Computational Photography

Photographie Algorithmique

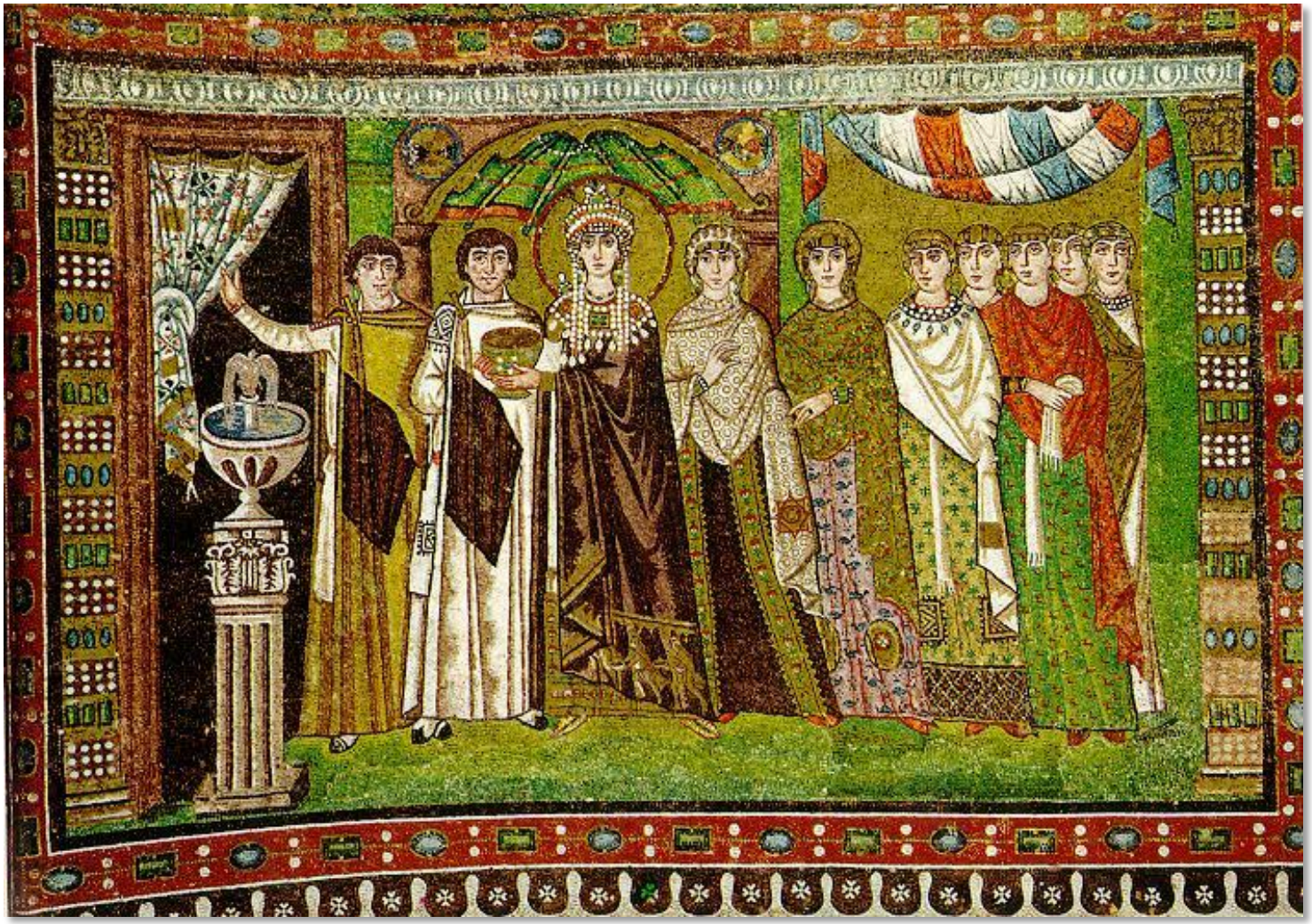
Crédits :

Alexei Efros, James Hays, Frédo Durand, Steve Seitz, Rick Szeliski, Paul Debevec, Stephen Palmer, Paul Heckbert, David Forsyth, Steve Marschner, Shai Avidan

Un peu d'histoire: les débuts

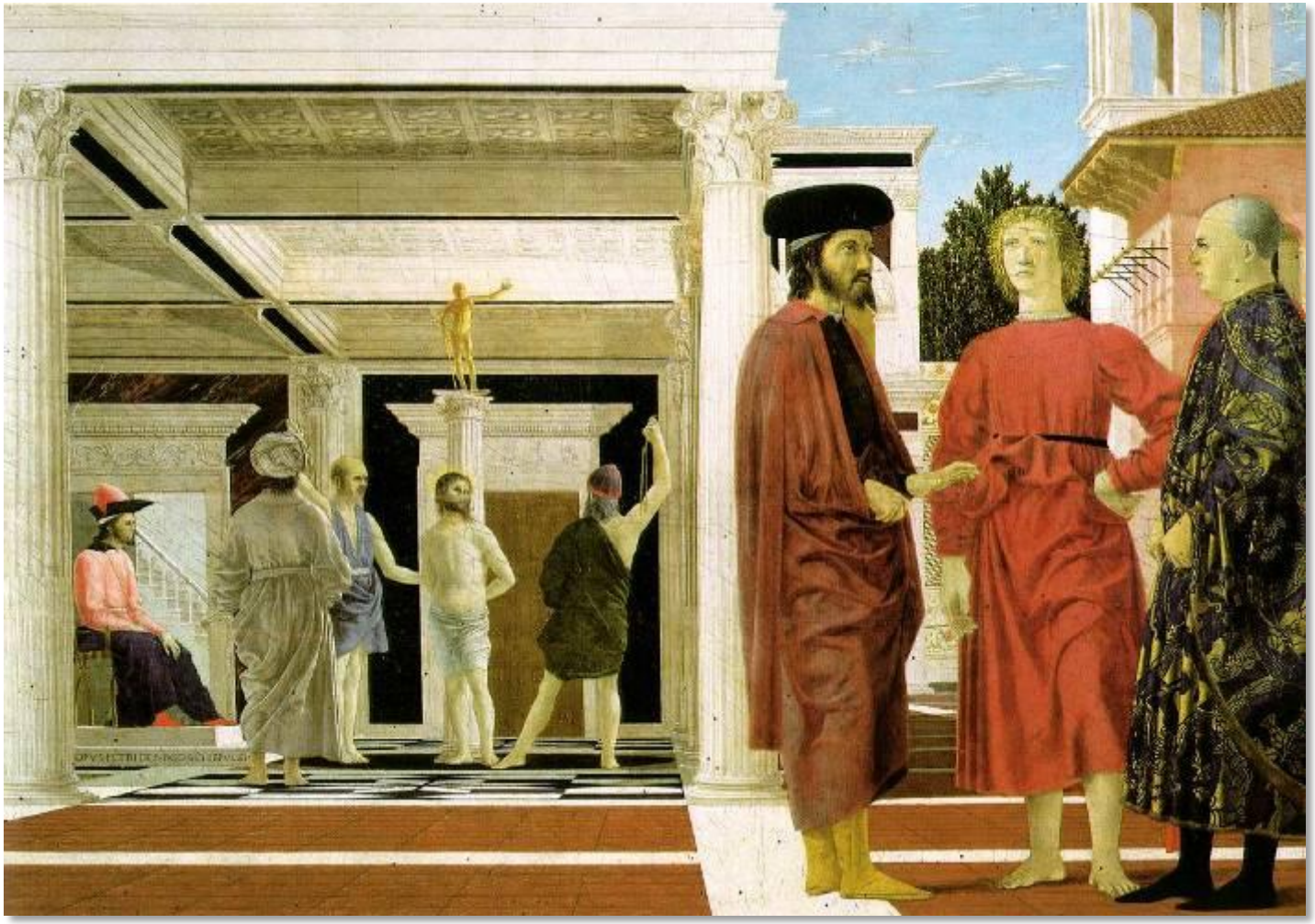
Peintures rupestres, Lascaux ~ 13,000 -- 15,000 B.C.

Le Moyen-Age



The Empress Theodora with her court.
Ravenna, St. Vitale 6th c.

Renaissance

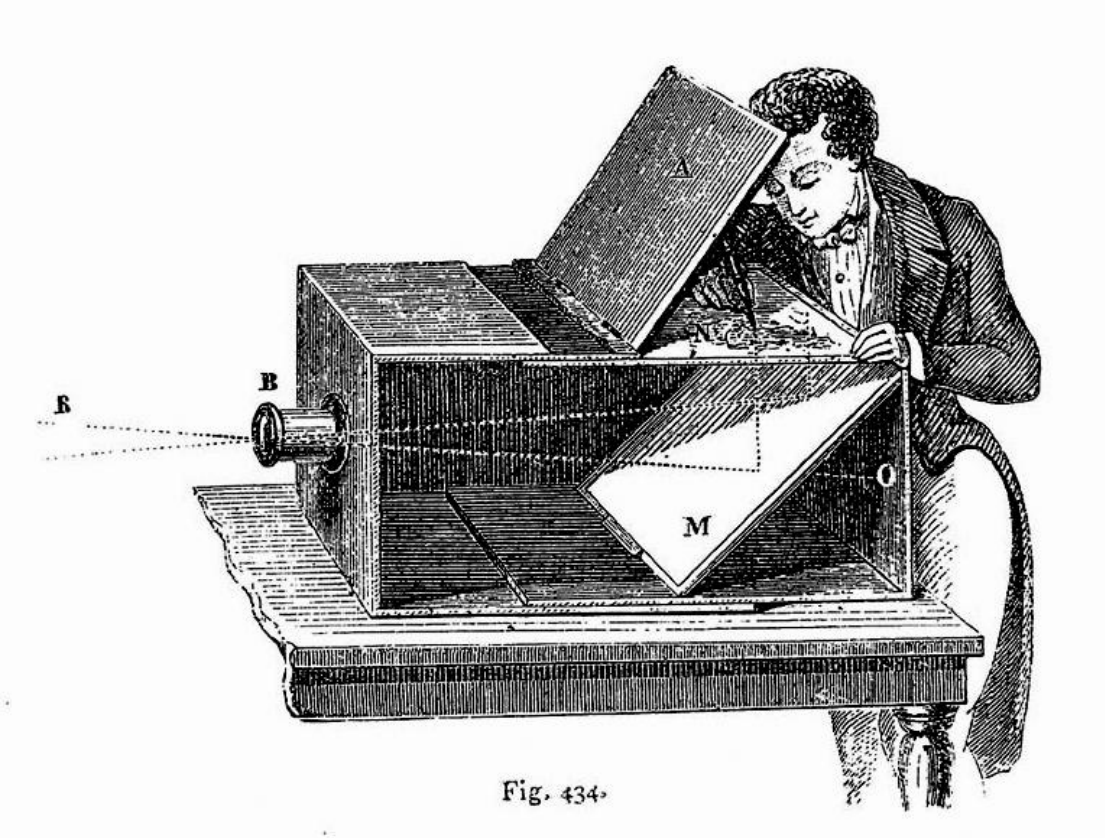


Piero della Francesca, The Flagellation (c.1469)

La quête de la « perfection »

Jan van Eyck, The Arnolfini Marriage (c.1434)

La quête de la « perfection »



Lens Based Camera Obscura, 1568

La « perfection »

Nature morte, **Louis-Jacques-Mandé Daguerre**, 1837

Perfection = réalité ?

Multiple viewpoints

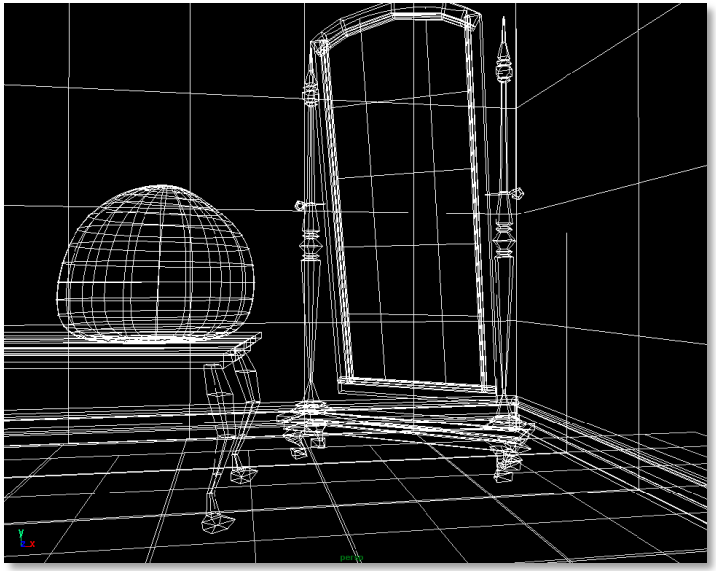
David Hockney,
Place Furstenberg, 1985

Single viewpoint

Alyosha Efros
Place Furstenberg, 2009

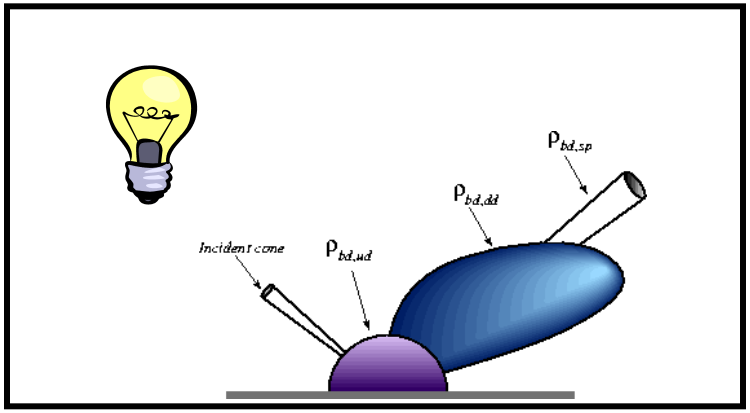
Arrivée de la synthèse d'image

Synthèse d'image



géometrie 3D

projection

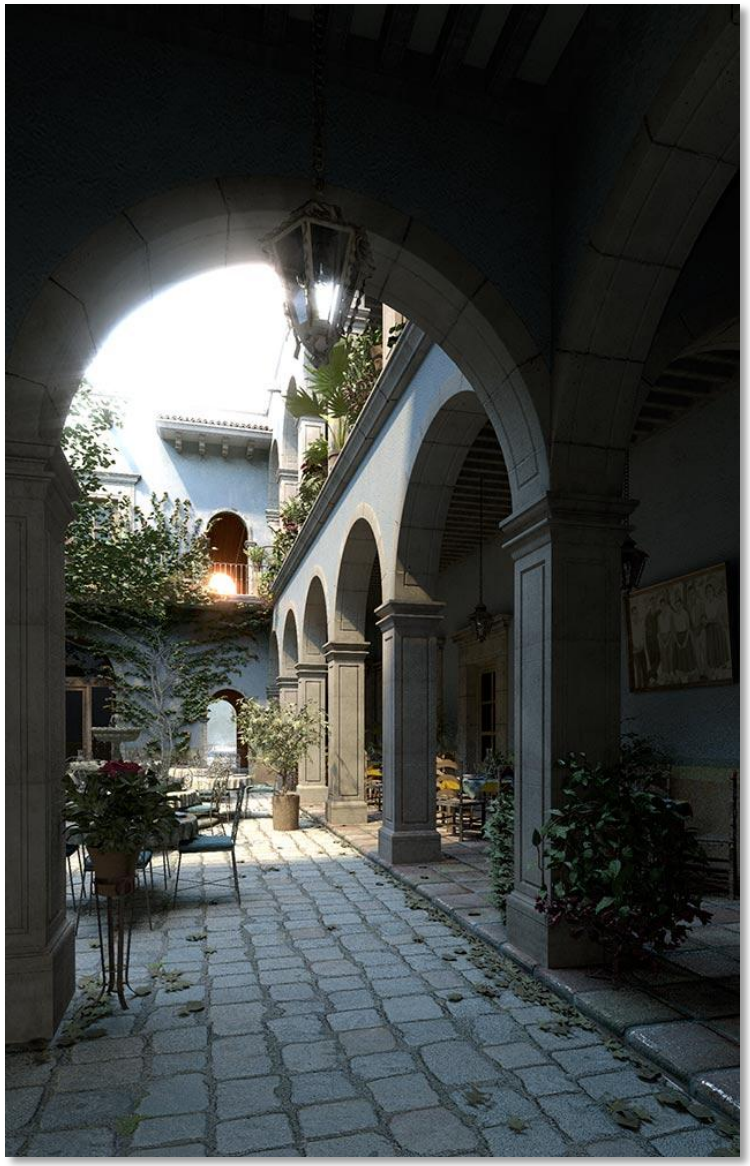


physique

Simulation

Synthèse d'image

- Hyper-réalisme
  - Modélisation
  - Éclairage
- Mais pas équivalent à la réalité : pourquoi ?



Couverture de PBRT par Guillermo M. Leal Ilaguno

La richesse du monde réel

Photo de Svetlana Lazebnik

Les êtres humains

"Final Fantasy"

dans le métro, Londres

Les visages / cheveux

"Final Fantasy"

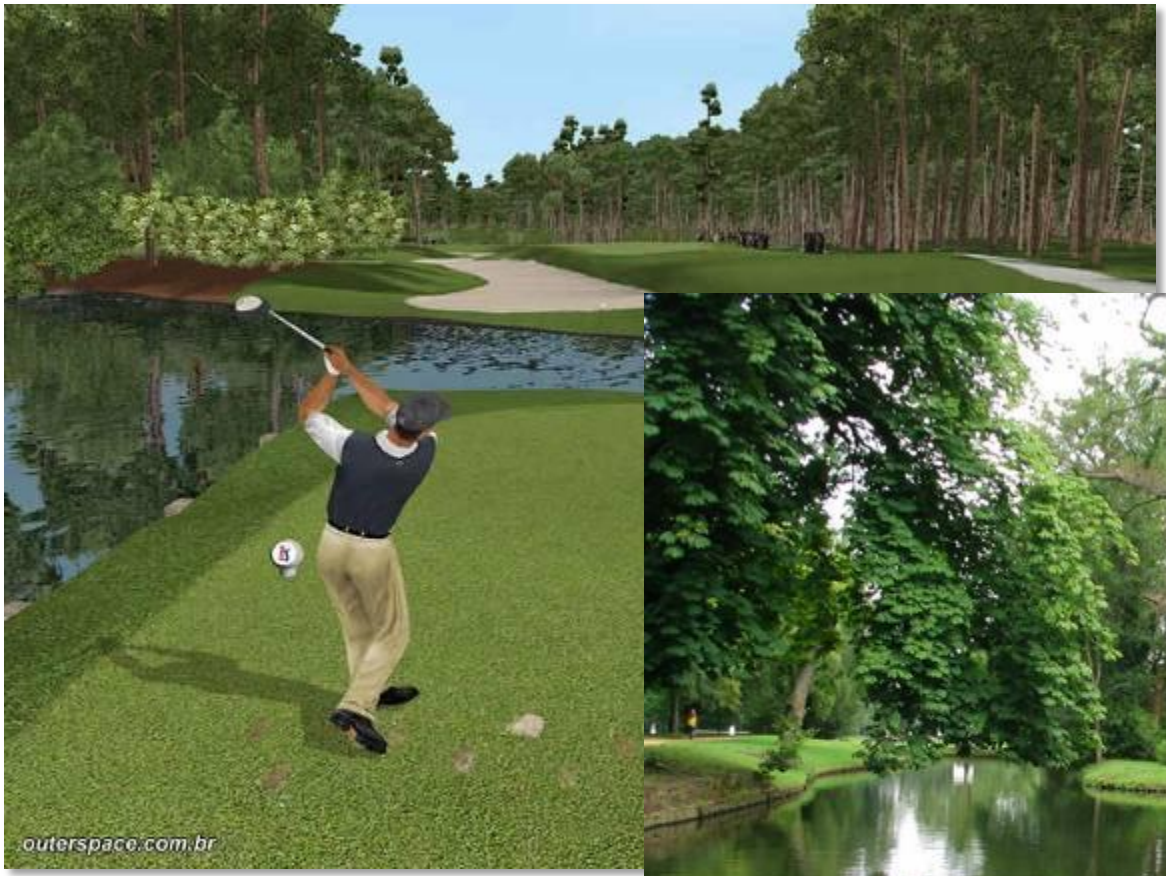
Photo de Joaquin Rosales Gomez

Les scène urbaines

Virtual LA (SGI)

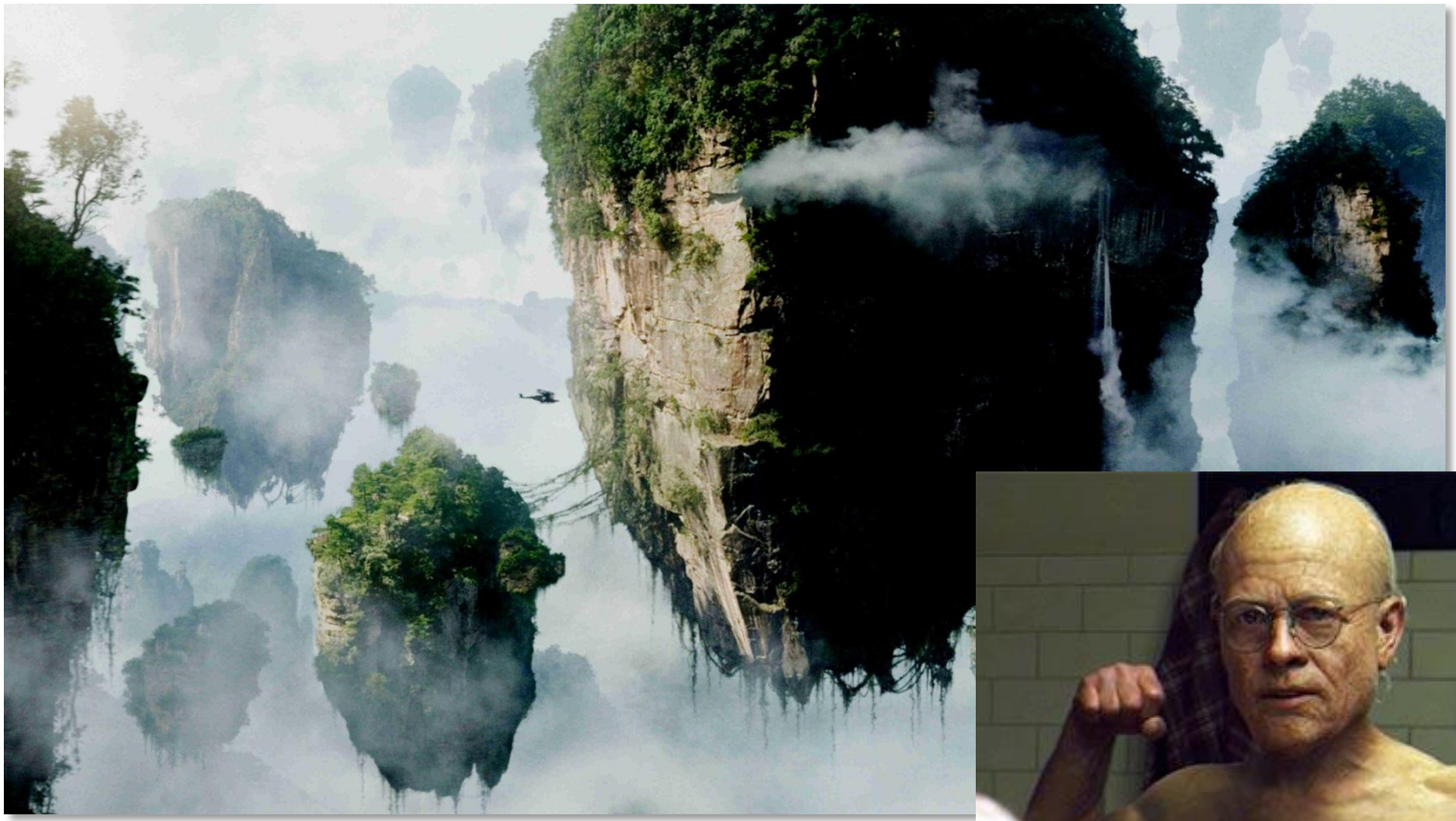
Photo de LA

La nature



River Cherwell, Oxford

Plus récemment…



Avatar

The Curious Case
of Benjamin Button

Le meilleur de deux mondes

Informatique graphique

+ facile de créer de nouveaux mondes

+ facile de manipuler les objets / le point de vue

– très difficile d'être réaliste

Computational Photography

Réaliste
Manipulations
Capture simple

Photographie

+ intrinsèquement réaliste

+ acquisition simple

– très difficile de manipuler les objets / le point de vue

Le meilleur de deux mondes

- Essayer de garder le meilleur de deux mondes
  - la capture du réel
  - La facilité d'édition
- A l'intersection de plusieurs disciplines
  - la photographie
  - le traitement d'image
  - la synthèse d'images
  - la vision par ordinateur

