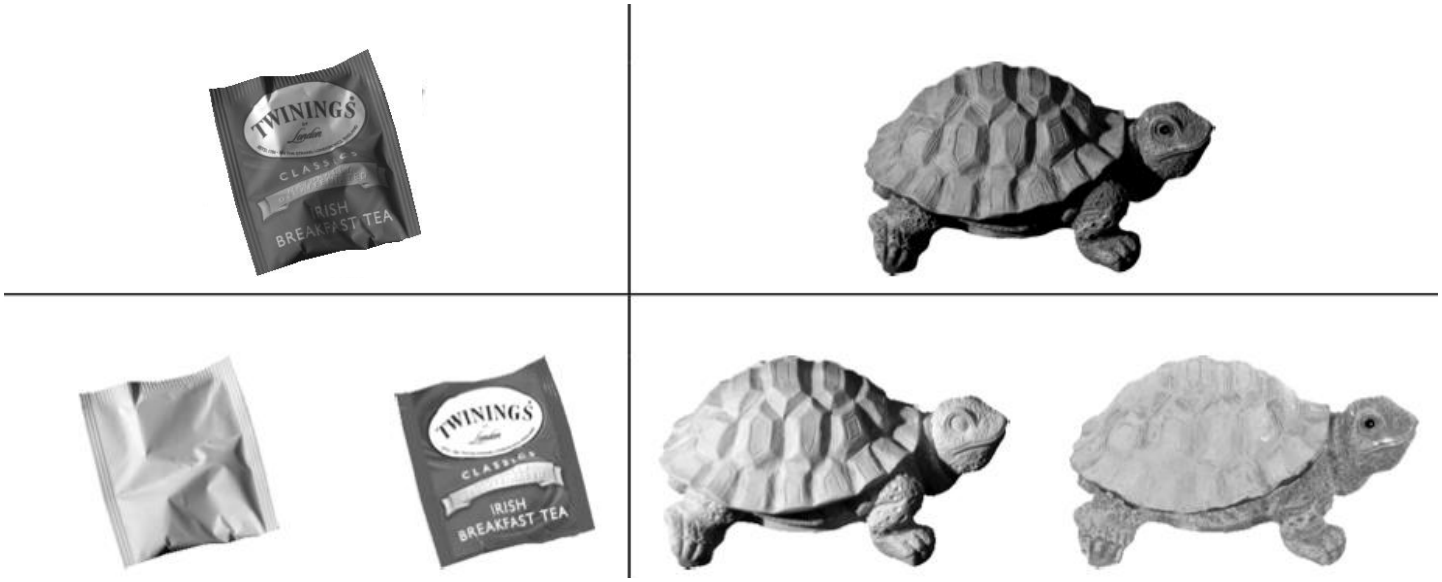
E. H. Land and J. J. McCann.
Lightness and retinex theory. *JOSA*, 1971.

Existing Work: Intrinsic Images



Horn. Determining lightness from an image. CGIP, 1974
Grosse et el., Ground-truth dataset and baseline evaluations for intrinsic image algorithms, ICCV, 2009

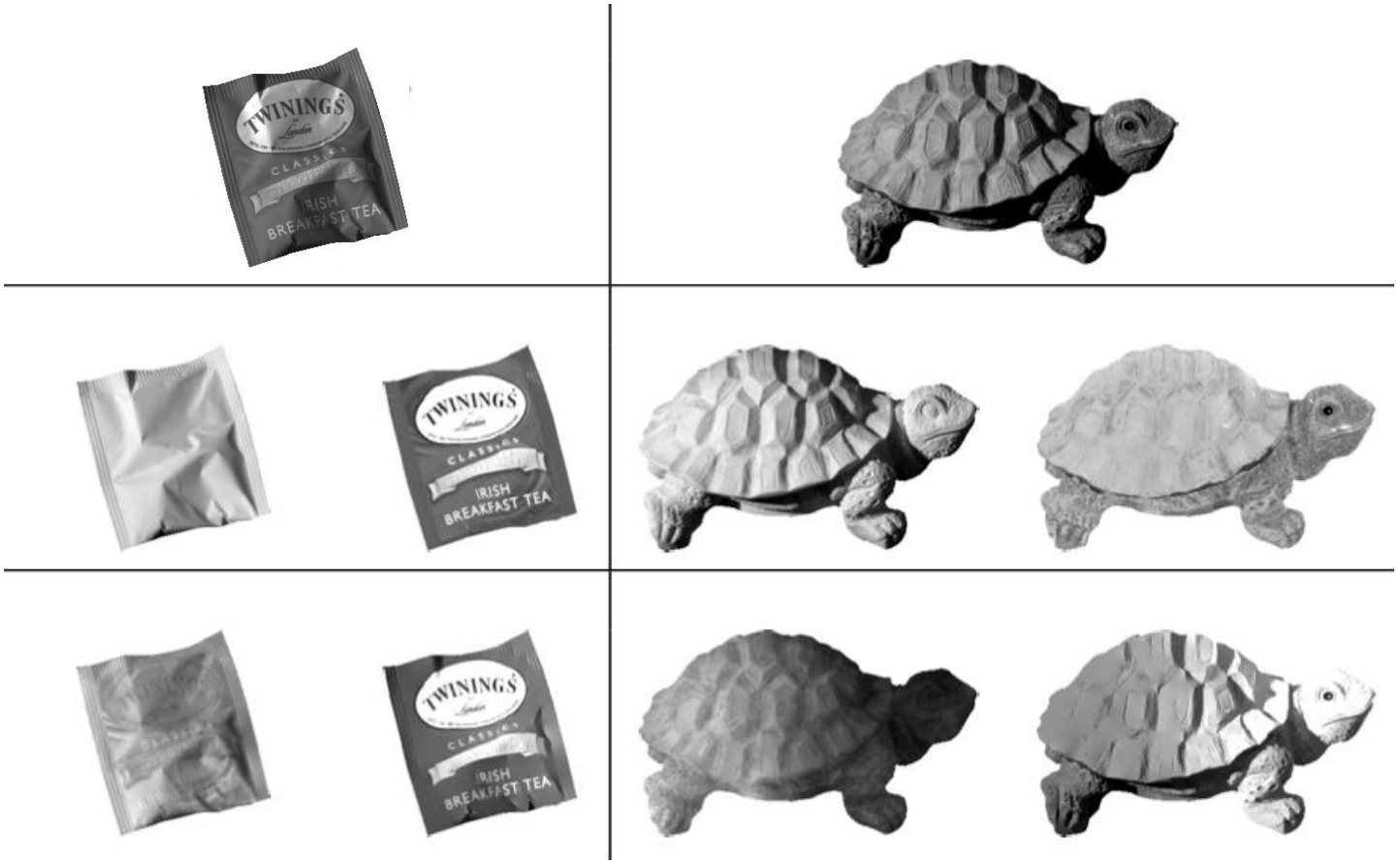
Existing Work: Intrinsic Images



Horn. Determining lightness from an image. CGIP, 1974

Grosse et el., Ground-truth dataset and baseline evaluations for intrinsic image algorithms, ICCV, 2009

Existing Work: Intrinsic Images



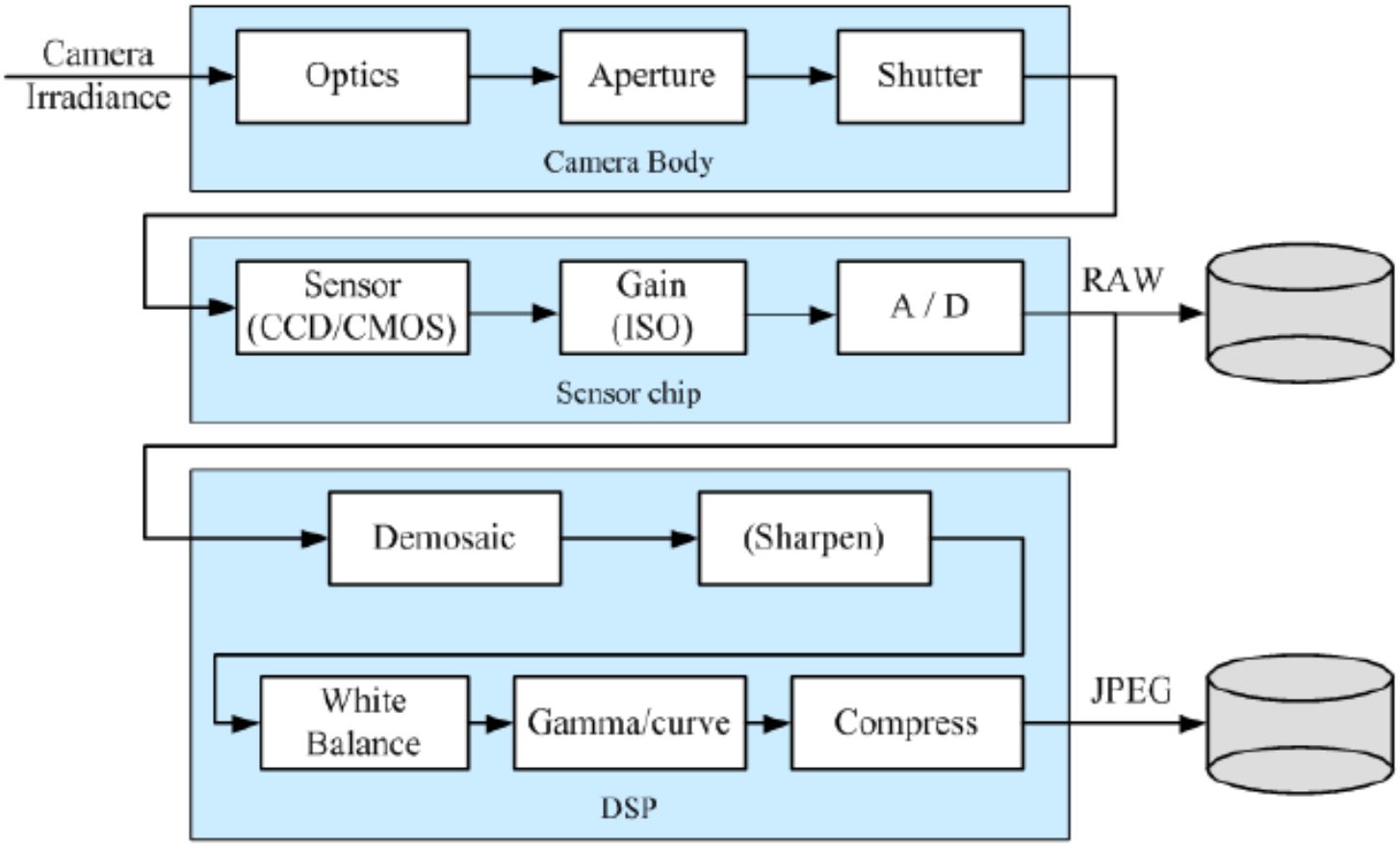
Horn. Determining lightness from an image. CGIP, 1974

Grosse et el., Ground-truth dataset and baseline evaluations for intrinsic image algorithms, ICCV, 2009

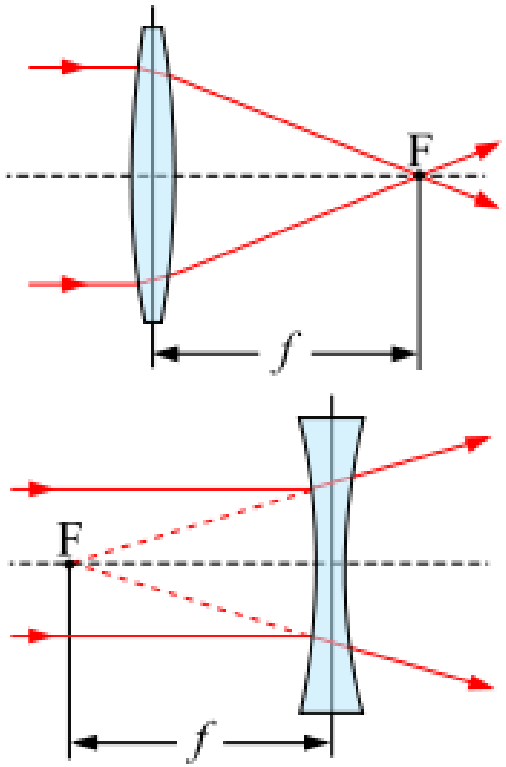
Existing Work: Color Constancy

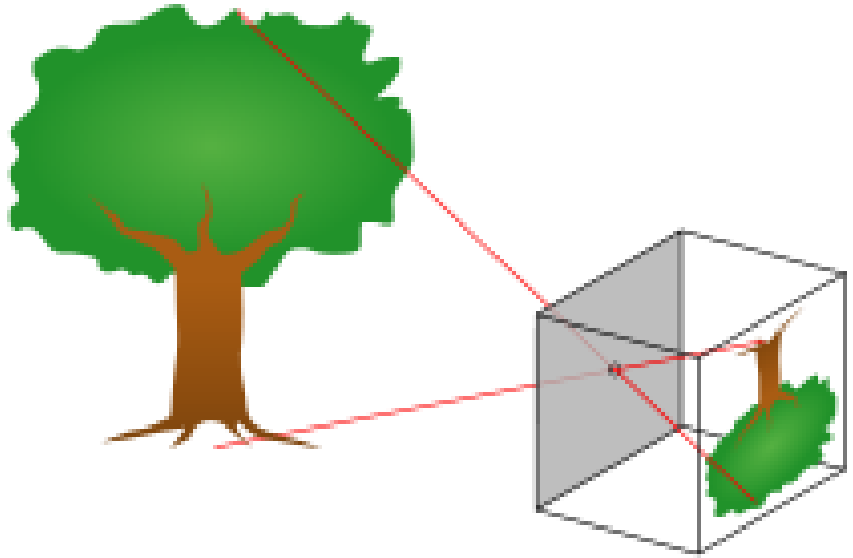
Where is an image from?

Digital Camera



Digital Camera

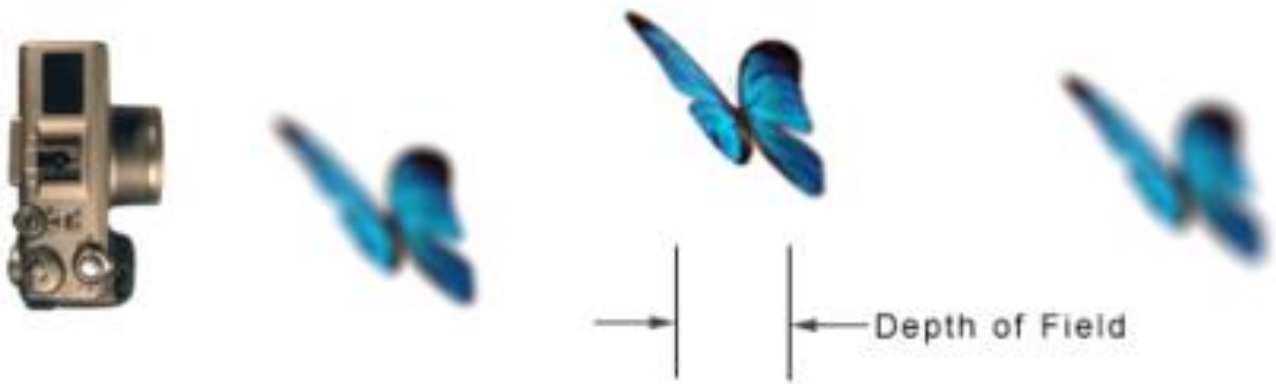




The focal point F and focal length *f* of a positive (convex) lens, a negative (concave) lens

Digital Camera

A 35 mm lens set to *f*/11. The depth-of-field scale (top) indicates that a subject which is anywhere between 1 and 2 meters in front of the camera will be rendered acceptably sharp. If the aperture were set to *f*/22 instead, everything from 0.7 meters to infinity would appear to be in focus.

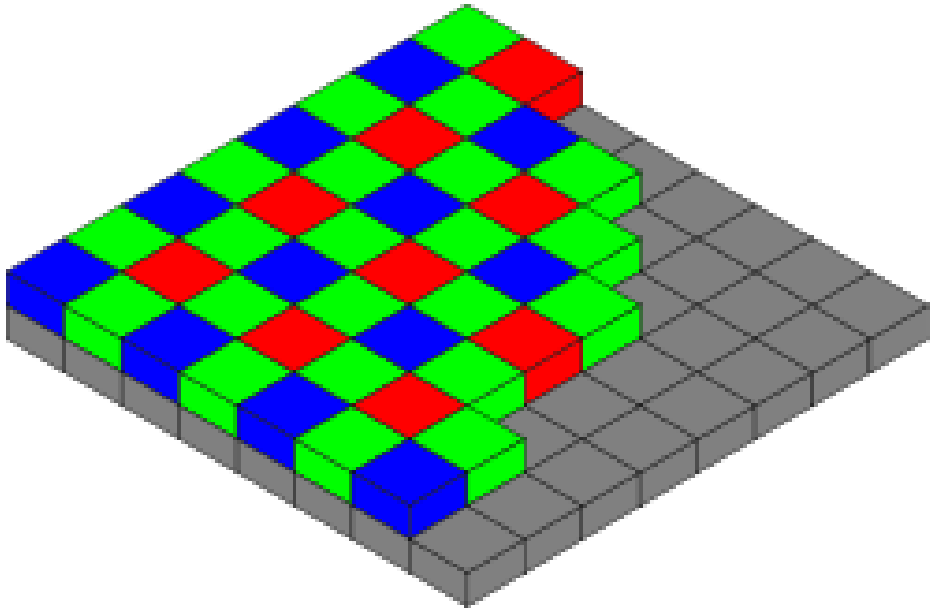


Digital Camera

Digital Camera

- **Demosaicing**

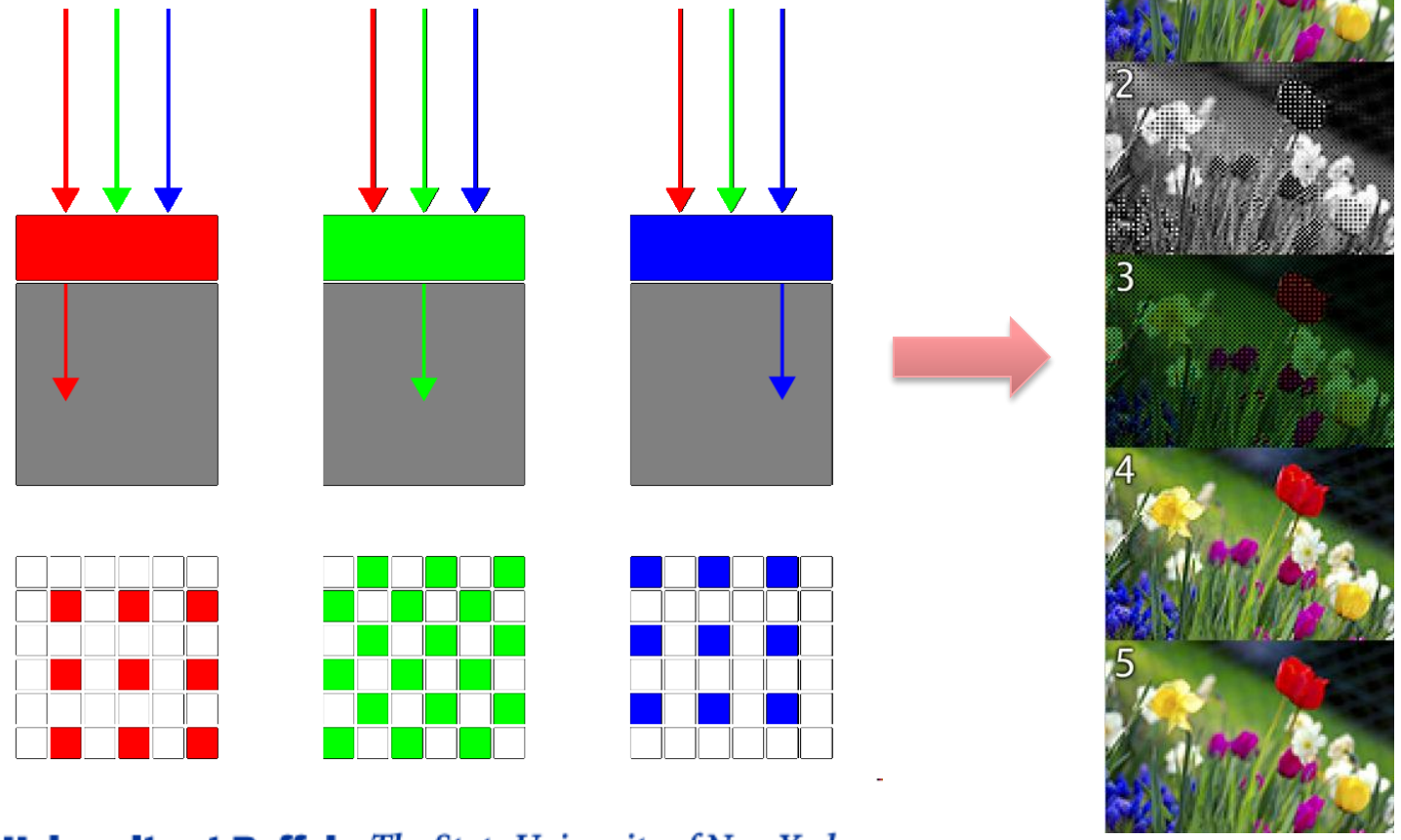
  - is to reconstruct a full color image (i.e. a full set of color triples) from the spatially undersampled color channels output from the CFA.



