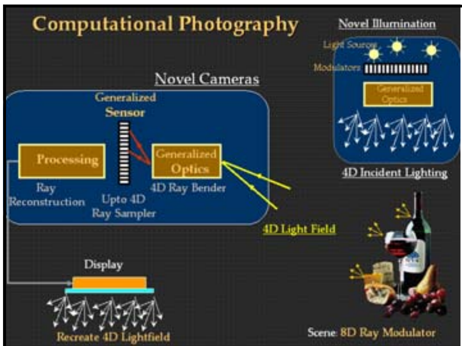
Computational Photography

Eduard Gröller

Most of material and slides courtesy of
Fredo Durand (http://people.csail.mit.edu/fredo/) and
Oliver Deussen (http://graphics.uni-konstanz.de/mitarbeiter/deussen.php)

What is computational photography

- **Convergence of image processing, computer vision, computer graphics and photography**
- **Digital photography:**
  - Simply replaces traditional sensors and recording by digital technology
  - Involves only simple image processing
- **Computational photography**
  - More elaborate image manipulation, more computation
  - New types of media (panorama, 3D, etc.)
  - Camera design that take computation into account



[Raskar and Tumblin]

http://web.media.mit.edu/~raskar/photo/

- Generalized Optics
- Generalized Sensor
- Processing, and
- Generalized Illumination

Computational Photography – Taxonomy (1)

- Computational illumination
  - Flash/no-flash imaging
  - Multi-flash imaging
  - Different exposures imaging
  - Image-based Relighting
  - Other uses of structured illumination
- Computational optics
  - Coded aperture imaging
  - Coded exposure imaging
  - Light field photography
  - Catadioptric imaging
  - Wavefront coding
  - Compressive imaging

Eduard Gröller

Computational Photography – Taxonomy (2)

- Computational processing
  - Panorama mosaicing
  - Matte extraction
  - Digital photomontage
  - High dynamic range imaging
  - All-focus imaging
- Computational sensors
  - Artificial retinas
  - High dynamic range sensors
  - Retinex sensors

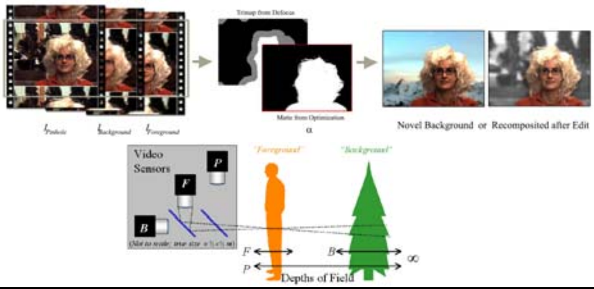
Eduard Gröller

Tone mapping

- **One of your assignments!**

Defocus Matting

- With Morgan McGuire, Wojciech Matusik, Hanspeter Pfister, John "Spike" Hughes
- Data-rich: use 3 streams with different focus

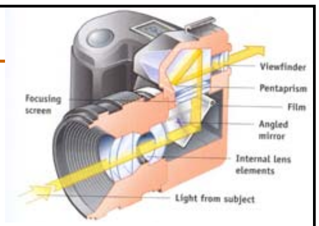


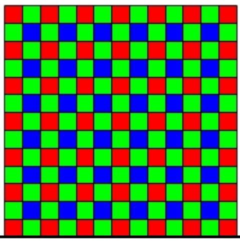
Motion magnification

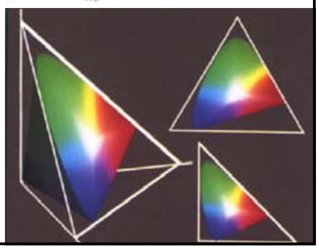
Video

Syllabus

- Image formation
- Color and color perception
- Demosaicing

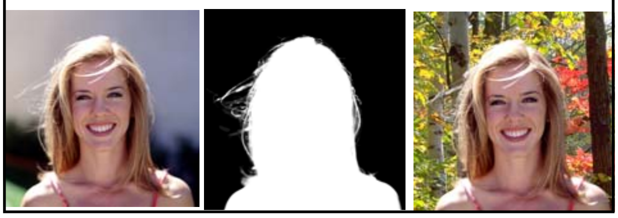






Syllabus

- High Dynamic Range Imaging
- Bilateral filtering and HDR display
- Matting



Syllabus

- Gradient image manipulation

sources/destinations cloning seamless cloning

Syllabus

- Non-parametric image synthesis, inpainting, analogies

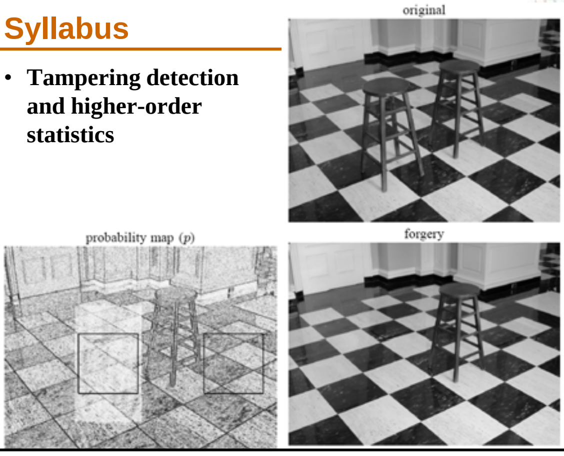
input images quilting results

A A′ B B′

Figure 1 An image analogy. Our problem is to compute a new "analogous" image B′ that relates to B in "the same way" as A′ relates to A. Here, A, A′, and B are inputs to our algorithm, and B′ is the output. The full-size images are shown in Figures 10 and 11.

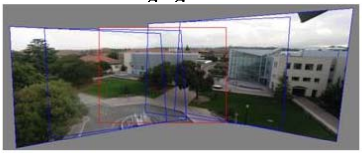
Syllabus

- **Tampering detection and higher-order statistics**

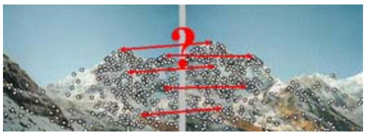


Syllabus

- **Panoramic imaging**



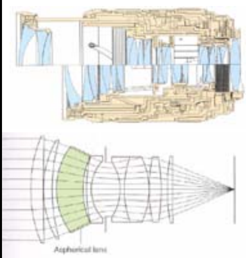
- **Image and video registration**



- **Spatial warping operations**

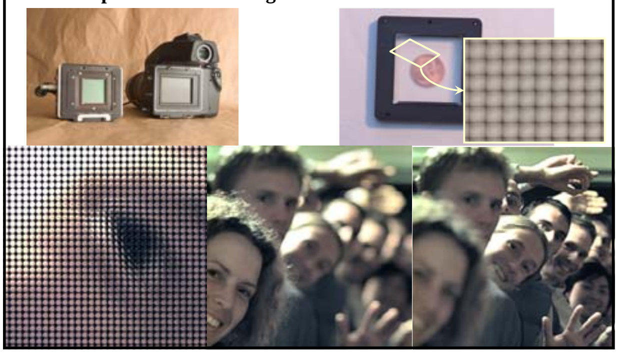
Syllabus

- **Active flash methods**
- **Lens technology**
- **Depth and defocus**



Syllabus

- **Future cameras**
- **Plenoptic function and light fields**



Modern cameras use image stacks (1)

[Deussen et al.]

Modern cameras use image stacks (1)

[Deussen et al.]

Modern cameras use image stacks (1)

[Deussen et al.]

Modern cameras use image stacks (1)

[Deussen et al.]

Modern cameras use image stacks (1)

[Deussen et al.]

Modern cameras use image stacks (2)

[Deussen et al.]

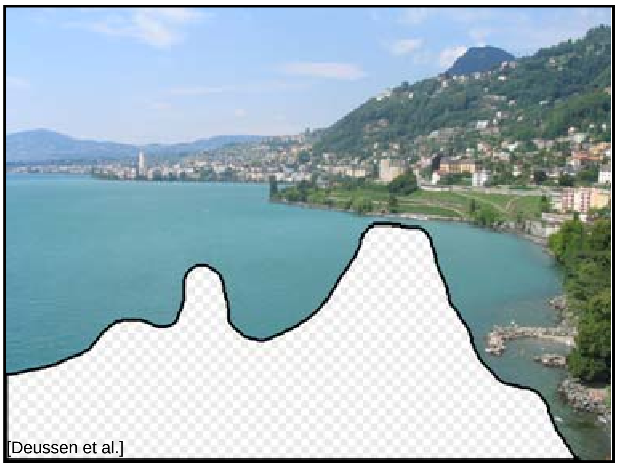
Modern cameras use image stacks (2)

[Deussen et al.]

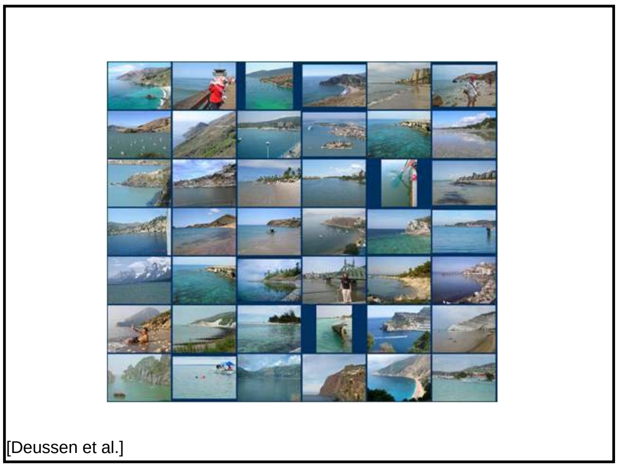
- The web: a completely new source of image information

[Deussen et al.]


[Deussen et al.]


[Deussen et al.]


[Deussen et al.]


[Deussen et al.]