« Retouche » d'images

1860

d'après **Photo Fakery**, D. Brugioni

« Retouche » d'images

1942

d'après **Photo Fakery**, D. Brugioni

« Retouche » d'images

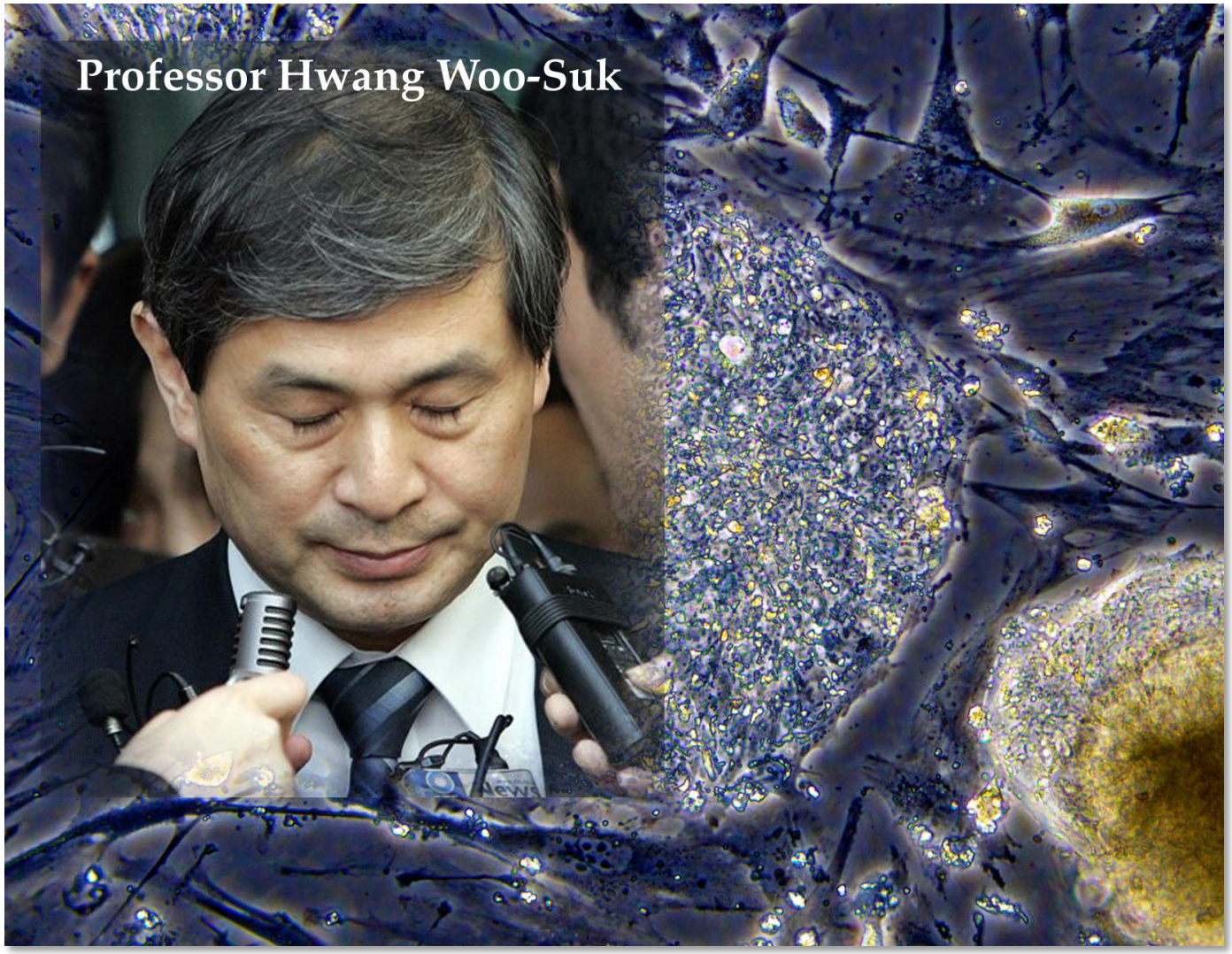
1942

d'après **Photo Fakery**, D. Brugioni

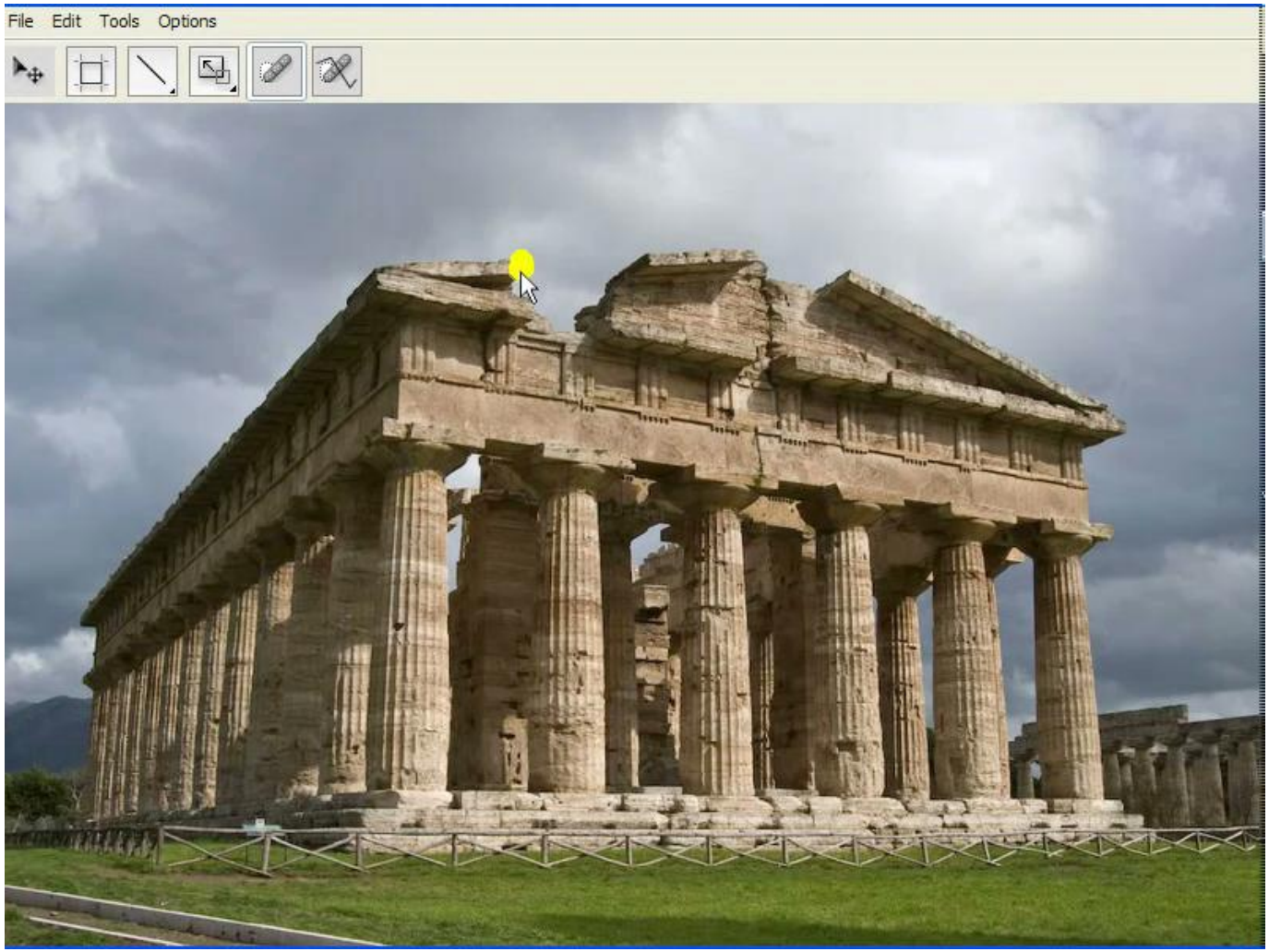
« Retouche » d'images

Dans la presse, juillet 2008

« Retouche » d'images



PatchMatch [2009]



Plan

- Computational processing
- Computational illumination
- Computational optics

Plan

- **Computational processing**
  - Image retargeting
  - Filtering
  - Image Warping & Morphing
  - Compositing & Matting
  - Gradient Editing
- Computational illumination
- Computational optics

Plan

- Computational processing
  - Image retargeting
  - Filtering
  - Image Warping & Morphing
  - Compositing & Matting
  - Gradient Editing
- Computational illumination
- Computational optics

Image retargeting

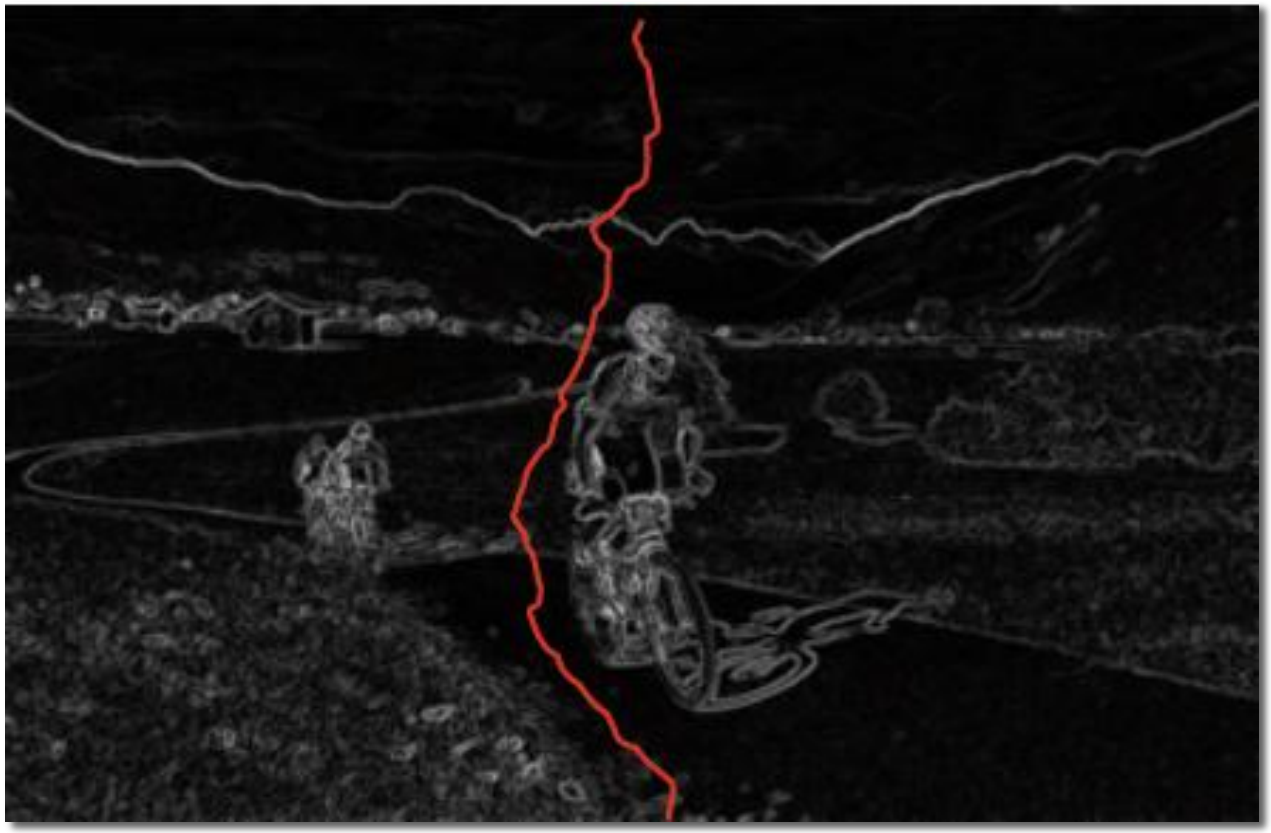
Cropping

Scaling

Retargeting

Seam carving

[Avidan et al. 2007]



$$s^* = \arg\min_s E(s)$$

Dynamic Programming

High cost

Low cost

Horizontal cost

Vertical cost

Energy / Error functions

HoG

L_1

Entropy

Seg.+L_1

Results

Seam Carving Scaling Cropping

Results

Change aspect ratio : enlarge

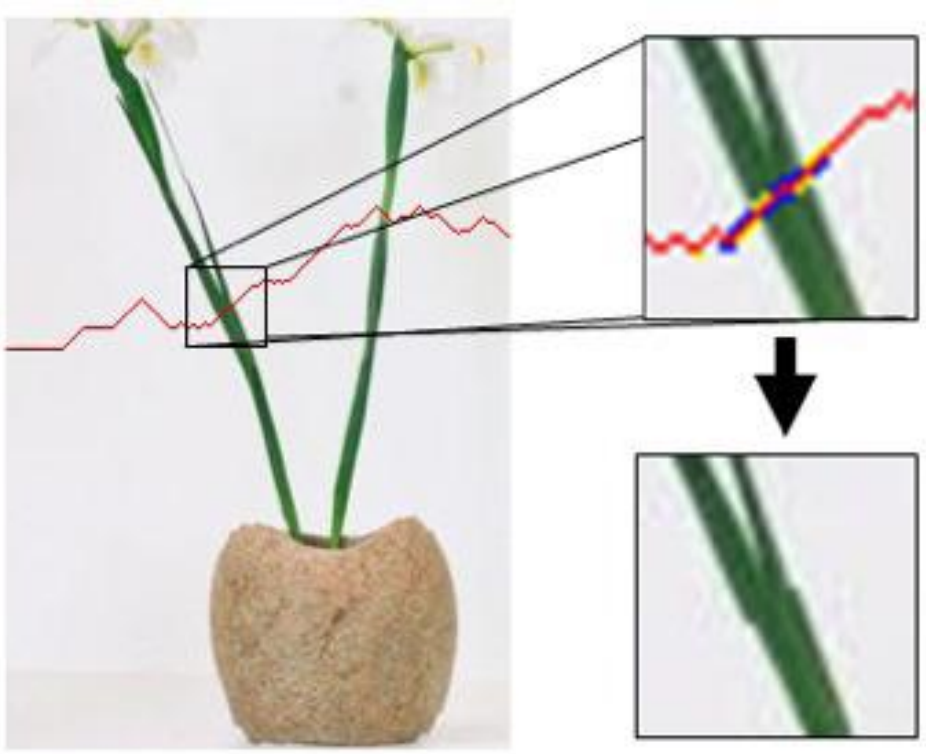
Objects removal

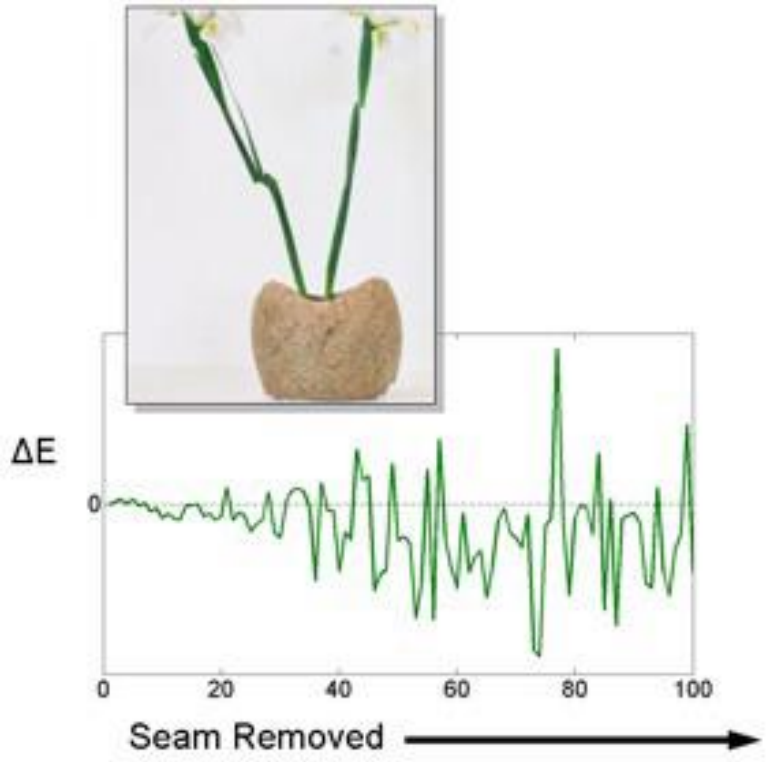
Limitations

Content

Structure

Limitations: why?





Energy inserted into the retargeted image **ignored**

Forward energy

Seam carving

+ forward energy

“Improved Seam Carving for Video Retargeting”

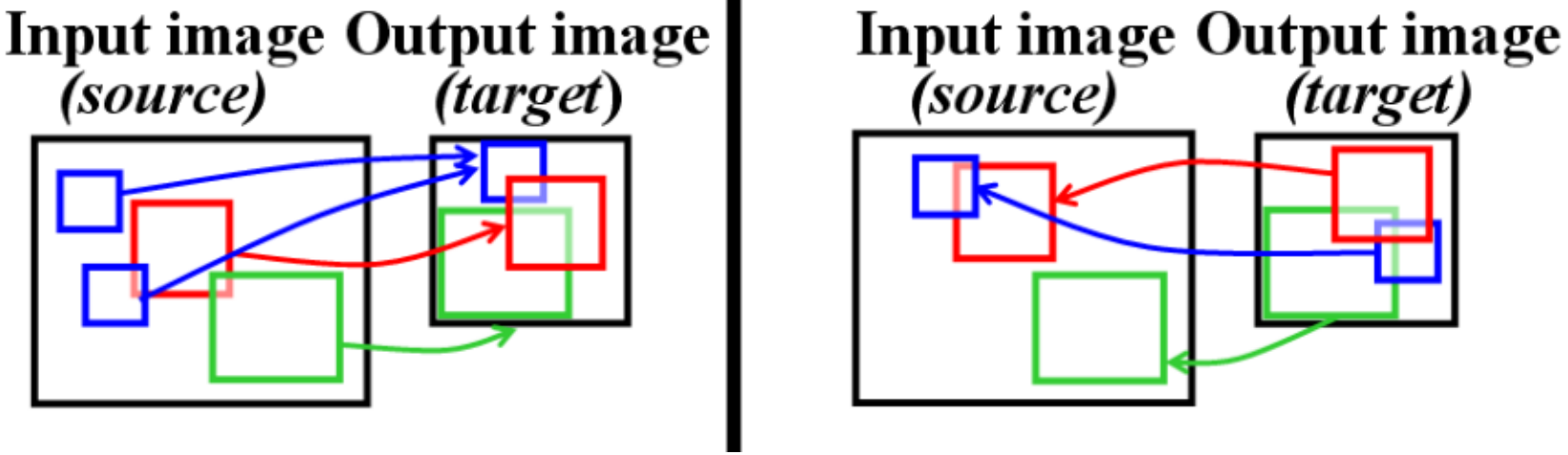
[Rubinstein et al. 2008]

Bidirectional similarity

[Simakov et al. 2008]

Completeness: | **Coherence:**

(a) The bidirectional spatial (image) similarity:



Plan

- **Computational processing**
  - Image retargeting
  - **Filtering**
  - Image Warping & Morphing
  - Compositing & Matting
  - Gradient Editing
- Computational illumination
- Computational optics

Filtrage

- Filtrage linéaire : convolution
- Filtrage non-linéaire : filtre bilatéral
- Opérateurs morphologiques

Convolution

image filtrée noyau image d'entrée

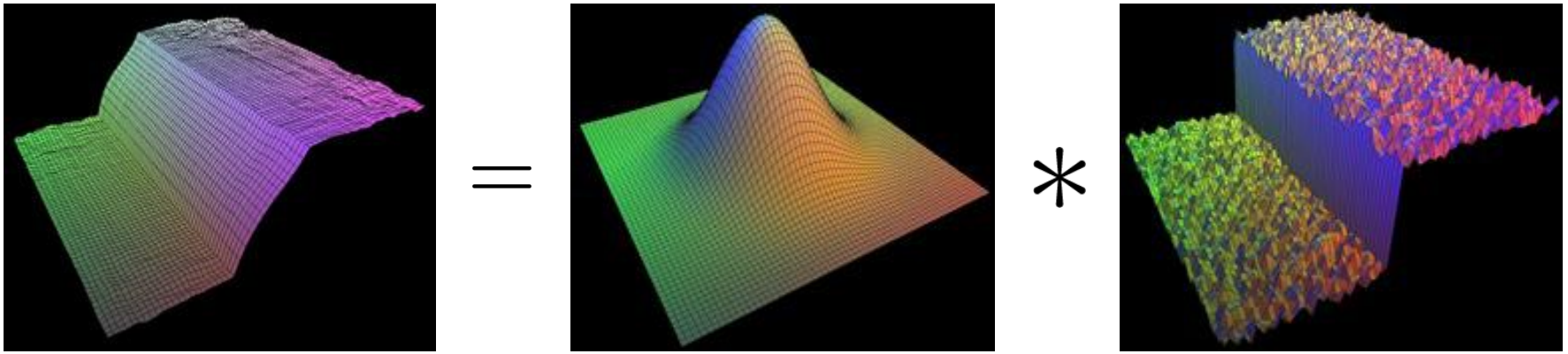
$$J = f * I$$

- Applications :
  - Rendre flou (blur)
  - Rendre plus net (sharpen)
  - Débruiter (denoising)
  - …

Filtre gaussien

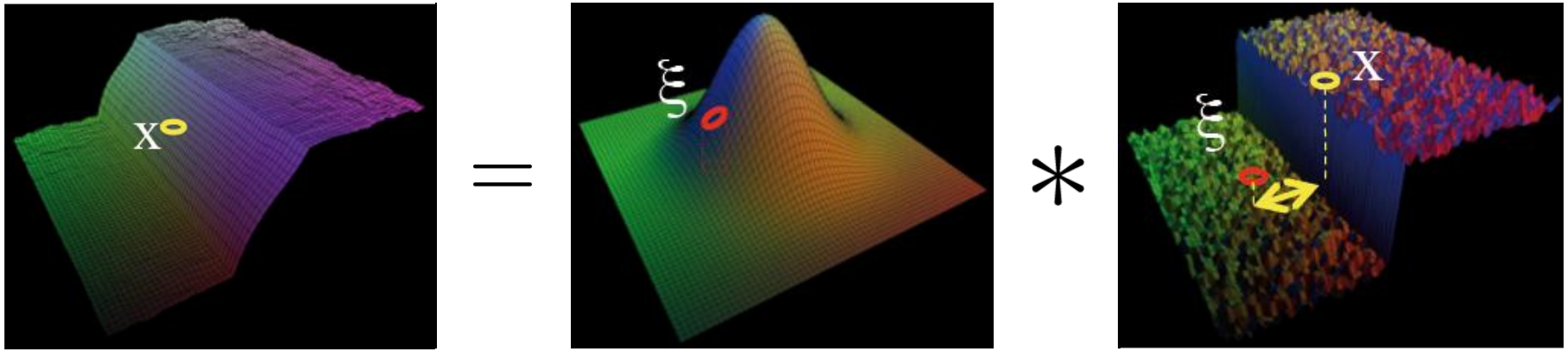
gaussienne marche + bruit

$$J = f * I$$



Filtre gaussien : somme pondérée

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



Filtre gaussien

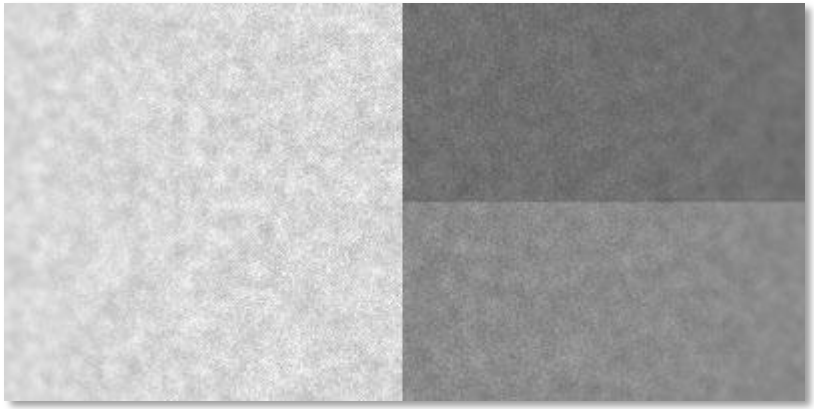
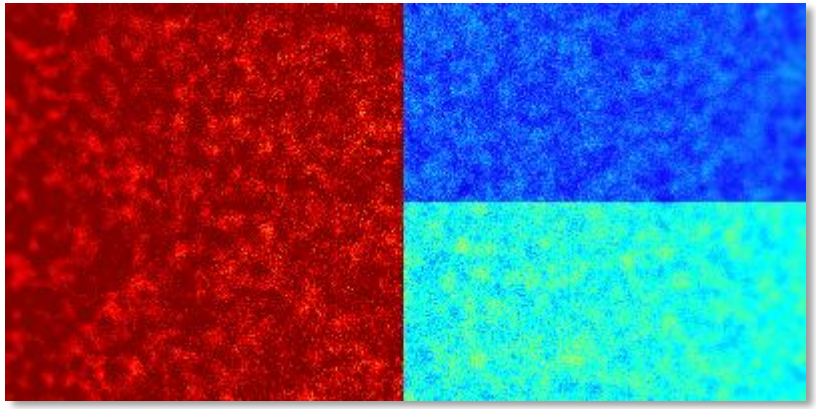
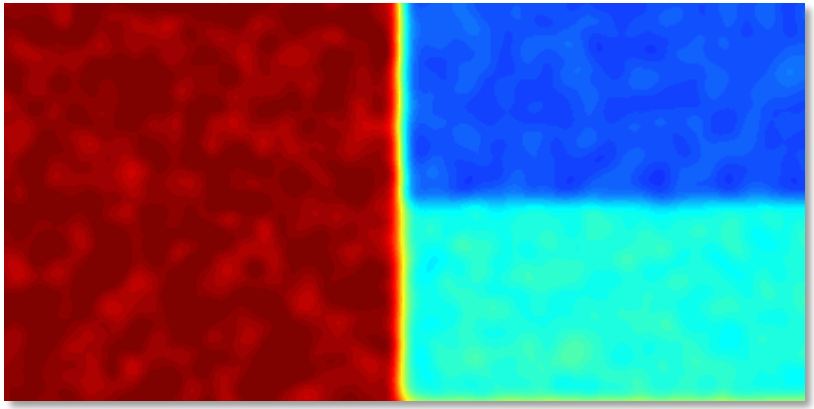