color filter array：Bayer mode

Digital Camera

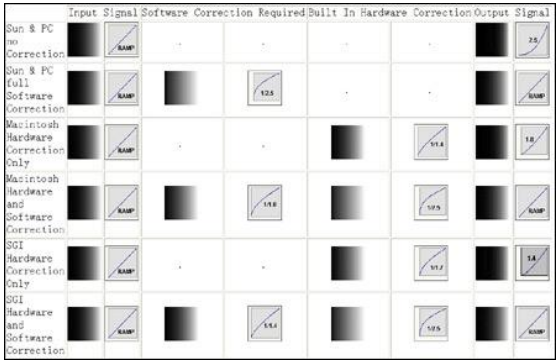
- Demosaicing



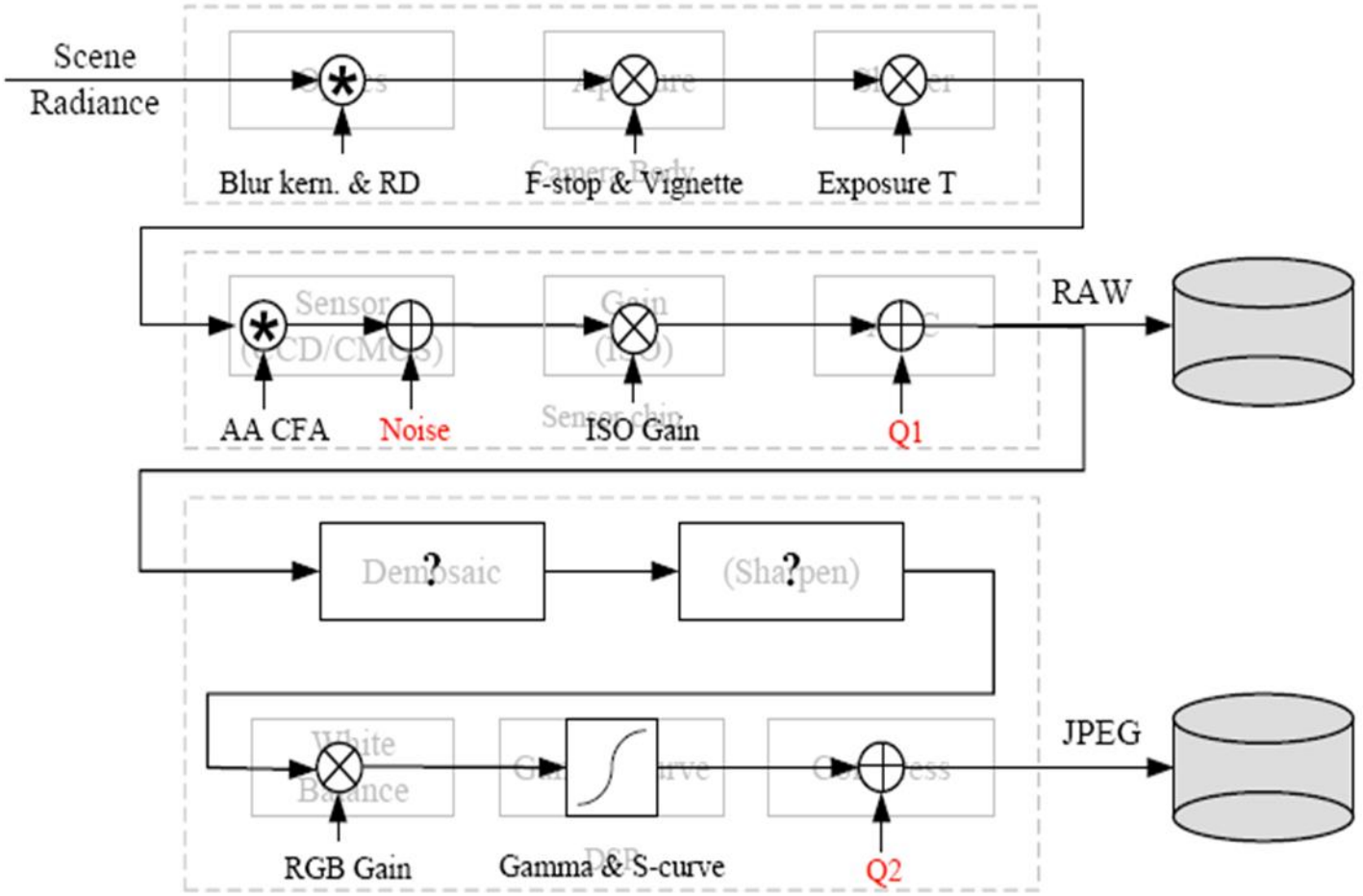
Digital Camera

- **White Balance**

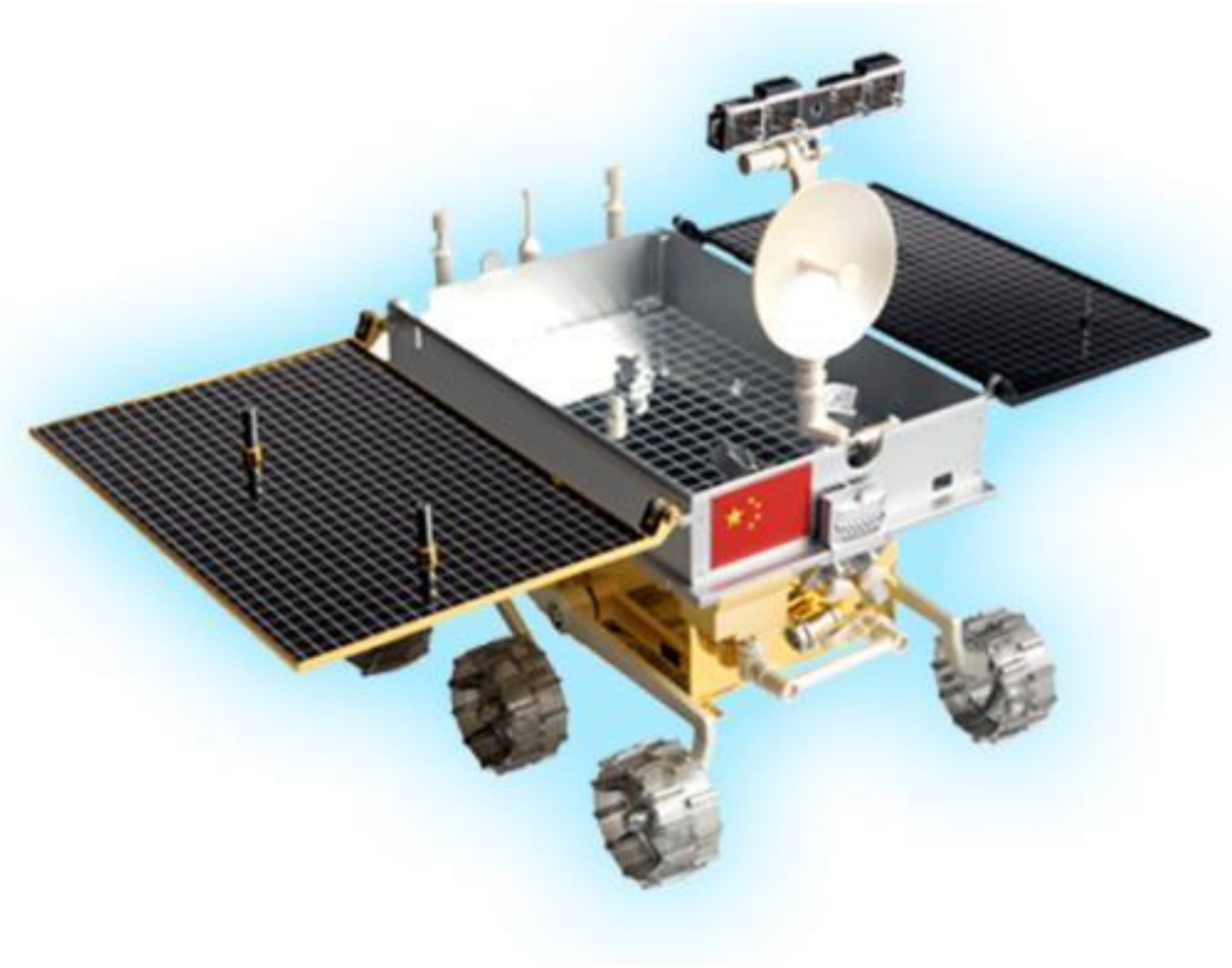
- **Gamma**



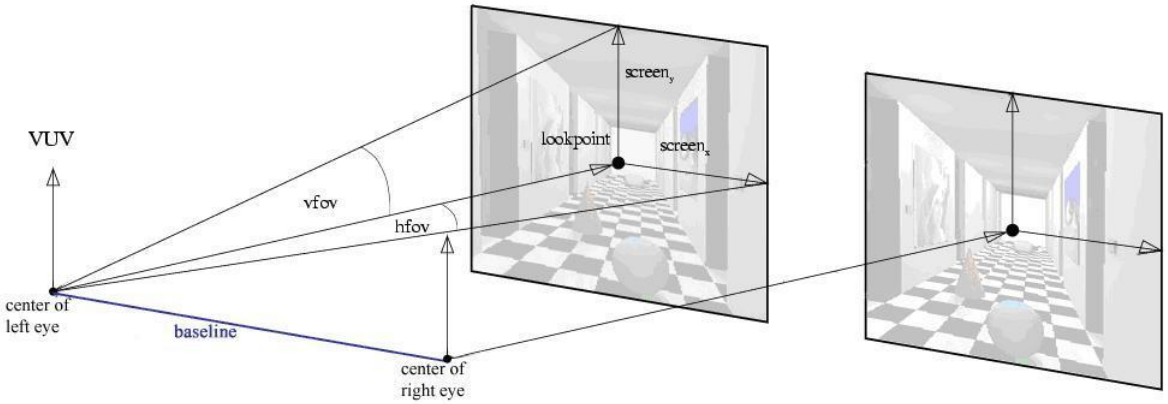
Digital Camera

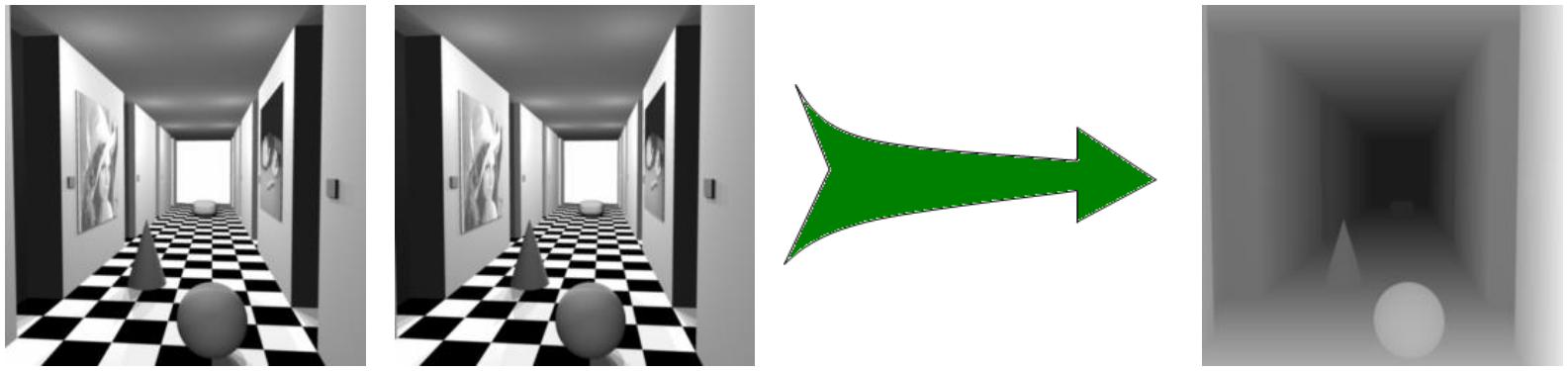


Stereo Camera

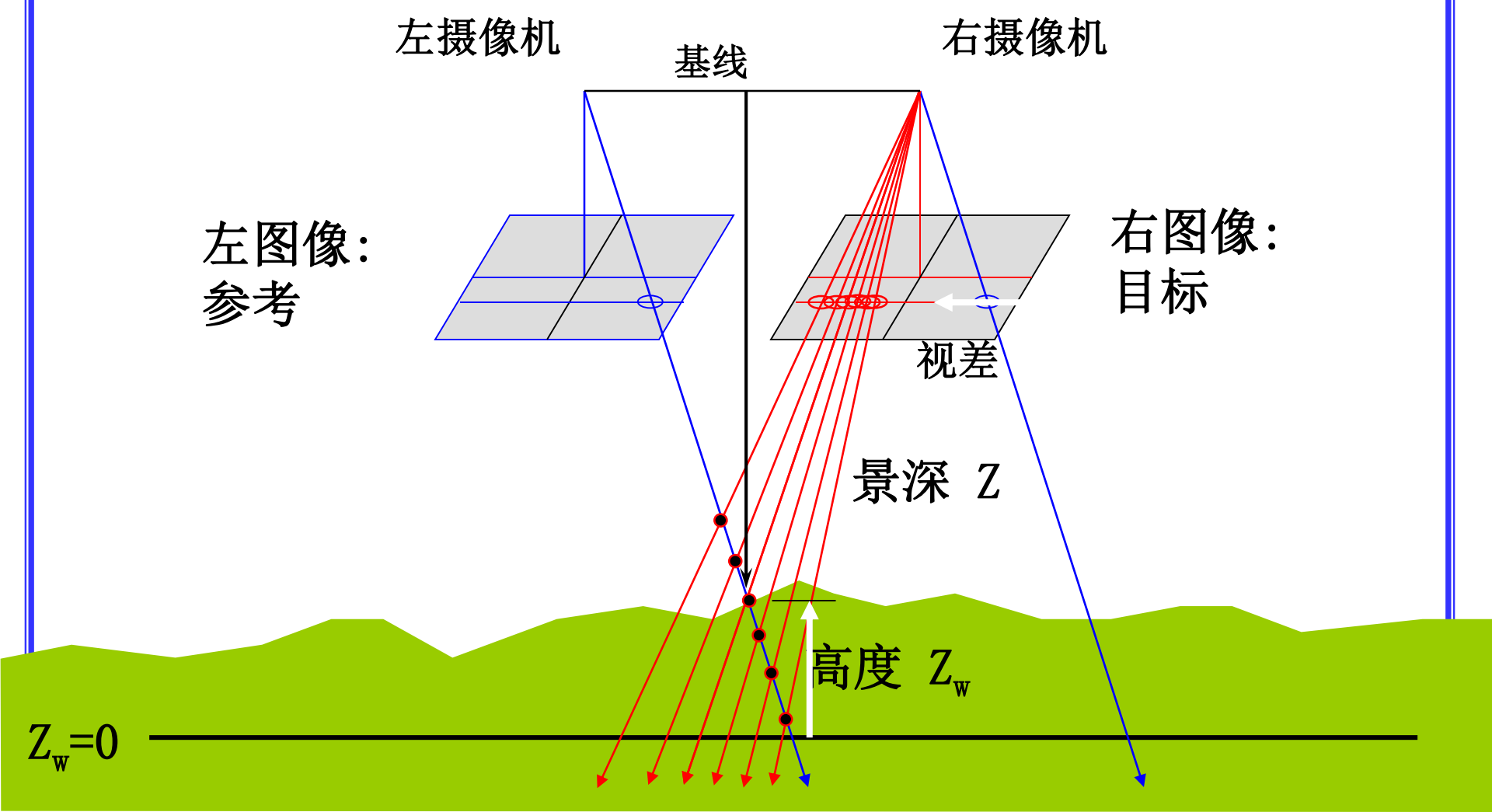


Stereo Camera

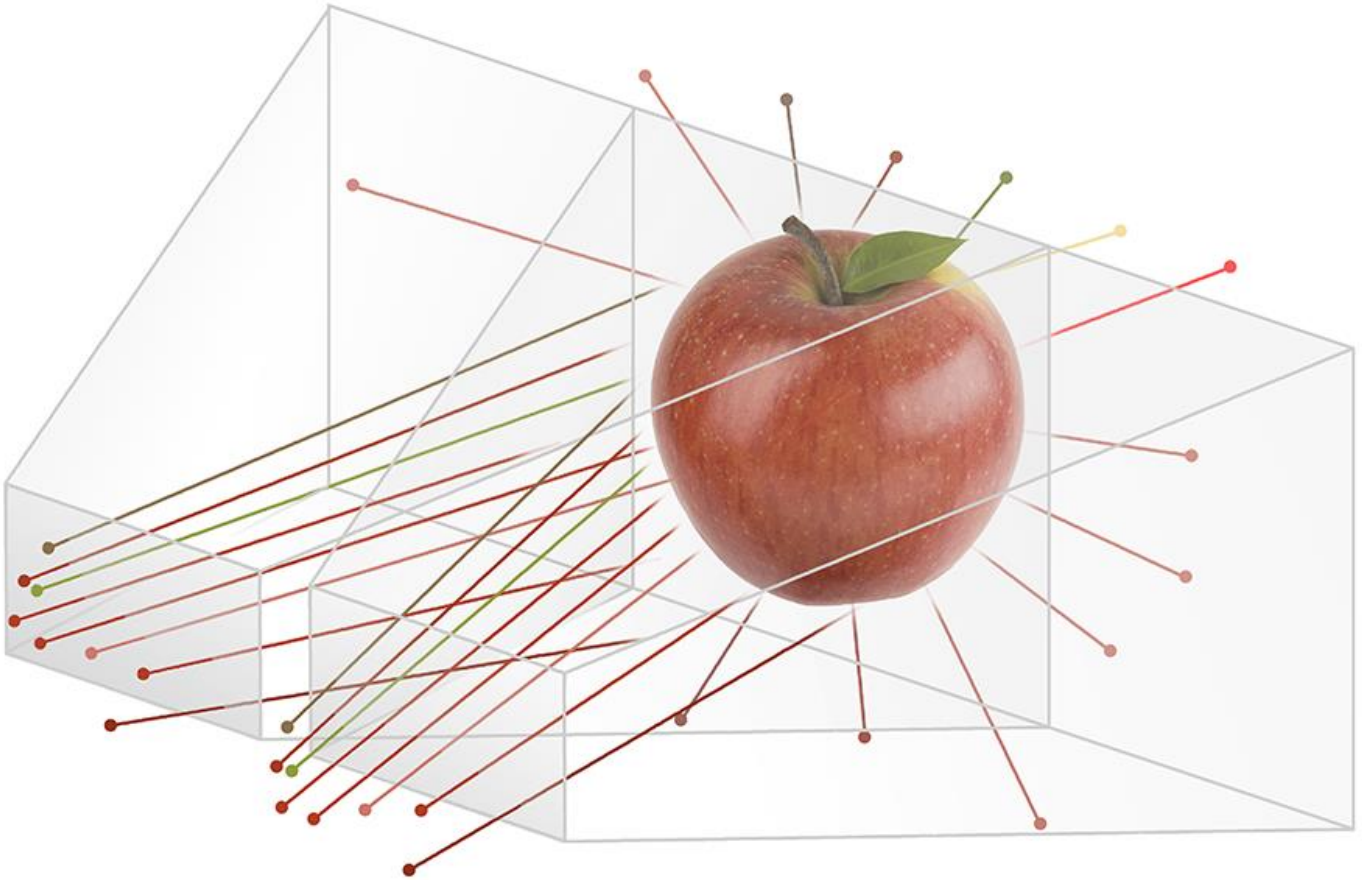