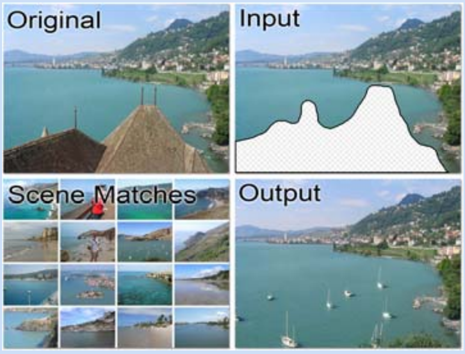

[Deussen et al.]


[Deussen et al.]

[Deussen et al.]

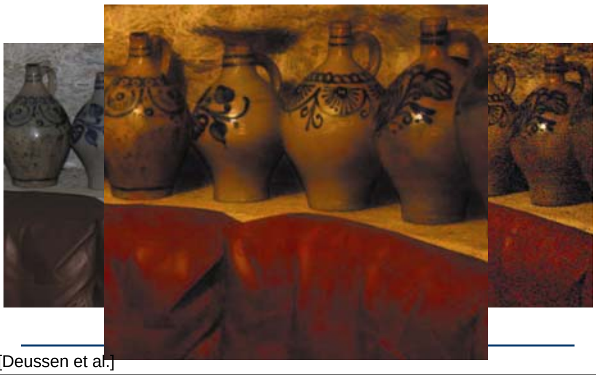
Scene Completion Using Millions of Photographs

Hays and Efros, Siggraph 2007



[Debevec, Raskar, Tumblin]

Flash/ No Flash Photography



[Deussen et al.]

Beautification

[Deussen et al.]

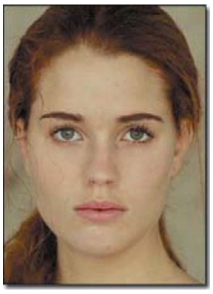
Beautification

[Deussen et al.]

Beautification

[Deussen et al.]

Beautification



[Deussen et al.]

Data-Driven Enhancement of Facial Attractiveness

Tommer Leyvand, Daniel Cohen-Or, Gideon Dror and Dani Lischinski

[Debevec, Raskar, Tumblin]

Gigapixel Images

[Deussen et al.]

Displaying Gigapixel – Tiled Displays

- Form a large display by combining several smaller ones
- Reverse process to stitching large images from smaller ones
- **Two types:**
  - Monitor walls (typically LCD)
  - Multi-projector back-projection systems

100 megapixel wall, Univ. Illinois

10 megapixel wall, Univ. Konstanz

[Deussen et al.]

Tiled Monitor Walls

- **Advantages**
  - Relatively cheap
  - Scalable in size
  - No calibration
- **Problems**
  - Clearly visible borders (mullions)
  - Compensate for borders
  - No multi-user stereo

HIPerSpace OptIPortal (UC San Diego): 220 million pixels (55 screens)

[Deussen et al.]

Capturing Gigapixel Images

[Deussen et al.]

Example: University of Konstanz / MSR

3,600,000,000 Pixels

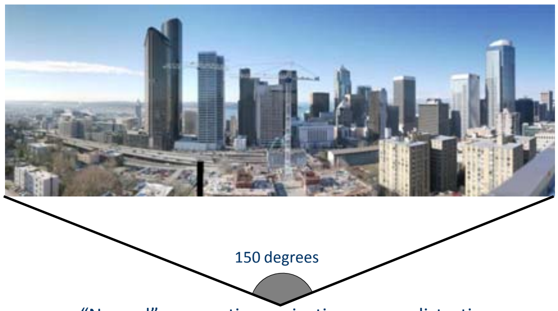
Created from about 800 8 MegaPixel Images

[Deussen et al.]

Example: University of Konstanz / MSR

[Deussen et al.]

Example: University of Konstanz / MSR

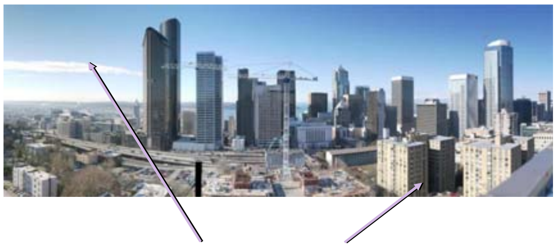


150 degrees

“Normal” perspective projections cause distortions.

[Deussen et al.]

Example: University of Konstanz / MSR



100X variation in Radiance

High Dynamic Range

[Deussen et al.]

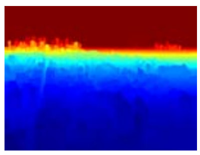
Deep Photo

- Can we change an normal outdoor photography after the shot?
- Yes, especially when depth information is available

[Deussen et al.]

Our Approach

Input Photo



Depth Map

Camara with GPS

Virtual Earth data

[Deussen et al.]

Applications

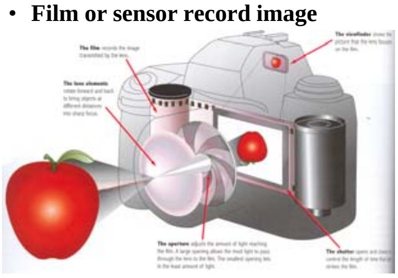
- Image Enhancement
  - Remove haze
  - Relighting
- Novel View synthesis
  - Expanding the FOV
  - Change view point
- Information visualization
  - Integration of GIS data

[Deussen et al.]

Photography - Basics

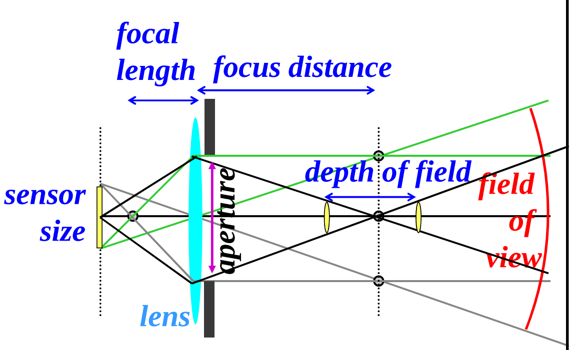
Overview

- **Lens and viewpoint determine perspective**
- **Aperture and shutter speed determine exposure**
- **Aperture and other effects determine depth of field**
- **Film or sensor record image**



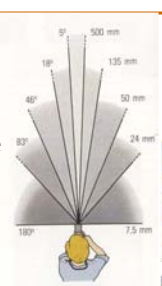
- **Focal length (in mm)**
  - Determines the field of view.
    wide angle (<30mm) to telephoto (>100mm)
- **Focusing distance**
  - Which distance in the scene is sharp
- **Depth of field**
  - Given tolerance, zone around the focus distance that is sharp
- **Aperture (in f number)**
  - Ratio of used diameter and focal lens.
    Number under the divider ➔ small number = large aperture
    (e.g. f/2.8 is a large aperture, f/16 is a small aperture)
- **Shutter speed (in fraction of a second)**
  - Reciprocity relates shutter speed and aperture
- **Sensitivity (in ISO)**
  - Linear effect on exposure
  - 100 ISO is for bright scenes, ISO 1600 is for dark scenes

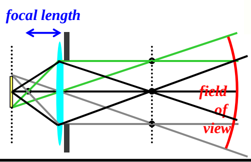
Quantities



Focal length

<30mm: wide angle
50mm: standard
>100mm telephoto
Affected by sensor size (crop factor)





Exposure

- **Aperture (f number)**
  - Expressed as ratio between focal length and aperture diameter: diameter = f / <f number>
  - f/2.0, f/2.8, f/4.0, f/5.6, f/8.0, f/11, f/16 (factor of sqrt (2))
  - Small f number means large aperture
  - Main effect: depth of field
  - A good standard lens has max aperture f/1.8. A cheap zoom has max aperture f/3.5
- **Shutter speed**
  - In fraction of a second
  - 1/30, 1/60, 1/125, 1/250, 1/500 (factor of 2)
  - Main effect: motion blur
  - A human can usually hand-hold up to 1/f seconds, where f is focal length
- **Sensitivity**
  - Gain applied to sensor
  - In ISO, bigger number, more sensitive (100, 200, 400, 800, 1600)
  - Main effect: sensor noise

Reciprocity between these three numbers: for a given exposure, one has two degrees of freedom.

Depth of field

- **The bigger the aperture (small f number), the shallower the DoF**
  - Just think Gaussian blur: bigger kernel → more blurry
  - This is the advantage of lenses with large maximal aperture: they can blur the background more
- **The closer the focus, the smaller the DoF**
- **Focal length has a more complex effect on DoF**
  - Distant background more blurry with telephoto
  - Near the focus plane, depth of field only depends on image size
- **Hyperfocal distance:**
  - Closest focusing distance for which the depth of field includes infinity
  - The largest depth of field one can achieve.
  - Depends on aperture.

What is an image?