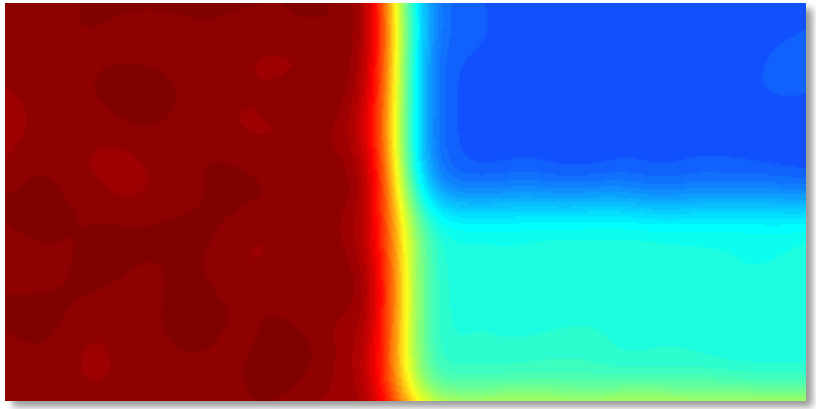
Image bruitée



Codage couleur



Filtre gaussien (σ=4)



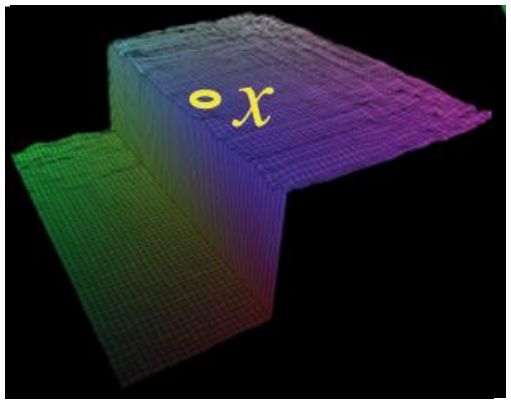
Filtre gaussien (σ=12)

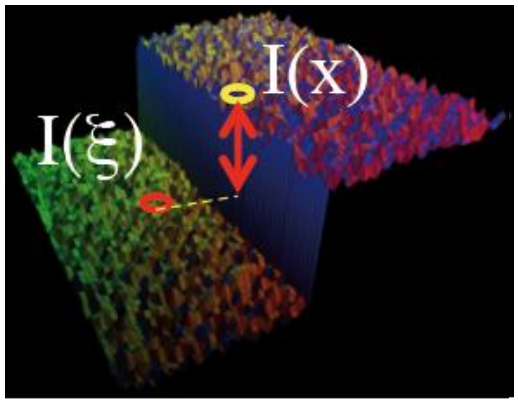
Filtre bilatéral

Tomasi and Manduci 1998

- Pénalité **g** en fonction des différences d'intensité

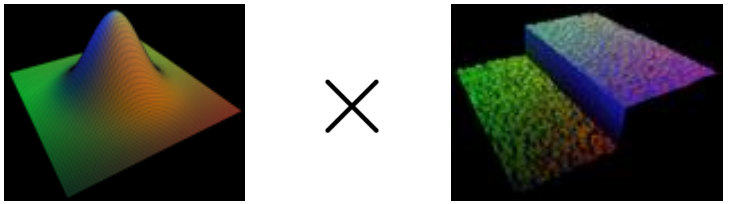
$$J(x) = \frac{1}{k(x)} \sum_{\xi} f(x,\xi)\ g(I(\xi) - I(x))\ I(\xi)$$

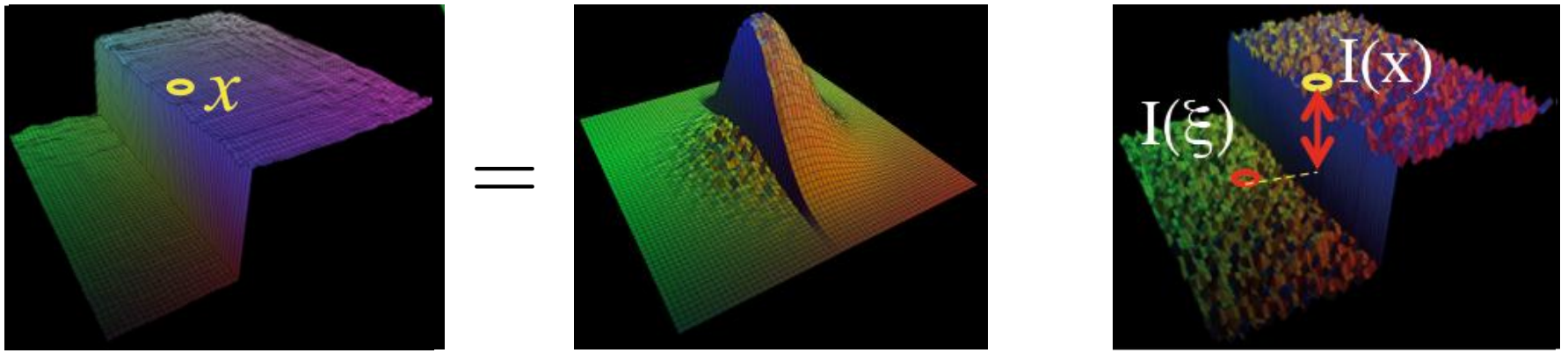




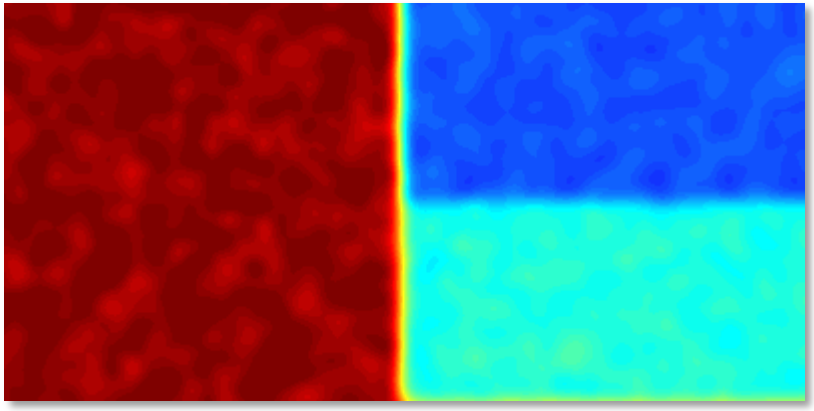
Filtre bilatéral

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

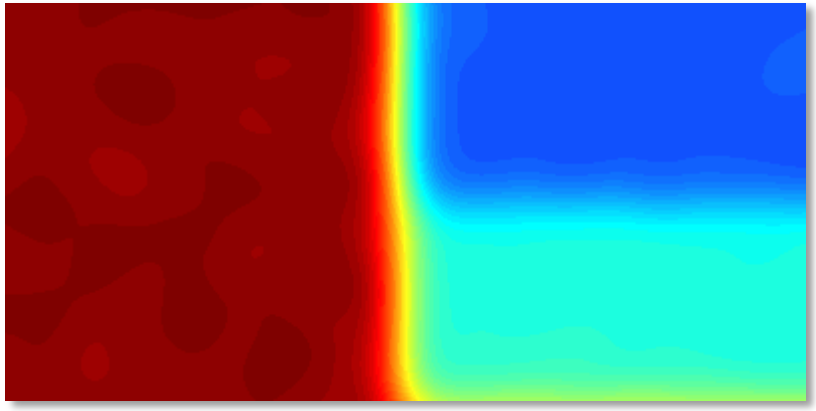




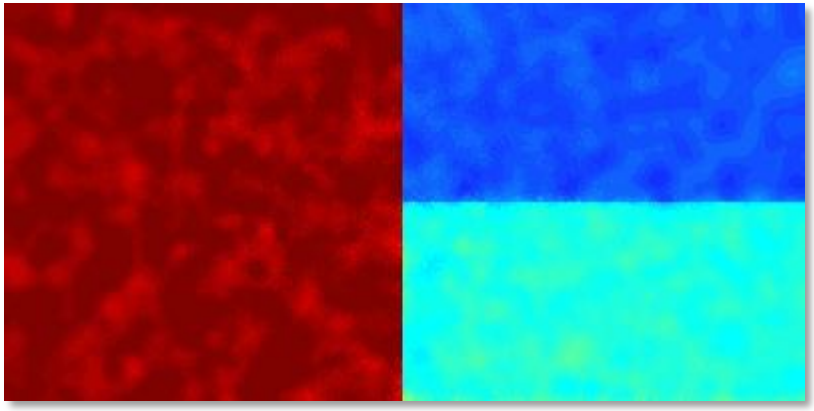
Filtre gaussien



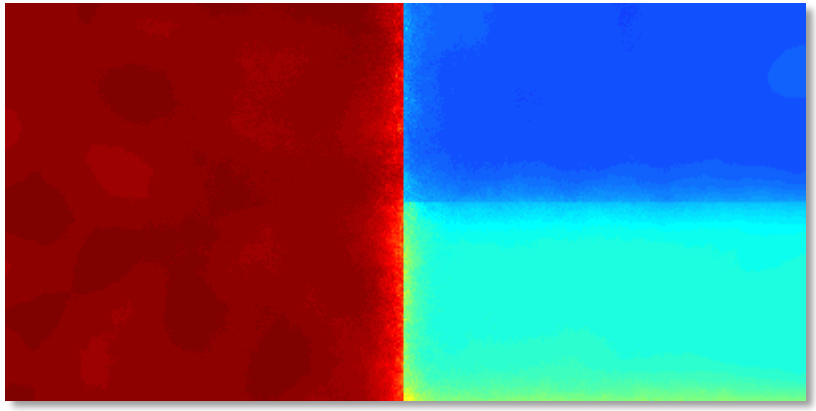
Filtre gaussien (σ=4)



Filtre gaussien (σ=12)



Filtre bilatéral (σ=4)



Filtre bilatéral (σ=12)

Débruitage

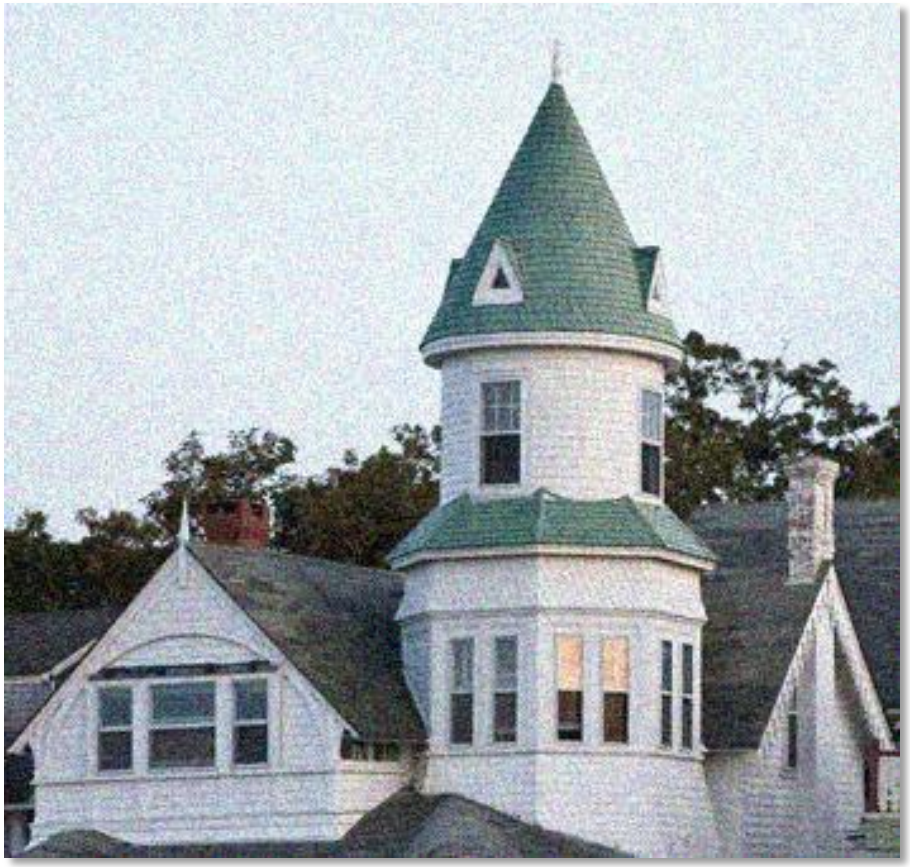


Image bruitée

Après filtrage gaussien

Débruitage

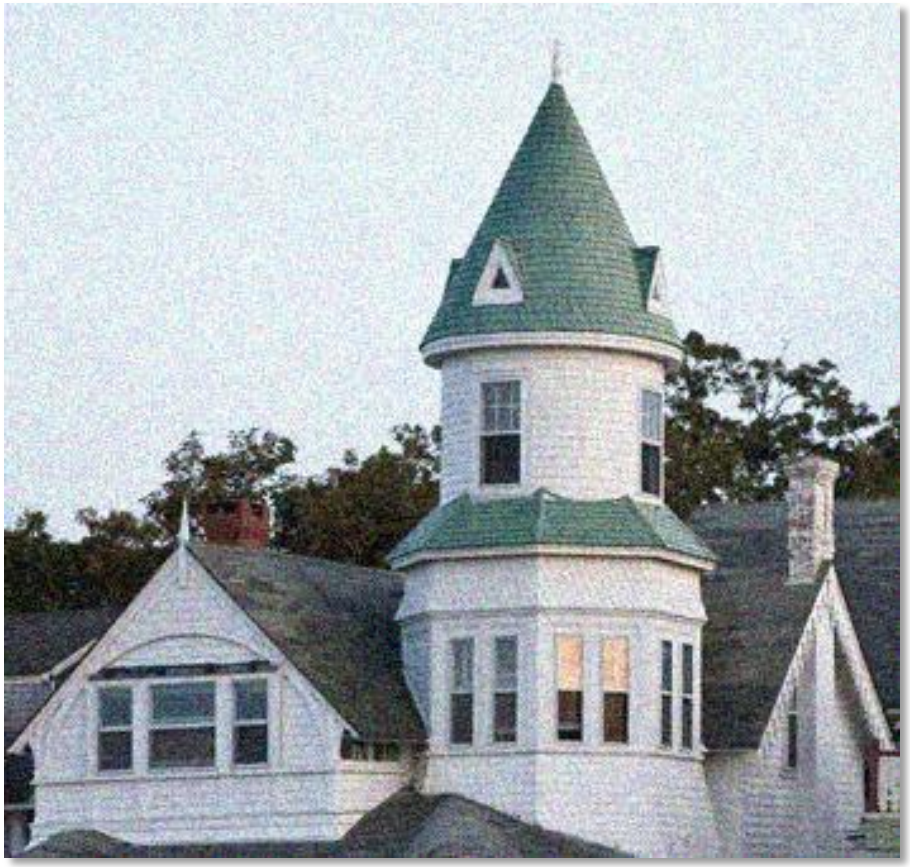
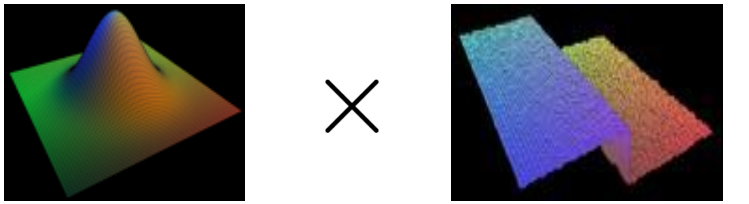


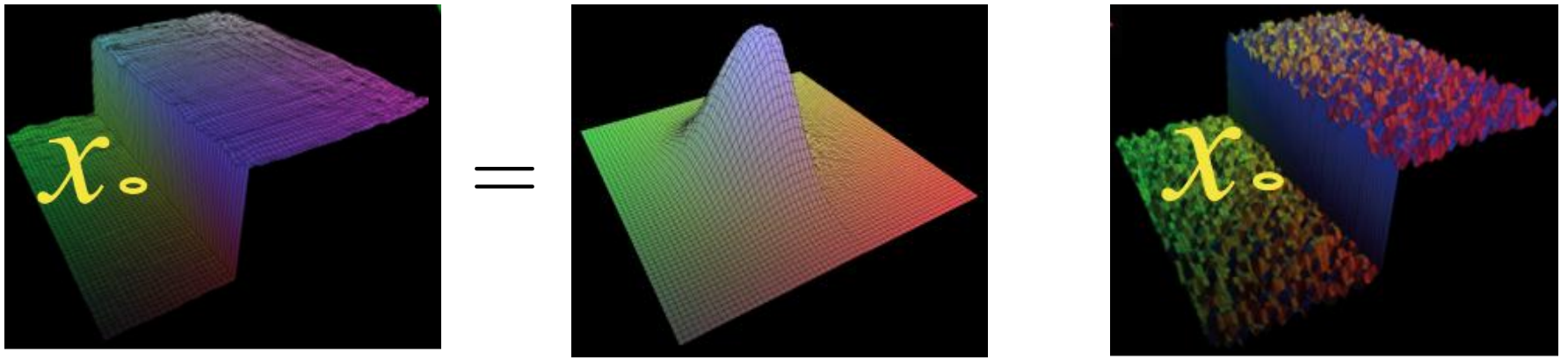
Image bruitée

Après filtrage bilatéral

Filtre non-linéaire

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

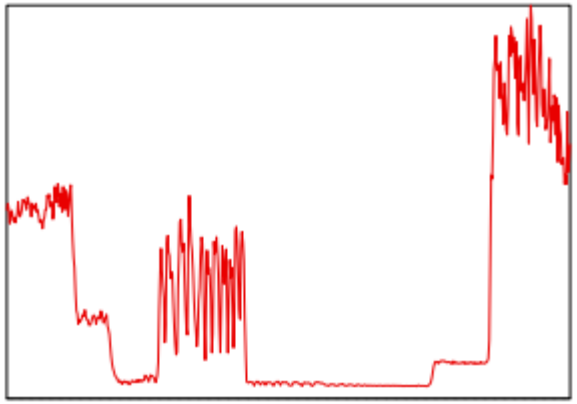


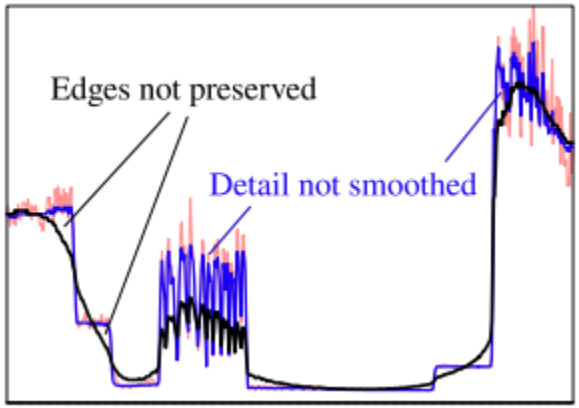


Limitation

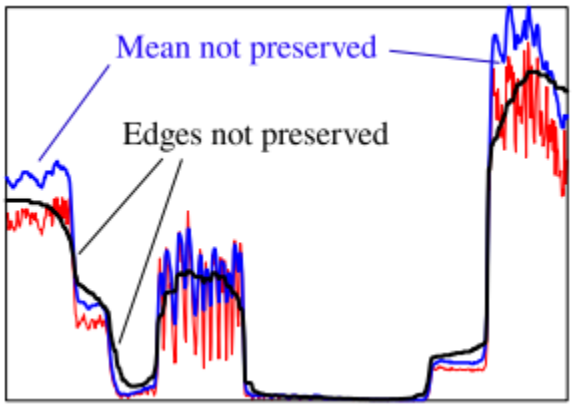
- Calcul « brute-force » (très) **lent**
  ⇨ bilateral grid [Chen et al. 2007]
- Préservation **partielle** des contours

Signal d'entrée

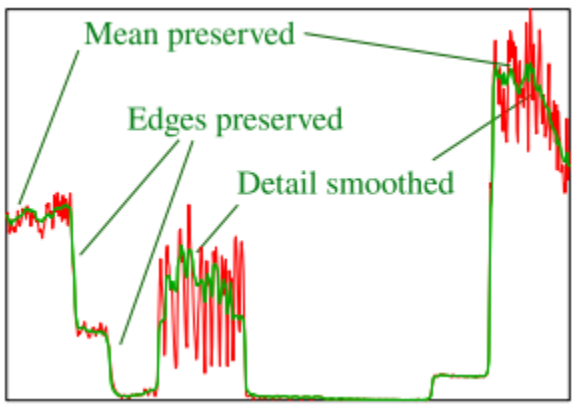




Filtre bilatéral



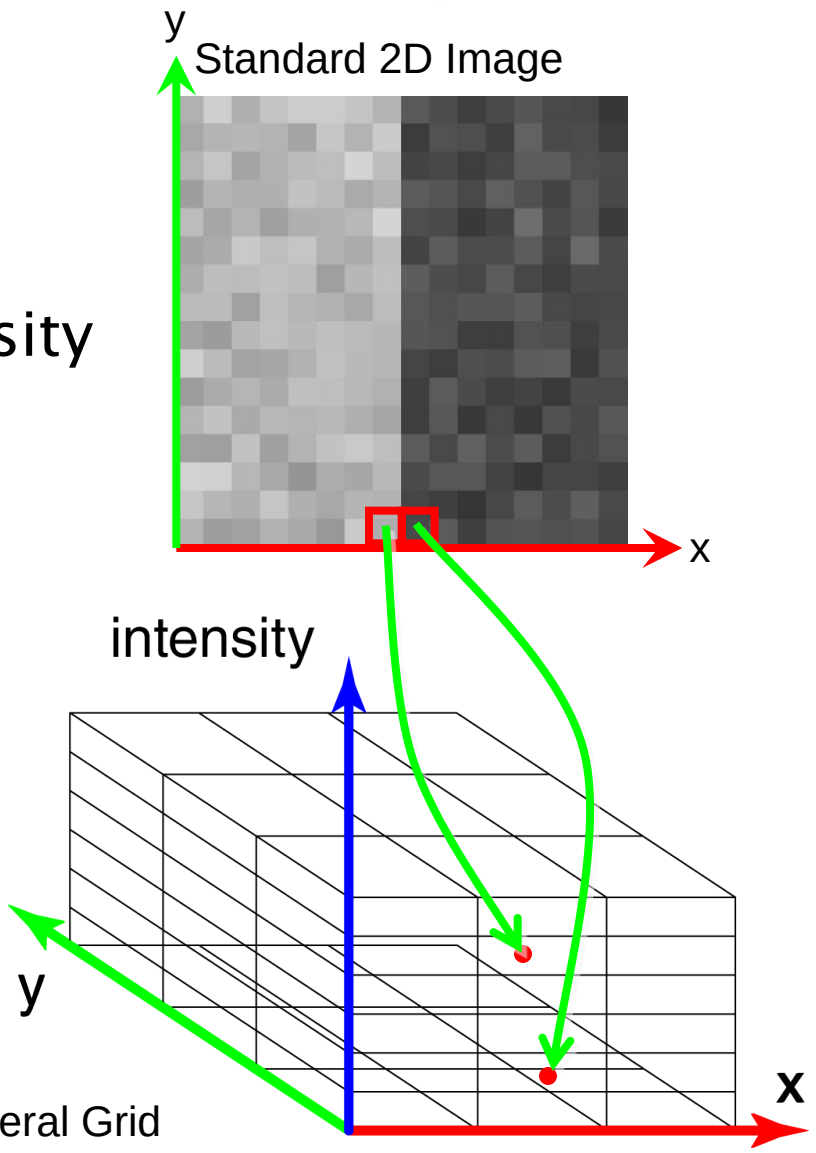
[Farbman et al. 2008]