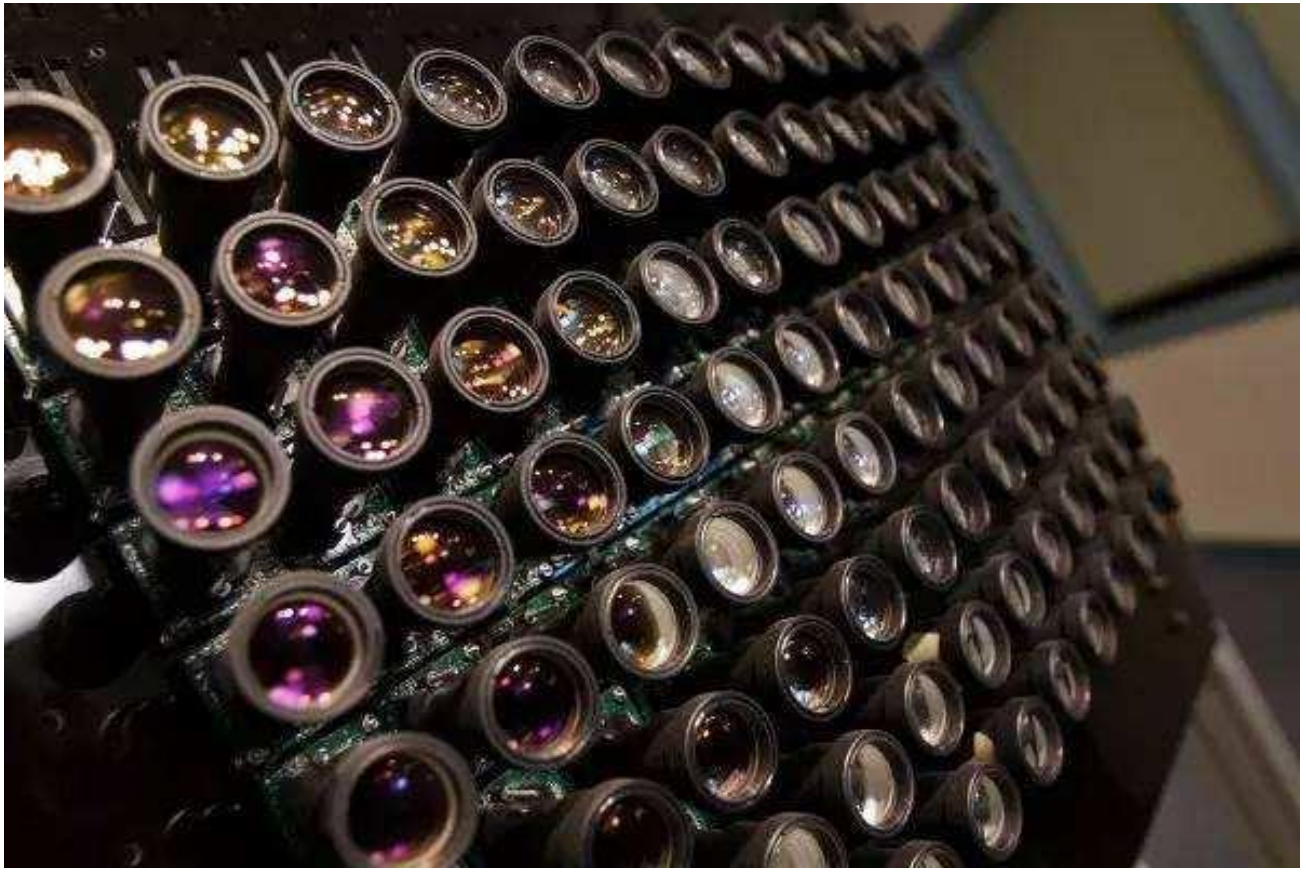
Stereo Matching



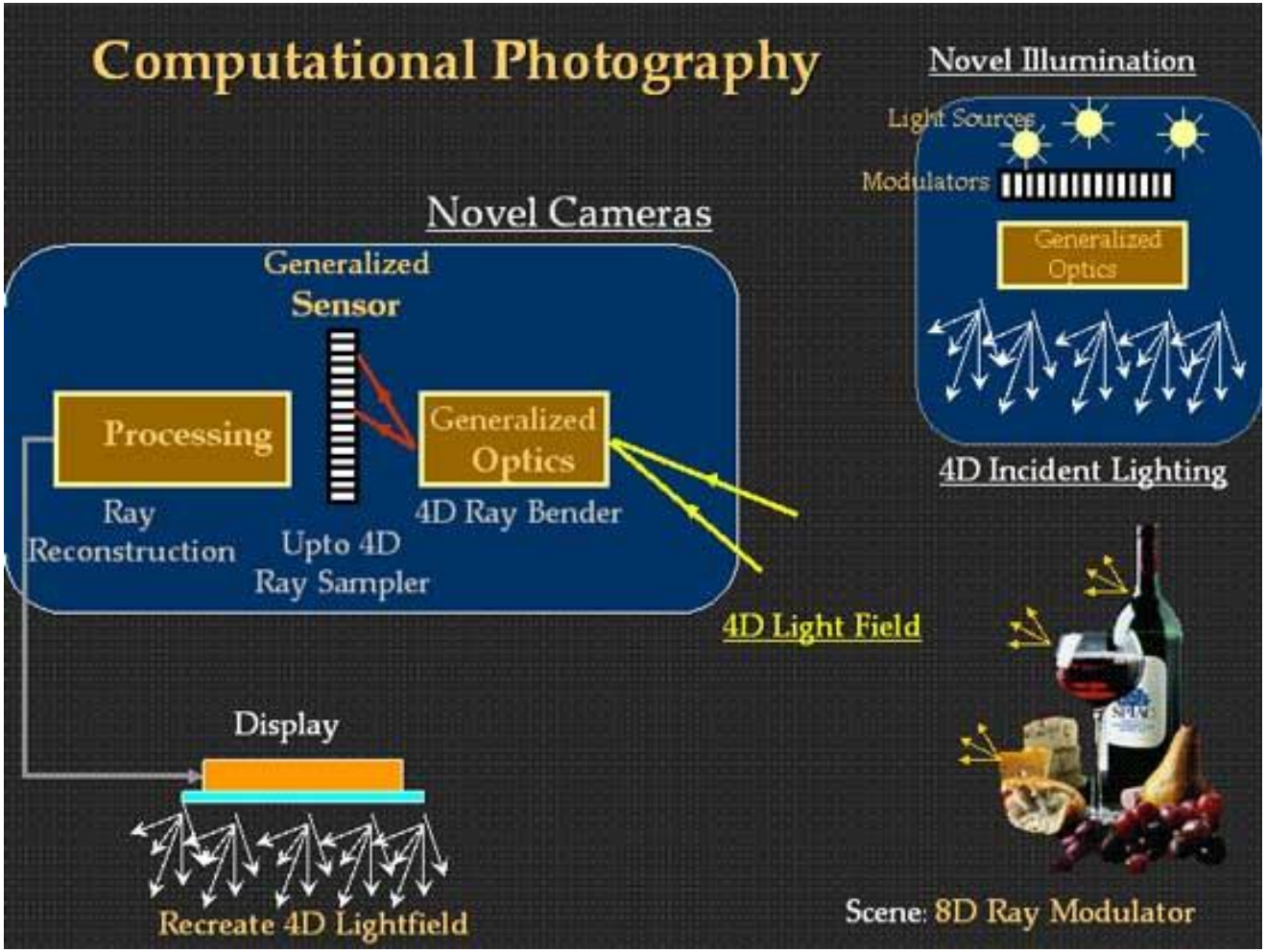
Light field



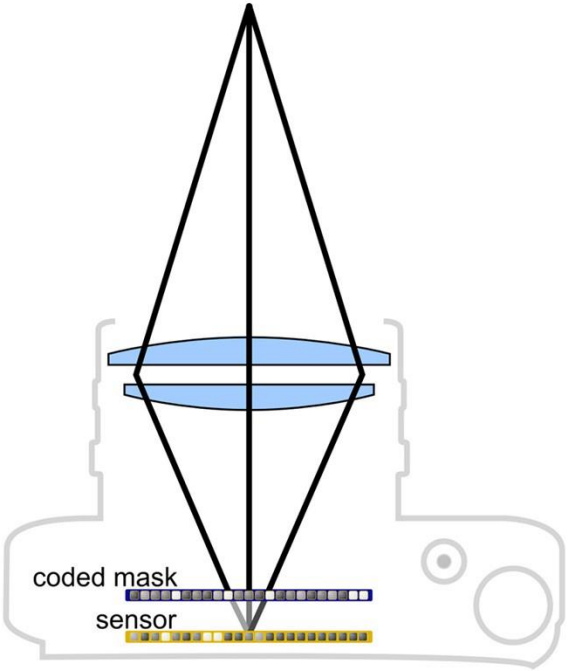
Camera Array



Computational Photography

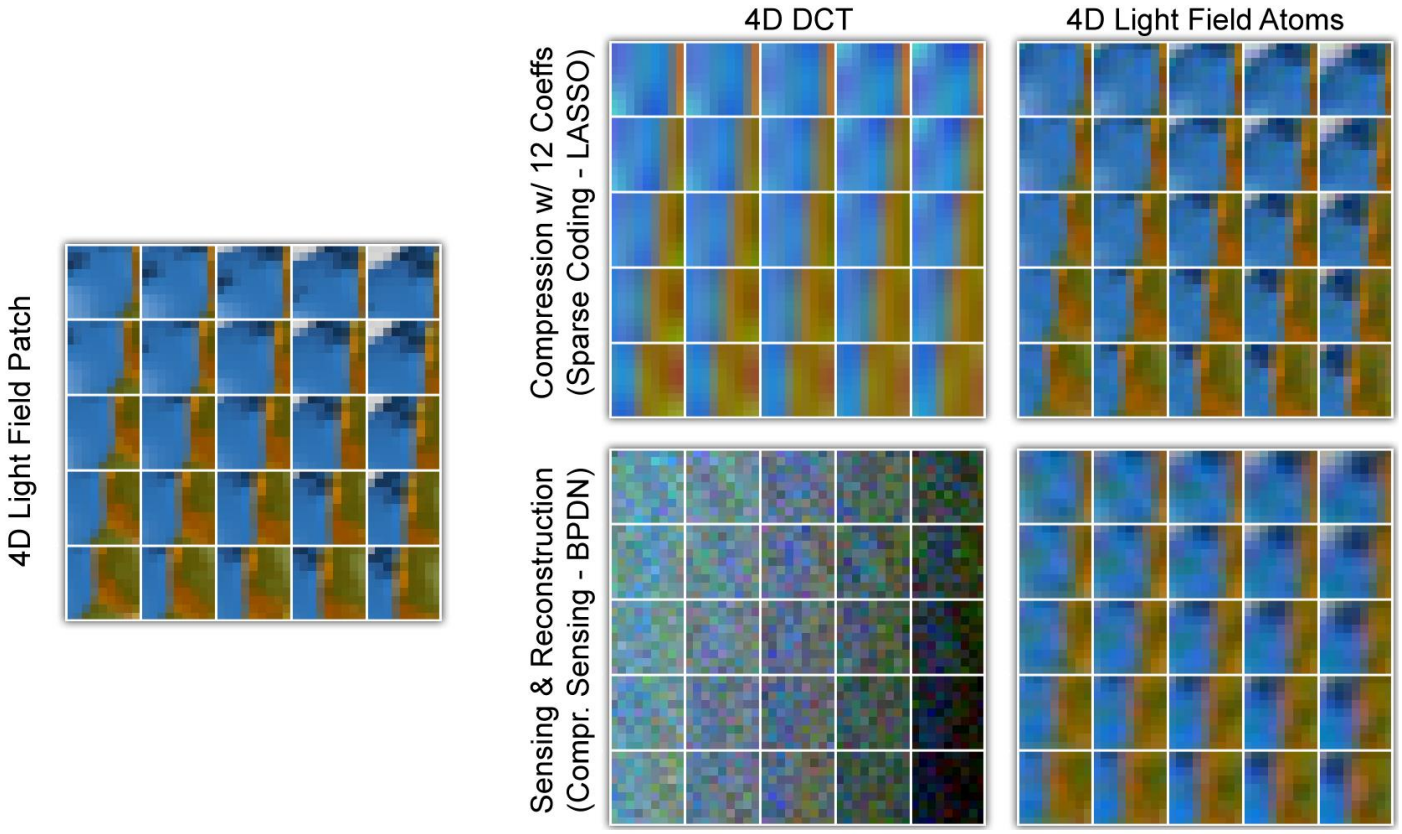


Light Field Camera

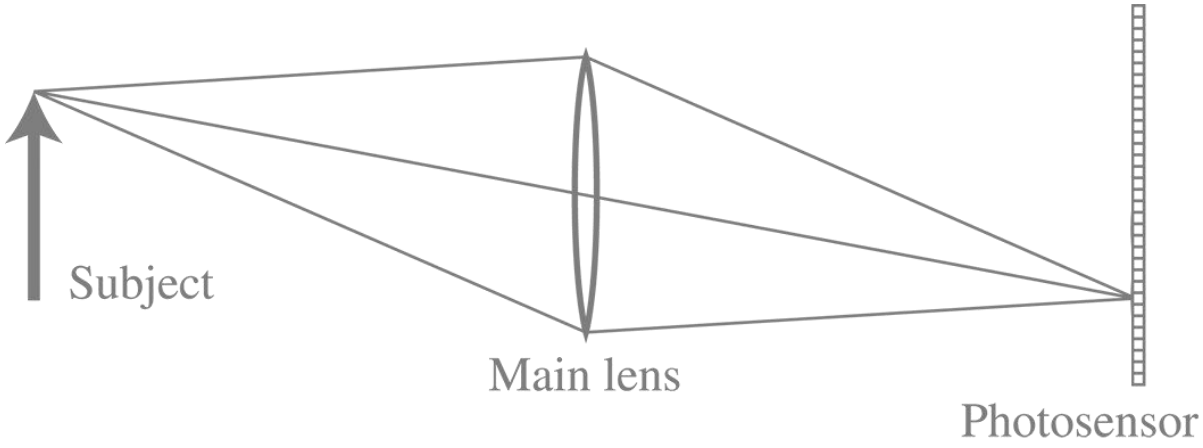


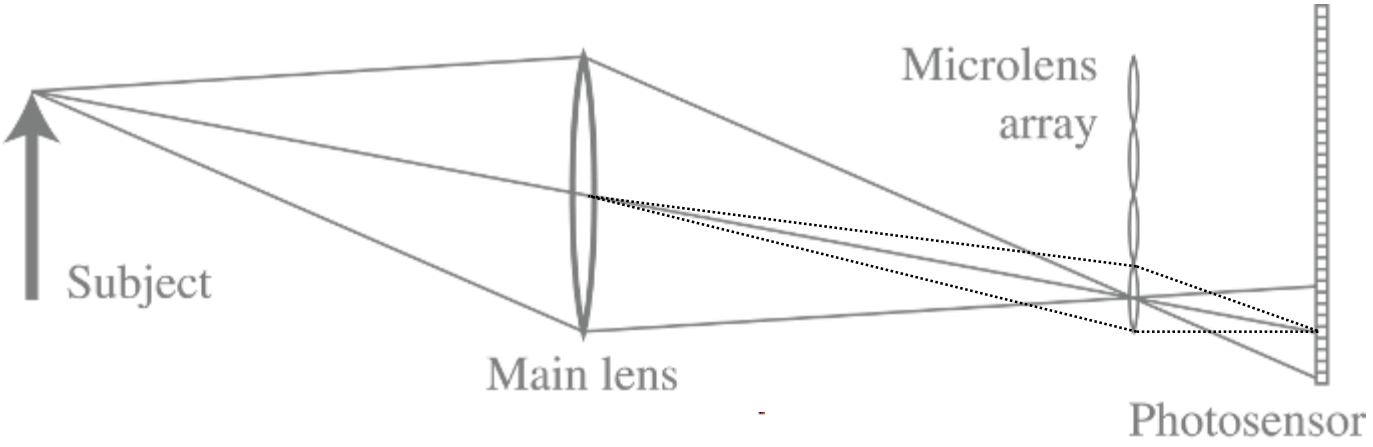
Light Field Camera

Compressibility & Reconstruction from Coded 2D Projections - Qualitative Evaluation