- We can think of an **image** as a function, *f*,
- from R^2 to R:
  - *f*(*x, y*) gives the **intensity** at position (*x, y*)
  - Realistically, we expect the image only to be defined over a rectangle, with a finite range:
    - $f: [a,b]\times[c,d] \rightarrow [0,1]$
- A color image is just three functions pasted together. We can write this as a “vector-valued” function:

$$f(x,y) = \begin{bmatrix} r(x,y) \\ g(x,y) \\ b(x,y) \end{bmatrix}$$

Images as functions

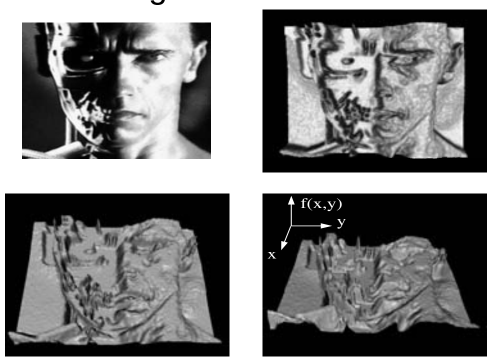
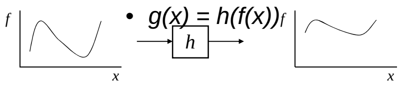


Image Processing

- image filtering: change ***range*** of image
  - *g(x) = h(f(x))*



- image warping: change ***domain*** of image
  - *g(x) = f(h(x))*

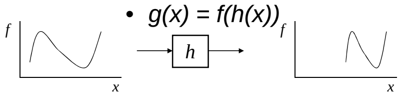
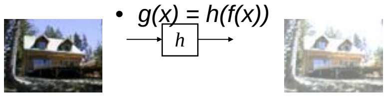
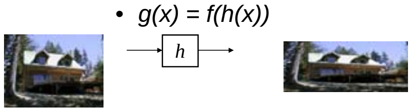


Image Processing

- image filtering: change ***range*** of image
  - *g(x) = h(f(x))*



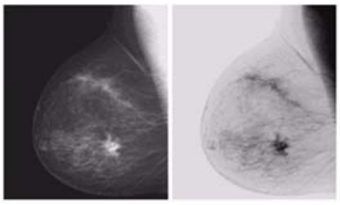
- image warping: change ***domain*** of image
  - *g(x) = f(h(x))*



Point Processing

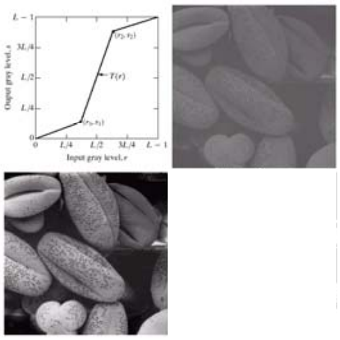
- The simplest kind of range transformations are these independent of position x,y:
  - $g = t(f)$
- This is called point processing.
- **Important:** every pixel for himself – spatial information completely lost!

Negative



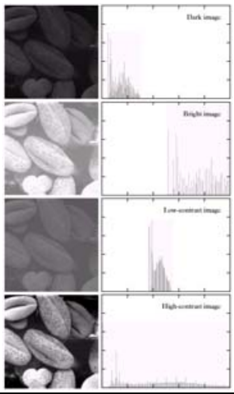
a b
FIGURE 3.4 (a) Original digital mammogram. (b) Negative image obtained using the negative transformation in Eq. (3.2-1). (Courtesy of G.E. Medical Systems.)

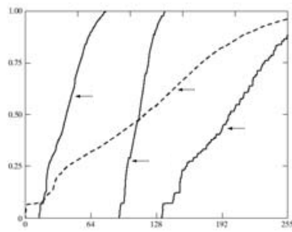
Contrast Stretching



a b
c d
FIGURE 3.10 Contrast stretching. (a) Form of transformation function. (b) A low-contrast image. (c) Result of contrast stretching. (d) Result of thresholding. (Original image courtesy of Dr. Roger Heady, Research School of Biological Sciences, Australian National University, Canberra, Australia.)

Image Histograms



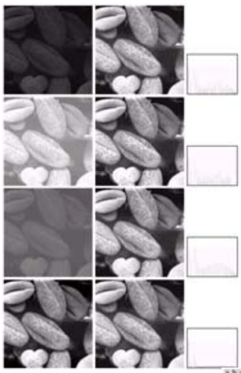


Cumulative Histograms

s = T(r)

a b
FIGURE 3.15 Four basic image types: dark, light, low contrast, high contrast, and their corresponding histograms. (Original image courtesy of Dr. Roger Heady, Research School of Biological Sciences, Australian National University, Canberra, Australia.)

Histogram Equalization



a b c
FIGURE 3.17 (a) Images from Fig. 3.15. (b) Results of histogram equalization. (c) Corresponding histograms.

Image Filtering – Change Range of Image

- Only range based: histogram manipulation
- Take only spatial neighborhood (domain) into account – averaging, median, Gaussian
- Domain + Range considered: Bilateral filtering (edge-preserving smoothing)

Eduard Gröller

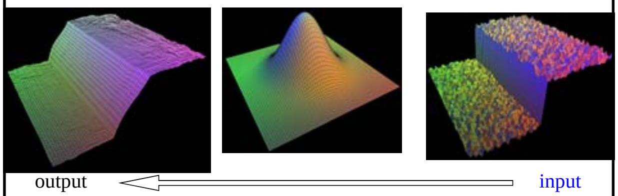
Bilateral filter

- **Tomasi and Manduci 1998**
  http://www.cse.ucsc.edu/~manduchi/Papers/ICCV98.pdf
- **Related to**
  - SUSAN filter
    [Smith and Brady 95]
    http://citeseer.ist.psu.edu/smith95susan.html
  - Digital-TV [Chan, Osher and Chen 2001]
    http://citeseer.ist.psu.edu/chan01digital.html
  - sigma filter
    http://www.geogr.ku.dk/CHIPS/Manual/f187.htm

Start with Gaussian filtering

- **Here, input is a step function + noise**

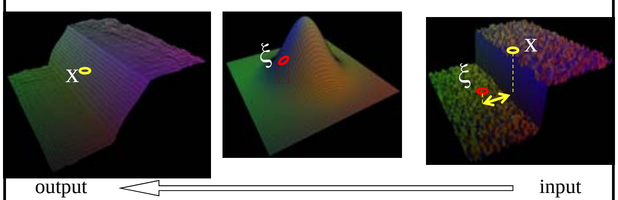
$$J \quad = \quad f \quad \otimes \quad I$$



Gaussian filter as weighted average

- **Weight of ξ depends on distance to x**

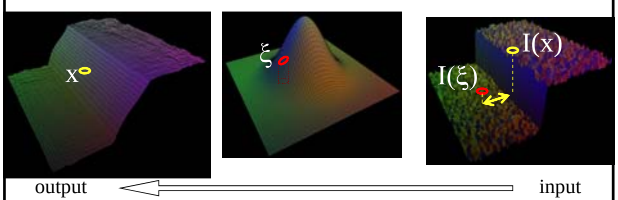
$$J(x) = \sum_{\xi} \quad f(x,\xi) \quad I(\xi)$$



The problem of edges

- **Here, $I(\xi)$ "pollutes" our estimate J(x)**
- **It is too different**

$$J(x) = \sum_{\xi} \quad f(x,\xi) \quad I(\xi)$$



Principle of Bilateral filtering

[Tomasi and Manduchi 1998]

- **Penalty g on the intensity difference**

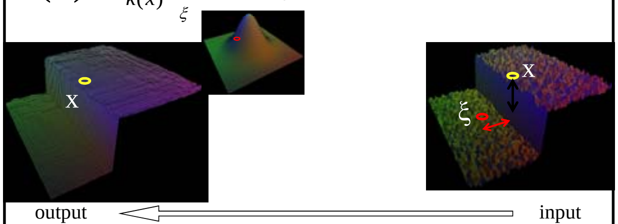
$$J(x) = \frac{1}{k(x)} \sum_{\xi} f(x,\xi) \quad g(I(\xi) - I(x)) \quad I(\xi)$$

Bilateral filtering

[Tomasi and Manduchi 1998]

- **Spatial Gaussian f**

$$J(x) = \frac{1}{k(x)} \sum_{\xi} f(x,\xi) g(I(\xi) - I(x)) \quad I(\xi)$$

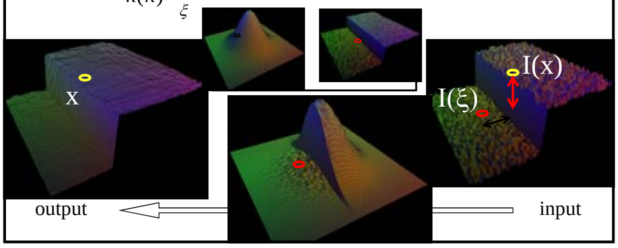


Bilateral filtering

[Tomasi and Manduchi 1998]

- Spatial Gaussian f
- Gaussian g on the intensity difference

$$J(x) = \frac{1}{k(x)} \sum_{\xi} f(x,\xi)\, g(I(\xi) - I(x))\, I(\xi)$$

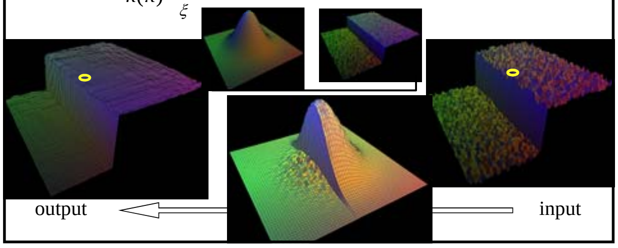


Normalization factor

[Tomasi and Manduchi 1998]

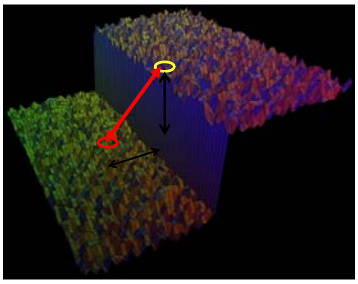
- k(x)= $\sum_{\xi} f(x,\xi) \quad g(I(\xi) - I(x))$

$$J(x) = \frac{1}{k(x)} \sum_{\xi} f(x,\xi) \quad g(I(\xi) - I(x)) \quad I(\xi)$$



Other view

- The bilateral filter uses the 3D distance



6.088 Digital and Computational Photography
6.882 Advanced Computational Photography

Dynamic Range and Contrast

Frédo Durand
MIT - EECS

Light, exposure and dynamic range

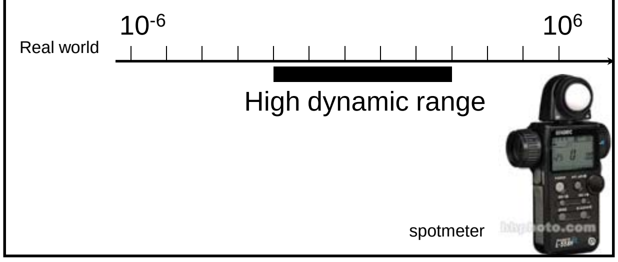
- Exposure: how bright is the scene overall
- Dynamic range: contrast in the scene
- Bottom-line problem: illumination level and contrast are not the same for a photo and for the real scene.