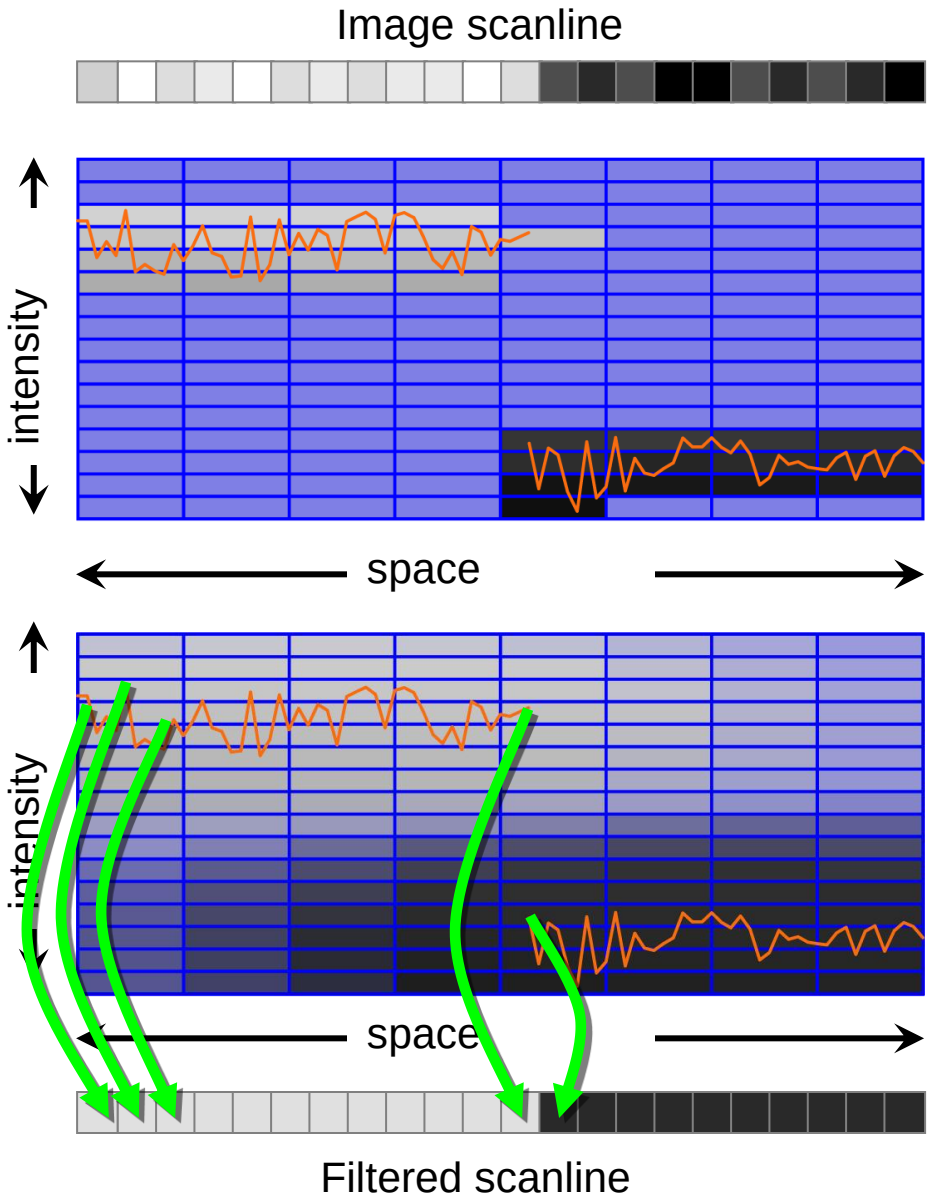
[Subr et al. 2009]

“Real-time Edge-Aware Image Processing with the Bilateral Grid”

- Bilateral grid = 3D array
  - x and y correspond to pixel position
  - z corresponds to pixel intensity
  - Euclidean distance accounts for edges
    - space distance (x,y) and intensity distance (z)
- Grid can be coarse
  - E.g., 70 x 70 x 10 for an 8 megapixel image



Bilateral Filter on the Bilateral Grid



Slice: query grid with input image

Performance

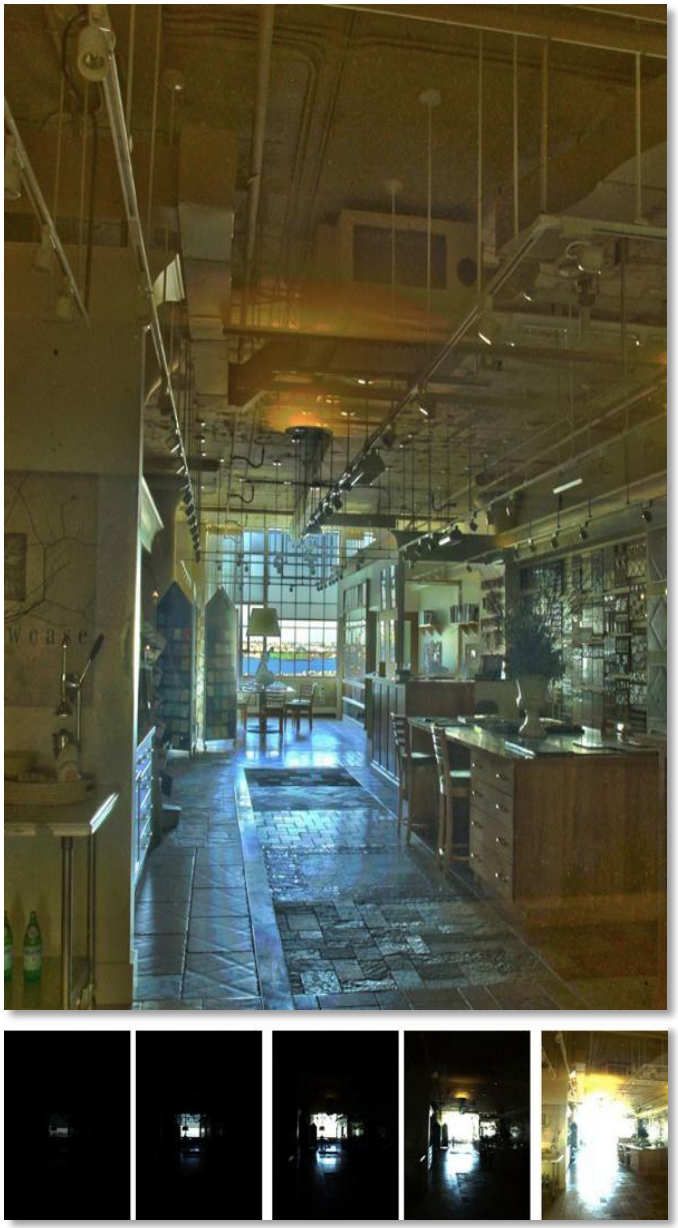
Image size: 2 MPixels

- CPU
  - Brute force: **10 minutes**
  - State of the art ’06: **1 second** [Weiss 06, Paris 06]
- Bilateral Grid with GPU
  - 2006 card (G80): **9 ms** (111 Hz)

Applications

- Égalisation locale d'histogrammes
- Tone mapping
- Abstraction de vidéos
- Transfert de style

[Durand 02]



“Real-time video abstraction”

[Willmöller et al. 2002]

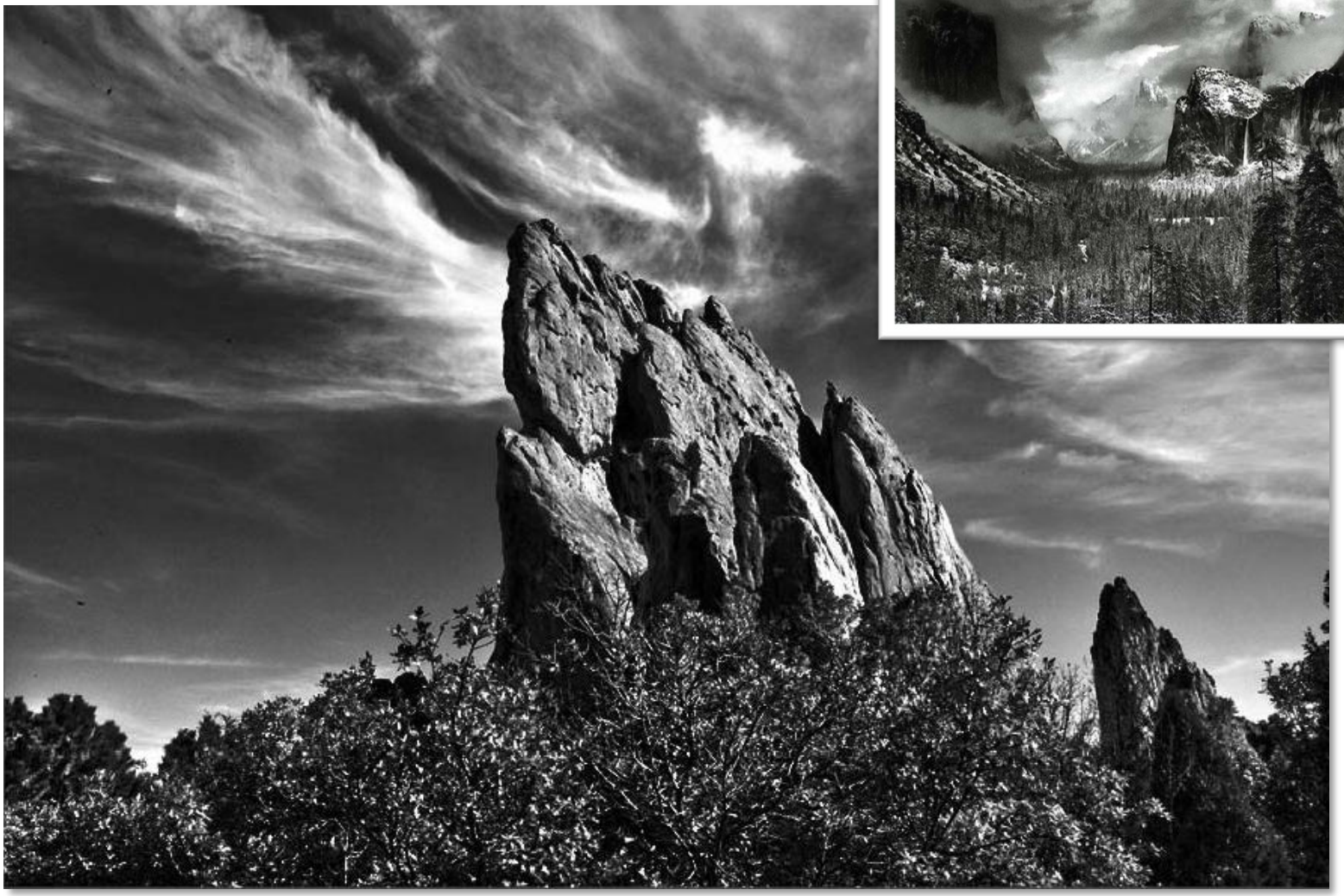
“Two-scale Tone Management for Photographic Look” [Bae et al. 2006]

Ansel Adams, *Clearing Winter Storm*

“Two-scale Tone Management for Photographic Look”

Photographie amateure

“Two-scale Tone Management for Photographic Look”



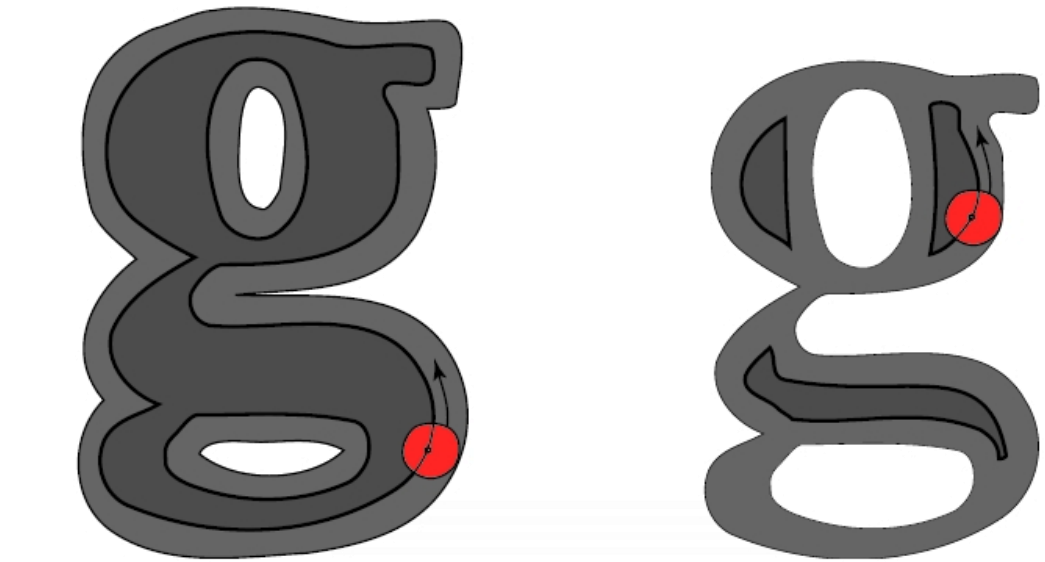
Apparence transférée

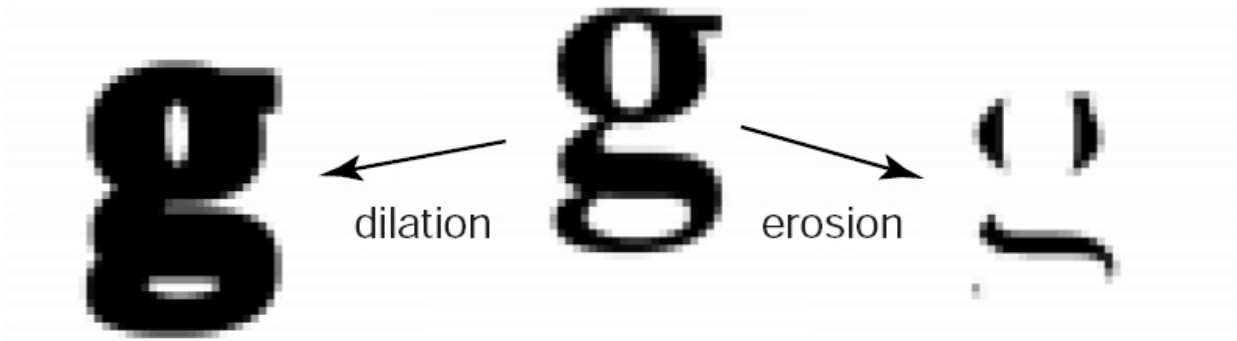
Opérateurs morphologiques

- Masques binaires de l'image
- Théorie des ensembles / logique binaire
- **Morpho-mathématiques** [Haralick87]

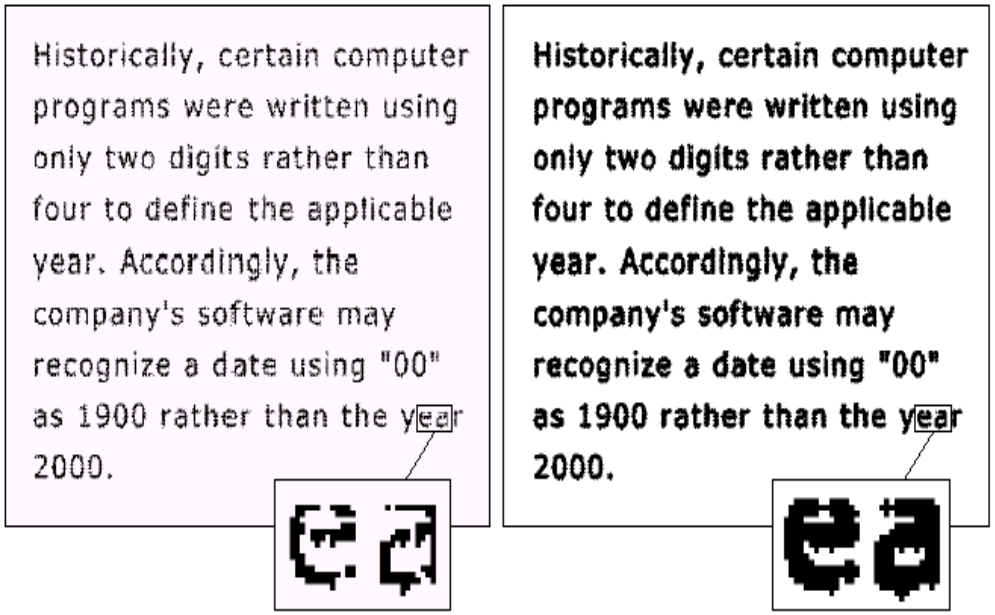
Dilatation / érosion

- Opérations de base sur l'image A selon le noyau B





Dilatation



« élément de structure » ↔ noyau :

0	1	0
1	1	1
0	1	0

Érosion

Image d'origine

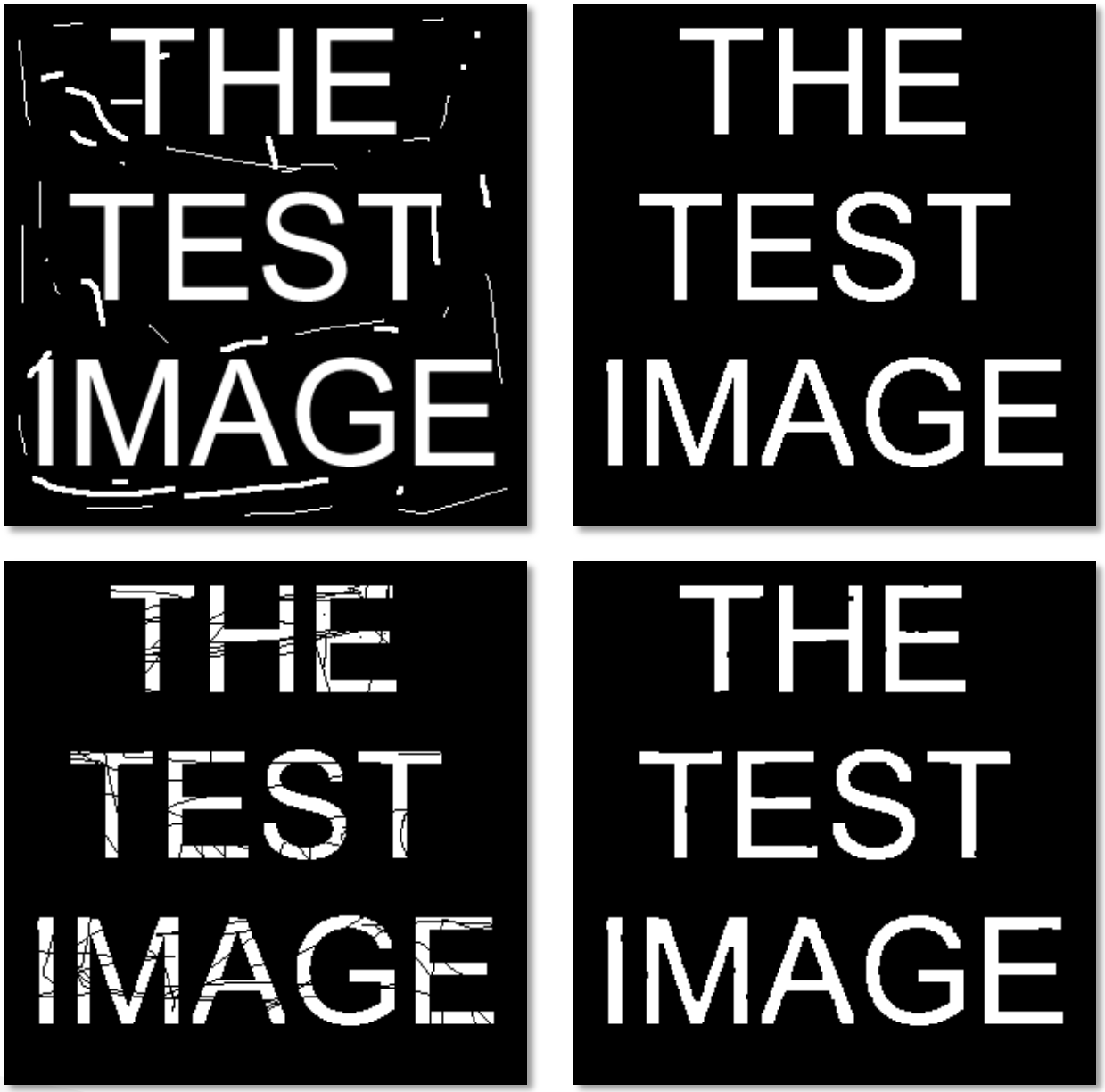
Image érodée 1x

Image érodée 2x

Ouverture et fermeture

Ouverture :
2 érosions puis
2 dilatations.

⇨ Suppression
du bruit



Fermeture:
dilatation
puis érosion.

⇨ Remplissage
des trous.