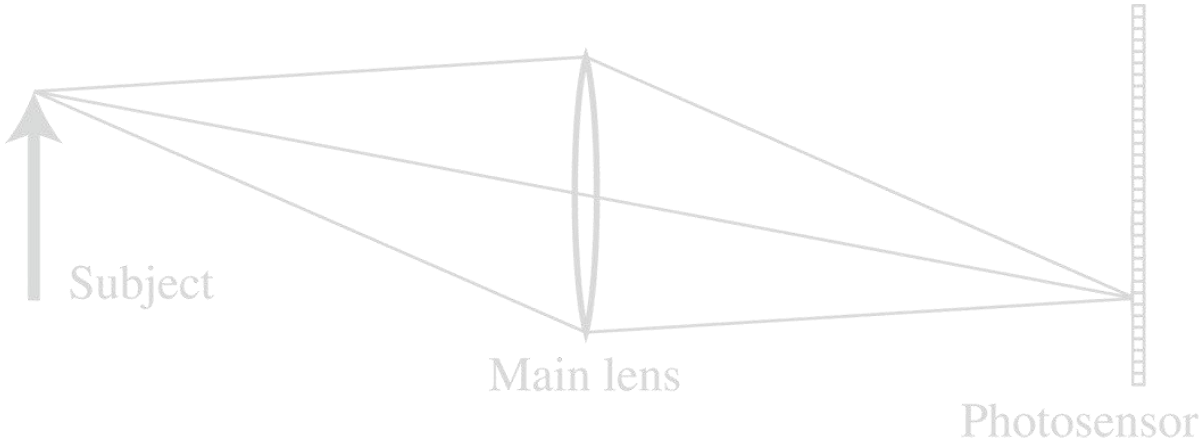


Light Field Camera



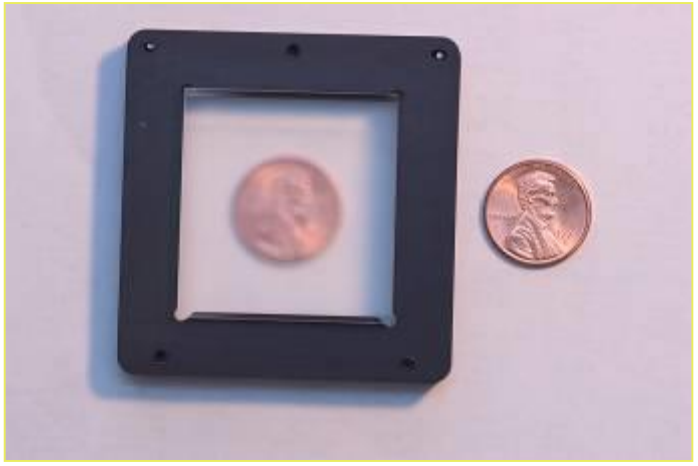


Light Field Camera

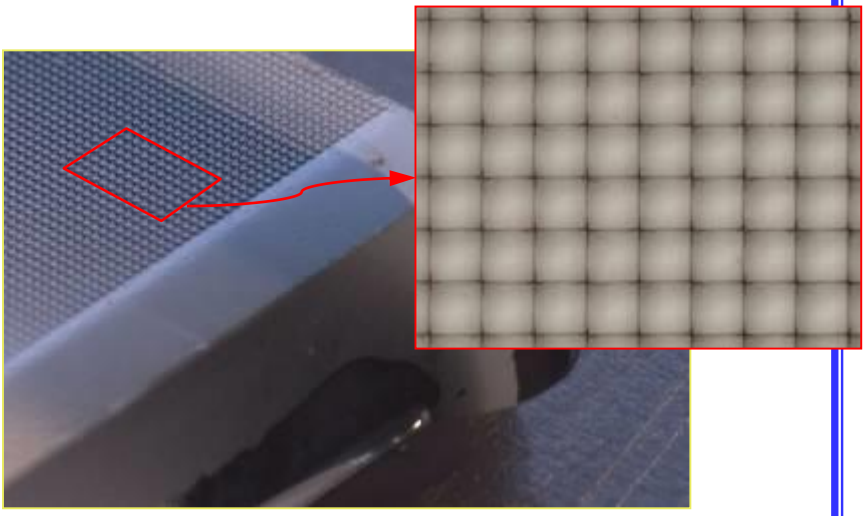


Contax medium format camera

Kodak 16-megapixel sensor

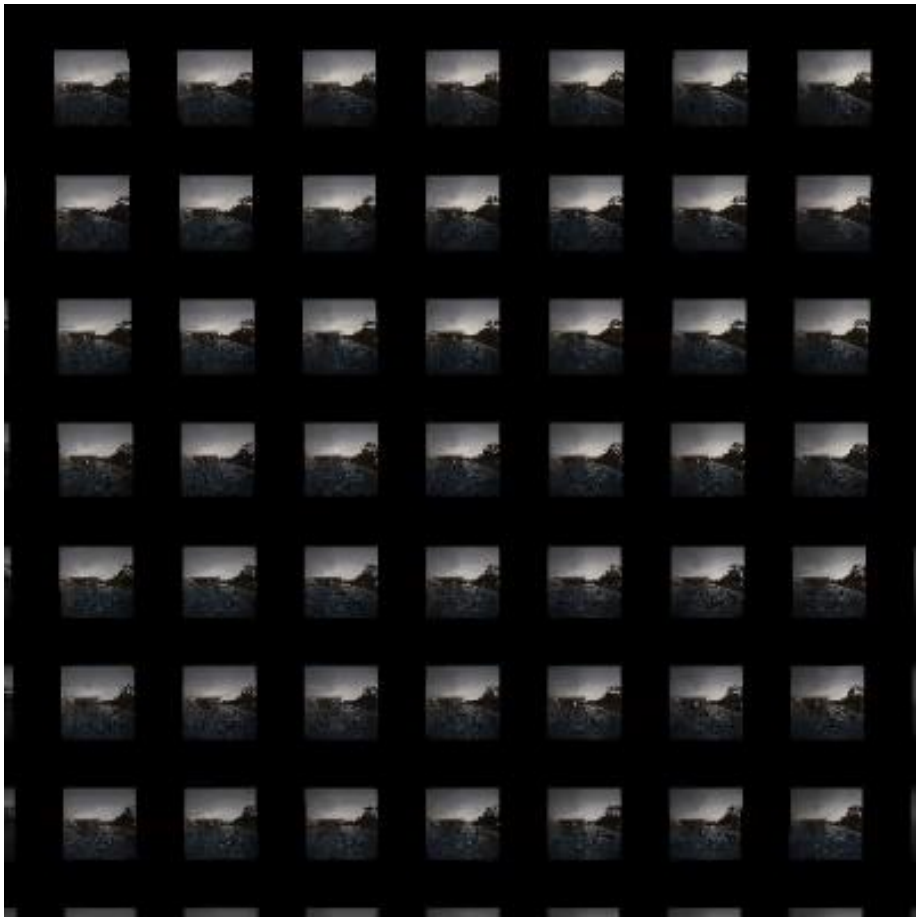


Adaptive Optics microlens array

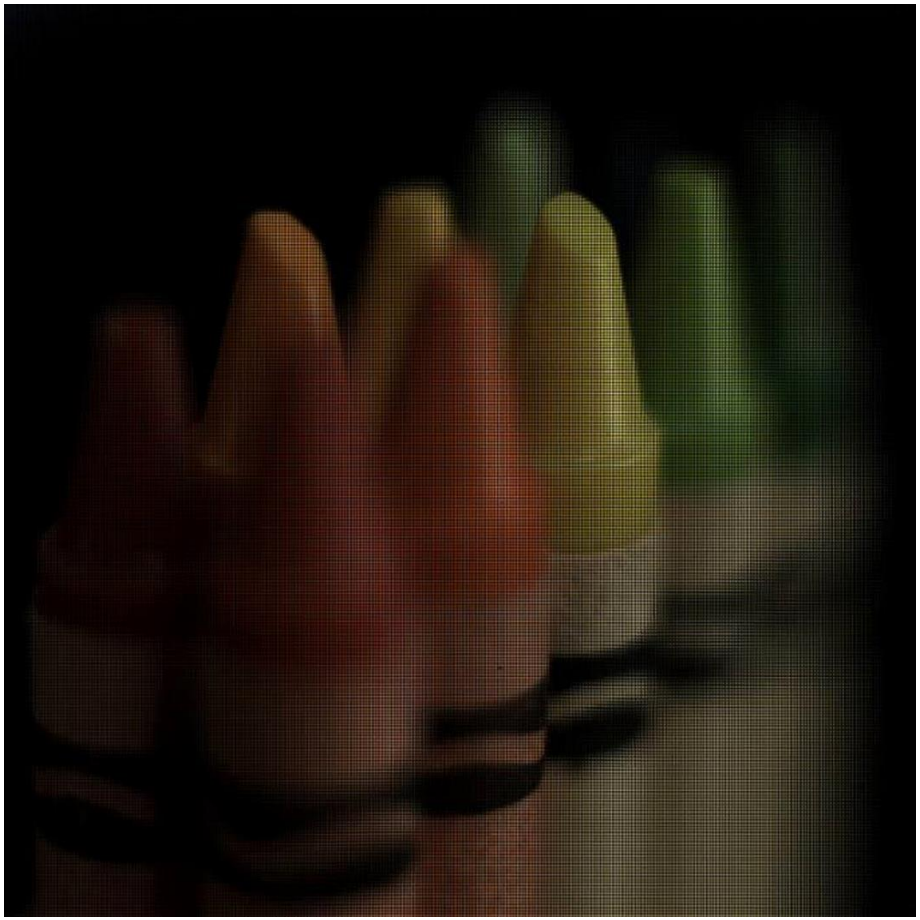


125µ square-sided microlenses

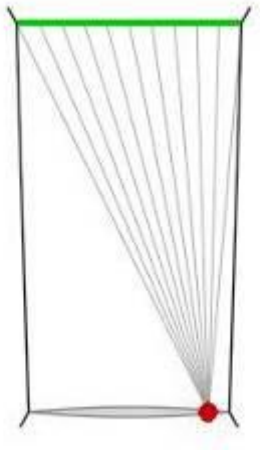
Light Field Camera



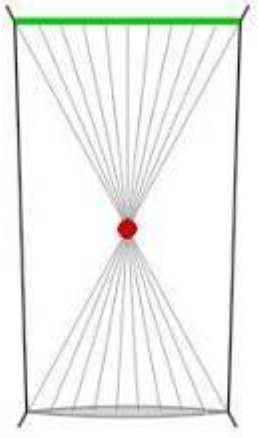
Light Field Camera



Light Field Camera



Light Field Camera



Light Field Camera

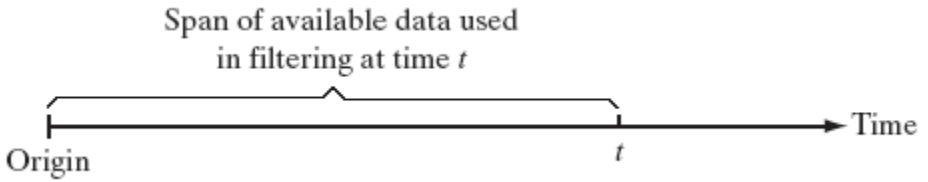
Applications of LF Techniques

How to process an image?

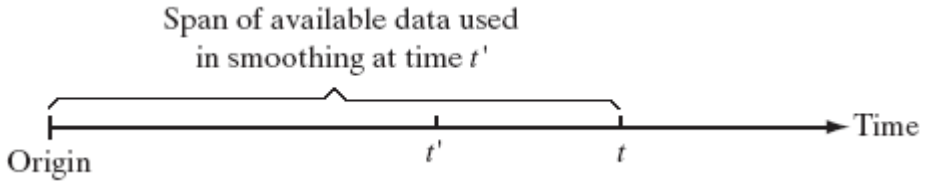
滤波与滤波器

在信号与图像处理领域，滤波器主要有三种应用

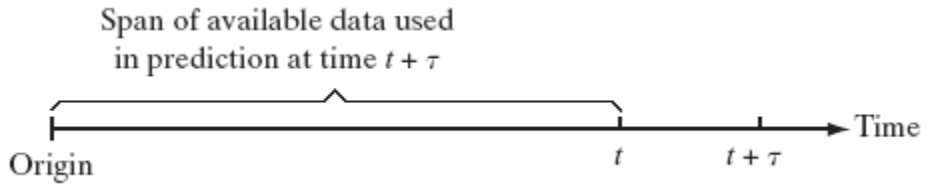
滤波去噪



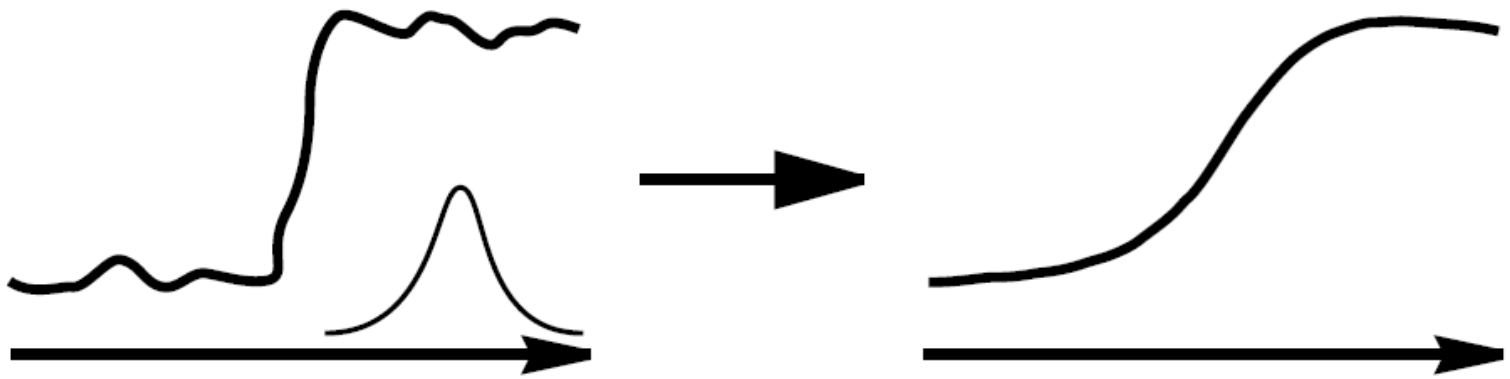
平滑内插



预测估计

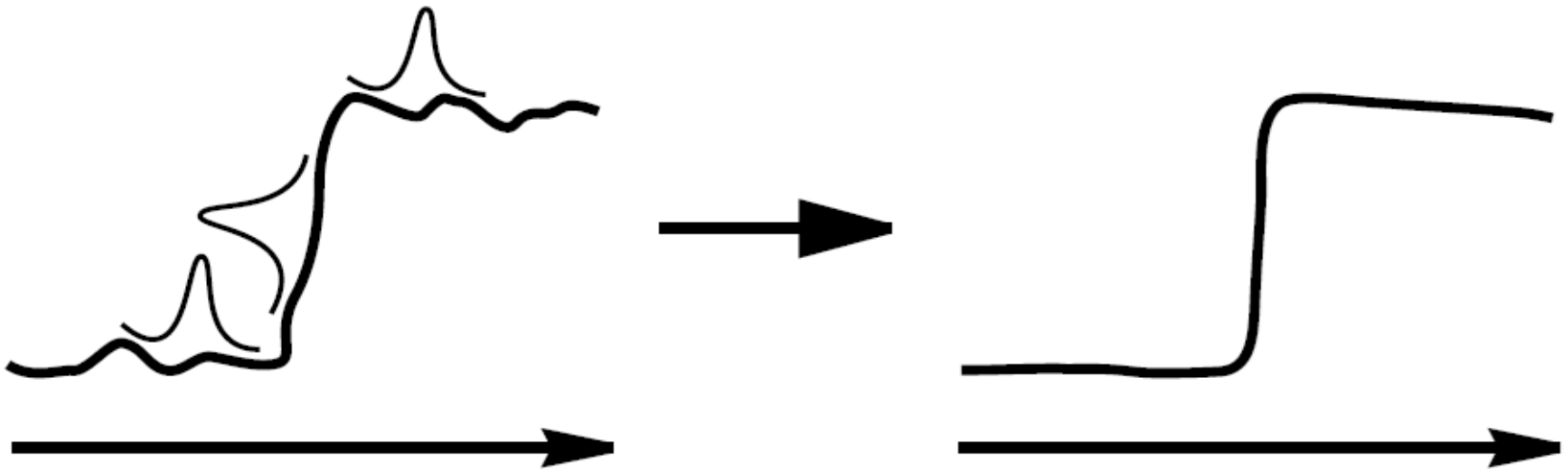


Bilateral Filter

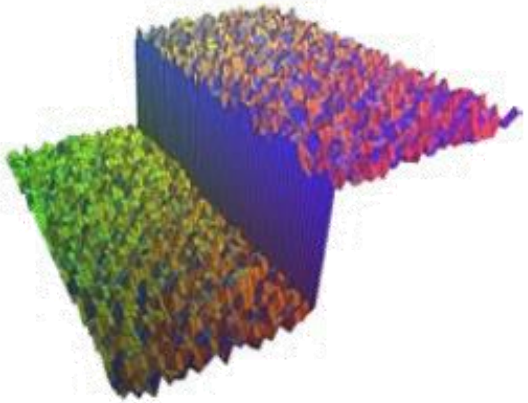


高斯滤波器的不足

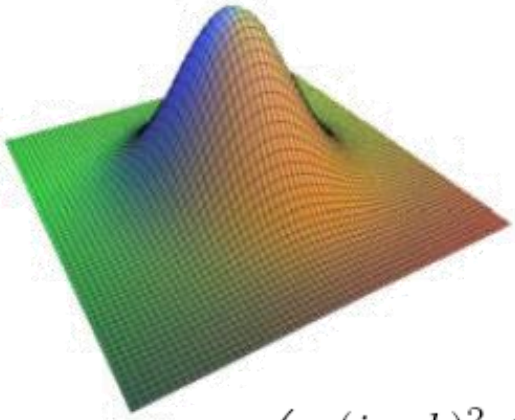
Bilateral Filter

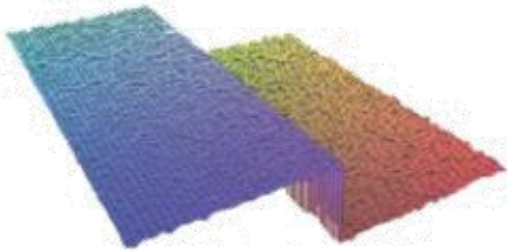


Bilateral Filter



noisy step edge input;





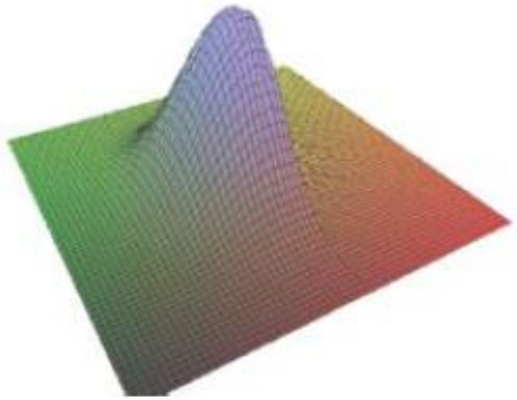
$$d(i,j,k,l) = \exp\left(-\frac{(i-k)^2+(j-l)^2}{2\sigma_d^2}\right), \qquad r(i,j,k,l) = \exp\left(-\frac{\|f(i,j)-f(k,l)\|^2}{2\sigma_r^2}\right).$$

domain filter (Gaussian); range filter (similarity to center pixel value);

Bilateral Filter

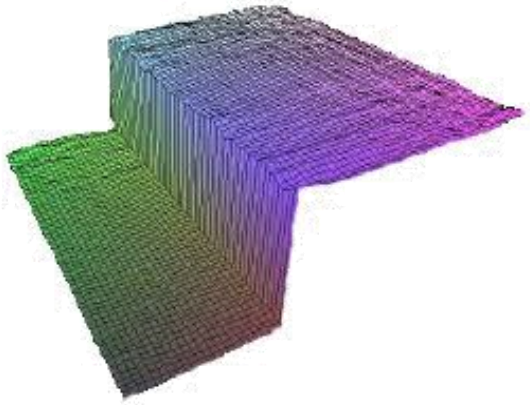
双边滤波器

$$w(i,j,k,l) = \exp\left(-\frac{(i-k)^2+(j-l)^2}{2\sigma_d^2} - \frac{\|f(i,j)-f(k,l)\|^2}{2\sigma_r^2}\right)$$



滤波输出

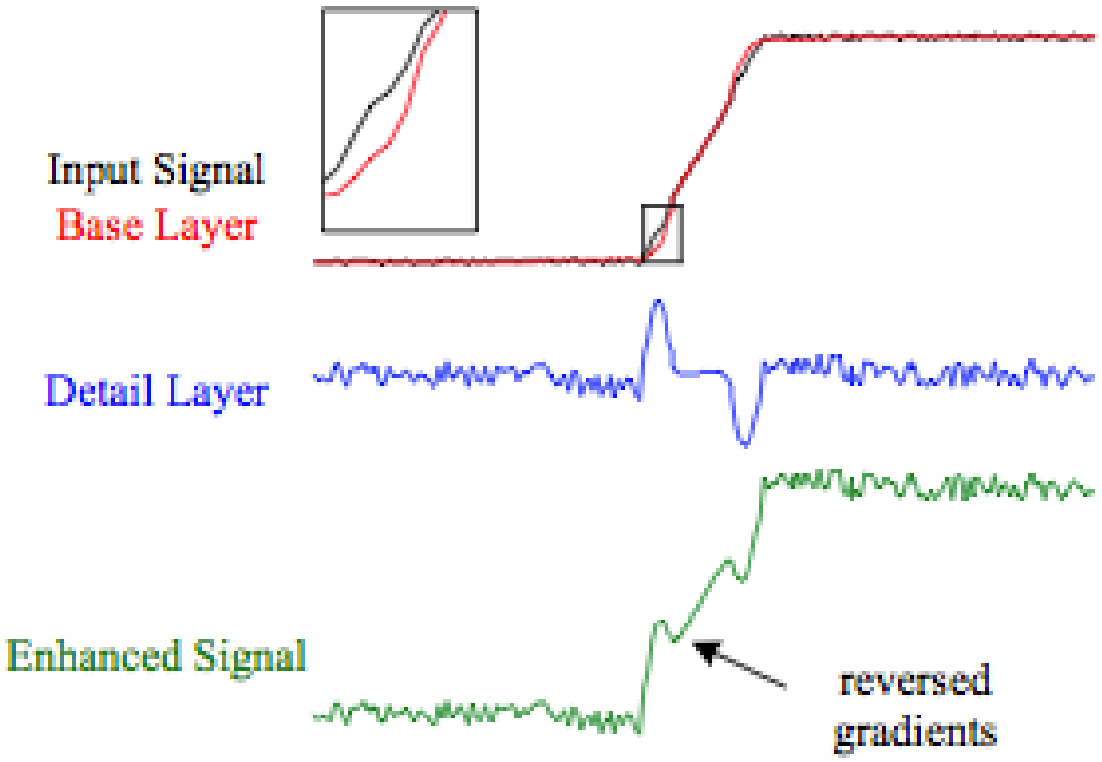
$$g(i,j) = \frac{\sum_{k,l} f(k,l)w(i,j,k,l)}{\sum_{k,l} w(i,j,k,l)}.$$