Example:

- Photo with a Canon G3
- Jovan is too dark
- Sky is too bright

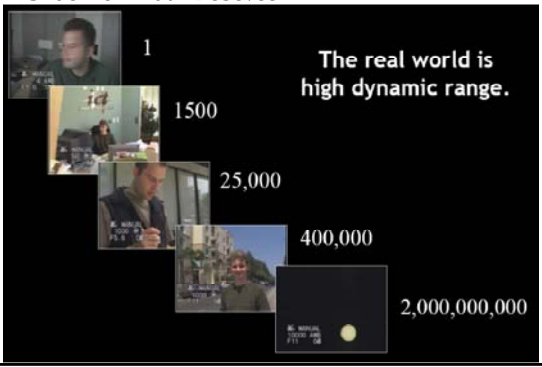
Real world dynamic range

- Eye can adapt from ~ 10^{-6} to 10^{6} cd/m^2
- Often 1 : 100,000 in a scene



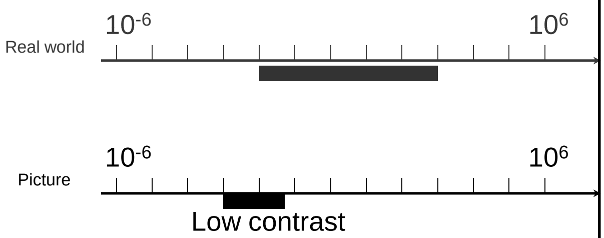
The world is high dynamic range

- Slide from Paul Debevec



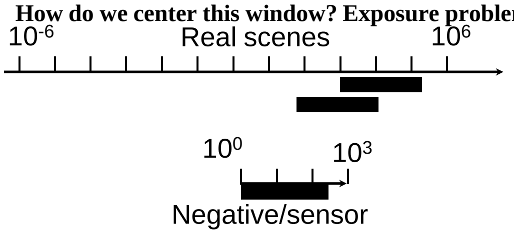
Problem 2: Picture dynamic range

- Typically 1: 20 or 1:50
  - Black ■ is ~ 50x darker than white □
- Max 1:500



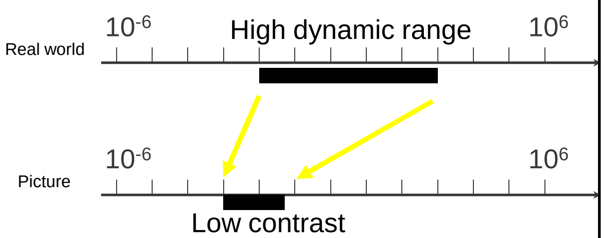
Problem 1

- The range of illumination levels that we encounter is 10 to 12 orders of magnitudes
- Negatives/sensors can record 2 to 3 orders of magnitude
- How do we center this window? Exposure problem.



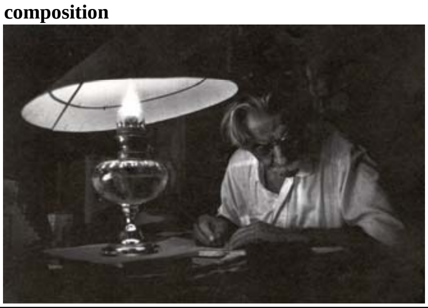
Contrast reduction

- Match limited contrast of the medium
- Preserve details



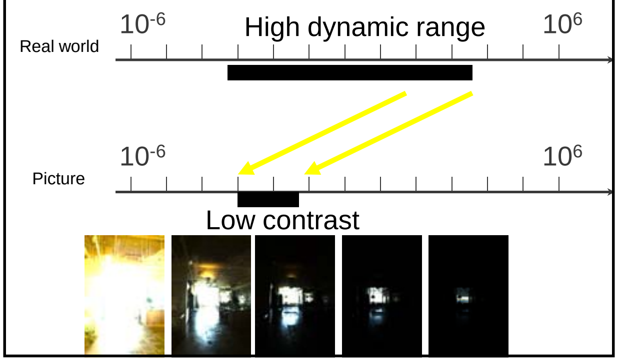
Limited dynamic range can be good!

- W. Eugene Smith photo of Albert Schweitzer
- 5 days to print!
- Things can be related because the intensity is more similar
- Balance, composition



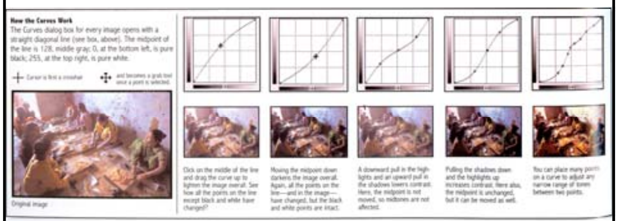
Multiple exposure photography

- Sequentially measure all segments of the range



Photoshop curves

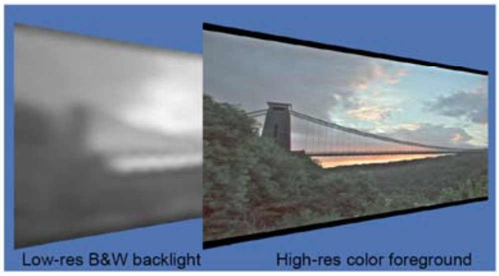
- Specify an arbitrary remapping curve
- Especially useful for black and white



From Photography by London et al.

Sunnybrook HDR display

- Use Bright Source + Two 8-bit Modulators
  - Transmission multiplies together
  - Over 10,000:1 dynamic range possible



Slide from the 2005 Siggraph course on HDR

Gradient Manipulation

Today: Gradient manipulation

Idea:

- Human visual system is very sensitive to gradient
- Gradient encode edges and local contrast quite well

- Do your editing in the gradient domain
- Reconstruct image from gradient

r

- Various instances of this idea, I'll mostly follow Perez et al. Siggraph 2003 http://research.microsoft.com/vision/cambridge/papers/perez_siggraph03.pdf

Problems with direct cloning

From Perez et al. 2003

Solution: clone gradient

sources/destinations

seamless cloning

Gradients and grayscale images

- Grayscale image: scalars
- Gradient: *2D vectors*
- Overcomplete!
- What's up with this?
- Not all vector fields are the gradient of an image!
- Only if they are curl-free (a.k.a. conservative)
  - But it does not matter for us

Seamless Poisson cloning

- Given vector field *v* (pasted gradient), find the value of *f* in unknown region that optimize:

Poisson equation with Dirichlet conditions

$$\min_f \iint_\Omega |\nabla f - \mathbf{v}|^2 \text{ with } f|_{\partial\Omega} = f^*|_{\partial\Omega}$$

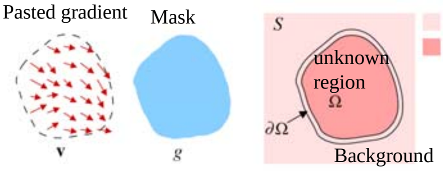
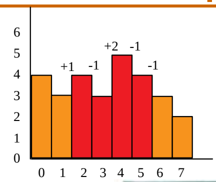


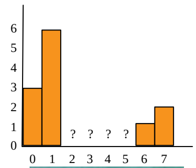
Figure 1: **Guided interpolation notations.** Unknown function f interpolates in domain Ω the destination function f^*, under guidance of vector field $\mathbf{v}$, which might be or not the gradient field of a source function g.

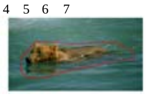
Discrete 1D example: minimization

- Copy



to



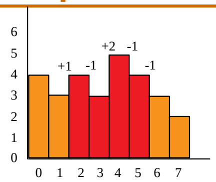


- Min $((f_2-f_1)-1)^2$
- Min $((f_3-f_2)-(-1))^2$
- Min $((f_4-f_3)-2)^2$
- Min $((f_5-f_4)-(-1))^2$
- Min $((f_6-f_5)-(-1))^2$

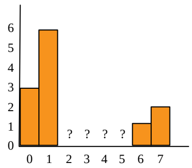
With
$f_1=6$
$f_6=1$

1D example: minimization

- Copy



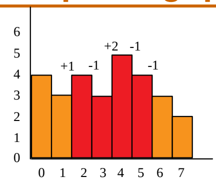
to



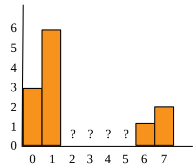
- Min $((f_2-6)-1)^2$ ==> $f_2^2+49-14f_2$
- Min $((f_3-f_2)-(-1))^2$ ==> $f_3^2+f_2^2+1-2f_3f_2 +2f_3-2f_2$
- Min $((f_4-f_3)-2)^2$ ==> $f_4^2+f_3^2+4-2f_3f_4 -4f_4+4f_3$
- Min $((f_5-f_4)-(-1))^2$ ==> $f_5^2+f_4^2+1-2f_5f_4 +2f_5-2f_4$
- Min $((1-f_5)-(-1))^2$ ==> $f_5^2+4-4f_5$

1D example: big quadratic

- Copy

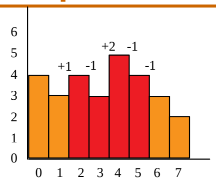
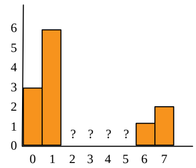


to



- Min ($f_2^2+49-14f_2$
  $+ f_3^2+f_2^2+1-2f_3f_2 +2f_3-2f_2$
  $+ f_4^2+f_3^2+4-2f_3f_4 -4f_4+4f_3$
  $+ f_5^2+f_4^2+1-2f_5f_4 +2f_5-2f_4$
  $+ f_5^2+4-4f_5$)
  Denote it Q