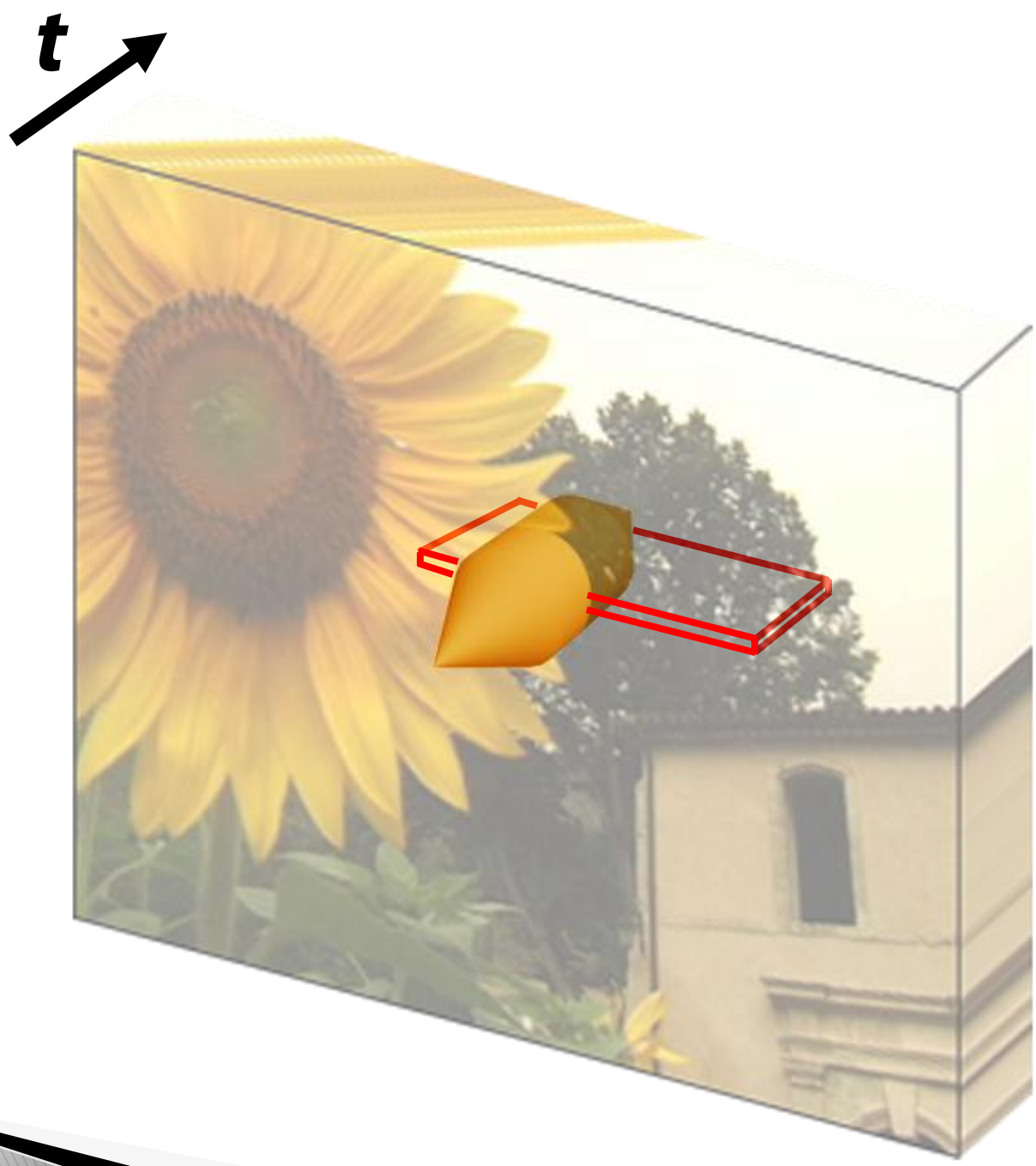
Application au NPR

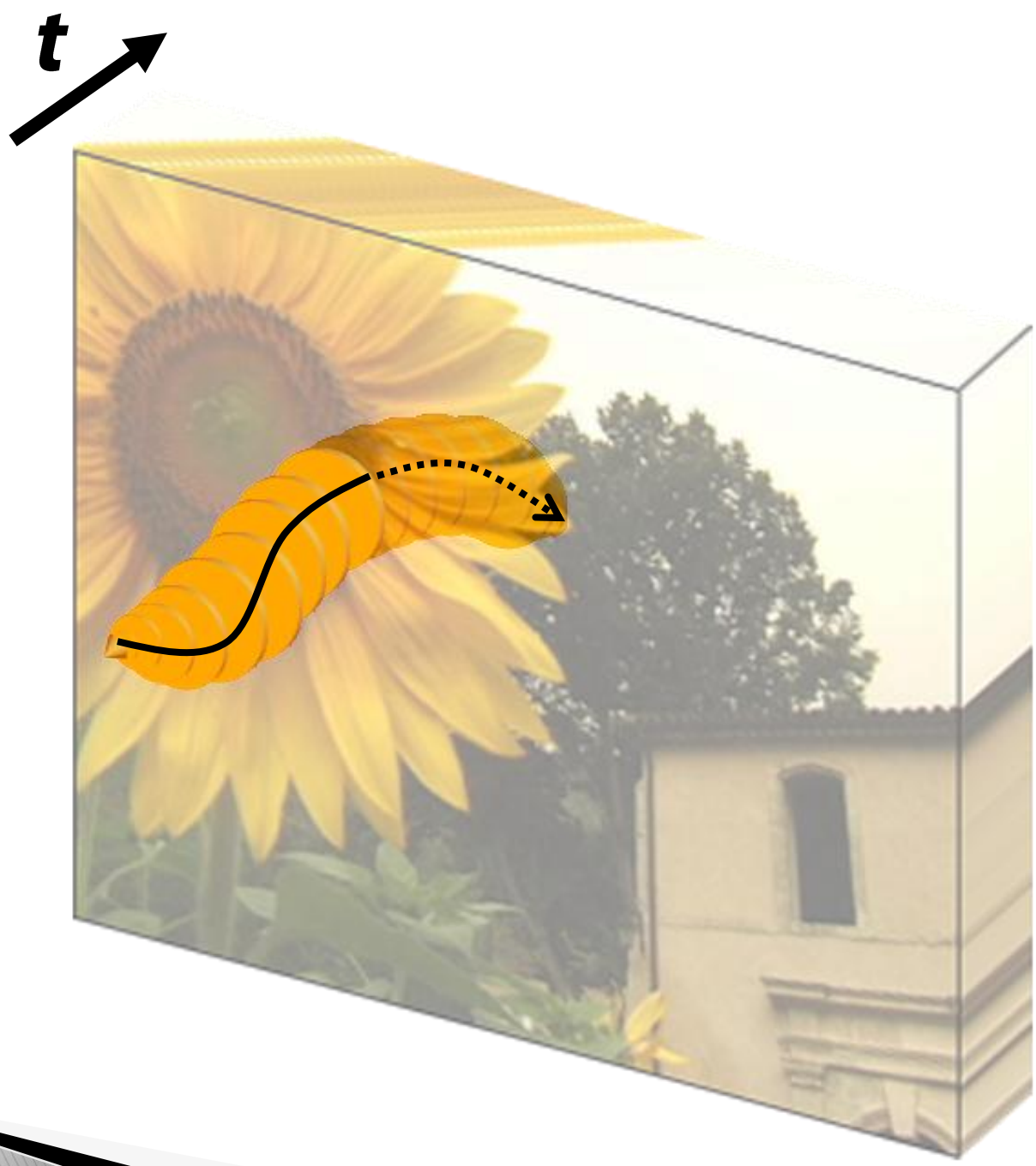
- “Video Watercolorization using Bidirectional Texture Advection” Bousseau et al. 2007

Ouverture

“Video Watercolorization using Bidirectional Texture Advection”

Filtrage spatial

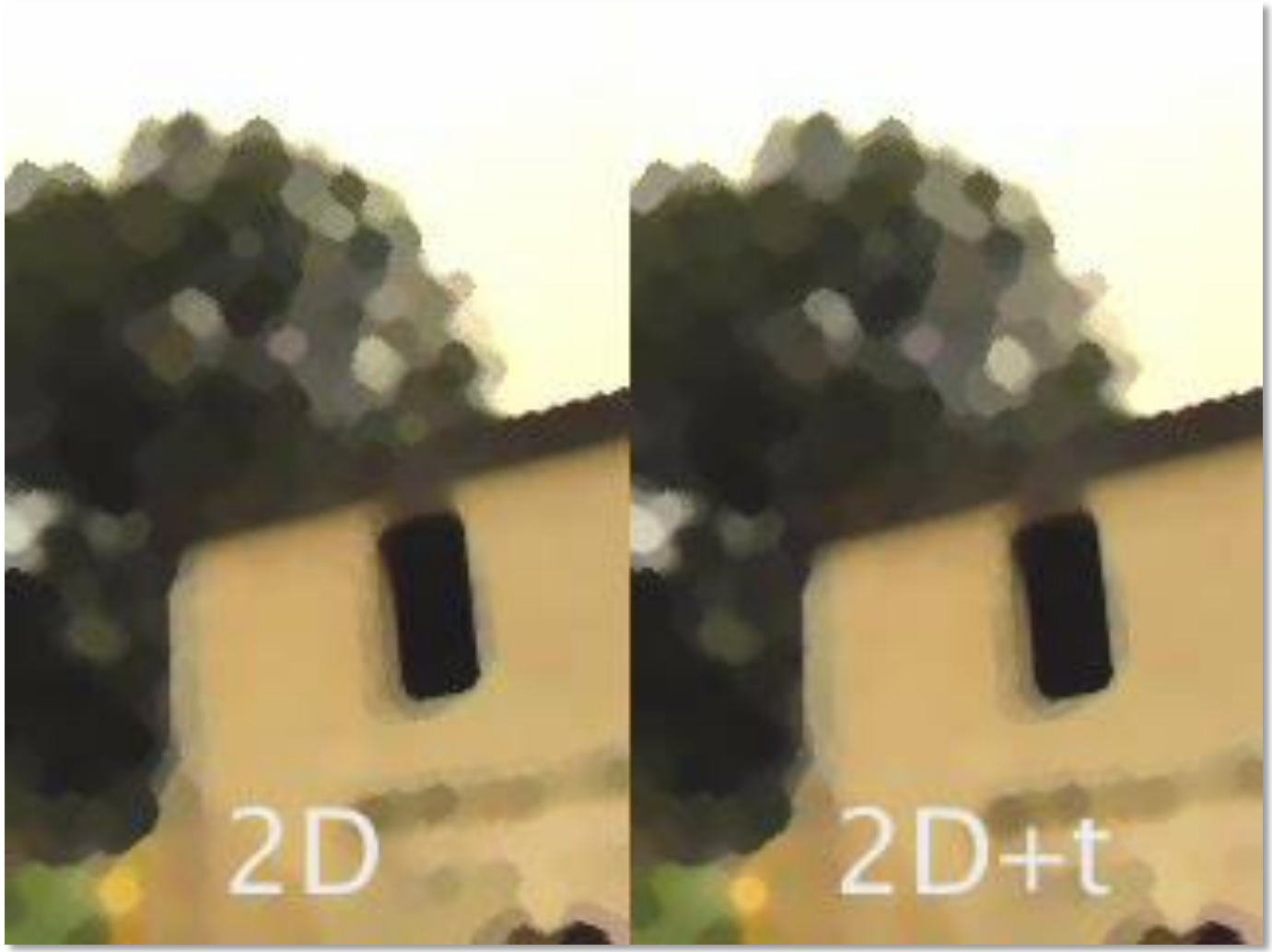

t


t

“Video Watercolorization using Bidirectional Texture Advection”

Filtrage spatio-temporel

“Video Watercolorization using Bidirectional Texture Advection”



Comparaison

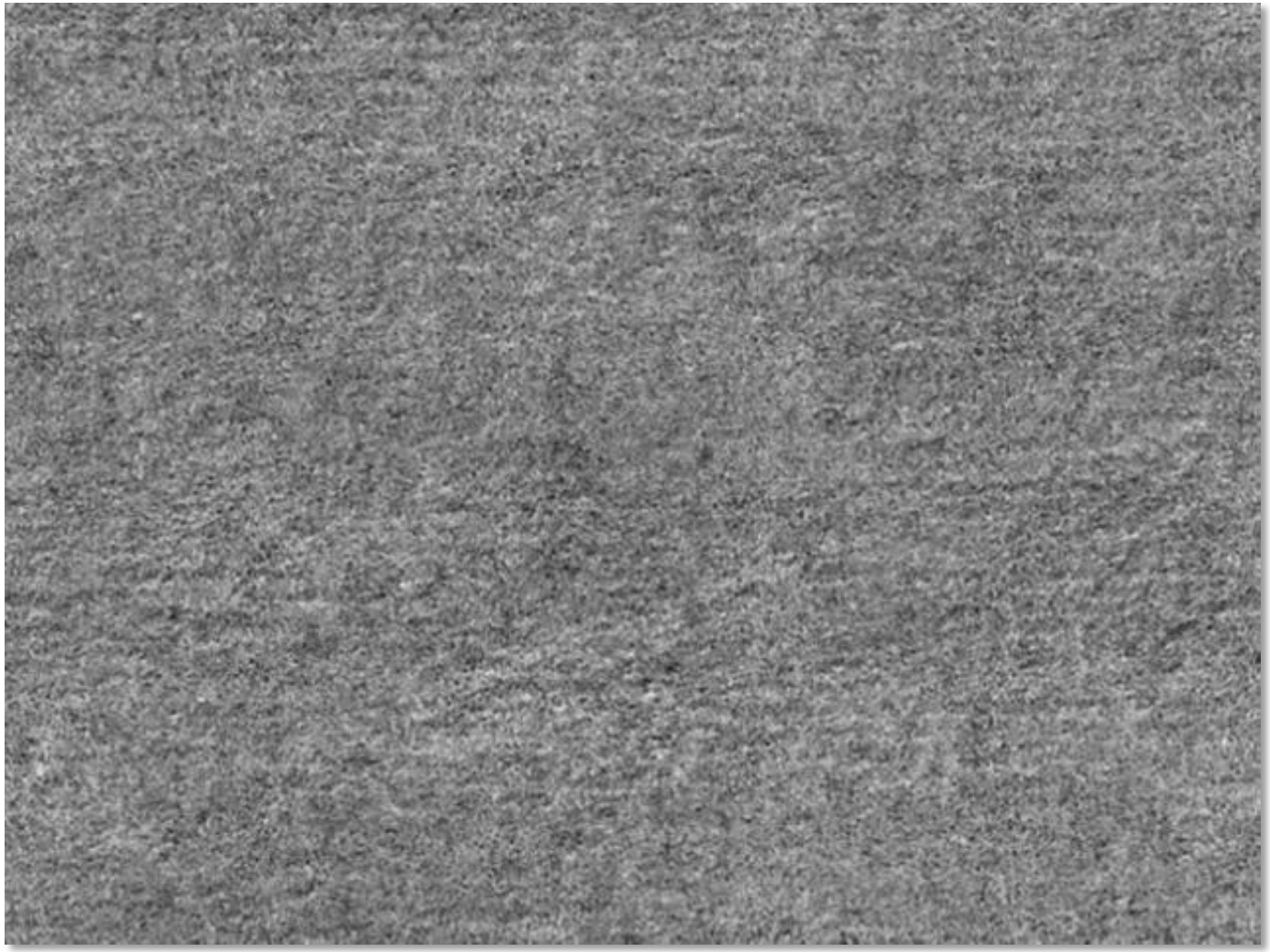
“Video Watercolorization using Bidirectional Texture Advection”

Vidéo d’origine

“Video Watercolorization using Bidirectional Texture Advection”

Filtrage spatio-temporel

“Video Watercolorization using Bidirectional Texture Advection”



Advection de texture

“Video Watercolorization using Bidirectional Texture Advection”

Stylisation aquarelle

Plan

- **Computational processing**
  - Image retargeting
  - Filtering
  - **Image Warping & Morphing**
  - Compositing & Matting
  - Gradient Editing
- Computational illumination
- Computational optics

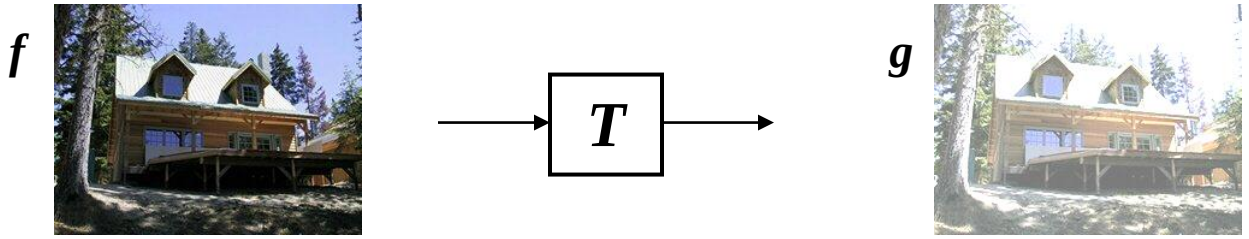
Image Warping & Morphing

http://youtube.com/watch?v=nUDIoN-_Hxs

Image Warping

- **Filtrage** : modifie la plage de valeur

$$g(x) = T(f(x))$$



- **Warping** : modifie le domaine de l'image

$$g(x) = f(T(x))$$

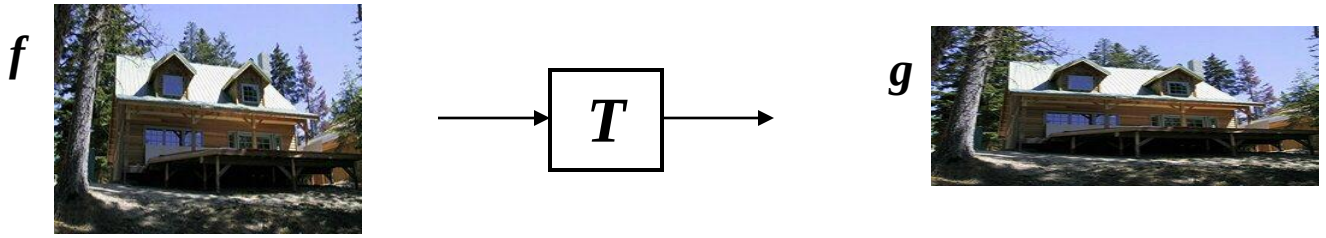
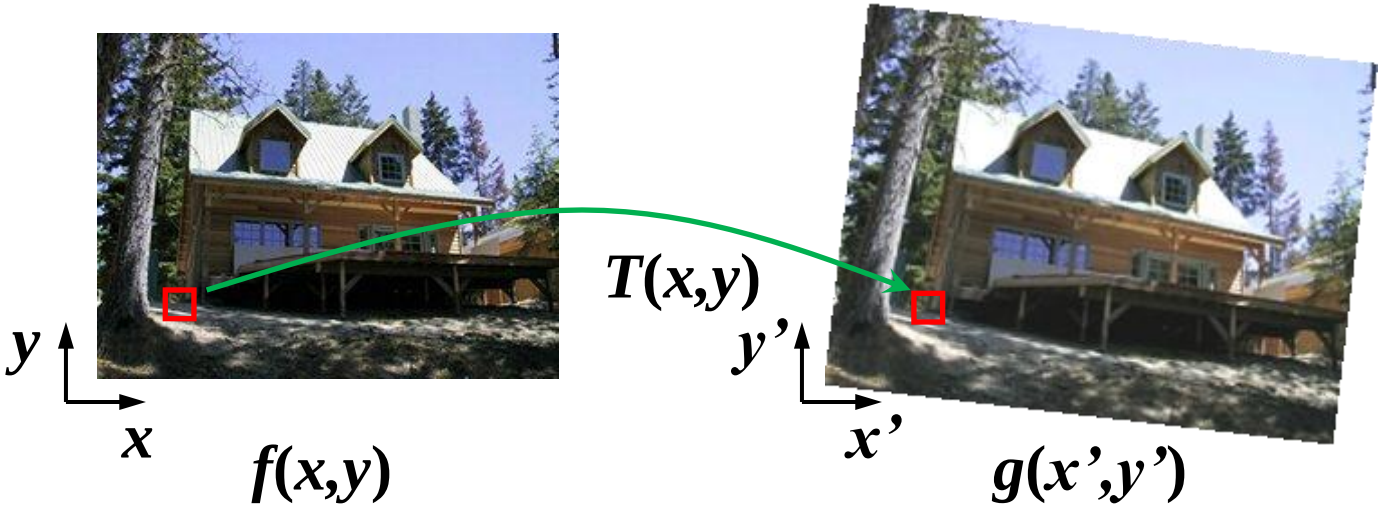
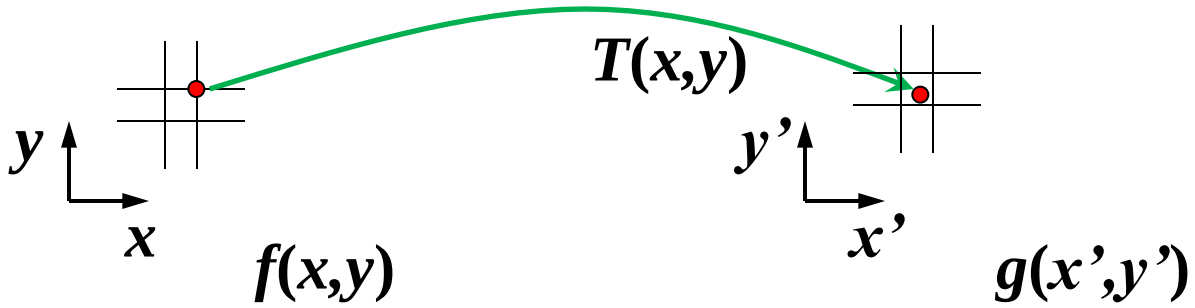


Image Warping



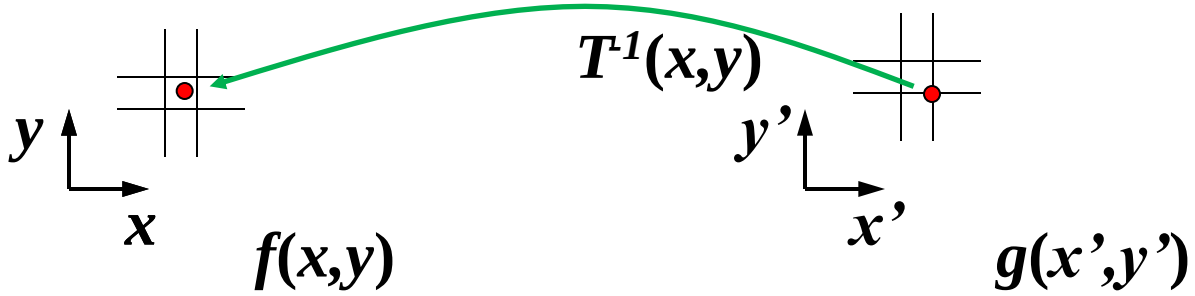
Connaissant la transformation (x',y')=T(x,y) et l'image source f(x,y), comment calculer l'image transformée g(x',y')=f(T(x,t)) ?

Forward warping



- Envoyer chaque pixel f(x,y) à la position (x',y') = T(x,y) dans la deuxième image
- Distribuer la couleur sur les pixels voisins (splating)

Inverse warping



- Chercher la position de chaque pixel g(x',y') dans l'image d'origine : (x,y) = T^{-1}(x',y')
- Interpoler entre les différentes couleur (plus proche voisin, bilinéaire…)

Forward vs. Inverse

- Quelle est la meilleure méthode ?

Forward vs. Inverse

- Quelle est la meilleure méthode ?
- Généralement le **warping inverse**
  - évite les trous
  - mais la fonction doit être inversible (ce qui n'est pas toujours le cas)

Morphing



- « Moyenne » entre deux images
  - Pas la moyenne de l'image des objets…
  - …mais une image de la moyenne des objets
  - et une moyenne évoluant au cours du temps.
- Comment savoir ce qu'est la bonne moyenne ?
  - On n'en sait rien !
  - Mais les artistes peuvent nous aider

Fondu : « Cross-dissolve »

- Interpolation de l'image complète
  - $I_t = (1-t) * I_1 + t * I_2$
- Mais que se passe-t-il si les images ne sont pas alignées ?

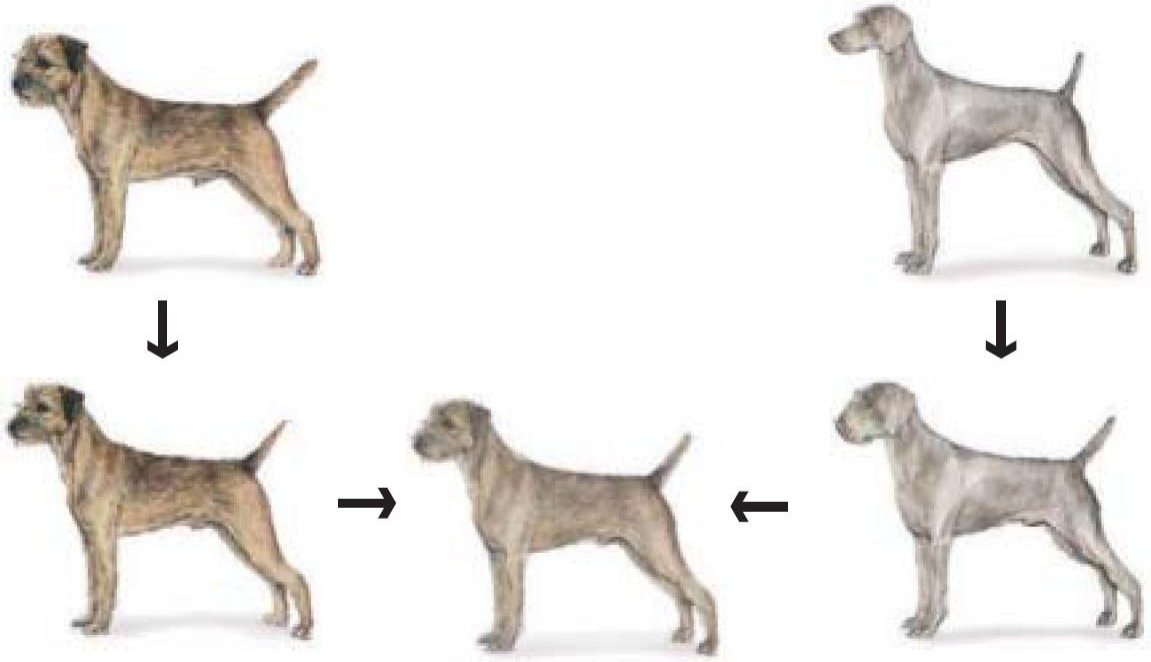
Alignement puis fondu

- Aligner d'abord (wrap global), puis faire un fondu

Et pour le chien ?

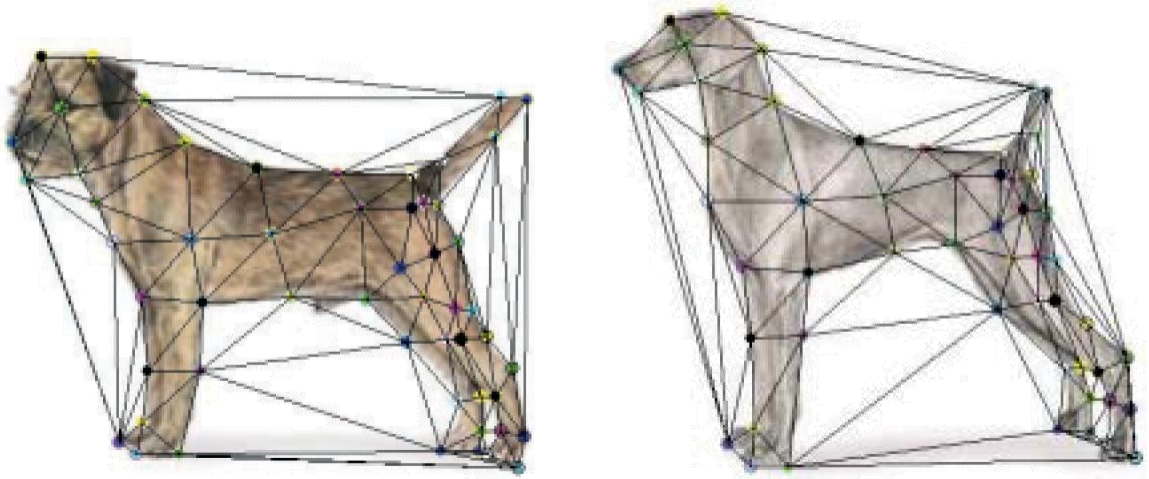
- Fondu ne marche pas
- Alignement global ne marche pas

Warping local, puis fondu



Comment spécifier la transformation ?