Bilateral Filter

原始图像　　高斯滤波　　双边滤波

Image guided Filter



Bilateral Filter

Image guided Filter

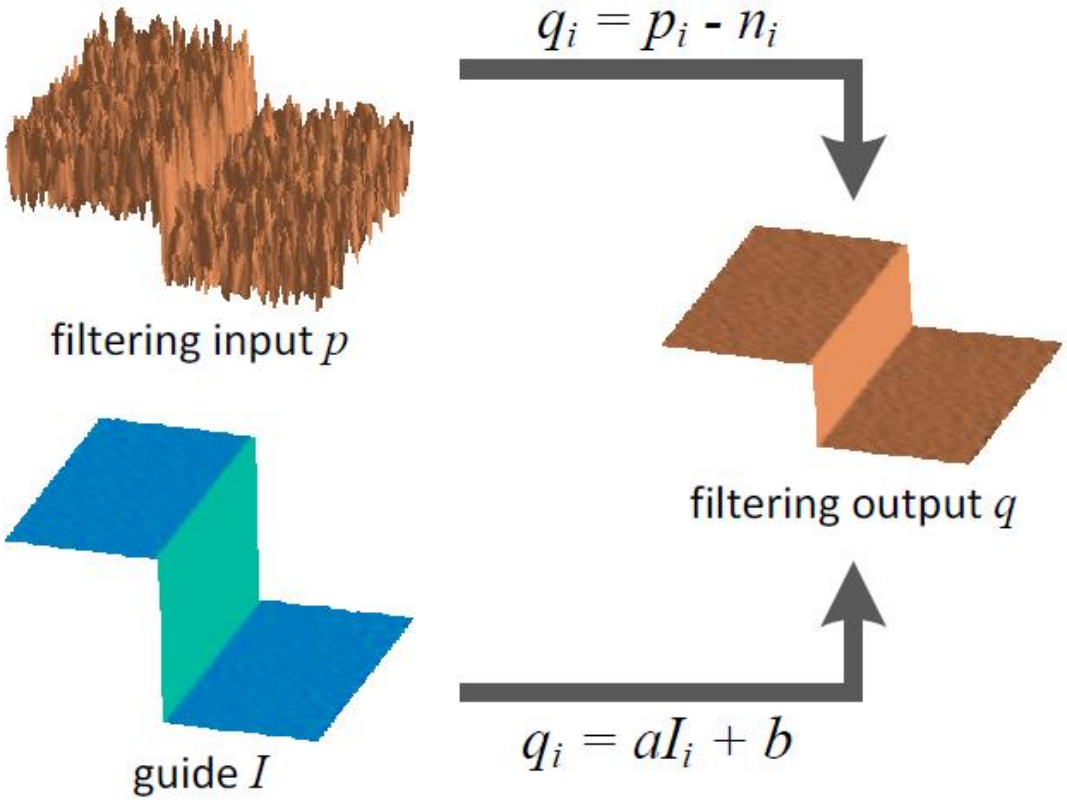


Image guided Filter

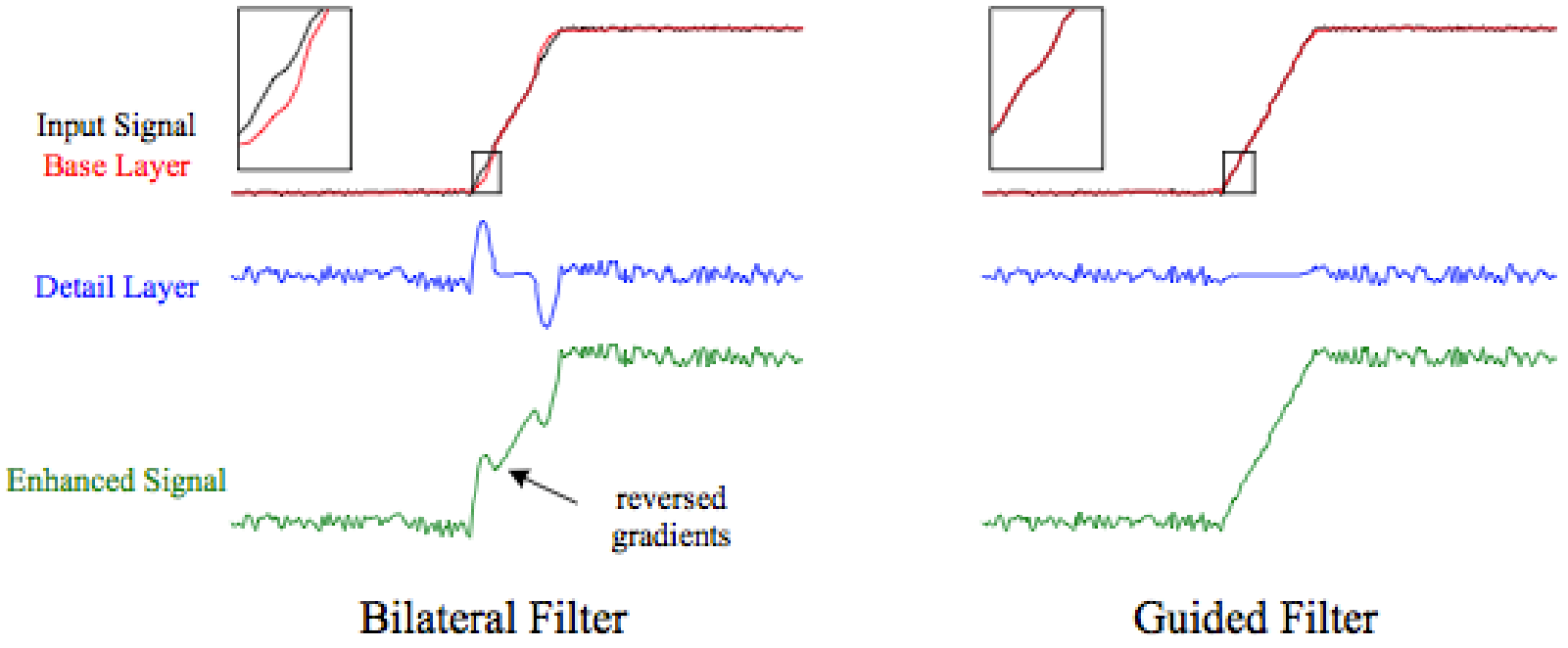
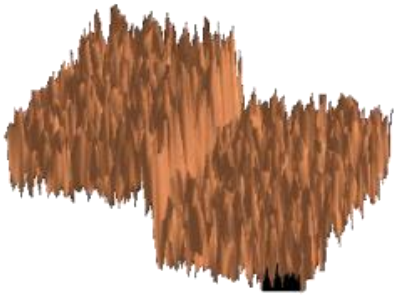


Fig. 4. 1-D illustration for detail enhancement. See the text for explanation.

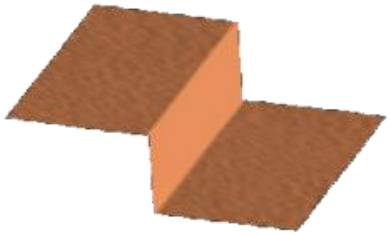
Image guided Filter



input p

$$q_i = p_i - n_i$$

n_i - noise / texture



output q

$$\min_{(a,b)} \sum_i (aI_i + b - p_i)^2 + \varepsilon a^2$$

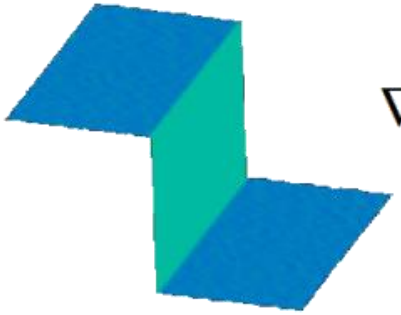
Linear regression

$$a = \frac{\mathrm{cov}(I,p)}{\mathrm{var}(I) + \varepsilon}$$

$$b = \bar{p} - a\bar{I}$$

$$q_i = aI_i + b$$

$$\nabla q_i = a \nabla I_i$$

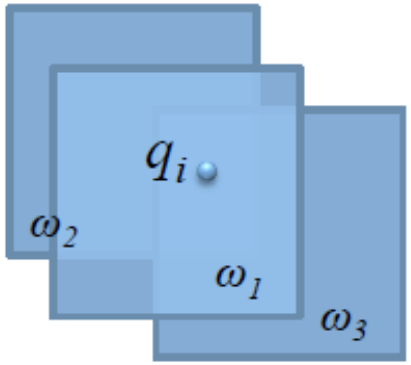


guide I

Image guided Filter

Extend to the entire image

- In all local windows ω_k,compute the linear coefficients
- Compute the average of $a_k I_i + b_k$ in all ω_k that covers pixel q_i



- Window radius r
- Regularization ε

Definition

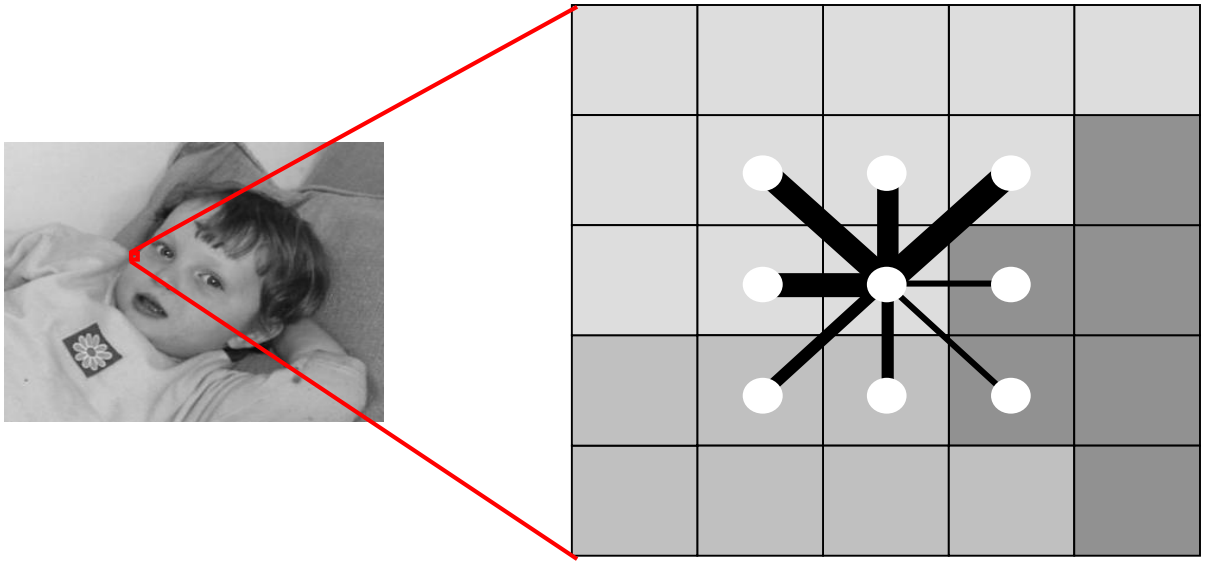
$$a_k = \frac{\mathrm{cov}_k(I,p)}{\mathrm{var}_k(I)+\varepsilon}$$

$$b_k = \bar{p}_k - a\bar{I}_k$$

$$q_i = \frac{1}{|\omega|}\sum_{k|i\in\omega_k}(a_k I_i + b_k)$$

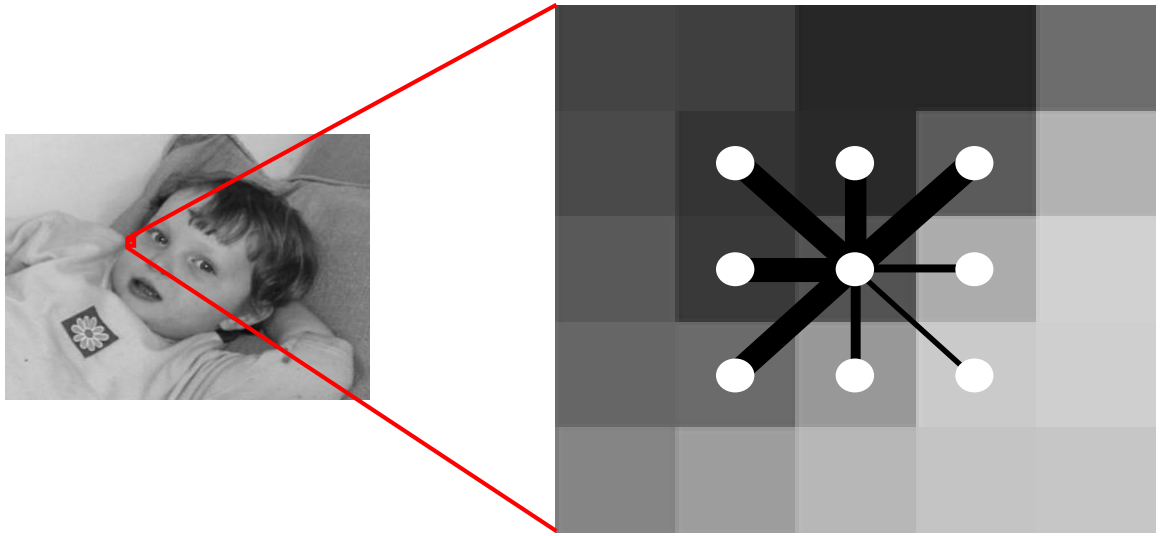
$$= \bar{a}_i I_i + \bar{b}_i$$

Image guided Filter



$$W_{GF}(i,j) = e^{-\|C_i - C_j\|^2/\sigma^2}$$

Image guided Filter



$$W_{IGF}(i,j) \propto \sum_{k|(i,j)\in w_k} \left(1+(C_i-\mu_k)^T(\Sigma_k+\varepsilon I_3)^{-1}(C_j-\mu_k)\right)$$

Image guided Filter

Guidance I

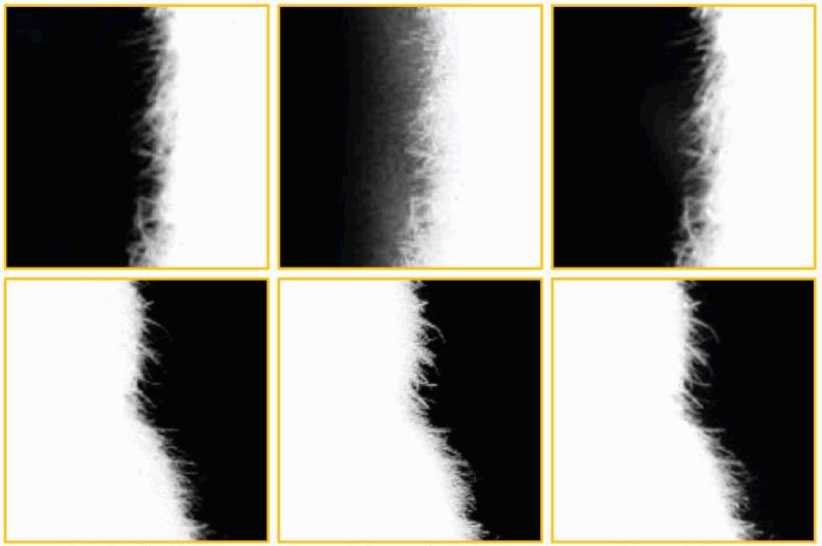
Guided Filter

Filter Input p

Joint Bilateral Filter

Image guided Filter

Guidance I | Binary Mask p | Guided Filter Output q



Guided Filter | Photoshop | Closed-form

图像退化过程

Blurred image G

$=$

Sharp image F

$\otimes$

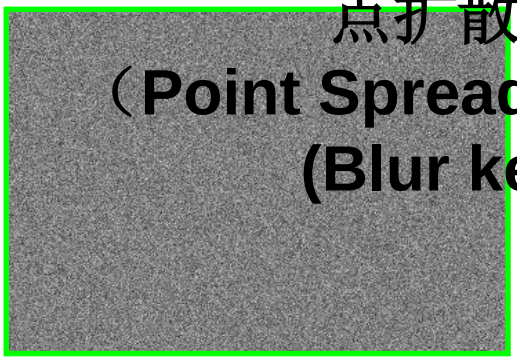
Blur kernel h

点扩散函数

（**Point Spread Function**）

(Blur kernel)

$+$



Camera Noise n

Lens and defocus

Lens' aperture

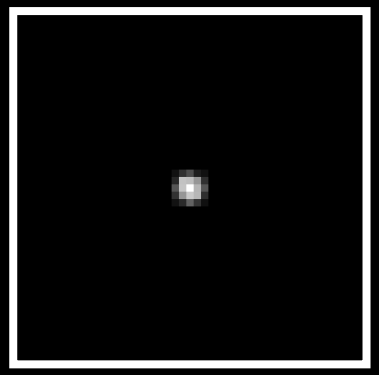
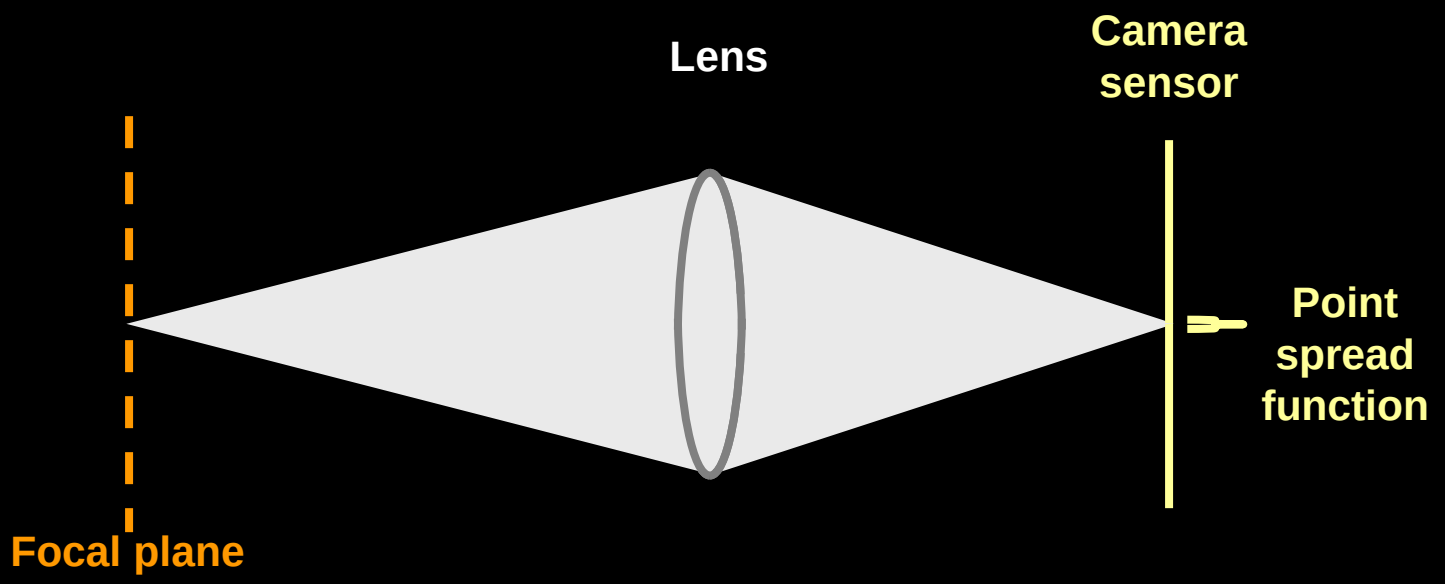


Image of a point light source



Lens and defocus

Lens' aperture

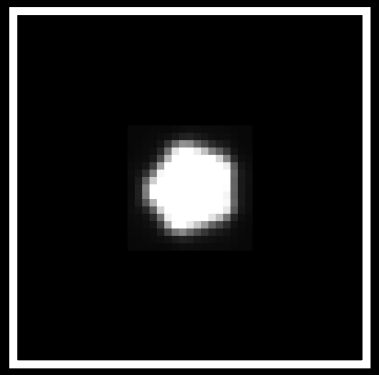
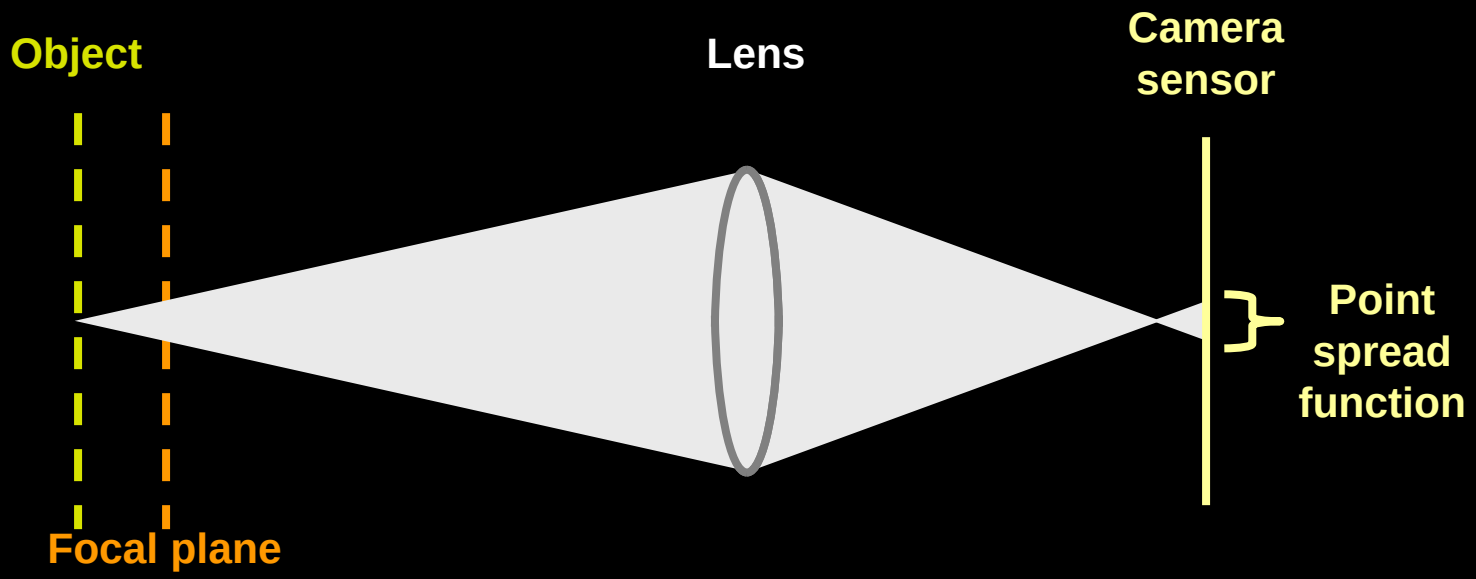