1D example: derivatives

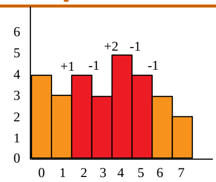
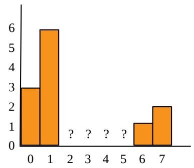
- **Copy**  **to** 

Min $(f_2^2+49-14f_2$
$+ f_3^2+f_2^2+1-2f_3f_2 +2f_3-2f_2$
$+ f_4^2+f_3^2+4-2f_3f_4 -4f_4+4f_3$
$+ f_5^2+f_4^2+1-2f_5f_4 +2f_5-2f_4$
$+ f_5^2+4-4f_5)$

Denote it Q

$$\frac{dQ}{df_2} = 2f_2 + 2f_2 - 2f_3 - 16$$
$$\frac{dQ}{df_3} = 2f_3 - 2f_2 + 2 + 2f_3 - 2f_4 + 4$$
$$\frac{dQ}{df_4} = 2f_4 - 2f_3 - 4 + 2f_4 - 2f_5 - 2$$
$$\frac{dQ}{df_5} = 2f_5 - 2f_4 + 2 + 2f_5 - 4$$

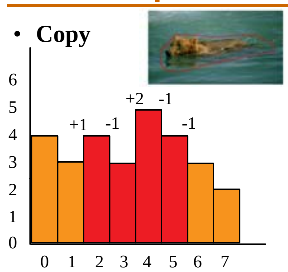
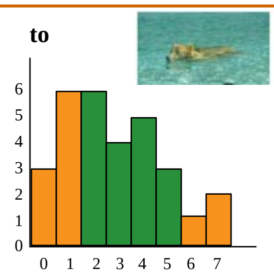
1D example: set derivatives to zero

- **Copy**  **to** 

$$\frac{dQ}{df_2} = 2f_2 + 2f_2 - 2f_3 - 16$$
$$\frac{dQ}{df_3} = 2f_3 - 2f_2 + 2 + 2f_3 - 2f_4 + 4$$
$$\frac{dQ}{df_4} = 2f_4 - 2f_3 - 4 + 2f_4 - 2f_5 - 2$$
$$\frac{dQ}{df_5} = 2f_5 - 2f_4 + 2 + 2f_5 - 4$$

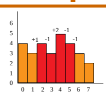
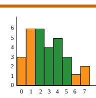
==> $$\begin{pmatrix} 4 & -2 & 0 & 0 \\ -2 & 4 & -2 & 0 \\ 0 & -2 & 4 & -2 \\ 0 & 0 & -2 & 4 \end{pmatrix} \begin{pmatrix} f_2 \\ f_3 \\ f_4 \\ f_5 \end{pmatrix} = \begin{pmatrix} 16 \\ -6 \\ 6 \\ 2 \end{pmatrix}$$

1D example

- **Copy**  **to** 

$$\begin{pmatrix} 4 & -2 & 0 & 0 \\ -2 & 4 & -2 & 0 \\ 0 & -2 & 4 & -2 \\ 0 & 0 & -2 & 4 \end{pmatrix} \begin{pmatrix} f_2 \\ f_3 \\ f_4 \\ f_5 \end{pmatrix} = \begin{pmatrix} 16 \\ -6 \\ 6 \\ 2 \end{pmatrix} \qquad \begin{pmatrix} f_2 \\ f_3 \\ f_4 \\ f_5 \end{pmatrix} = \begin{pmatrix} 6 \\ 4 \\ 5 \\ 3 \end{pmatrix}$$

1D example: remarks

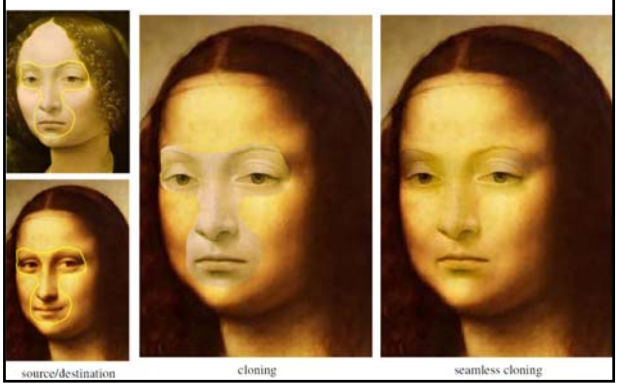
- **Copy**  **to**  $\begin{pmatrix} 4 & -2 & 0 & 0 \\ -2 & 4 & -2 & 0 \\ 0 & -2 & 4 & -2 \\ 0 & 0 & -2 & 4 \end{pmatrix} \begin{pmatrix} f_2 \\ f_3 \\ f_4 \\ f_5 \end{pmatrix} = \begin{pmatrix} 16 \\ -6 \\ 6 \\ 2 \end{pmatrix}$

- **Matrix is sparse**
- **Matrix is symmetric**
- **Everything is a multiple of 2**
  - because square and derivative of square
- **Matrix is a convolution (kernel -2 4 -2)**
- **Matrix is independent of gradient field. Only RHS is**
- **Matrix is a second derivative**

Recap

- **Find image whose gradient best approximates the input gradient**
  - least square Minimization
- **Discrete case: turns into linear equation**
  - Set derivatives to zero
  - Derivatives of quadratic ==> linear
- **Continuous: turns into Euler-Lagrange form**
  - $\Delta f = \text{div } v$
- **When gradient is null, membrane interpolation**
  - Linear interpolation in 1D

Result (eye candy)



source/destination — cloning — seamless cloning

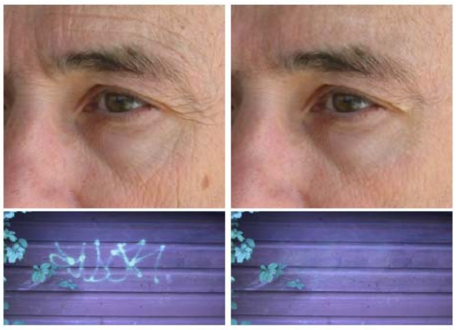
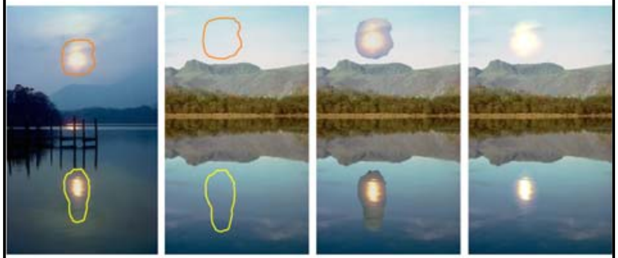


Figure 2: **Concealment**. By importing seamlessly a piece of the background, complete objects, parts of objects, and undesirable artifacts can easily be hidden. In both examples, multiple strokes (not shown) were used.



sources | destinations | cloning | seamless cloning

Seamless Image Stitching in the Gradient Domain

- Anat Levin, Assaf Zomet, Shmuel Peleg, and Yair Weiss
  http://www.cs.huji.ac.il/~alevin/papers/eccv04-blending.pdf
  http://eprints.pascal-network.org/archive/00001062/01/tips05-blending.pdf
- **Various strategies (optimal cut, feathering)**

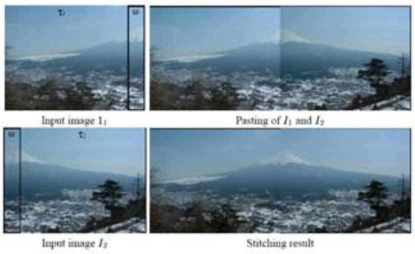


Fig. 1. Image stitching. On the left are the input images. ω is the overlap region. On top right is a simple pasting of the input images. On the bottom right is the result of the GIST1 algorithm.

Poisson Matting

- **Sun et al. Siggraph 2004**
- **Assume gradient of F & B is negligible**
- **Plus various image-editing tools to refine matte**

$$I = \alpha F + (1-\alpha)B$$

$$\nabla I = (F-B)\nabla\alpha + \alpha\nabla F + (1-\alpha)\nabla B$$

$$\nabla\alpha \approx \frac{1}{F-B}\nabla I$$

Figure 1: Pulling of matte from a complex scene. From left to right: a complex natural image for existing matting techniques where the color background is complex, a high quality matte generated by Poisson matting, a composite image with the extracted koala and a constant-color background, and a composite image with the extracted koala and a different background.

Poisson-ish mesh editing

- **http://portal.acm.org/citation.cfm?id=1057432.1057456**
- **http://www.cad.zju.edu.cn/home/xudong/Projects/mesh_editing/main.htm**
- **http://people.csail.mit.edu/sumner/research/deftransfer/**

Figure 1: An unknown mythical creature. Left: mesh components for merging and deformation (the arm). Right: final editing result.

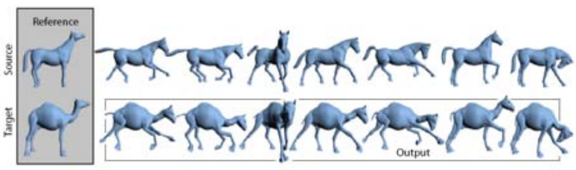


Figure 1: Deformation transfer copies the deformations exhibited by a source mesh onto a different target mesh. In this example, deformations of the reference horse mesh are transfered to the reference camel, generating seven new camel poses. Both gross skeletal changes as well as more subtle skin deformations are successfully reproduced.