Solution 1 : Maillage



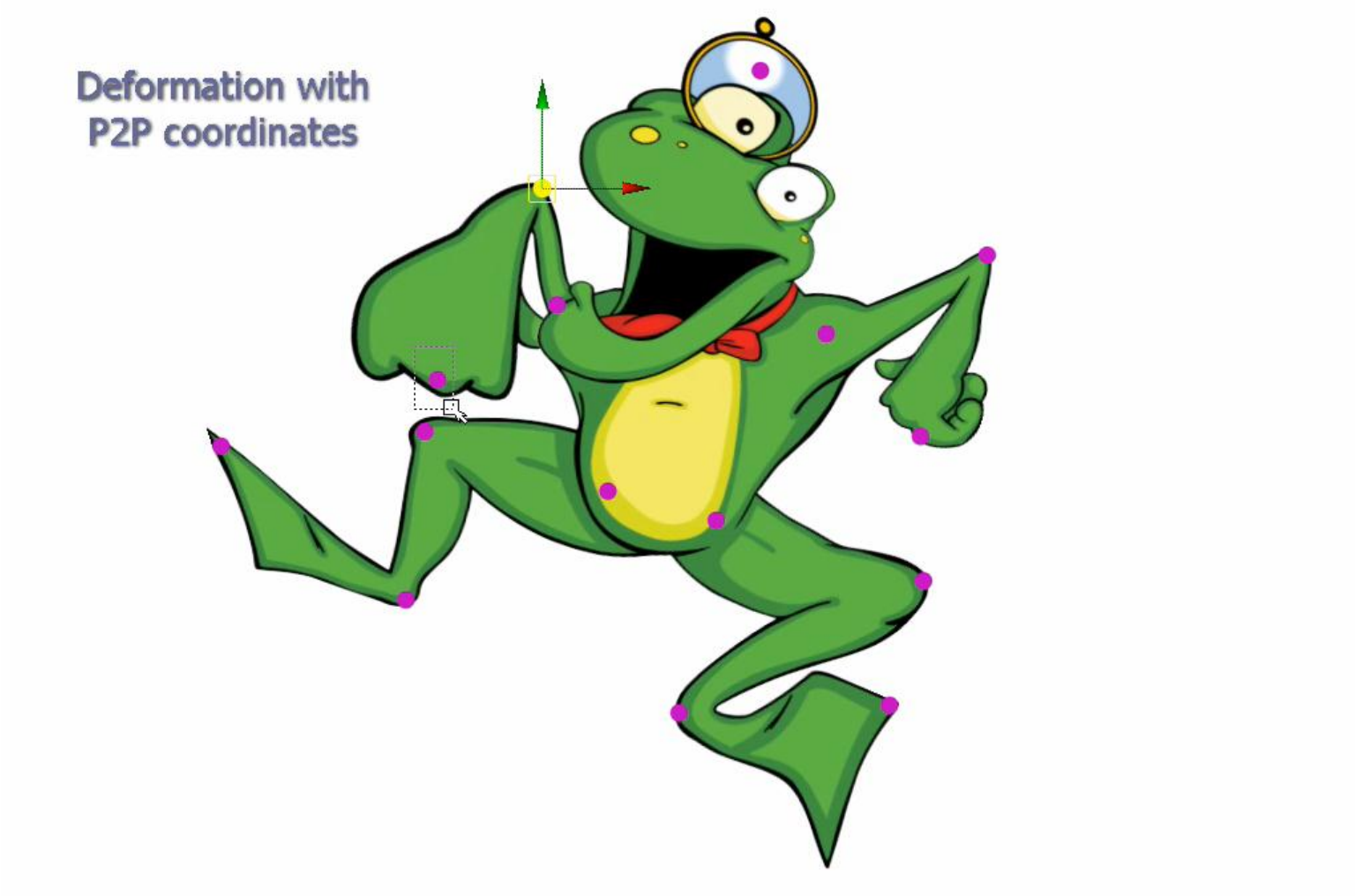
- Définir des points de correspondance
- En déduire une triangulation (Delaunay)
- Et la déformation de chaque triangle (transformation affine = texture mapping)

Solution 2 : autres coordonnées

- “Mean Value Coordinates” [Floater03]
- “Harmonic Coordinates” [DM07]
- “Green Coordinates” [LLC08]
- “Complex Barycentric Coordinates” [OBG09]

Solution 2 : autres coordonnées

Solution 2 : autres coordonnées



Plan

- **Computational processing**
  - Image retargeting
  - Filtering
  - Image Warping & Morphing
  - **Compositing & Matting**
  - Gradient Editing
- Computational illumination
- Computational optics

Comment volent les super-héros ?

Superpouvoirs ?

ou

Image matting ?

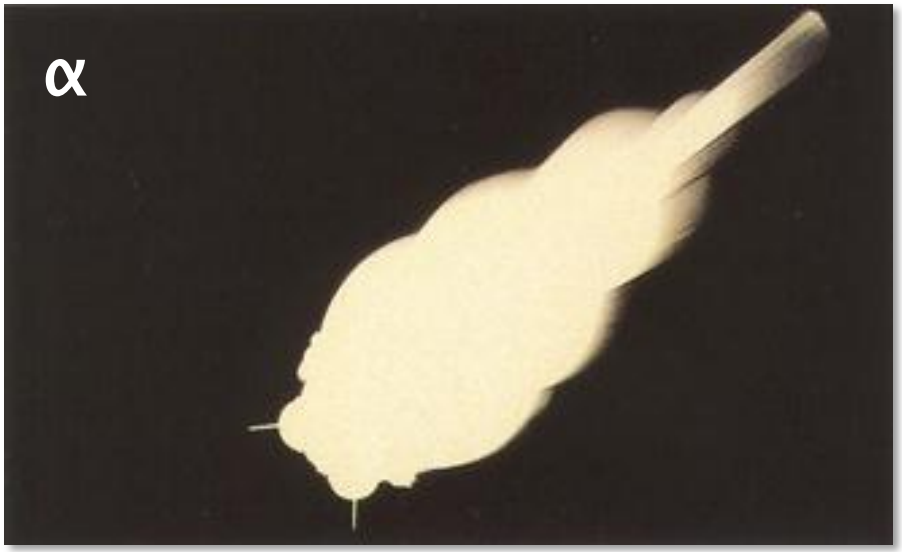
Image matting – Compositing

Cinefex

Compositing

Digital Domain

- Alpha-mask
- $\alpha \in [0;1]$
- $C=\alpha F+(1-\alpha)B$



Alpha binaire

Digital Domain

Alpha continue

Digital Domain

L’écran bleu

L'écran bleu

- Matting le plus courant à la TV et au cinéma
- Inventé par **Petros Vlahos** dans les année 50 (Technical Academy Award, 1995)
- Limitation : pas de bleu dans le premier plan
- Heuristique : $\alpha = 1 - p_1(b - p_2\, g)$
  - $b < p_2\, g$, avec a_2 entre 0.5 et 1.5
  - p_1 et p_2 définis par l'utilisateur

Natural matting : ambiguïté

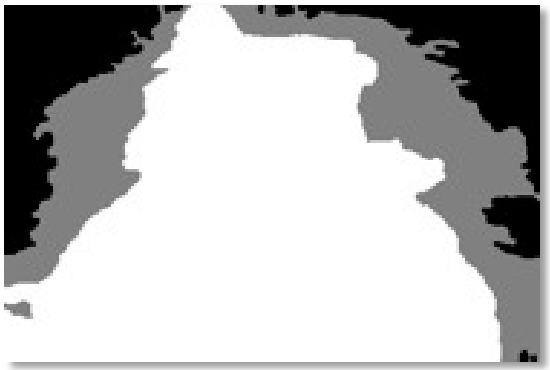
- **7 inconnues** :
  - α
  - $F=(R_F, G_F, B_F)$
  - $B=(R_B, G_B, B_B)$
- **3 équations** : une par canal de couleur

Natural matting

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

- Image à l'arrière plan quelconque
- **Trimap** grossière fournie par l'utilisateur
  - B connu en noir,
  - F connu en blanc
  - Inconnu en gris
- **Objectif** : estimer F, B, α dans la zone inconnue

Image d'entrée



Trimap

Bayesian matting

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

le paramètre à estimer | ce qu'on observe

fonction de vraisemblance

probabilité a priori

constante vis-à-vis de x

Bayesian matting

- Ce qu'on observe : **la couleur C**

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

le paramètre à estimer | fonction de vraisemblance | constante vis-à-vis de x

ce qu'on observe | probabilité a priori

Bayesian matting

- Ce qu'on observe : la couleur C
- Ce qu'on veut estimer : **F, B, α**

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

le paramètre à estimer

ce qu'on observe

fonction de vraisemblance

probabilité a priori

constante vis-à-vis de x

Bayesian matting

- Ce qu'on observe : la couleur C
- Ce qu'on veut estimer : F, B, α
- **Fonction de vraisemblance**
  - Connaissant F, B, α, probabilité d'observer C
  - Mesures parfaites ⇨ $C=\alpha F+(1-\alpha)B$
  - En pratique, hypothèse de bruit Gaussien de variance σ_C (+lissage de la probabilité)

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

le paramètre à estimer

ce qu'on observe

fonction de vraisemblance

probabilité a priori

constante vis-à-vis de x

Bayesian matting

- Ce qu'on observe : la couleur C
- Ce qu'on veut estimer : F, B, α
- Fonction de vraisemblance
- **Probabilité a priori**
  - Construire une distribution de probabilité d'après les régions connues de la **trimap**
  - Cœur du « Bayesian matting »

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

le paramètre à estimer

ce qu'on observe

fonction de vraisemblance

probabilité a priori

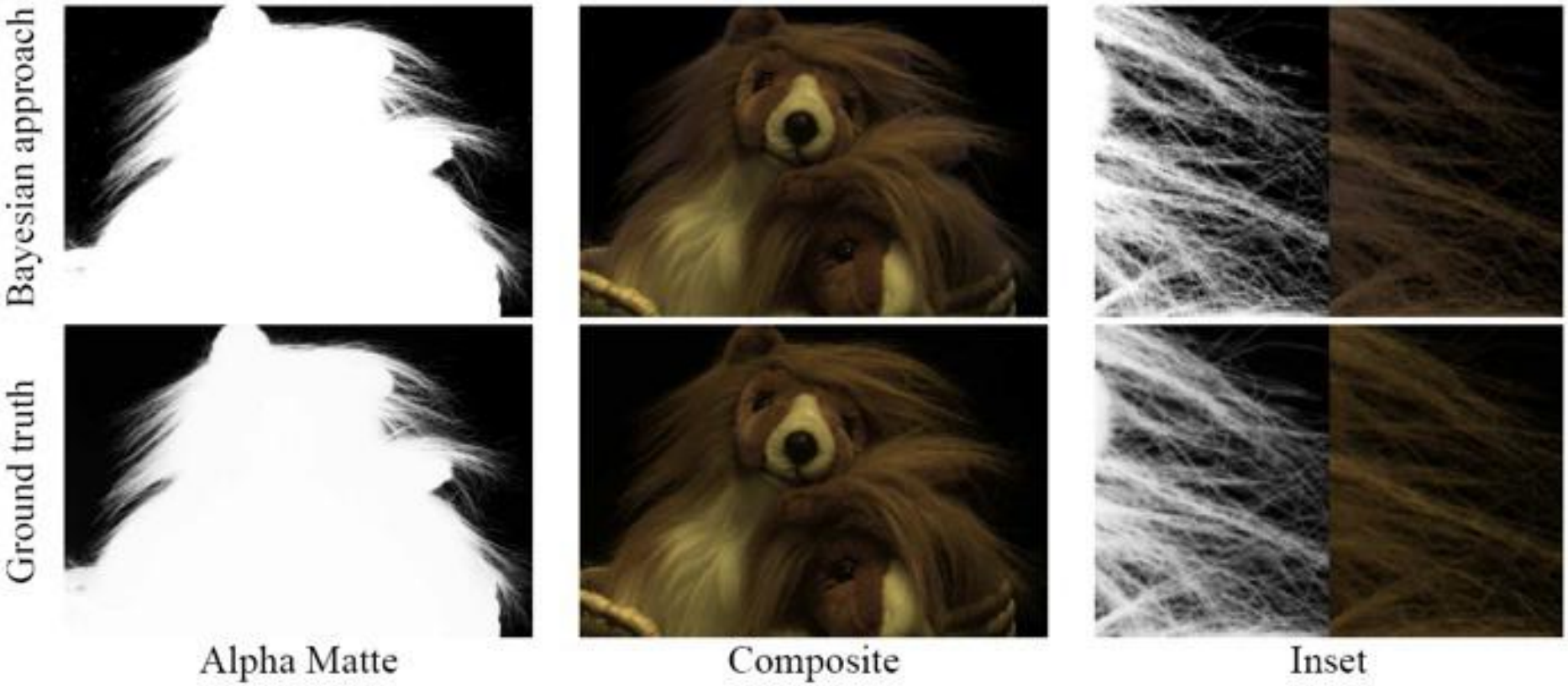
constante vis-à-vis de x

Bayesian matting

- Hypothèse : F, B, α **indépendants**
  $P(F,B,\alpha|C) = P(C|F,B,\alpha)\ P(F,B,\alpha)\ /\ P(C)$
  $= P(C|F,B,\alpha)\ \mathbf{P(F)\ P(B)\ P(\alpha)}/P(C)$
- Passage au log (supprimer les multiplications)
  $\mathbf{L(F,B,\alpha|C) = L(C|F,B,\alpha) + L(F) + L(B) + L(\alpha) - L(C)}$
- Ignorer la constante L(C)
- Maximiser itérativement :
  $\mathbf{L(C|F,B,\alpha) + L(F) + L(B) + L(\alpha)}$

Bayesian matting

Chuang et al. 2001



Bayesian matting

Chuang et al. 2001

Alpha Matte

Composite

Inset

Environment matting

- Effets optiques plus complexes
- Chaque pixel peut dépendre de **plusieurs pixels** de l'arrière plan

[Chuang, Zongker, Hindorff, Curless, Salesin and Szeliski *1999-2000*]

Environment Matting Equation

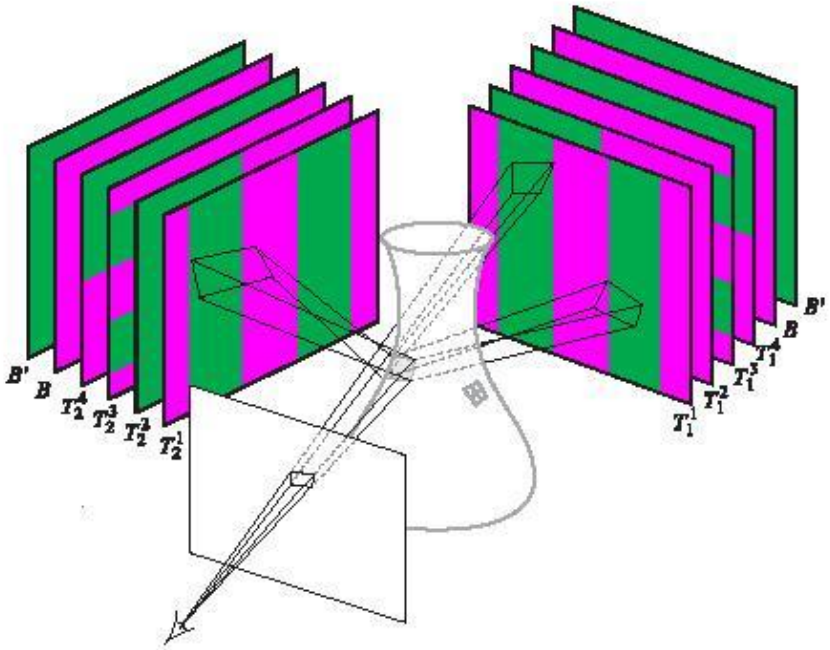
$$C = F + (1 - \alpha)B + \Phi$$

- α ~ amount of light that passes through the foreground
- Φ ~ contribution of light from Environment that travels through the object

Explanation of Φ

$$\Phi = \sum_{i=1}^{m} \int R_i(\boldsymbol{x})\, T_i(\boldsymbol{x})\, d\boldsymbol{x}$$

R – reflectance image
T – Texture image



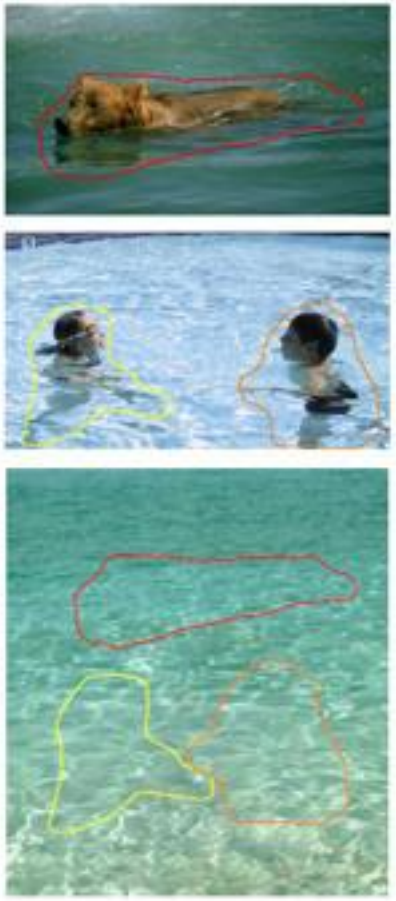
Environment matting

Alpha Matte Environment Matte Photographie

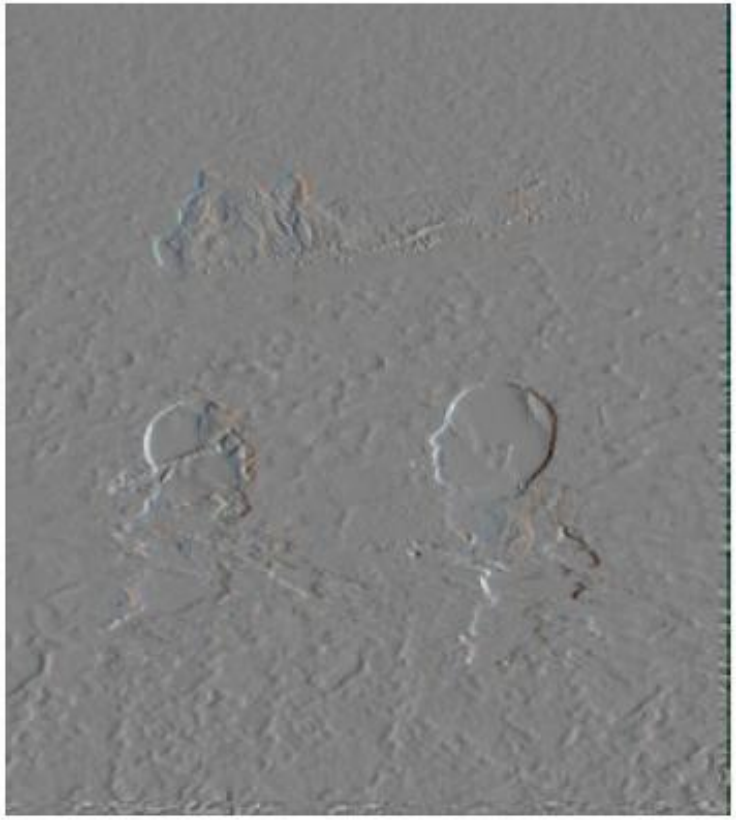
Plan

- **Computational processing**
  - Image retargeting
  - Filtering
  - Image Warping & Morphing
  - Compositing & Matting
  - **Gradient Editing**
- Computational illumination
- Computational optics

Gradient Domain Blending



sources/destination



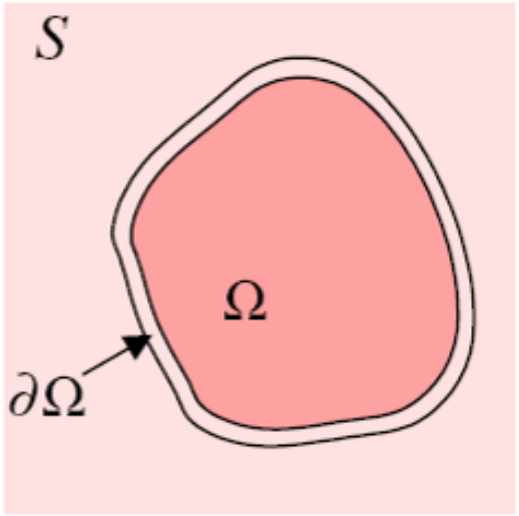
gradient visualization

seamless cloning

[Perez et al. 2003]

Gradient Domain Blending

- **Copier le gradient** de l'image source (sélection Ω) dans l'image destination S
- Rendre le nouveau gradient **aussi proche que possible** du gradient source en tenant compte des **valeurs au bord** ∂Ω

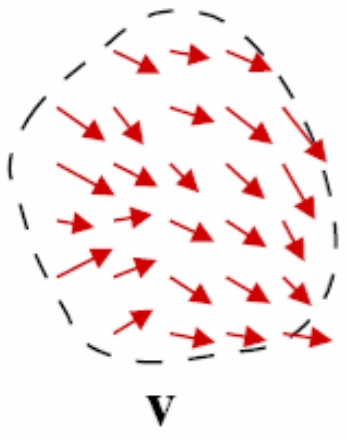


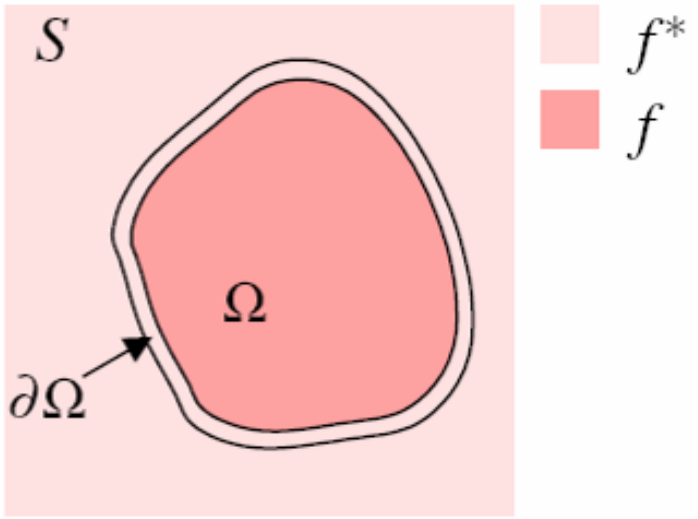
Gradient Domain Blending

- Connaissant le gradient source (champ de vecteur **v**), trouver **f** dans la zone **g** qui minimise :

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

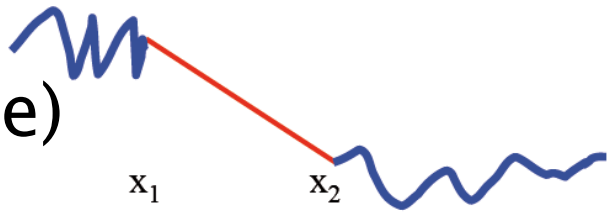
⇨ équation de Poisson avec conditions au bord de Dirichlet

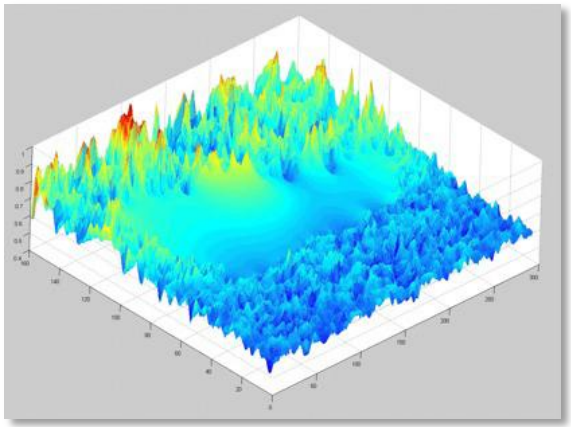




Gradient Domain Blending

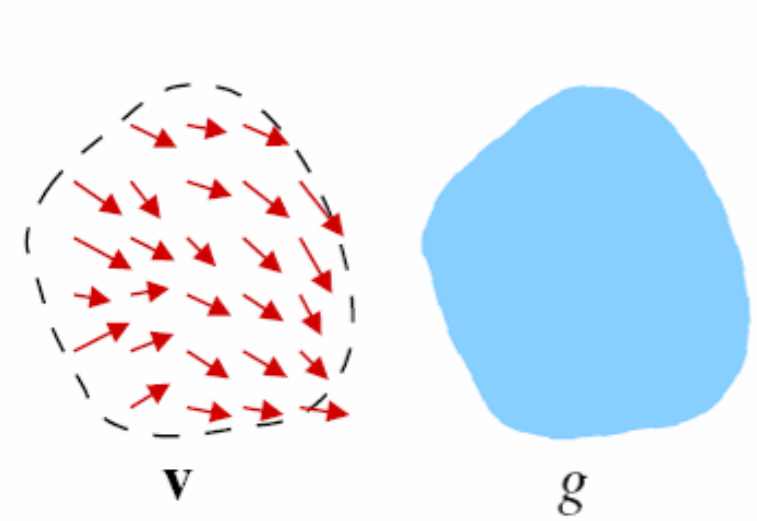
- Si **v** est nul
- Equation de Laplace (membrane)
- En 1D = interpolation linéaire
- En 2D :