Image of a defocused point light source



Lens and defocus

Lens' aperture

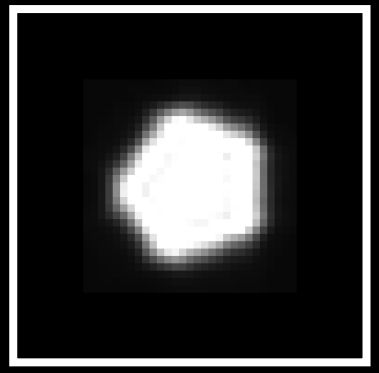
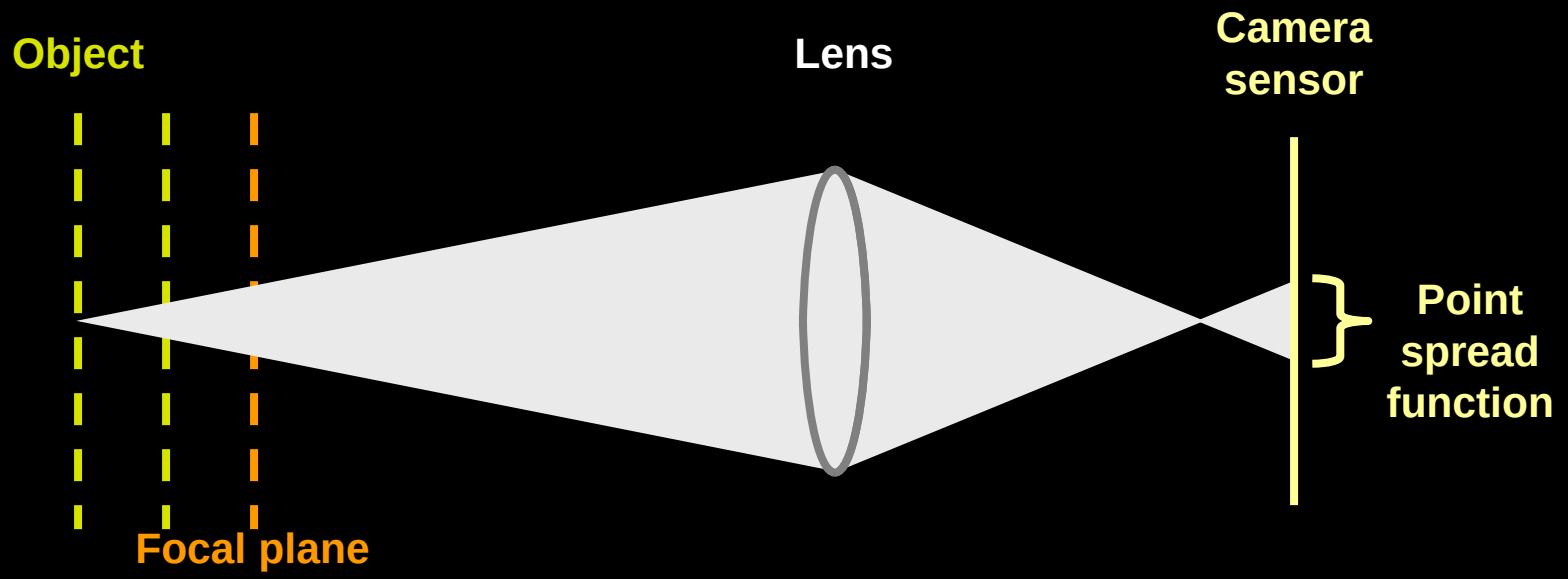


Image of a defocused point light source



Lens and defocus

Lens' aperture

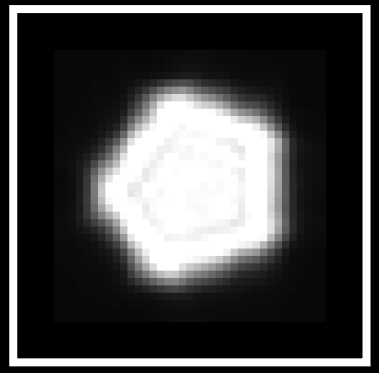
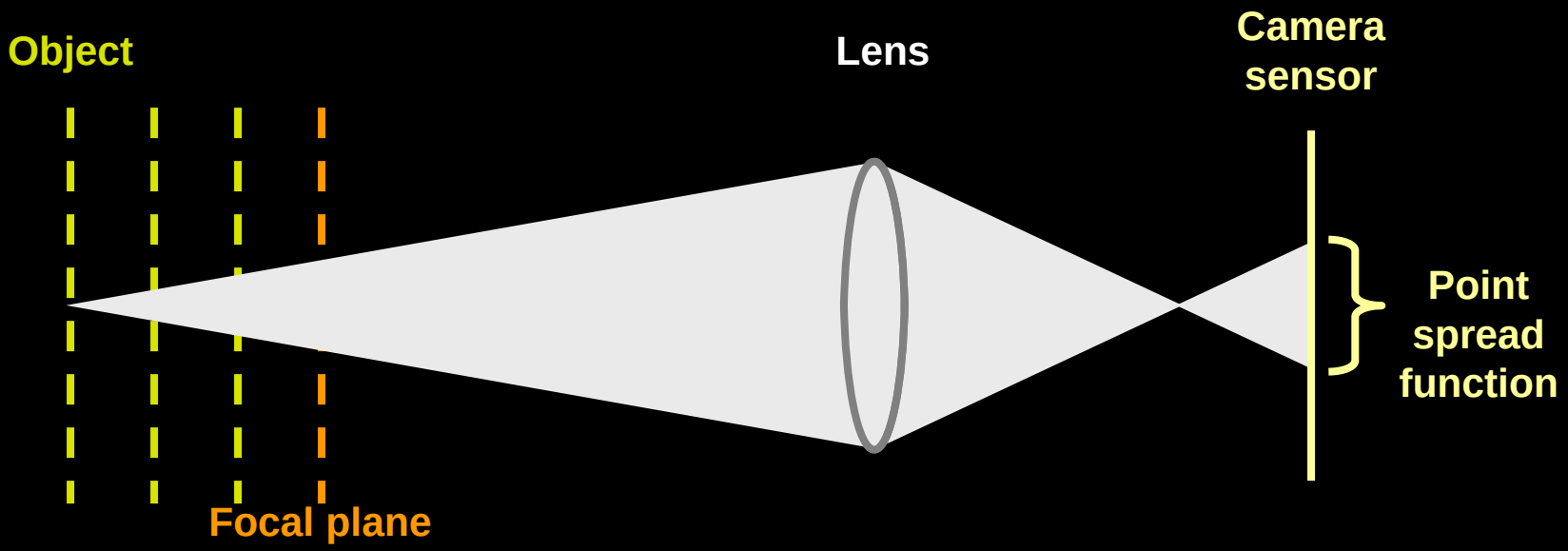


Image of a defocused point light source



Lens and defocus

Lens' aperture

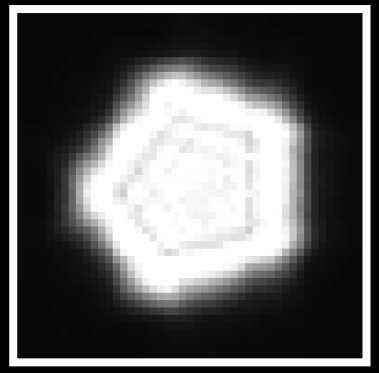
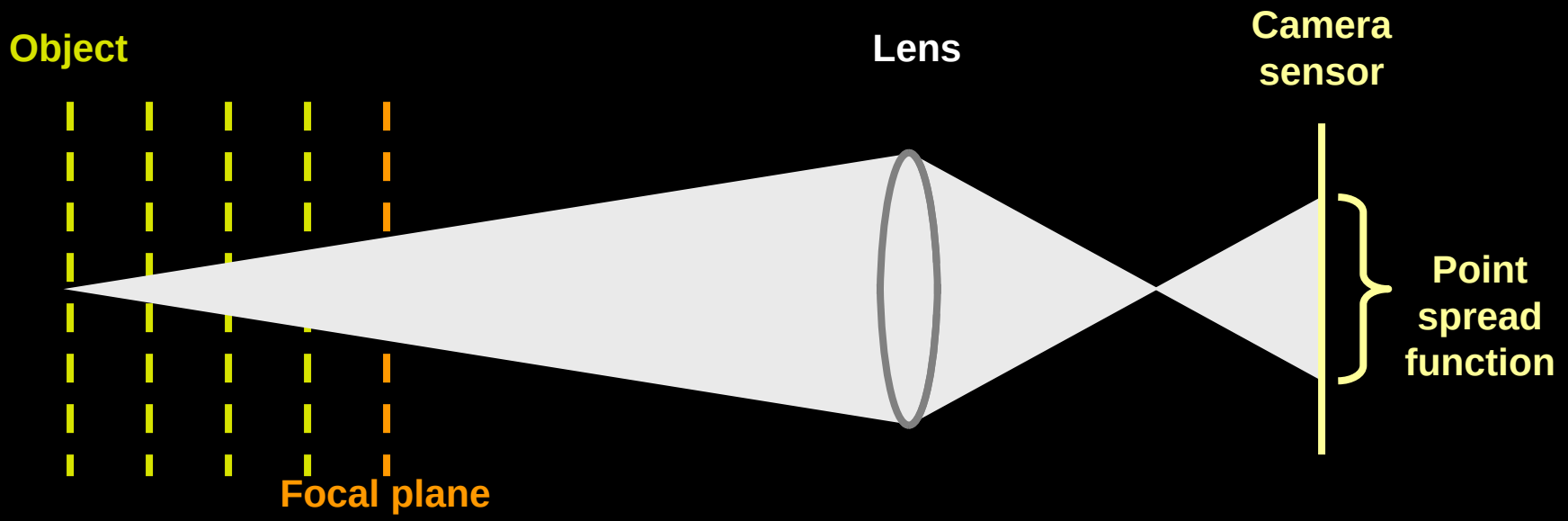
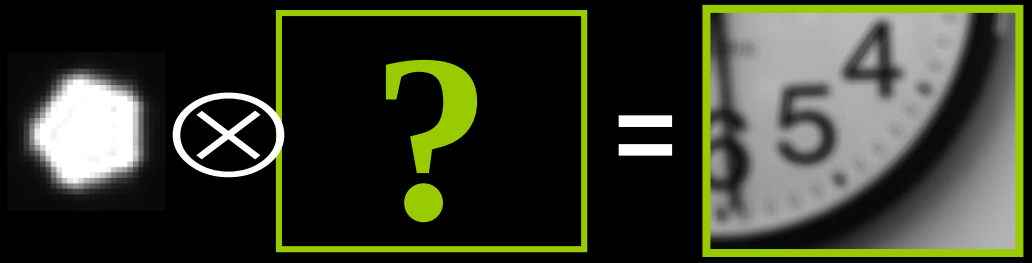


Image of a defocused point light source



Deconvolution is ill posed

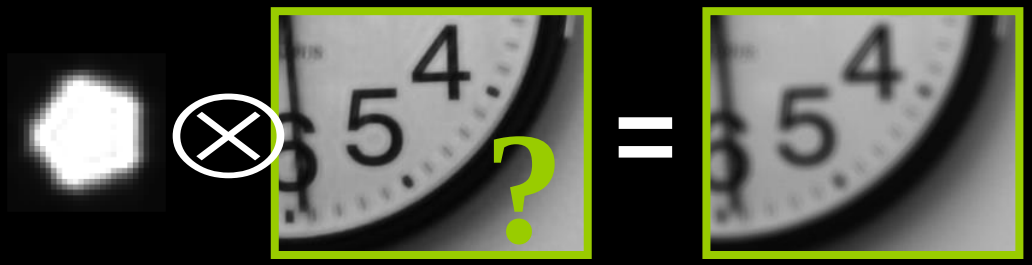
$$f \otimes x = y$$



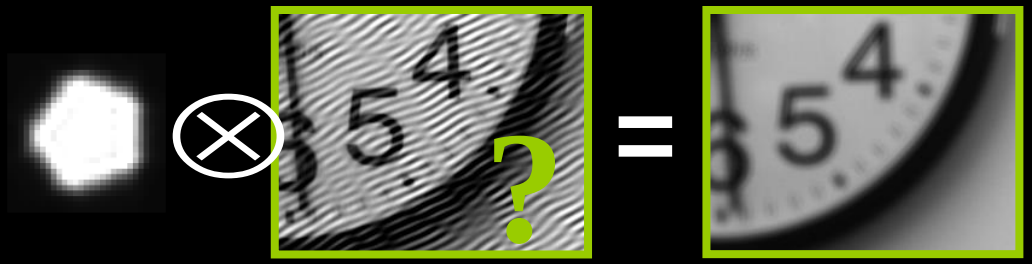
Deconvolution is ill posed

$$f \otimes x = y$$

Solution 1:



Solution 2:



Idea: Natural images prior

What makes images special?

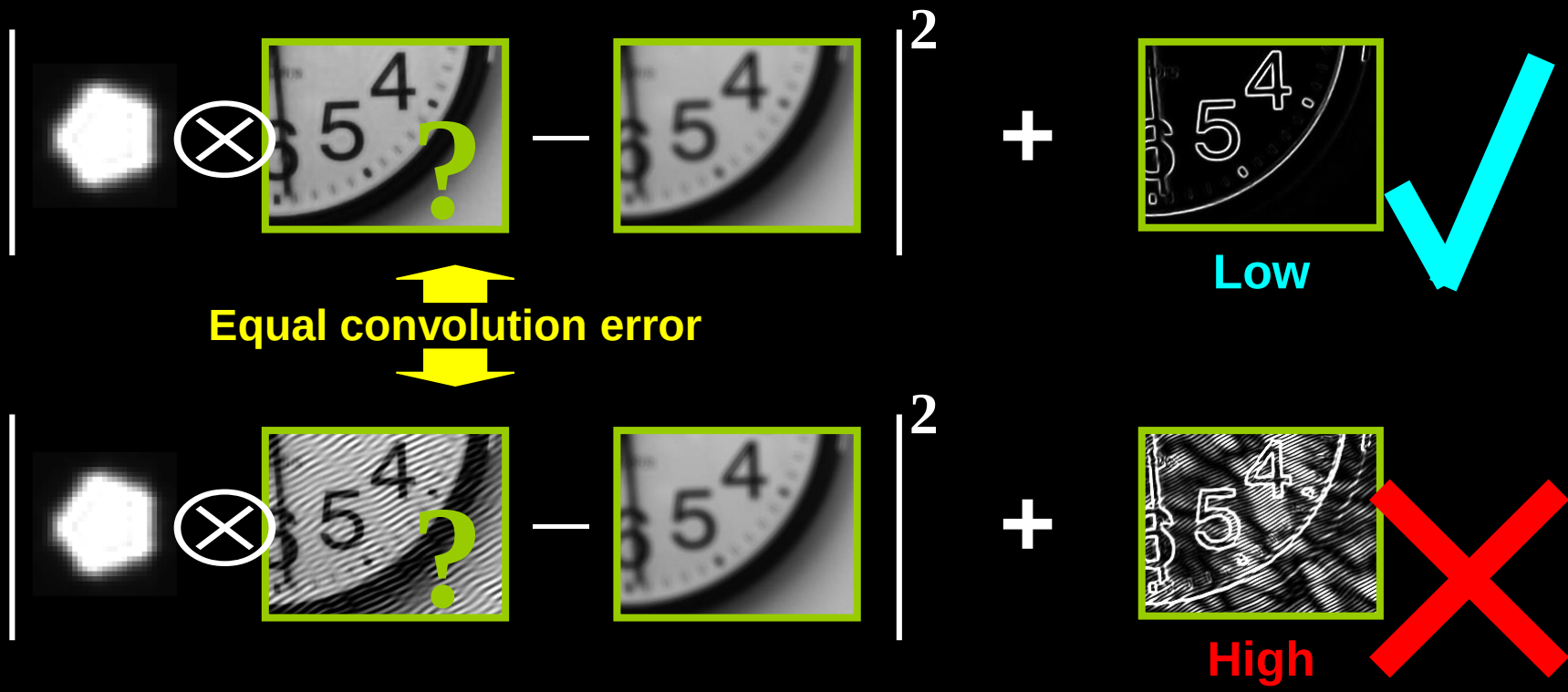
	Natural	Unnatural	
Image			
gradient			

Natural images have sparse gradients

➡ put a penalty on gradients

Deconvolution with prior

$$x = \arg\min \underbrace{\left| f \otimes x - y \right|^2}_{\text{Convolution error}} + \lambda \underbrace{\sum_i \rho(\nabla x_i)}_{\text{Derivatives prior}}$$



Comparing deconvolution algorithms

(Non blind) deconvolution code available online:
http://groups.csail.mit.edu/graphics/CodedAperture/

Input

$\rho(\nabla x) = \|\nabla x\|^2$

"spread" gradients

$\rho(\nabla x) = \|\nabla x\|^{0.8}$

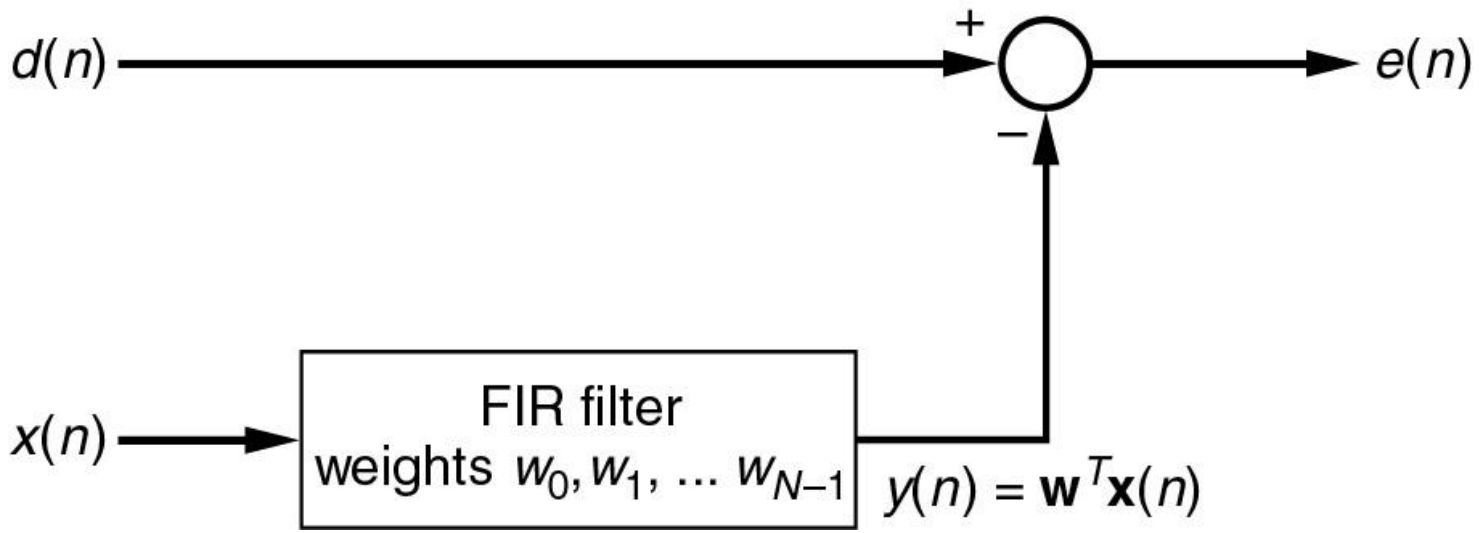
"localizes" gradients

Richardson-Lucy

Gaussian prior

Sparse prior

维纳滤波



Parameters

x(n) = input of FIR filter
y(n) = output of FIR filter
d(n) = desired response
e(n)= d(n)-y(n) = estimation error