Inpainting

- **More elaborate energy functional/PDEs**
- **http://www-mount.ee.umn.edu/~guille/inpainting.htm**

6.098 Digital and Computational Photography
6.882 Advanced Computational Photography

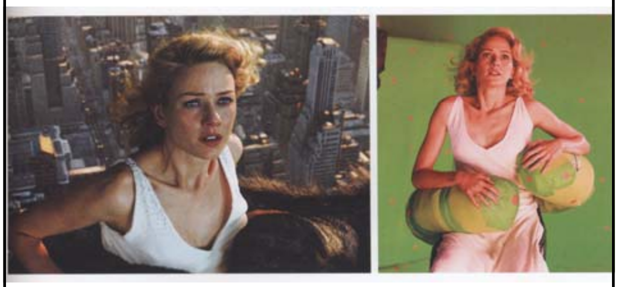
Matting & Compositing

Frédo Durand
MIT - EECS

Motivation: compositing

Combining multiple images. Typically, paste a foreground object onto a new background

- **Movie special effect**
- **Multi-pass CG**
- **Combining CG & film**
- **Photo retouching**
  - Change background
  - Fake depth of field
  - Page layout: extract objects, magazine covers



From Cinefex

Photo editing

- **Edit the background independently from foreground**

Photo editing

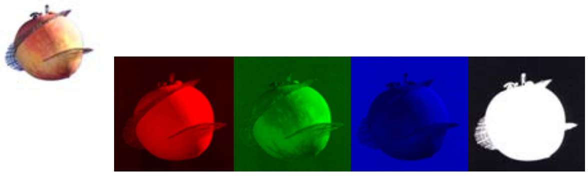
- **Edit the background independently from foreground**

Technical Issues

- **Compositing**
  - How exactly do we handle transparency?
- **Smart selection**
  - Facilitate the selection of an object
- **Matte extraction**
  - Resolve sub-pixel accuracy, estimate transparency
- **Smart pasting**
  - Don't be smart with copy, be smart with paste
  - See gradient manipulation
- **Extension to video**
  - Where life is always harder

Alpha

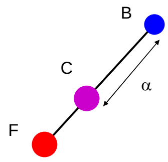
- α: 1 means opaque, 0 means transparent
- 32-bit images: R, G, B, α



From the Art & Science of Digital Compositing

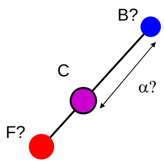
Compositing

- *Non* premultiplied version:
  Given the foreground color $F=(R_F, G_F, B_F)$, the background color (R_B, G_B, B_B) and α for each pixel
- The over operation is: $C=\alpha F+(1-\alpha)B$
  - (in the premultiplied case, omit the first α)



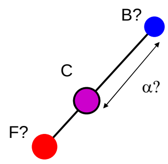
Matting problem

- Inverse problem:
  Assume an image is the *over* composite of a foreground and a background
- Given an image color C, find F, B and α so that $C=\alpha F+(1-\alpha)B$



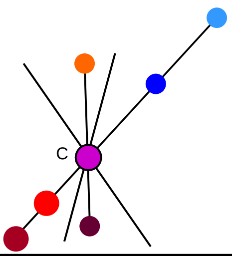
Matting ambiguity

- $C=\alpha F+(1-\alpha)B$
- How many unknowns, how many equations?



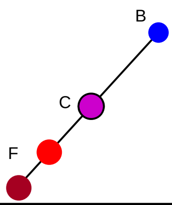
Matting ambiguity

- $C=\alpha F+(1-\alpha)B$
- 7 unknowns: α and triplets for F and B
- 3 equations, one per color channel

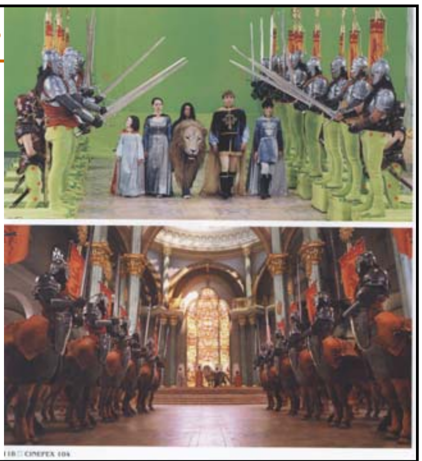


Matting ambiguity

- $C=\alpha F+(1-\alpha)B$
- 7 unknowns: α and triplets for F and B
- 3 equations, one per color channel
- With known background (e.g. blue/green screen): 4 unknowns, 3 equations



Questions?



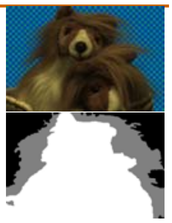
From Cinefex

Questions?

Natural matting

[Ruzon & Tomasi 2000, Chuang et al. 2001]

- **Given an input image with arbitrary background**
- **The user specifies a coarse *Trimap* (known Foreground, known background and unknown region)**
- **Goal: Estimate F, B, alpha in the unknown region**
  - We don't care about B, but it's a byproduct/unkown



images from Chuang et al

Now, what tool do we know to estimate something, taking into account all sorts of known probabilities?

Bayes theorem for matting

P(x|y) = P(y|x) P(x) / P(y)

The parameters you want to estimate — P(x|y)

What you observe — y

Likelihood function — P(y|x)

Prior probability — P(x)

Constant w.r.t. parameters x. — P(y)

Matting and Bayes

- **What do we observe?**

P(x|y) = P(y|x) P(x) / P(y)

The parameters you want to estimate — P(x|y)

What you observe — y

Likelihood function — P(y|x)

Prior probability — P(x)

Constant w.r.t. parameters x. — P(y)

Matting and Bayes

- **What do we observe?**
  - Color C at a pixel

P(x|C) = P(C|x) P(x) / P(C)

The parameters you want to estimate — P(x|C)

Color you observe — C

Likelihood function — P(C|x)

Prior probability — P(x)

Constant w.r.t. parameters x. — P(C)

Matting and Bayes

- What do we observe: Color C
- **What are we looking for?**

$$P(x|C) = P(C|x)\, P(x) / P(C)$$

The parameters you want to estimate — Color you observe — Likelihood function — Prior probability — Constant w.r.t. parameters x.

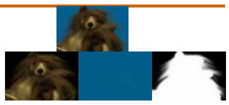
Matting and Bayes

- What do we observe: Color C
- **What are we looking for: F, B, α**

$$P(F,B,\alpha|C) = P(C|F,B,\alpha)\, P(F,B,\alpha) / P(C)$$

Foreground, background, transparency you want to estimate — Color you observe — Likelihood function — Prior probability — Constant w.r.t. parameters x.

Matting and Bayes

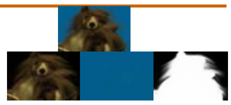


- What do we observe: Color C
- What are we looking for: F, B, α
- **Likelihood probability?**
  - Given F, B and Alpha, probability that we observe C

$$P(F,B,\alpha|C) = P(C|F,B,\alpha)\, P(F,B,\alpha) / P(C)$$

Foreground, background, transparency you want to estimate — Color you observe — Likelihood function — Prior probability — Constant w.r.t. parameters x.

Matting and Bayes



- What do we observe: Color C
- What are we looking for: F, B, α
- **Likelihood probability?**
  - Given F, B and Alpha, probability that we observe C
  - If measurements are perfect, non-zero only if $C=\alpha F+(1-\alpha)B$
  - But assume Gaussian noise with variance σ_C

$$P(F,B,\alpha|C) = P(C|F,B,\alpha)\, P(F,B,\alpha) / P(C)$$

Foreground, background, transparency you want to estimate — Color you observe — Likelihood function — Prior probability — Constant w.r.t. parameters x.

Matting and Bayes

- What do we observe: Color C
- What are we looking for: F, B, α
- Likelihood probability: Compositing equation + Gaussian noise with variance σ_C
- **Prior probability:**
  - How likely is the foreground to have color F? the background to have color B? transparency to be α?

$$P(F,B,\alpha|C) = P(C|F,B,\alpha)\, P(F,B,\alpha) / P(C)$$

Foreground, background, transparency you want to estimate — Color you observe — Likelihood function — Prior probability — Constant w.r.t. parameters x.

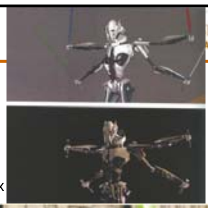
Matting and Bayes

- What do we observe: Color C
- What are we looking for: F, B, α
- Likelihood probability: Compositing equation + Gaussian noise with variance σ_C
- **Prior probability:**
  Build a probability distribution from the known regions
  - ***This is the heart of Bayesian matting***

$$P(F,B,\alpha|C) = P(C|F,B,\alpha)\, P(F,B,\alpha) / P(C)$$

Foreground, background, transparency you want to estimate — Color you observe — Likelihood function — Prior probability — Constant w.r.t. parameters x.

Questions?



From Cinefex

From Chuang et al 2001

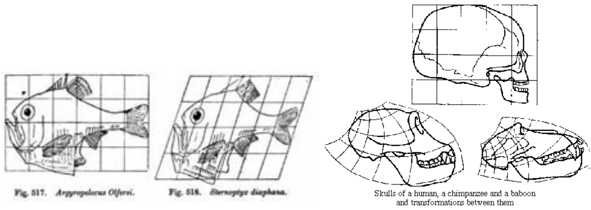
6.098 Digital and Computational Photography
6.882 Advanced Computational Photography

Image Warping and Morphing

Frédo Durand
Bill Freeman
MIT - EECS

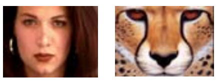
Intelligent design & image warping

- **D'Arcy Thompson**
  http://www-groups.dcs.st-and.ac.uk/~history/Miscellaneous/darcy.html
  http://en.wikipedia.org/wiki/D'Arcy_Thompson
- **Importance of shape and structure in evolution**



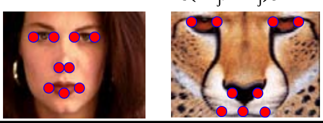
Morphing

- **Input: two images I_0 and I_N**



- **Expected output: image sequence I_i, with $i \in \{1..N-1\}$**

- **User specifies sparse correspondences on the images**
  - Pairs of vectors $\{(P^0_j, P^N_j)\}$



Morphing

- **For each intermediate frame I_t**
  - Interpolate feature locations $P^t_i = (1-t) P^0_i + t P^1_i$
  - Perform **two** warps: one for I_0, one for I_1
    - Deduce a dense warp field from the pairs of features
    - Warp the pixels
  - Linearly interpolate the two warped images

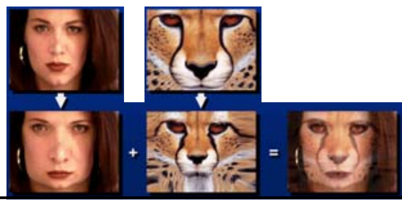
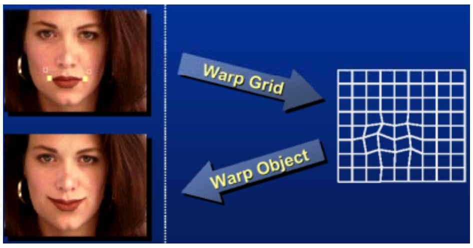


Image Warping – parametric

- **Move control points to specify a spline warp**
- **Spline produces a smooth vector field**



Slide Alyosha Efros

Warp specification - dense

- **How can we specify the warp?**

  Specify corresponding *spline control points*

  - *interpolate* to a complete warping function

But we want to specify only a few points, not a grid

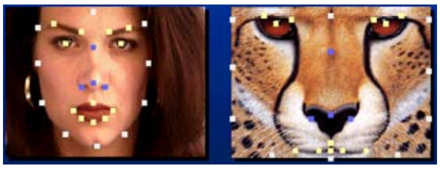
Slide Alyosha Efros

Warp specification - sparse

- **How can we specify the warp?**

  Specify corresponding *points*

  - *interpolate* to a complete warping function
  - How do we do it?