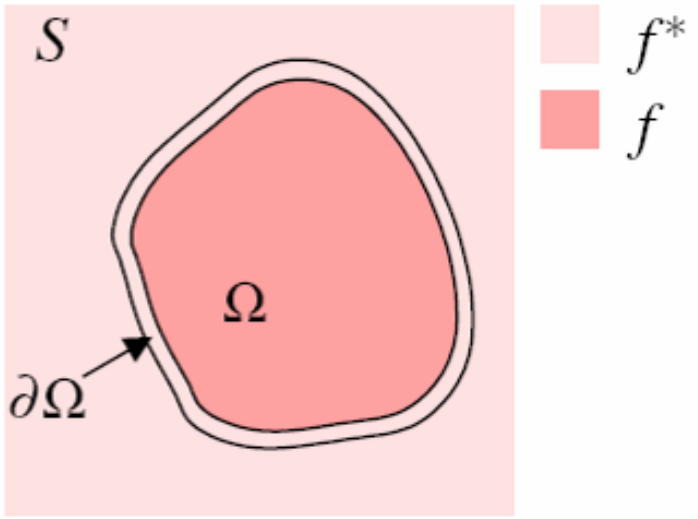




Gradient Domain Blending

- Si **v** n'est pas nul et conservatif (gradient de g)
- Fonction de correction $\hat{f}$ telle que $f = g + \hat{f}$
- $\hat{f}$ interpolant membrane de $(f^* - g)$ sur Ω

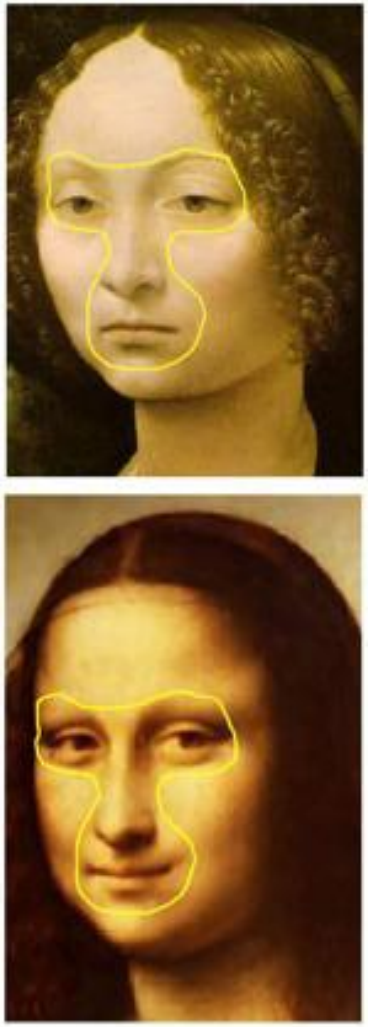




Gradient Domain Blending

- Minimisation au sens des moindres carrés
- Discrétisation ⇨ gros système d'équations linéaire creux à résoudre
  - Région de 100x100 pixels
  - 10 000 inconnues
  - matrice 10 000x10 000 !!!
- Iterative solvers, FFT, deconvolution, multigrid solvers…

Gradient Domain Blending



sources/destination

cloning

seamless cloning

Local color changes

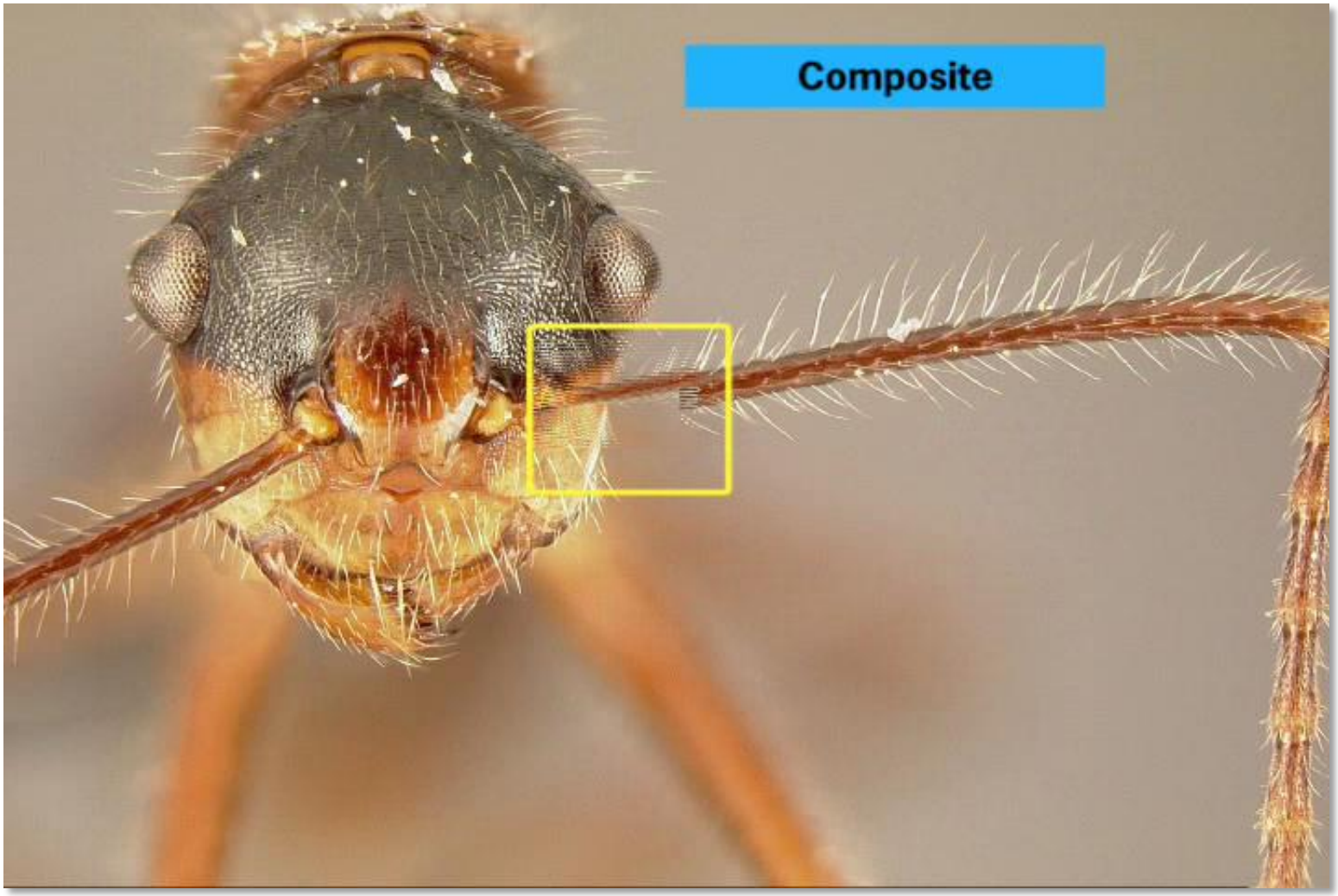
Limitations

- Pas d'inversion du contraste
  (gris sur noir => gris sur blanc)
- "Bleeding"
- Alignement quasi parfait des images

Applications

- Montage photo [Agrawala et al. 2002]
  - compositing, depth of field, panorama…

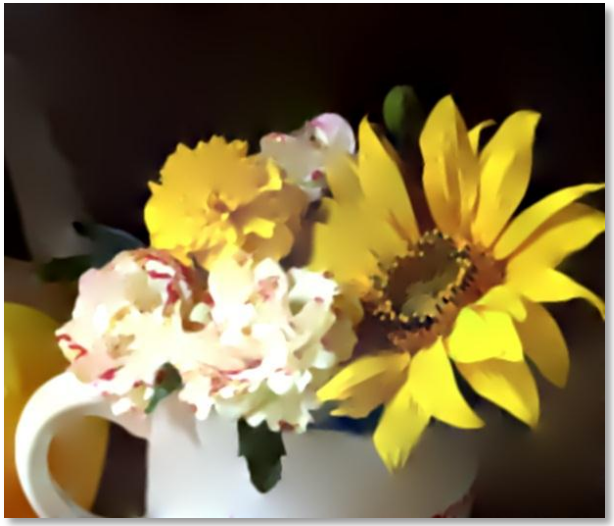
“Interactive Digital Photomontage”



[Agarwala et al. 2004]

Applications

- Montage photo [Agrawala et al. 2002]
  - compositing, depth of field, panorama…
- Tone-mapping [Fattal et al. 2002]
- Abstraction + stylisation [Orzan et al. 2007]


Applications

- Montage photo [Agrawala et al. 2002]
  - compositing, depth of field, panorama…
- Tone-mapping [Fattal et al. 2002]
- Abstraction + stylisation [Orzan et al. 2007]
- Painting [McCann et al. 2008]
  Drawing [Orzan et al. 2008]

[McCann et al. 2008]

[Orzan et al. 2008]

Plan

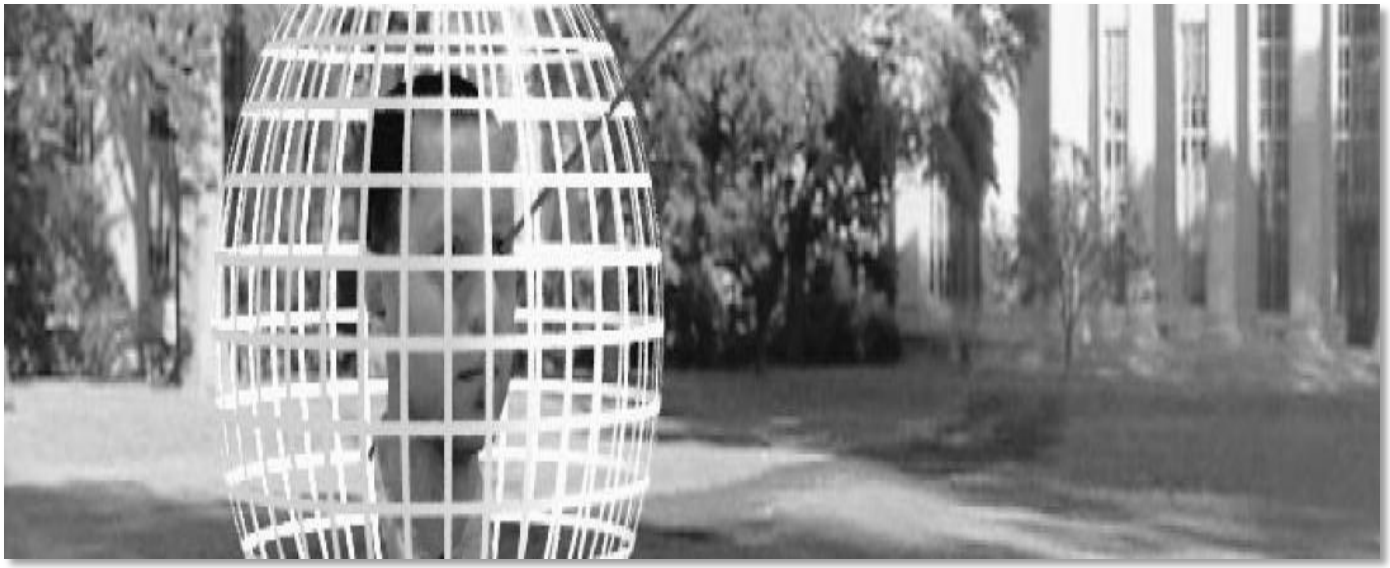
- Computational processing
- **Computational illumination**
  - **Light fields & Lumigraphs**
- Computational optics

The Plenoptic Function [Adelson & Bergen 91]

Crédit: Leonard McMillan

- Comment **décrire (et capturer) toutes les images possibles** autour de nous ?
- Essayons de trouver une paramétrisation pour une personne statique

Grayscale snapshot



$$P(\theta,\phi)$$

is intensity of light

- seen from a single view point
- at a single time
- averaged over the wavelengths of the visible spectrum

(can also do P(x,y), but spherical coordinate are nicer)

Color snapshot

$$P(\theta,\phi,\lambda)$$

is intensity of light

- seen from a single view point
- at a single time
- as a function of wavelength

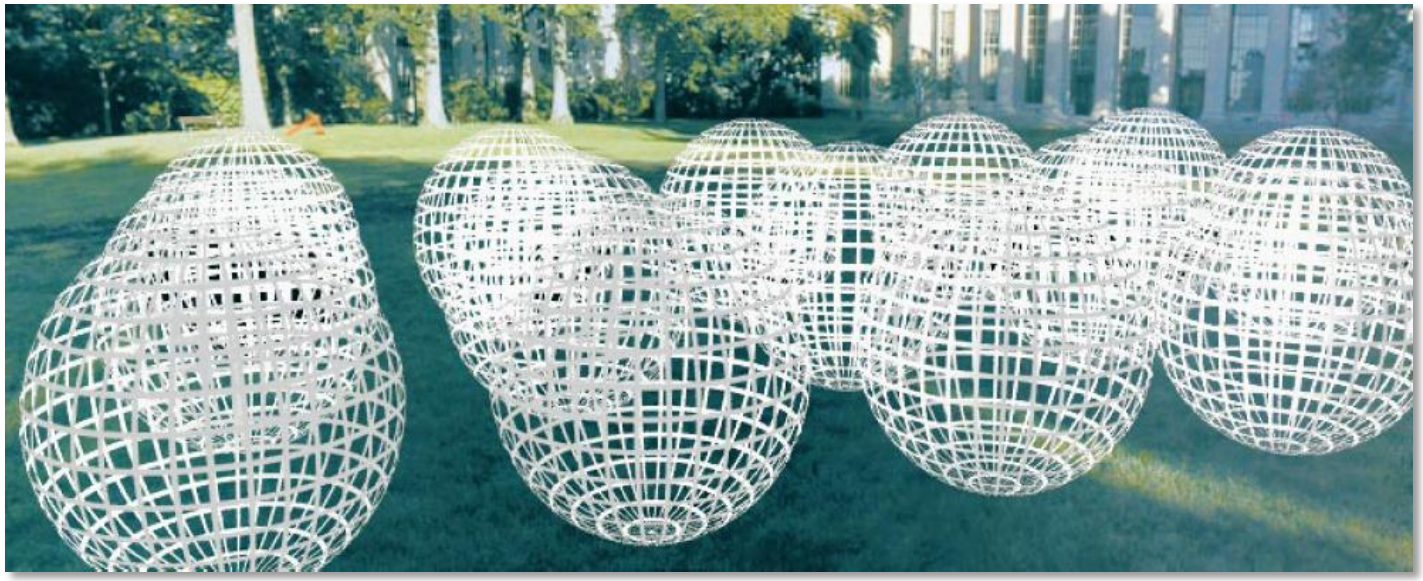
A movie

$$P(\theta,\phi,\lambda,t)$$

is intensity of light

- seen from a single view point
- **over time**
- as a function of wavelength

Holographic movie

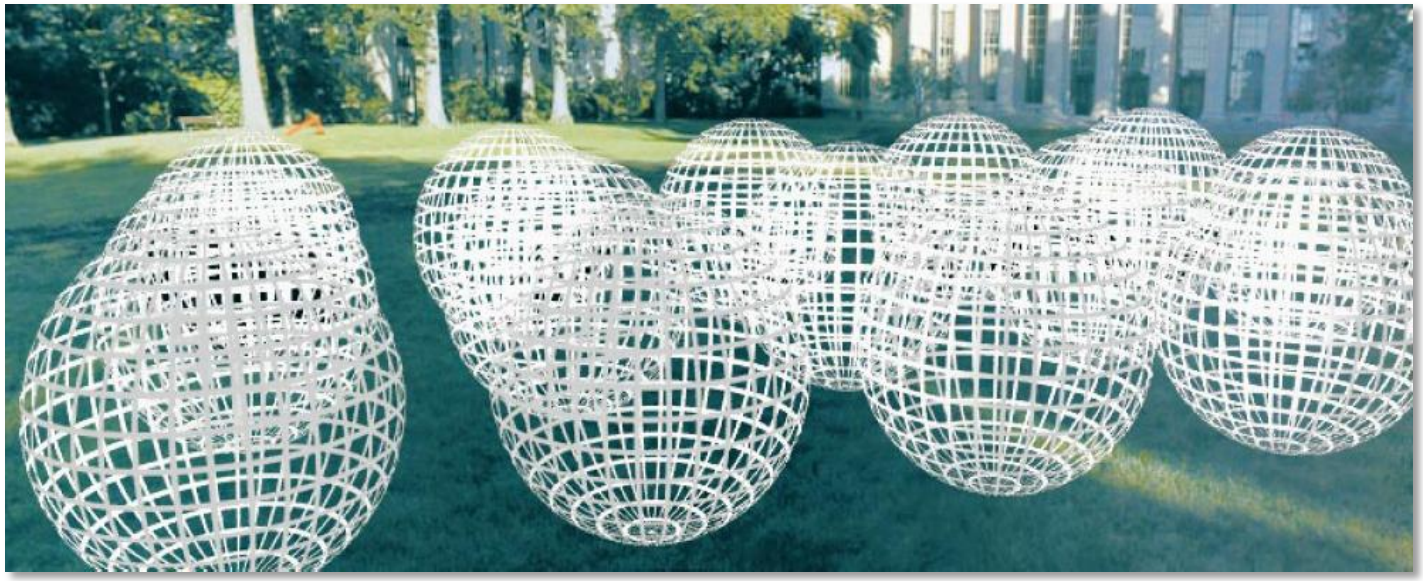


$$P(\theta,\phi,\lambda,t,V_X,V_Y,V_Z)$$

is intensity of light

- seen from **ANY** viewpoint
- over time
- as a function of wavelength

The Plenoptic Function



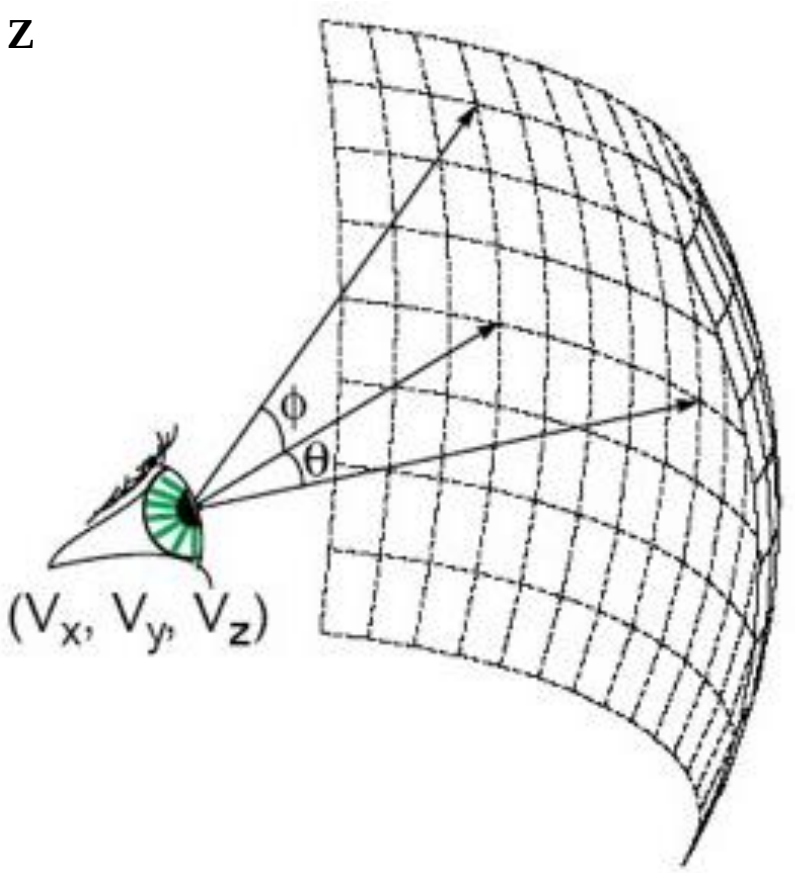
$$P(\theta,\phi,\lambda,t,V_X,V_Y,V_Z)$$

- Can reconstruct **every possible view**, at **every moment**, from **every position**, at **every wavelength**
- Contains every photograph, every movie, everything that anyone has ever seen! **It completely captures our visual reality!** Not bad for a function…

The Plenoptic Function

7D

- 3D for viewpoint: V_X, V_Y, V_Z
- 2D for ray direction: θ, ϕ
- 1D for wavelength: λ
- 1D for time: t



McMillan 95

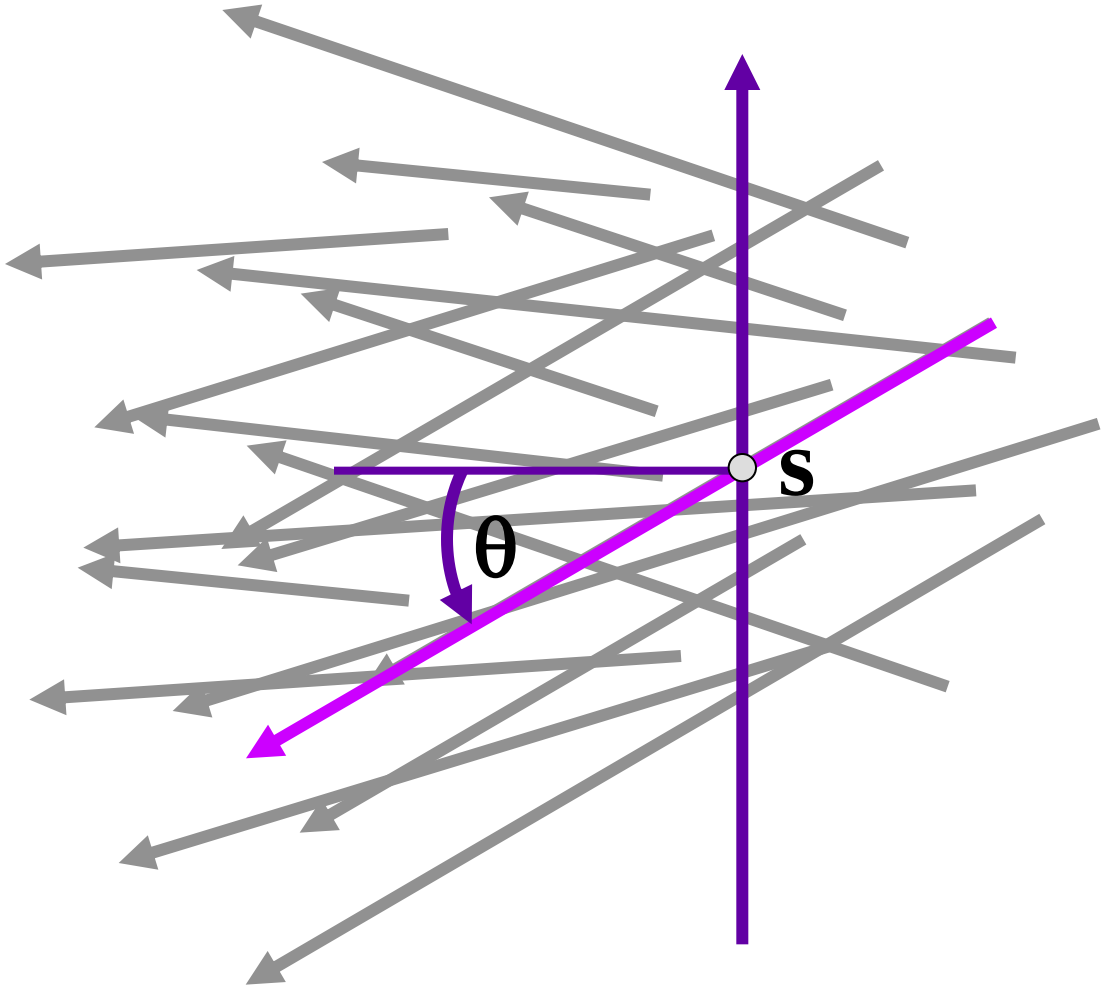
Lumigraph / Light fields

- Extérieur = enveloppe convexe de la scène
- Pour chaque rayon dans cet espace, capturer et stocker la radiance (couleur RGB)
- **Rendu = lookup**
- 2 publications en1996
  - “Light field rendering” [Levoy & Hanrahan]
  - “The Lumigraph” [Gortler et al.]