FIR filter:

$y(n) = w_0x(n)+w_1x(n-1)+\ldots+w_{N-1}x(n-N+1)$
$= \boldsymbol{W}^T\boldsymbol{X}(n)$

$\boldsymbol{W}^T=[\ w_0\ w_1\ \ldots\ w_{N-1}\ ]$

$\boldsymbol{X}(n)=[x(n)\ x(n-1)\ldots,x(n-N+1)]$

维纳滤波

目标：找到最佳的滤波器参数使得均方误差 $E\{e^2(n)\}$最小

定义目标函数

$$\begin{aligned} J(\mathbf{w}) &= E\{e^2(n)\} \\ &= E\{[d(n) - y(n)]^2\} \\ &= E\{d^2(n) - 2d(n)y(n) + y^2(n)\}. \end{aligned}$$

将滤波器定义代入，可得

$$\begin{aligned} J(\mathbf{w}) &= E\{d^2(n) - 2d(n)y(n) + y^2(n)\} \\ &= E\{d^2(n)\} - 2E\{d(n)\mathbf{w}^T\mathbf{x}(n)\} + E\{\mathbf{w}^T\mathbf{x}(n)\mathbf{x}^T(n)\mathbf{w}\}. \end{aligned}$$

维纳滤波

$$J(\mathbf{w}) = E\{d^2(n) - 2d(n)y(n) + y^2(n)\}$$
$$= E\{d^2(n)\} - 2E\{d(n)\mathbf{w}^T\mathbf{x}(n)\} + E\{\mathbf{w}^T\mathbf{x}(n)\mathbf{x}^T(n)\mathbf{w}\}.$$

因为W而是滤波器参数而不是随机变量，所以上式可以简化为：

$$J(\mathbf{w}) = E\{d^2(n)\} - 2\mathbf{w}^T E\{d(n)\mathbf{x}(n)\} + \mathbf{w}^T E\{\mathbf{x}(n)\mathbf{x}^T(n)\}\mathbf{w}.$$

定义 P = E{d(n) X(n)} 为互相关矩阵， R = E{X(n) X^T (n) 为自相关矩阵，则有

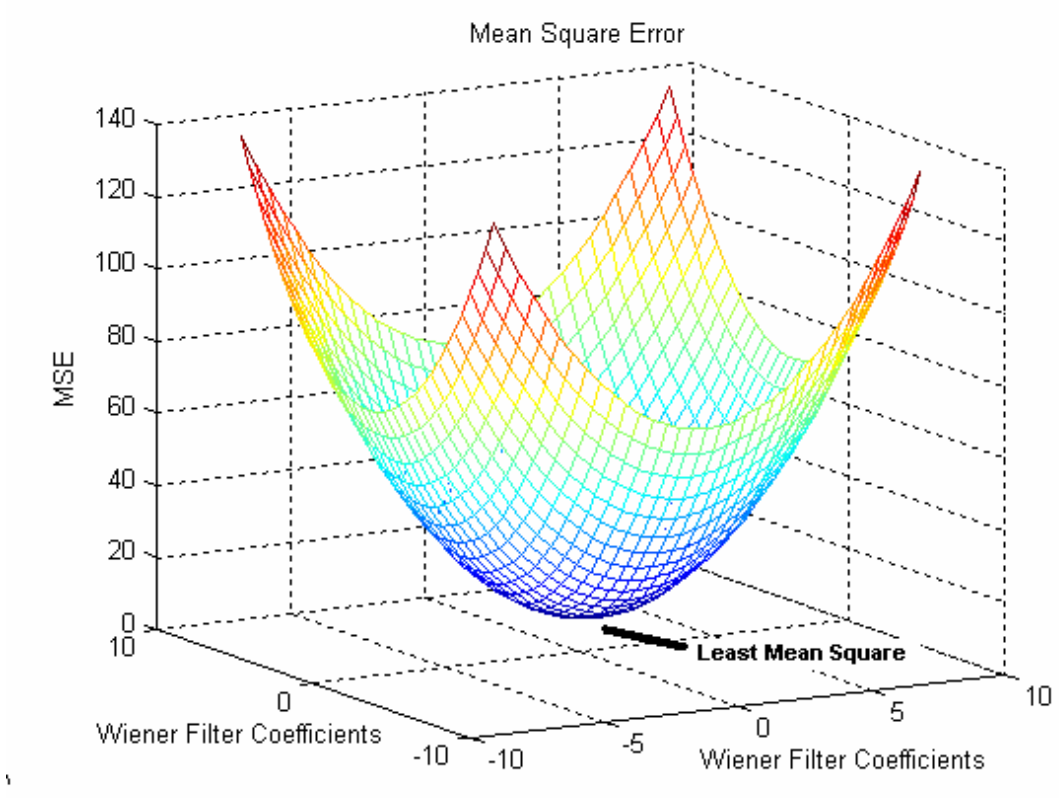
$$J(\mathbf{w}) = \sigma_d^2 - 2\mathbf{w}^T\mathbf{p} + \mathbf{w}^T\mathbf{R}\mathbf{w}.$$

维纳滤波

以二元维纳滤波为例，代价函数为：

$$
\begin{aligned}
J(w_0, w_1) &= E\left\{d^2(n)\right\} - 2E\left\{d(n)y(n)\right\} + E\left\{y^2(n)\right\} \\
&= \sigma_d^2 - 2E\left\{d(n)[w_0 x(n) + w_1 x(n-1)]\right\} + E\left\{[w_0 x(n) + w_1 x(n-1)]^2\right\} \\
&= \sigma_d^2 - 2w_0 E\left\{d(n)x(n)\right\} - 2w_1 E\left\{d(n)x(n-1)\right\} \\
&\quad + w_0^2 E\left\{x^2(n)\right\} + 2w_0 w_1 E\left\{x(n)x(n-1)\right\} + w_1^2 E\left\{x^2(n-1)\right\} \\
&= \sigma_d^2 - 2w_0 p(0) - 2w_1 p(1) + w_0^2 r(0) + 2w_0 w_1 r(1) + w_1^2 r(0).
\end{aligned}
$$

二次型代价函数


Mean Square Error
MSE
140
120
100
80
60
40
20
0
10
0
-10
Wiener Filter Coefficients
-10
-5
0
5
10
Wiener Filter Coefficients
Least Mean Square

维纳滤波

对代价函数分别求两个权重系数的梯度

$$\frac{\partial J}{\partial w_0} = -2p(0) + 2w_0 r(0) + 2w_1 r(1)$$

$$\frac{\partial J}{\partial w_1} = -2p(1) + 2w_0 r(1) + 2w_1 r(0).$$

将梯度写成矩阵形式：

$$\nabla \mathbf{J}(w_0, w_1) = \begin{bmatrix} \frac{\partial J}{\partial w_0} \\ \frac{\partial J}{\partial w_1} \end{bmatrix} = -2 \begin{bmatrix} p(0) \\ p(1) \end{bmatrix} + 2 \begin{bmatrix} r(0) & r(1) \\ r(1) & r(0) \end{bmatrix} \begin{bmatrix} w_0 \\ w_1 \end{bmatrix}.$$

维纳滤波

对于通用的N-1元维纳滤波， 可以采用类似策略来计算梯度，

$$\nabla J(\mathbf{w}) = -2\mathbf{p} + 2\mathbf{R}\mathbf{w}.$$

通过计算上式等于0，可以得到维纳滤波的表达式：

$$\mathbf{w}_{opt} = \mathbf{R}^{-1}\mathbf{p}$$

其中， $P = E\{d(n)\, X(n)\}$ 为互相关矩阵， $R = E\{X(n)\, X^T(n)\}$ 为自相关矩阵

维纳滤波

维纳滤波的表达式：

$$\mathbf{w}_{opt} = \mathbf{R}^{-1}\mathbf{p}$$

其中，$P = E\{d(n)\, X(n)\}$ 为互相关矩阵，$R = E\{X(n)\, X^T(n)\}$ 为自相关矩阵

自适应滤波？

性能分析：

- 利用平稳随机过程的相关特性和频谱特性，对混有噪声的信号进行滤波，奠定了最优滤波理论的基础。
- 当输入过程是广义平稳的，且统计特性已知时，能够取得较好的结果。
- 但是，输入过程取决于外界环境的信号和干扰，其统计特性常常是未知的、变化的。

4 最小二乘滤波（LMS Filter）

- 梯度下降算法（Steepest Descent Algorithm）
- 最小二乘算法（Least-mean-square algorithm）
- 性能分析

最小二乘滤波（LMS Filter）

- 由Widrow & Hoff 在1959年提出
- 位列自适应算法的Top10
  - ✔ 算法简单，自适应调整过程速度快
  - ✔ 属于梯度下降算法，可以保证收敛性

最小二乘滤波（LMS Filter）

- LMS filtering分为两个步骤：
  - ✔ Filtering, producing
    1) output signal
    2) estimation error
  - ✔ Adaptive process, automatic adjustment of filter tap weights

最小二乘滤波（LMS Filter）

- LMS filtering分为两个步骤：
  - ✔ Filtering, producing
    1) output signal
    2) estimation error
  - ✔ Adaptive process, automatic adjustment of filter tap weights

最小二乘滤波（LMS Filter）

基于递归机制的参数自适应

设W(n)为 第n次迭代后得到的滤波器参数，则第n+1次迭代的参数估计可表示为：

$$\mathbf{w}(n+1) = \mathbf{w}(n) + \mathbf{g}(n)$$

其中，g(n) 为方向向量，其作用是最小化目标函数J(W(n+1)) =J(W(n)+g(n)).

为实现目标函数的最小化，可以对J(W(n+1))进行泰勒级数展开。

最小二乘滤波（LMS Filter）

J(W(n+1)) 的泰勒级数展开为：

$$J(\mathbf{w}+\mathbf{g}) = J(\mathbf{w}) + \sum_i g_i \frac{\partial J(\mathbf{w})}{\partial w_i} + \frac{1}{2}\sum_i \sum_j g_i g_j \frac{\partial^2 J(\mathbf{w})}{\partial w_i \partial w_j} + \cdots$$

以二元滤波器为例，

$$J(\mathbf{w}+\mathbf{g}) = J(\mathbf{w}) + g_0 \frac{\partial J(\mathbf{w})}{\partial w_0} + g_1 \frac{\partial J(\mathbf{w})}{\partial w_1} + \frac{1}{2}[g_0^2 \frac{\partial^2 J(\mathbf{w})}{\partial w_0 \partial w_0} + g_0 g_1 \frac{\partial^2 J(\mathbf{w})}{\partial w_0 \partial w_1} +$$

$$g_0 g_1 \frac{\partial^2 J(\mathbf{w})}{\partial w_1 \partial w_0} + g_1^2 \frac{\partial^2 J(\mathbf{w})}{\partial w_1 \partial w_1}].$$

最小二乘滤波（LMS Filter）

对上式分别对$\mathbf{g}_0$ 和 $\mathbf{g}_1$ 求梯度，可得

$$\frac{\partial J(\mathbf{w}+\mathbf{g})}{\partial g_0} = \frac{\partial J(\mathbf{w})}{\partial w_0} + g_0 \frac{\partial^2 J(\mathbf{w})}{\partial w_0 \partial w_0} + g_1 \frac{\partial^2 J(\mathbf{w})}{\partial w_0 \partial w_1}$$

$$\frac{\partial J(\mathbf{w}+\mathbf{g})}{\partial g_1} = \frac{\partial J(\mathbf{w})}{\partial w_1} + g_0 \frac{\partial^2 J(\mathbf{w})}{\partial w_0 \partial w_1} + g_1 \frac{\partial^2 J(\mathbf{w})}{\partial w_1 \partial w_1}.$$

其矩阵形式为：

$$\frac{\partial J(\mathbf{w}+\mathbf{g})}{\partial \mathbf{g}} = \nabla \mathbf{J} + \begin{bmatrix} \frac{\partial^2 J(\mathbf{w})}{\partial w_0 \partial w_0} & \frac{\partial^2 J(\mathbf{w})}{\partial w_0 \partial w_1} \\ \frac{\partial^2 J(\mathbf{w})}{\partial w_0 \partial w_1} & \frac{\partial^2 J(\mathbf{w})}{\partial w_1 \partial w_1} \end{bmatrix} \begin{bmatrix} g_0 \\ g_1 \end{bmatrix}.$$

Hessian Matrix

最小二乘滤波（LMS Filter）

上式中的矩阵叫做Hessian Matrix，表示为 *H*.

为实现J(W+g)的最小化，令上式等于 0，可以得到：

$$\mathbf{g} = -\mathbf{H}^{-1}\nabla\mathbf{J}.$$

可得到牛顿迭代法（一种梯度下降策略）的表达式：

$$\mathbf{w}(n+1) = \mathbf{w}(n) - \mathbf{H}^{-1}\nabla\mathbf{J}.$$

最小二乘滤波（LMS Filter）

将上式中的Hessian矩阵近似为 H=2I，则可以得到最速下降法的通用表达式：

$$\mathbf{w}(n+1) = \mathbf{w}(n) - \frac{\mu}{2}\nabla \mathbf{J}.$$

这里, 常数μ 用于控制收敛速度。

The al arized in the

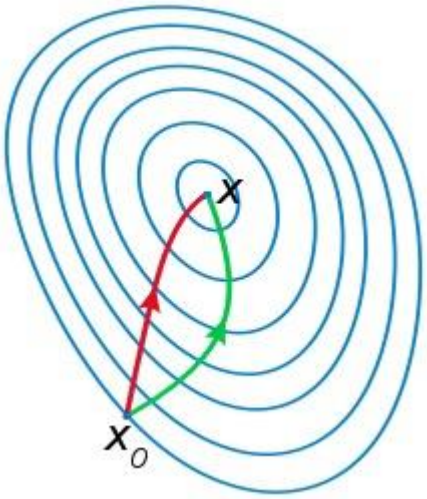


TABLE 7.1 Steepest Descent Algorithm

Initialization

$$\mathbf{w}(0) = \mathbf{0}$$

Algorithm

For $n = 0, 1, 2, \ldots$

$$\mathbf{w}(n+1) = \mathbf{w}(n) + \mu[\mathbf{p} - \mathbf{R}\mathbf{w}(n)]$$

最小二乘滤波（LMS Filter）

- 将维纳滤波器的表达式代入到最速下降滤波中，可得:

$$\mathbf{w}(n+1) = \mathbf{w}(n) + \frac{1}{2}\mu[-\nabla J(n)]$$

$$= \mathbf{w}(n) + \mu[\mathbf{p} - \mathbf{R}\mathbf{w}(n)]$$

- 其中，

$$\hat{\mathbf{R}}(n) = \mathbf{u}(n)\mathbf{u}^{\mathrm{H}}(n), \qquad \hat{\mathbf{p}}(n) = \mathbf{u}(n)d^{*}(n)$$

- 上式也有其它的表现形式，比如使用瞬态误差 $|e(n)|^2$

最小二乘滤波（LMS Filter）

- 将瞬态误差 $|e(n)|^2$ 代入到最速下降滤波中，

$$\begin{aligned}\mathbf{w}(n+1) &= \mathbf{w}(n) + \mu\left[\mathbf{p} - \mathbf{R}\mathbf{w}(n)\right] \\ &\approx \mathbf{w}(n) + \mu\left[\mathbf{u}(n)d^*(n) - \mathbf{u}(n)\underbrace{\mathbf{u}^{\mathrm{H}}(n)\mathbf{w}(n)}_{y^*(n)}\right] \\ &= \mathbf{w}(n) + \mu\mathbf{u}(n)\left[d^*(n) - y^*(n)\right] \\ &= \mathbf{w}(n) + \mu\mathbf{u}(n)e^*(n)\end{aligned}$$

滤波器输出：$\mathbf{y(n)} = \boldsymbol{W}^{\mathrm{T}}\mathbf{u(n)}$

最小二乘滤波（LMS Filter）

- 由于只能得到关于***R***和***p***的粗略估计，

  每一步的自适应步长是随机的。

- 算法中所采用的递归机制，相当于在参数的自适应调整过程中对于每一次的粗略估计进行了有效的平均处理，提高了整体算法的有效性。

Solution for ill-conditioned regression

- 假设我们有个方程组AX=b，我们需要求解X。如果A或者b稍微的改变，会使得X的解发生很大的改变，那么这个方程组系统就是ill-condition的。

equations	solution	equations	solution
$\begin{bmatrix}1 & 2\\ 2 & 3.999\end{bmatrix}\begin{bmatrix}x\\ y\end{bmatrix}=\begin{bmatrix}4\\ 7.999\end{bmatrix}$	$\begin{bmatrix}x\\ y\end{bmatrix}=\begin{bmatrix}2\\ 1\end{bmatrix}$	$\begin{bmatrix}1 & 2\\ 2 & 3\end{bmatrix}\begin{bmatrix}x\\ y\end{bmatrix}=\begin{bmatrix}4\\ 7\end{bmatrix}$	$\begin{bmatrix}x\\ y\end{bmatrix}=\begin{bmatrix}2\\ 1\end{bmatrix}$
$\begin{bmatrix}1 & 2\\ 2 & 3.999\end{bmatrix}\begin{bmatrix}x\\ y\end{bmatrix}=\begin{bmatrix}4.001\\ 7.998\end{bmatrix}$	$\begin{bmatrix}x\\ y\end{bmatrix}=\begin{bmatrix}-3.999\\ 4.000\end{bmatrix}$	$\begin{bmatrix}1 & 2\\ 2 & 3\end{bmatrix}\begin{bmatrix}x\\ y\end{bmatrix}=\begin{bmatrix}4.001\\ 7.001\end{bmatrix}$	$\begin{bmatrix}x\\ y\end{bmatrix}=\begin{bmatrix}1.999\\ 1.001\end{bmatrix}$
$\begin{bmatrix}1.001 & 2.001\\ 2.001 & 3.998\end{bmatrix}\begin{bmatrix}x\\ y\end{bmatrix}=\begin{bmatrix}4\\ 7.999\end{bmatrix}$	$\begin{bmatrix}x\\ y\end{bmatrix}=\begin{bmatrix}3.994\\ 0.001388\end{bmatrix}$	$\begin{bmatrix}1.001 & 2.001\\ 2.001 & 3.001\end{bmatrix}\begin{bmatrix}x\\ y\end{bmatrix}=\begin{bmatrix}4\\ 7\end{bmatrix}$	$\begin{bmatrix}x\\ y\end{bmatrix}=\begin{bmatrix}2.003\\ 0.997\end{bmatrix}$

What is ill-condition?

- 线性系统 Ax = b 为什么会病态？归根到底是由于 A 矩阵列向量线性相关性过大，表示的特征太过于相似以至于容易混淆所产生的。例如：

$$A = \begin{pmatrix} 1000 & 1000 \\ 0 & 0.001 \end{pmatrix}$$

- 在二维空间上，这两个列向量夹角非常小。假设第一次检测得到数据 b = [1000, 0]^T, 这个点正好在第一个列向量所在的直线上，解集是 [1, 0]^T。
- 现在再次检测，由于有轻微的误差，得到的检测数据是 b = [1000, 0.001]， 这个点正好在第二个列向量所在的直线上，解集是 [0, 1]^T。两次求解得到了差别迥异的解集。

What is ill-condition?