How do we go from feature points to pixels?

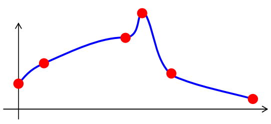
Slide Alyosha Efros

Warp as interpolation

- **We are looking for a warping field**
  - A function that given a 2D point, returns a warped 2D point
- **We have a sparse number of correspondences**
  - These specify values of the warping field
- **This is an interpolation problem**
  - Given sparse data, find smooth function

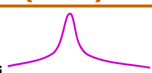
Interpolation in 1D

- **We are looking for a function *f***
- **We have N data points: x_i, y_i**
  - Scattered: spacing between x_i is non-uniform
- **We want *f* so that**
  - For each i, $f(x_i)=y_i$
  - *f* is smooth
- **Depending on notion of smoothness, different *f***

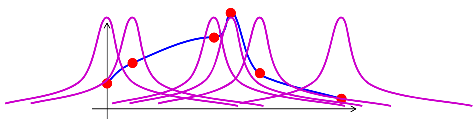


Radial Basis Functions (RBF)

- **Place a smooth kernel R centered on each data point x_i**

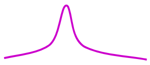


- $f(z) = \Sigma\, \alpha_i\, R(z, x_i)$

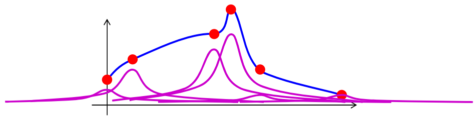


Radial Basis Functions (RBF)

- **Place a smooth kernel R centered on each data point x_i**



- **$f(z) = \Sigma\, \alpha_i\, R(z, x_i)$**
- **Find weights α_i to make sure we interpolate the data**
  for each i, $f(x_i)=y_i$



Kernel

- **Many choices**
- **e.g. inverse multiquadric**

$$R(z, x_i) = \frac{1}{\sqrt{c + \|z - x_i\|^2}}$$

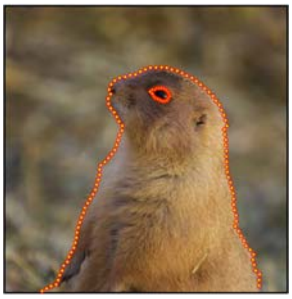
- **where *c* controls falloff**
- **Lazy way: set *c* to an arbitrary constant (pset 4)**
- **Smarter way: *c* is different for each kernel. For each x_i, set c as the squared distance to the closest other x_j**

Variations of RBF

- **Lots of possible kernels**
  - Gaussians $e^{-r^2/2\sigma}$
  - Thin-plate splines $r^2 \log r$
- **Sometimes add a global polynomial term**

Input images


Feature correspondences

- **The feature locations will be our x_i**
- **Yes, in this example, the number of features is excessive**

Interpolate feature location

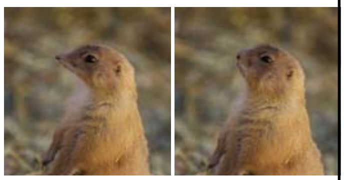
- **Provides the y_i**

Warp each image to intermediate location

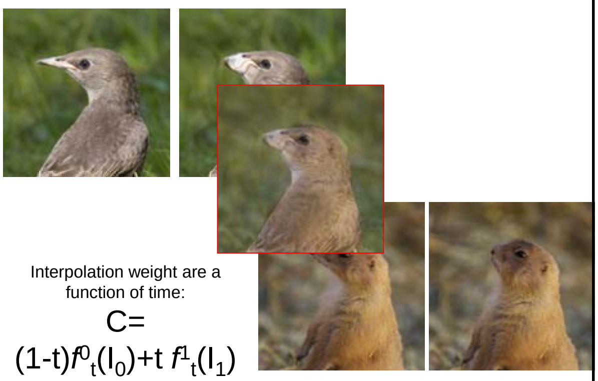
Two different warps:
Same target location, different source location
i.e. the y_i are the same (intermediate locations), the x_i are different (source feature locations)

Note: the x_i do not change along the animation, but the y_i are different for each intermediate image
Here we show t=0.5
(the y_i are in the middle)

Warp each image to intermediate location



Interpolate colors linearly



Interpolation weight are a function of time:

$$C=(1-t)f^0_t(I_0)+t\, f^1_t(I_1)$$

Uniform morphing

Figure 4. Uniform metamorphosis

6.088 Digital and Computational Photography
6.882 Advanced Computational Photography

Panoramas

Frédo Durand
MIT - EECS

Lots of slides stolen from Alyosha Efros, who stole them from Steve Seitz and Rick Szeliski

Why Mosaic?

- **Are you getting the whole picture?**
  - Compact Camera FOV = 50 x 35°

Slide from Brown & Lowe

Why Mosaic?

- **Are you getting the whole picture?**
  - Compact Camera FOV = 50 x 35°
  - Human FOV = 200 x 135°

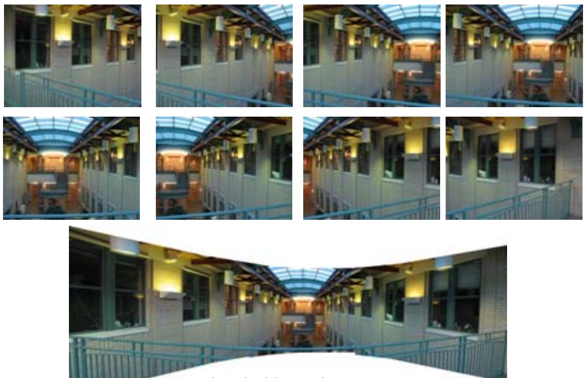
Slide from Brown & Lowe

Why Mosaic?

- **Are you getting the whole picture?**
  - Compact Camera FOV = 50 x 35°
  - Human FOV = 200 x 135°
  - Panoramic Mosaic = 360 x 180°

Slide from Brown & Lowe

Mosaics: stitching images together

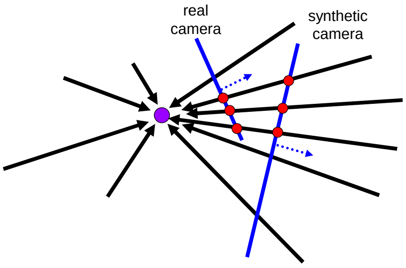


virtual wide-angle camera

How to do it?

- **Basic Procedure**
  - Take a sequence of images from the same position
    - Rotate the camera about its optical center
  - Compute transformation between second image and first
  - Transform the second image to overlap with the first
  - Blend the two together to create a mosaic
  - If there are more images, repeat
- **…but** wait**, why should this work at all?**
  - What about the 3D geometry of the scene?
  - Why aren't we using it?

A pencil of rays contains all views

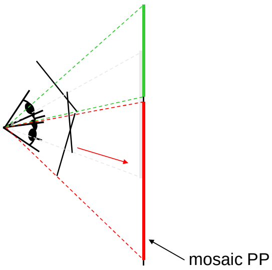


Can generate any synthetic camera view
as long as it has **the same center of projection**!

Aligning images: translation

Translations are not enough to align the images

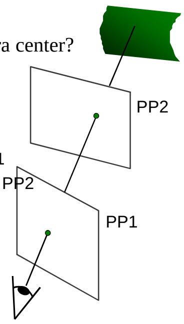
Image reprojection



- **The mosaic has a natural interpretation in 3D**
  – The images are reprojected onto a common plane
  – The mosaic is formed on this plane
  – Mosaic is a *synthetic wide-angle camera*

Image reprojection

- **Basic question**
  – How to relate 2 images from same camera center?
    - how to map a pixel from PP1 to PP2



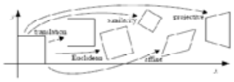
- **Answer**
  – Cast a ray through each pixel in PP1
  – Draw the pixel where that ray intersects PP2

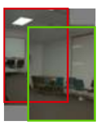
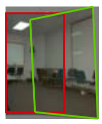
But don't we need to know the geometry of the two planes in respect to the eye?

Observation:
Rather than thinking of this as a 3D reprojection, think of it as a 2D **image warp** from one image to another

Back to Image Warping

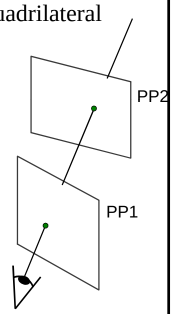
Which t-form is the right one for warping PP1 into PP2?
e.g. translation, Euclidean, affine, projective



Translation	Affine	Perspective
		
2 unknowns	6 unknowns	8 unknowns

Homography

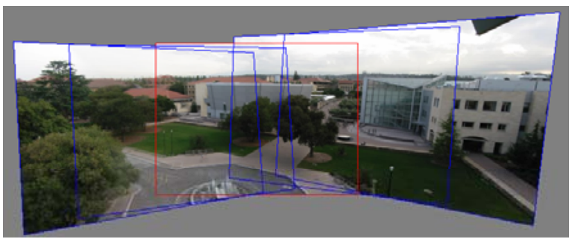
- **Projective – mapping between any two PPs with the same center of projection**
  – rectangle should map to arbitrary quadrilateral
  – parallel lines aren't
  – but must preserve straight lines
  – same as: project, rotate, reproject
- **called Homography**



$$\underbrace{\begin{bmatrix} wx' \\ wy' \\ w \end{bmatrix}}_{\mathbf{p'}} = \underbrace{\begin{bmatrix} * & * & * \\ * & * & * \\ * & * & * \end{bmatrix}}_{\mathbf{H}} \underbrace{\begin{bmatrix} x \\ y \\ 1 \end{bmatrix}}_{\mathbf{p}}$$

To apply a homography **H**
- Compute **p'** = **Hp** (regular matrix multiply)
- Convert **p'** from homogeneous to image coordinates

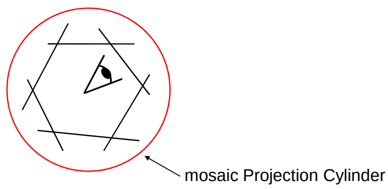
Panoramas



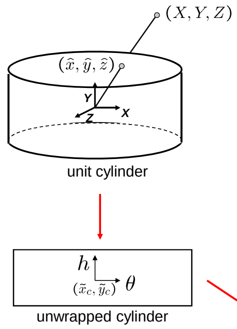
1. **Pick one image (red)**
2. **Warp the other images towards it (usually, one by one)**
3. **blend**

Full Panoramas

- **What if you want a 360° field of view?**



Cylindrical projection



- Map 3D point (X,Y,Z) onto cylinder

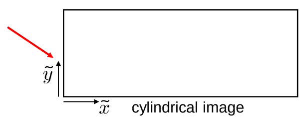
$$(\hat{x}, \hat{y}, \hat{z}) = \frac{1}{\sqrt{X^2 + Z^2}}(X, Y, Z)$$

- Convert to cylindrical coordinates

$$(sin\theta, h, cos\theta) = (\hat{x}, \hat{y}, \hat{z})$$

- Convert to cylindrical image coordinates

$$(\tilde{x}, \tilde{y}) = (f\theta, fh) + (\tilde{x}_c, \tilde{y}_c)$$



Full-view (360°) panoramas

Blending the mosaic

An example of image compositing:
the art (and sometime science) of
combining images together…

Multi-band Blending

Multi-band Blending

- **Burt & Adelson 1983**
  - Blend frequency bands over range $\propto \lambda$

References Links

- Links to Fredo
- New upcoming textbook

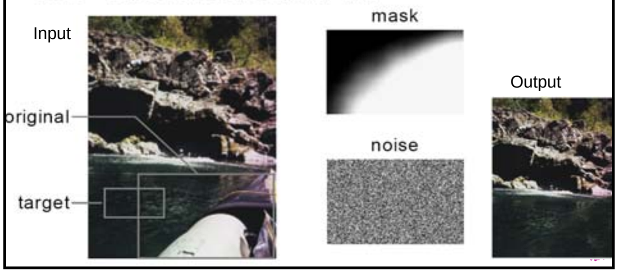
Eduard Gröller

Further Examples

Image Replacement through Texture Synthesis

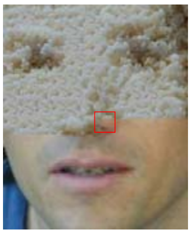
Homan Igehy Lucas Pereira
Computer Science Department
Stanford University

```
exture-Replace (noise, original, mask, target)
 original = Blend(Mean(target), original, mask)
 noise = Match-Histogram (noise, target)
 noise = Composite(noise, original, mask)
 analysis-pyr = Make-Pyramid (target)
 Loop for several iterations
    synthesis-pyr = Make-Pyramid (noise)
    Loop for a-band in sub-bands of analysis-pyr
          for s-band in sub-bands of synthesis-pyr
      s-band = Match-Histogram (s-band, a-band)
    noise = Collapse-Pyramid (synthesis-pyr)
    noise = Match-Histogram (noise, texture)
    noise = Composite(noise, original, mask)
```

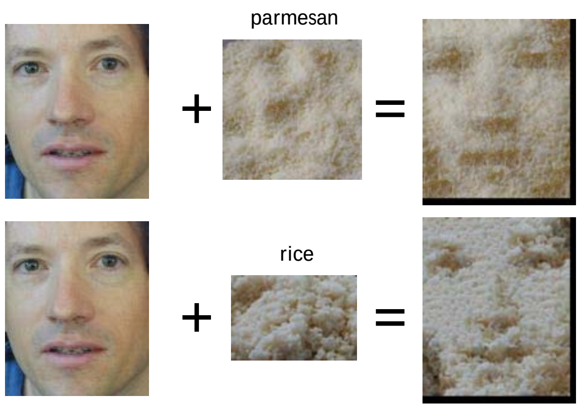


Texture Transfer

- Take the texture from one object and “paint” it onto another object
  - This requires separating texture and shape
  - That’s HARD, but we can cheat
  - Assume we can capture shape by boundary and rough shading



Then, just add another constraint when sampling: similarity to underlying image at that spot



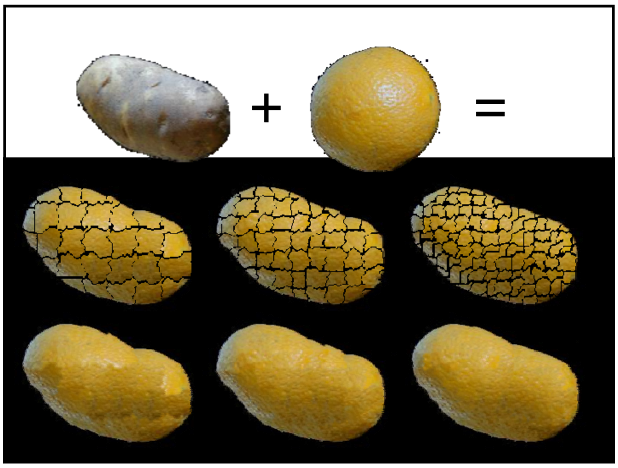


Image analogies

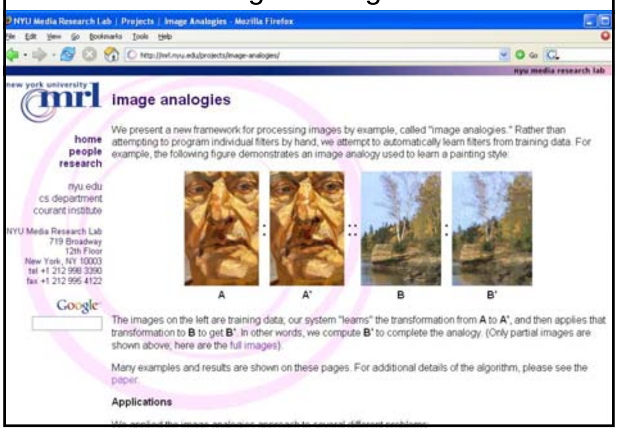
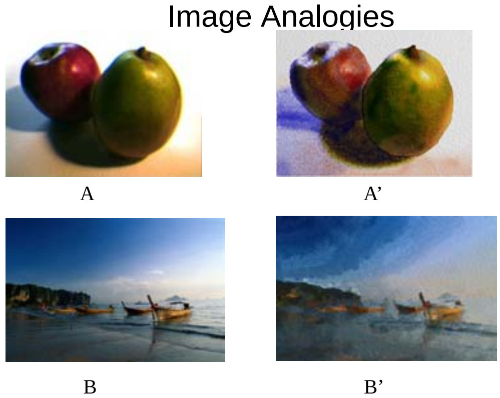
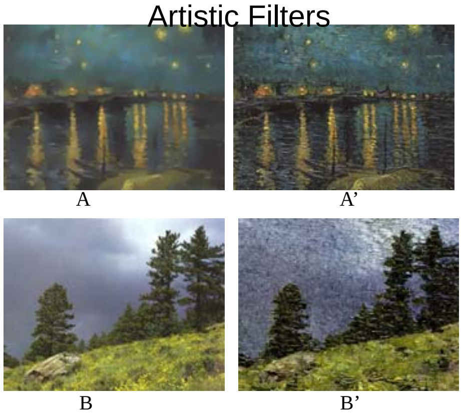


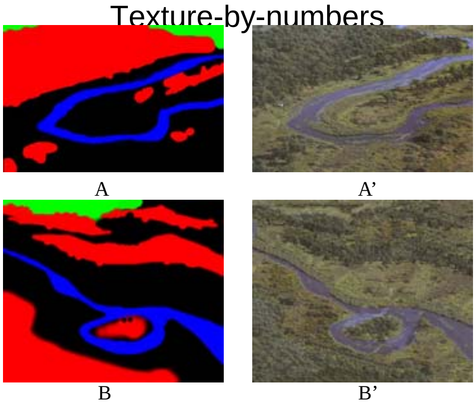
Image Analogies



Artistic Filters



Texture-by-numbers



Summary

- Modern algorithms enable qualitatively new imaging techniques
- Some of these algorithms will be integrated in cameras soon
- Former times: **physical capturing of light at a time**
- Today/future: **capturing the moment** (M. Cohen)

[Deussen et al.]

Interesting Links

- http://people.csail.mit.edu/fredo/)
- http://web.media.mit.edu/~raskar/photo/
- http://computationalphotography.org/
- http://en.wikipedia.org/wiki/Computational_photography