- 假设 A 的两个单位特征向量是 x1, x2, 根据特征向量的性质：

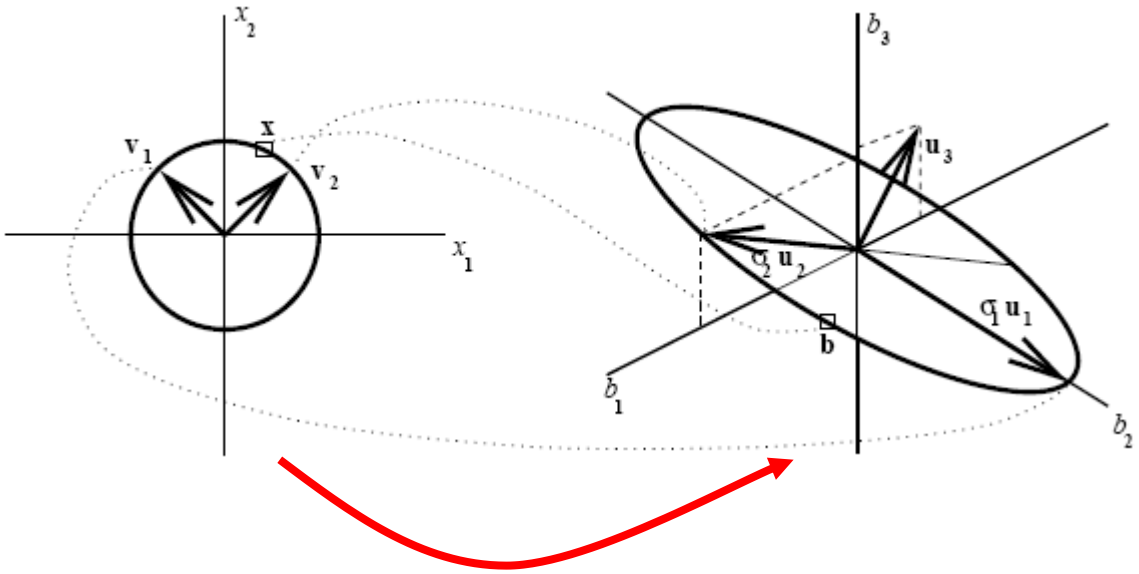
$$Ax_1 = \lambda_1 x_1, Ax_2 = \lambda_2 x_2$$

- 对于平面上的某一个向量 b，可以分解为两个特征向量的线性组合：

$$b = mx_1 + nx_2 = \frac{m}{\lambda_1}\lambda_1 x_1 + \frac{n}{\lambda_2}\lambda_2 x_2 = A(\frac{m}{\lambda_1}x_1 + \frac{n}{\lambda_2}x_2) = Ax$$

- 如果 λ1远远大于λ2， 当 b 点在 x1 方向发生移动, m 值改变， 解集 x 变化不明显； 反之， 如果在 x2 方向移动， n 值改变，解集 x 变化非常大！

SVD



$$\mathbf{b} = A\mathbf{x} \qquad A = \frac{1}{\sqrt{2}} \begin{bmatrix} \sqrt{3} & \sqrt{3} \\ -3 & 3 \\ 1 & 1 \end{bmatrix}$$

http://www.uwlax.edu/faculty/will/svd/index.html

SVD

Theorem 3.2.1 *If A is a real $m \times n$ matrix then there exist orthogonal matrices*

$$U = \left[\begin{array}{ccc} \mathbf{u}_1 & \cdots & \mathbf{u}_m \end{array} \right] \in \mathcal{R}^{m \times m}$$
$$V = \left[\begin{array}{ccc} \mathbf{v}_1 & \cdots & \mathbf{v}_n \end{array} \right] \in \mathcal{R}^{n \times n}$$

such that

$$U^T A V = \Sigma = \mathrm{diag}(\sigma_1, \ldots, \sigma_p) \in \mathcal{R}^{m \times n}$$

where $p = \min(m, n)$ *and* $\sigma_1 \geq \ldots \geq \sigma_p \geq 0$. *Equivalently,*

$$A = U \Sigma V^T \ .$$

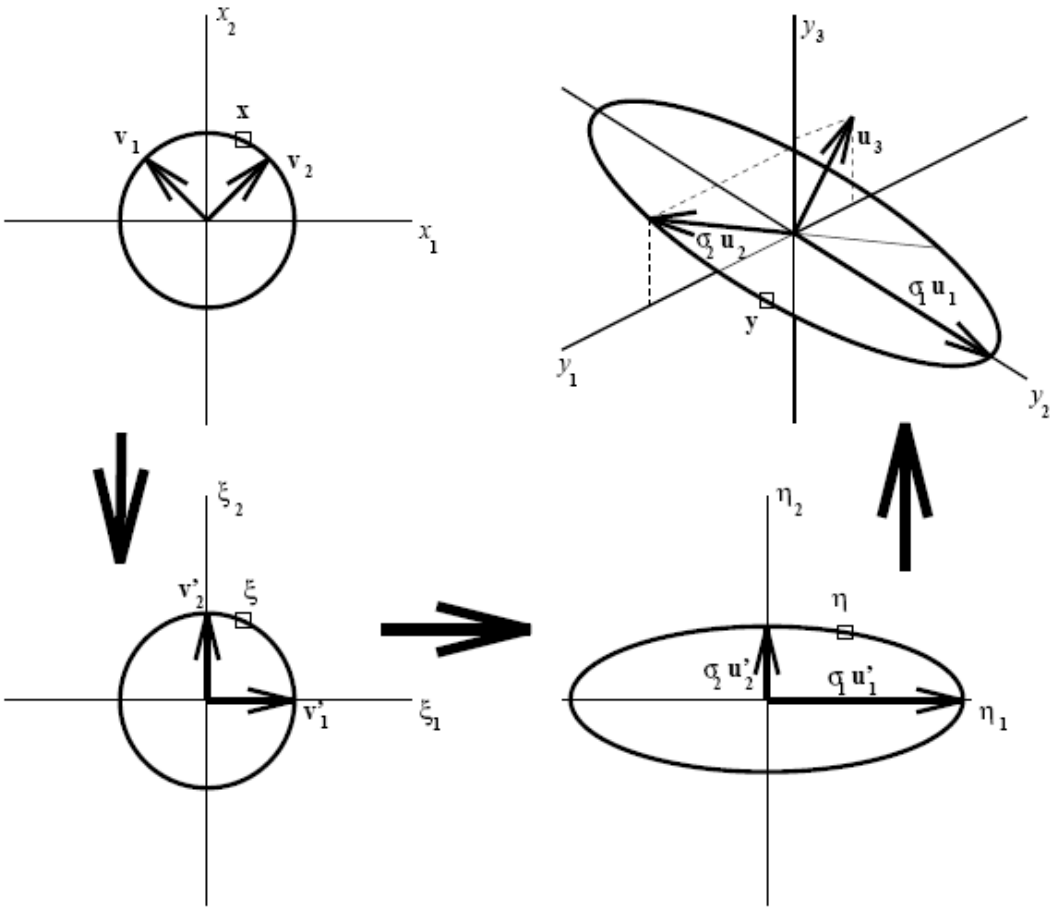
The SVD reveals a great deal about the structure of a matrix. If we define r by

$$\sigma_1 \geq \ldots \geq \sigma_r > \sigma_{r+1} = \ldots = 0 \ ,$$

that is, if σ_r is the smallest nonzero singular value of A, then

$$\mathrm{rank}(A) \quad = \quad r$$

SVD



SVD

Theorem 3.3.1 *The minimum-norm least squares solution to a linear system* $A\mathbf{x} = \mathbf{b}$*, that is, the shortest vector* $\mathbf{x}$ *that achieves the*

$$\min_{\mathbf{x}} \|A\mathbf{x} - \mathbf{b}\| ,$$

is unique, and is given by

$$\hat{\mathbf{x}} = V\Sigma^{\dagger}U^{T}\mathbf{b}$$

where

$$\Sigma^{\dagger} = \begin{bmatrix} 1/\sigma_1 & & & & & 0 & \cdots & 0 \\ & \ddots & & & & & & \\ & & 1/\sigma_r & & & \vdots & & \vdots \\ & & & 0 & & & & \\ & & & & \ddots & & & \\ & & & & & 0 & 0 & \cdots & 0 \end{bmatrix}$$

is an $n \times m$ *diagonal matrix.*

The matrix

$$A^{\dagger} = V\Sigma^{\dagger}U^{T}$$

is called the *pseudoinverse* of A.

Solution for ill-conditioned regression

- 病态矩阵解集的不稳定性是由于解集空间包含了自由度过大的方向，解决这个问题的关键就是将这些方向去掉，而保留 scaling 较大的方向，从而把解集局限在一个较小的区域内。
- A 矩阵的特征向量不一定正交，不适合做新基， SVD 分解正好分解出了正交基，可以选前 k 个 v^T 向量作为正交基。

$$\min_y \|AZ_k y - b\|_2.$$

- 真正的自由是建立在规范的基础上的，在数据项的基础上增加正则项。

线性回归+L1正则项：**Lasso** 回归
线性回归+L2正则项：**Ridge** 回归（岭回归）

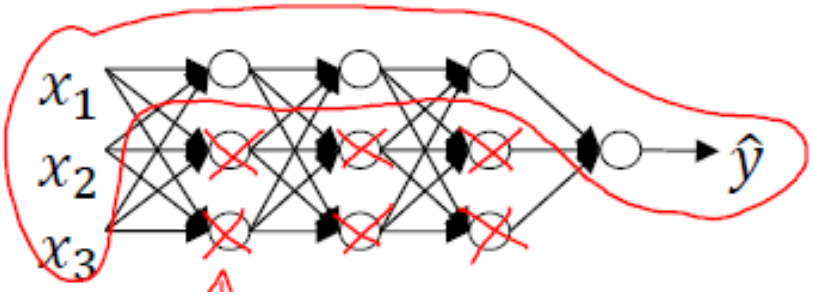
L1&L2 Regularization

考虑如下一般形式的目标函数：

$$w^* = \arg\min_w \sum L(y_i, f(x_i; w)) + \lambda\Omega(w)$$

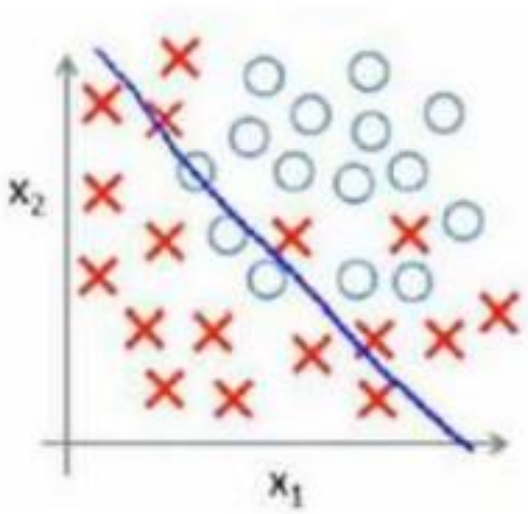
既要让误差（上式第一项）最小，又想让模型尽可能地简单（上式第二项）。

一个朴素的想法：那就让权重W多几个为0（或者接近于0，说明该节点影响很小）不就好了，这样模型就变简单了。



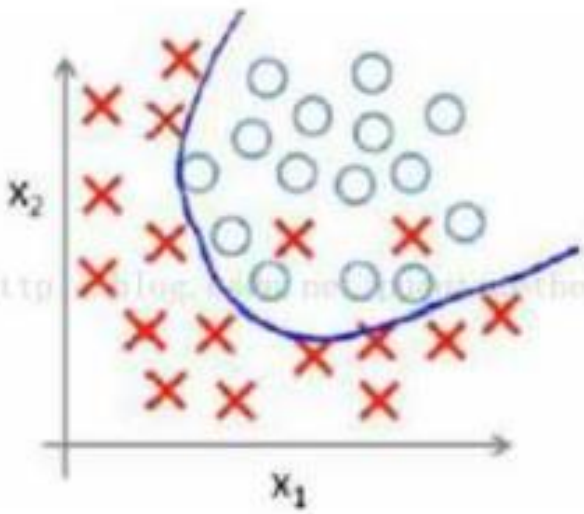
以训练网络模型为例

L1&L2 Regularization

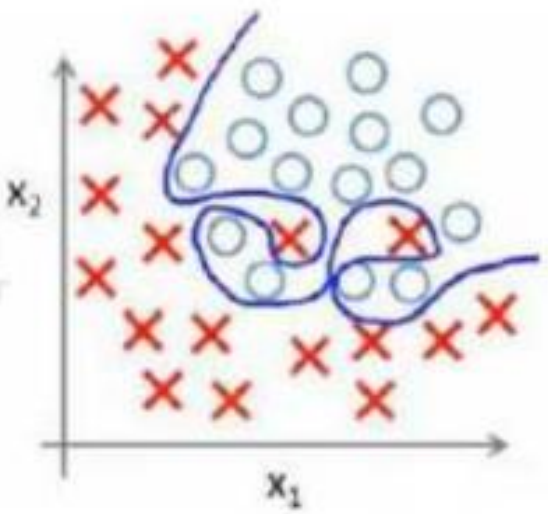


欠拟合

高偏差



恰好



过拟合

高方差

就是为了防止过拟合！！！

L1&L2 Regularization

为了让W多几个为0，基于如下3种范数定义正则化项

- L0范数：$\|w\|_0$ ，指向量中非0的元素的个数
- L1范数：$\|w\|_1$ ，指向量中各个元素绝对值之和
- L2范数：$\|w\|_2$ ，即各元素的平方和再开方

•线性回归+L1正则项：Lasso 回归

•线性回归+L2正则项：Ridge 回归（岭回归）

L1&L2 Regularization

- 如果我们用L0范数来正则化一个参数矩阵W的话，就是希望W的大部分元素都是0，让参数W是稀疏的，“压缩感知”、“稀疏编码”就是通过L0来实现的
- 那为什么用L1去稀疏，而不用L0呢，因为L0范数很难优化求解
- L1范数是L0范数的最优凸近似，而且它比L0范数要容易优化求解

$$\min \|x\|_0 \quad \text{s.t. } Ax = b$$

在一定条件下，以概率1意义下等价

$$\min \|x\|_1 \quad \text{s.t. } Ax = b$$

L1&L2 Regularization

那么L2范数与L1范数有什么区别呢？

1、L2范数更有助于计算病态的问题

2、L1相对于L2能够产生更加稀疏的模型

3、从概率角度进行分析，很多范数约束相当于对参数添加先验分布，其中L2范数相当于参数服从高斯先验分布；L1范数相当于拉普拉斯分布。

L1&L2 Regularization

- L2范数有助于处理病态问题，对于线性回归来说，其最优解为：

$$\hat{\mathbf{w}} = \left(X^T X\right)^{-1} X^T \mathbf{y}$$

- 当我们的样本X的数目比每个样本的维度还要小的时候，矩阵XTX将会不是满秩的，也就是XTX会变得不可逆
- 但如果加上L2正则项，就变成了下面这种情况，就可以直接求逆了：

$$w^* = \left(X^T X + \lambda I\right)^{-1} X^T y$$

L1&L2 Regularization

- 此外，还可以将范数约束看成带有参数的约束优化问题。带有参数惩罚的优化目标为：

$$\tilde{J}(\theta;, X, y) = J(\theta; X, y) + \alpha\Omega(\theta)$$

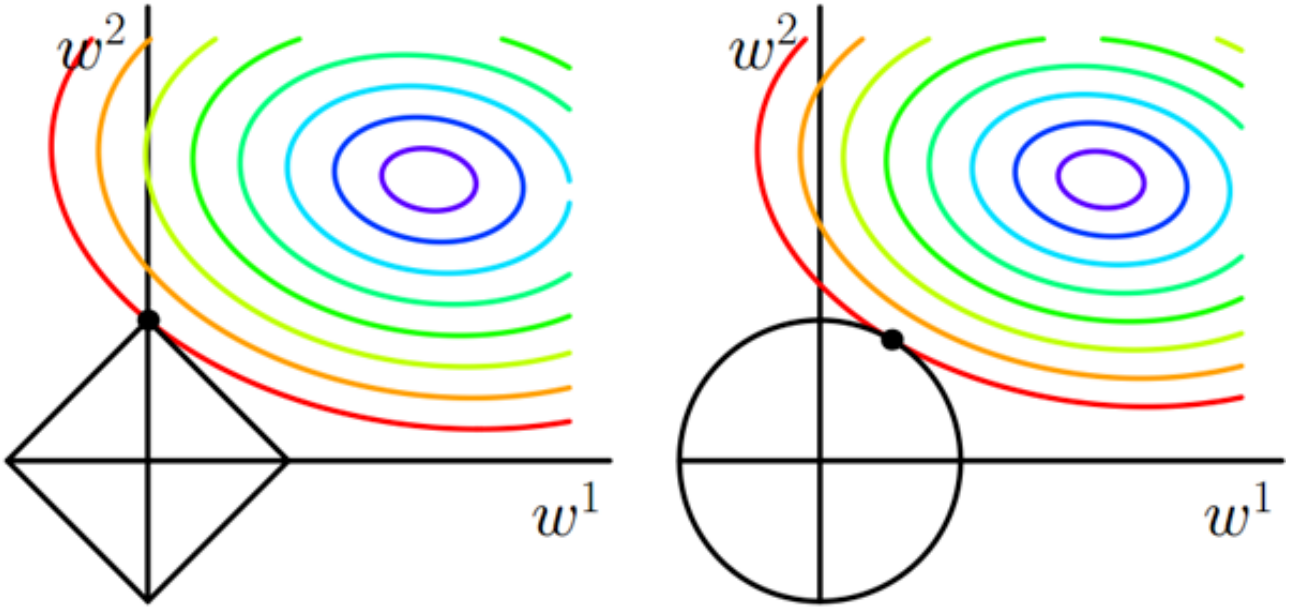
- 带约束的最优问题，可以表示为：

$$\min J(\theta; X, y)$$
$$s.t. \Omega(\theta) < k$$

- 通过KKT条件进行求解时，对应的拉格朗日函数为：

$$L(\theta, \alpha; x, y) = J(\theta; x, y) + \alpha(\Omega(\theta) - k)$$

L1&L2 Regularization

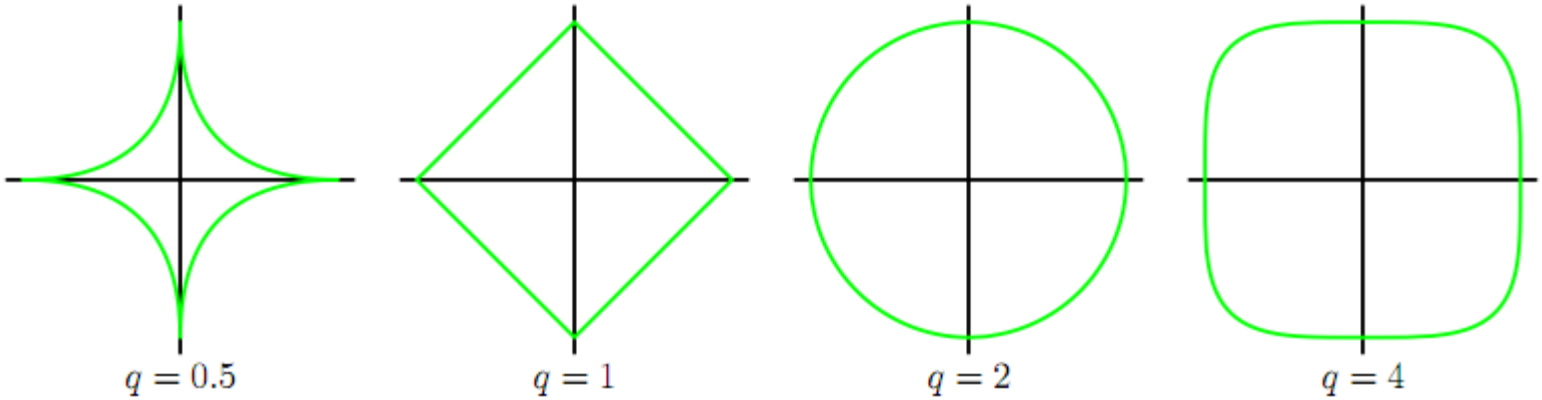


(a) ℓ_1-ball meets quadratic function. ℓ_1-ball has corners. It's very likely that the meet-point is at one of the corners.

(b) ℓ_2-ball meets quadratic function. ℓ_2-ball has no corner. It is very unlikely that the meet-point is on any of axes."

如果 Ω是L2范数，那么权重就是被约束在一个L2球中；如果Ω是L1范数，那么权重就是约束在L1范数限制的区域中；另外也可以得出L1得到的解比L2稀疏

L1&L2 Regularization



如果 Ω是L2范数，那么权重就是被约束在一个L2球中；如果Ω是L1范数，那么权重就是约束在L1范数限制的区域中；另外也可以得出L1得到的解比L2稀疏