Lumigraph – Organization

2D position: **s**

2D direction: θ

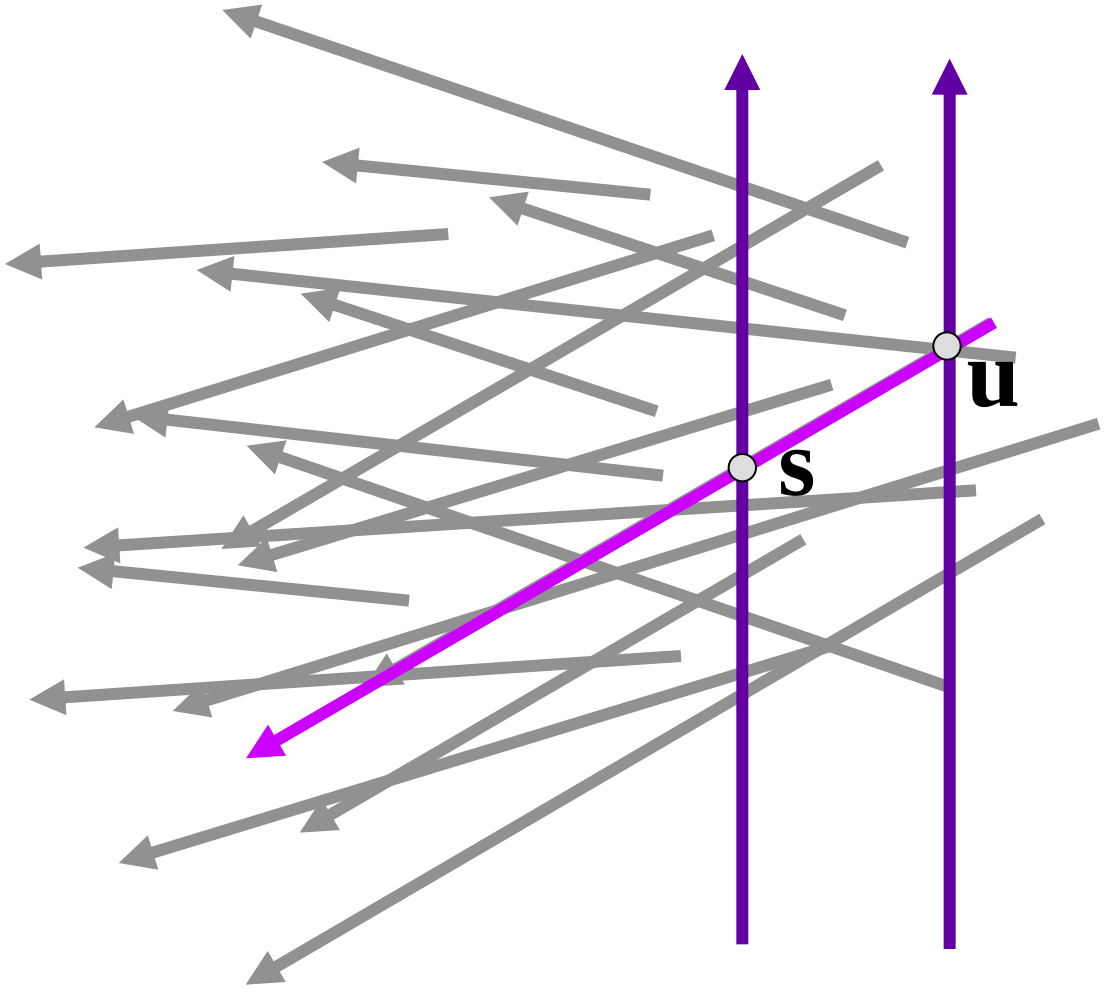


Slide by Rick Szeliski and Michael Cohen

Lumigraph – Organization

2D position: **s**

2D position: **u**



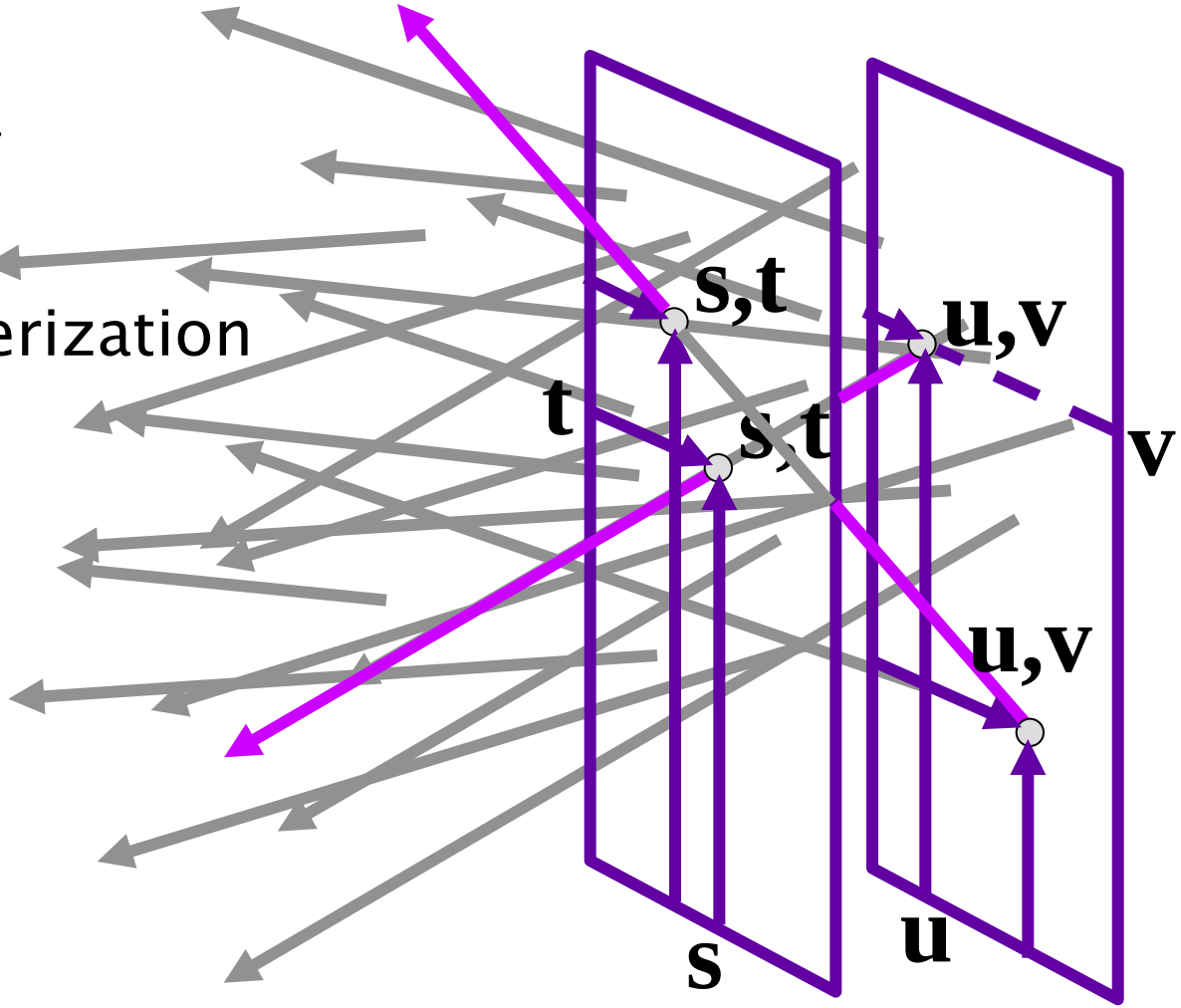
Slide by Rick Szeliski and Michael Cohen

Lumigraph – Organization

2D position: **s,t**

2D position: **u,v**

2 plane parameterization



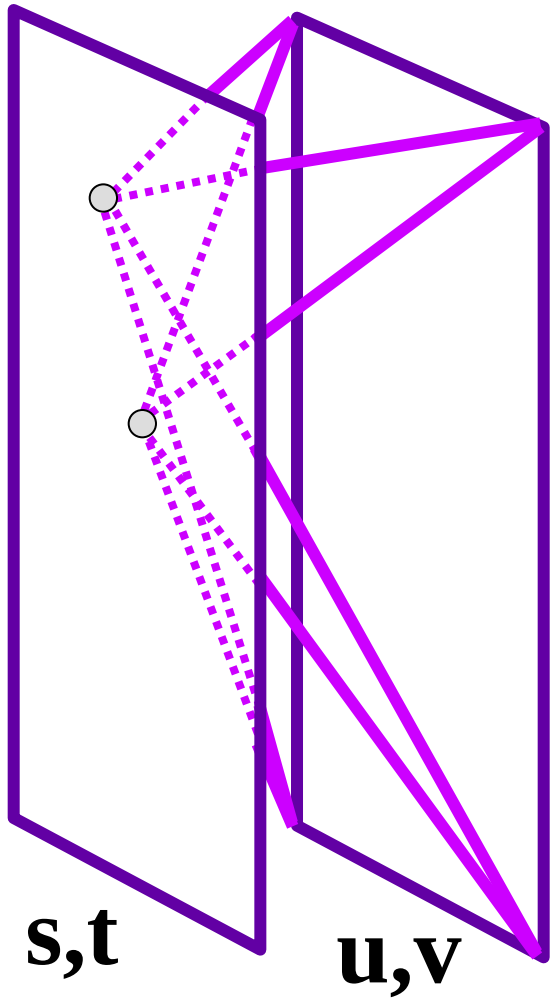
Slide by Rick Szeliski and Michael Cohen

Lumigraph – Organization

Hold **s,t** constant

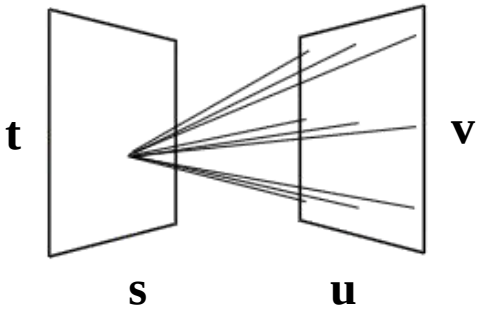
Let **u,v** vary

⇨ an image

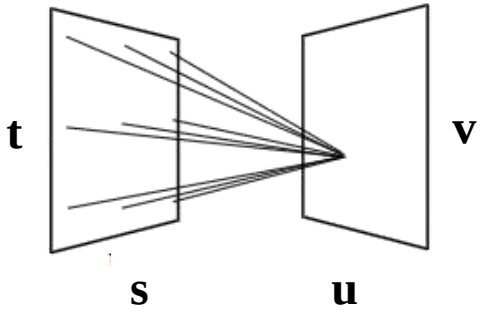


Slide by Rick Szeliski and Michael Cohen

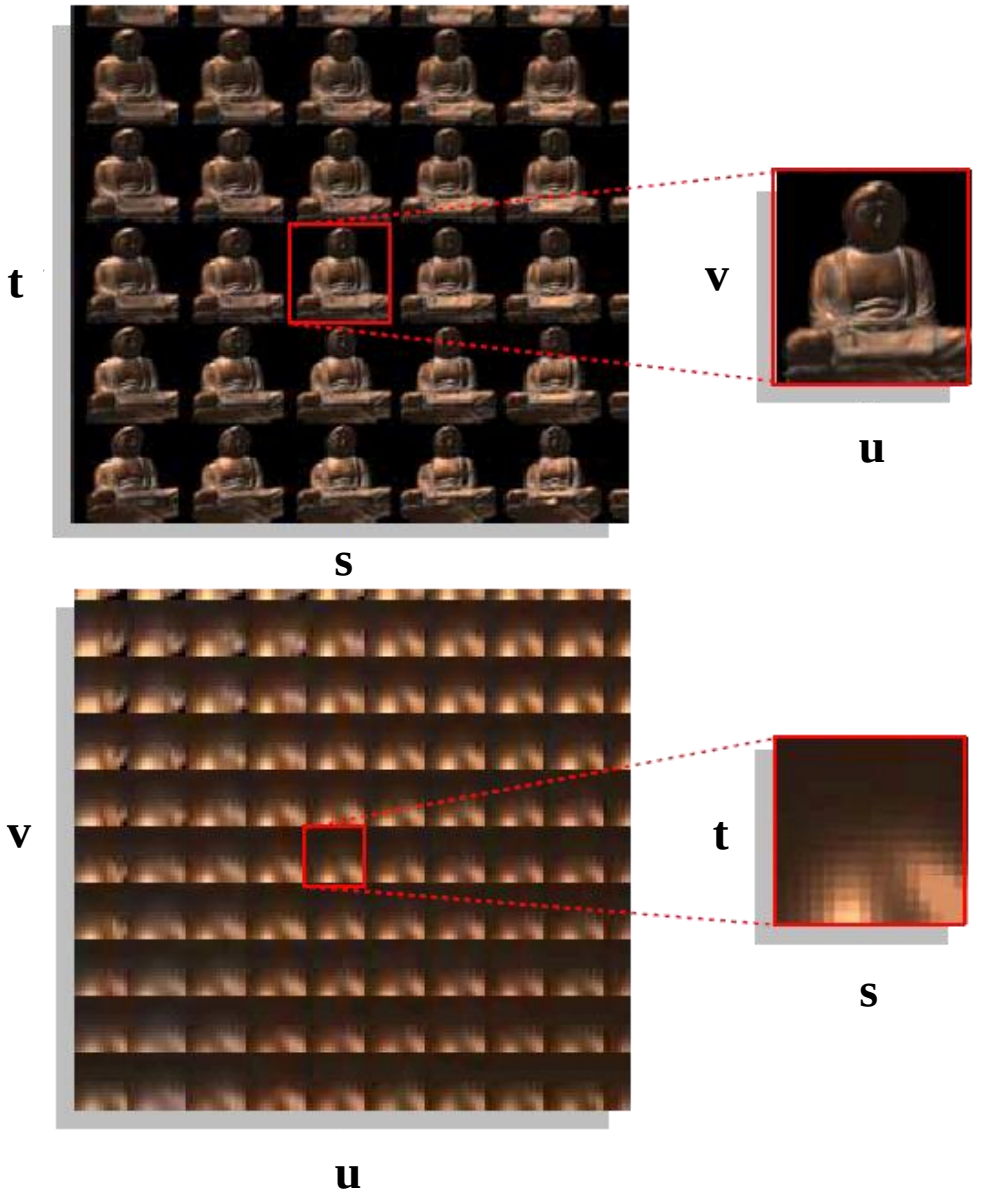
Lumigraph / Light field



(a)

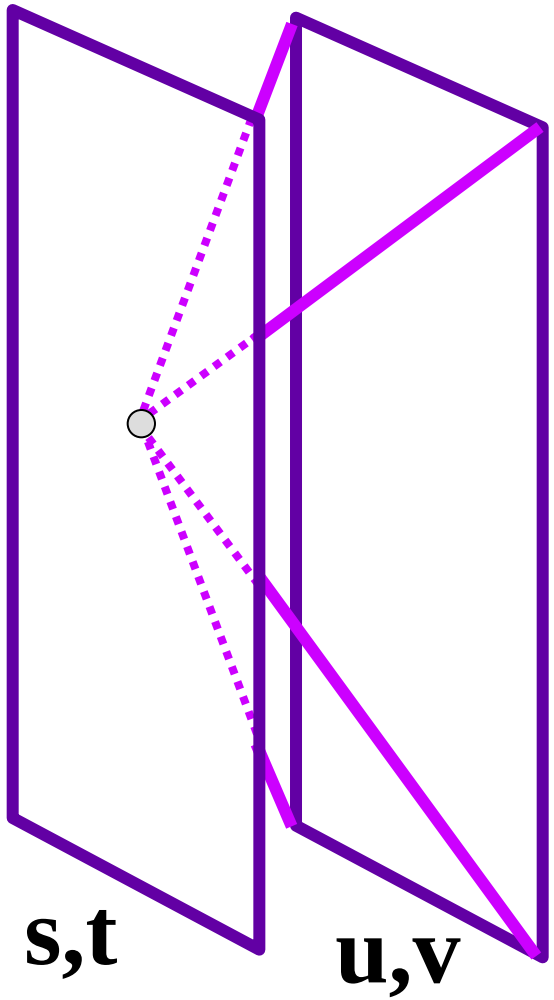


(b)



Lumigraph – Capture

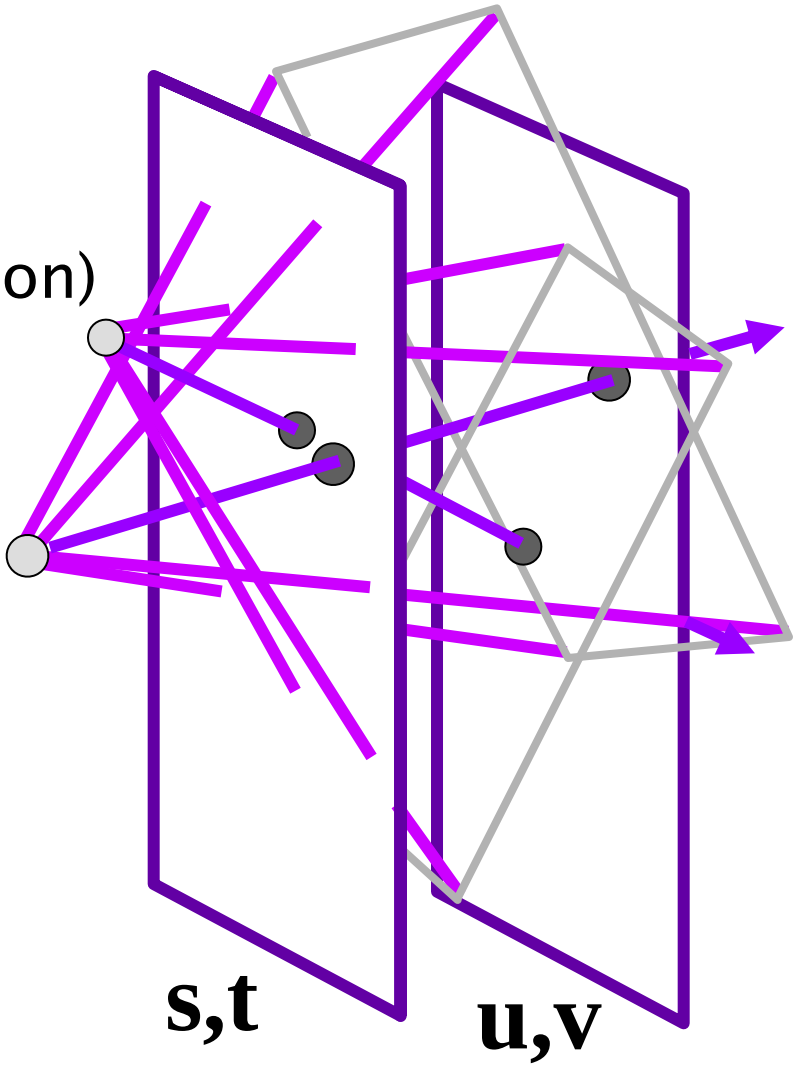
- Idea 1
  - Move camera carefully over s,t plane



Slide by Rick Szeliski and Michael Cohen

Lumigraph – Capture

- Idea 2
  - Move camera anywhere
  - Rebinning
    (calibration & pose estimation)



Slide by Rick Szeliski and Michael Cohen

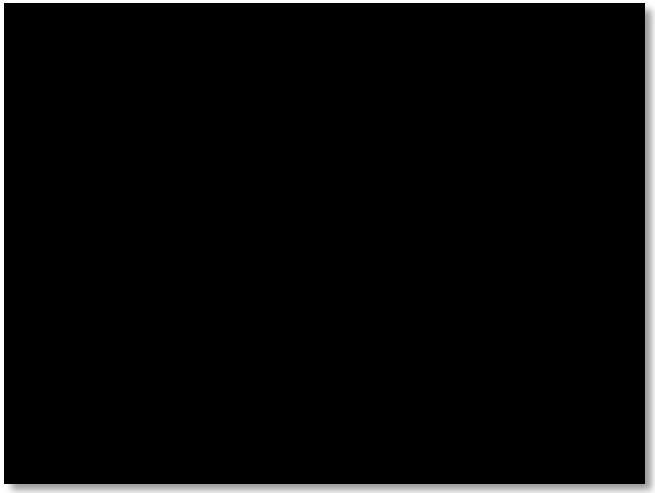
Stanford multi-camera array

- 640 × 480 pixels
  × 30 fps × 128 cameras
- Synchronized timing
- Continuous streaming
- Flexible arrangement

http://graphics.stanford.edu/papers/CameraArray/

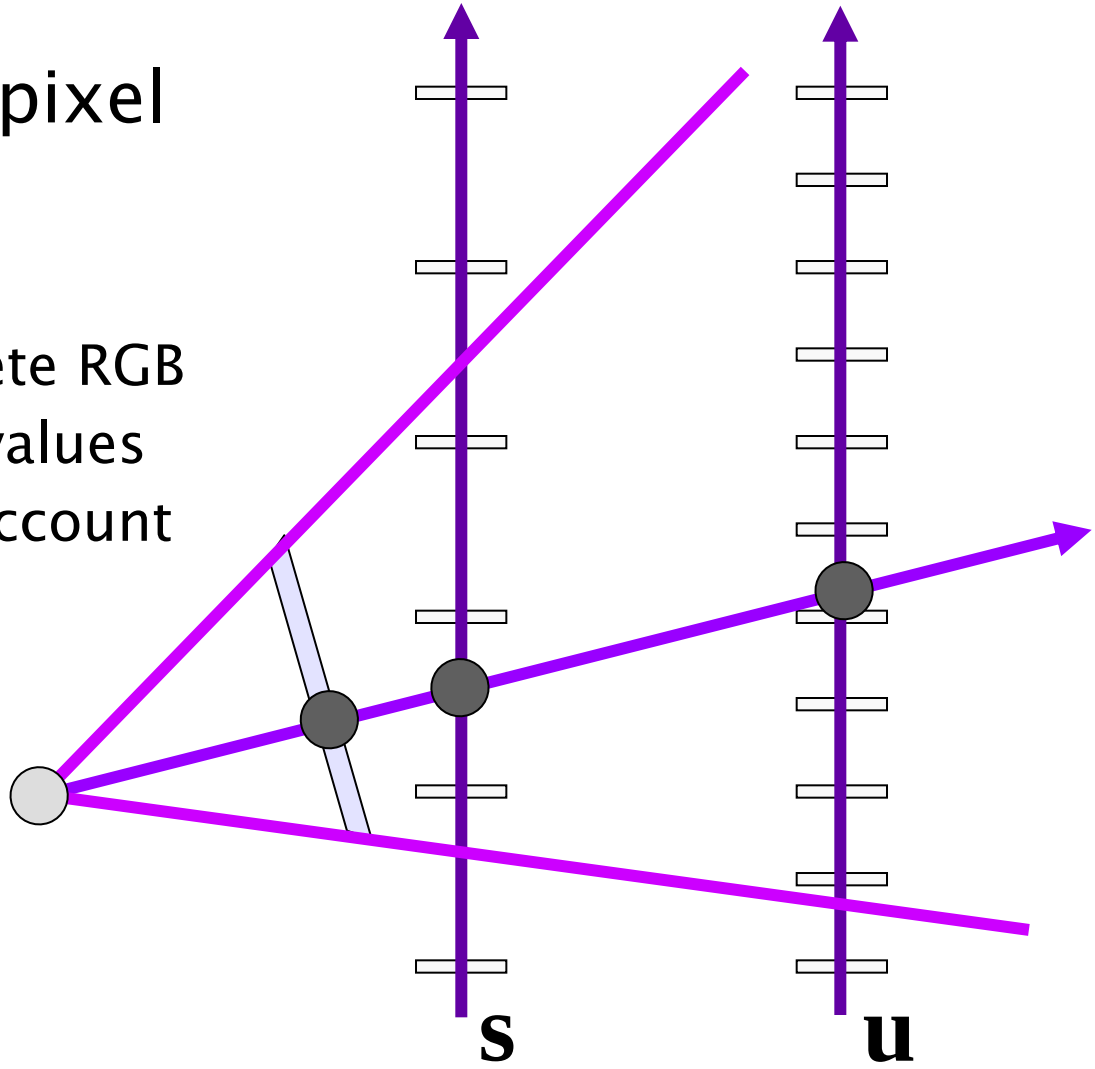
Time splice

http://www.timesplice.com.au/index.html



Lumigraph – Rendering

- For each output pixel
  - determine **s,t,u,v**
  - either
    - use closest discrete RGB
    - interpolate near values
    - take a lens into account



Slide by Rick Szeliski and Michael Cohen

Démo lightfield viewer

http://lightfield.stanford.edu/lfs.html

Plan

- Computational processing
- Computational illumination
- **Computational optics**
  - Digital Refocusing
  - Coded Aperture

Image capture

- Un capteur seul ne capture pas une image

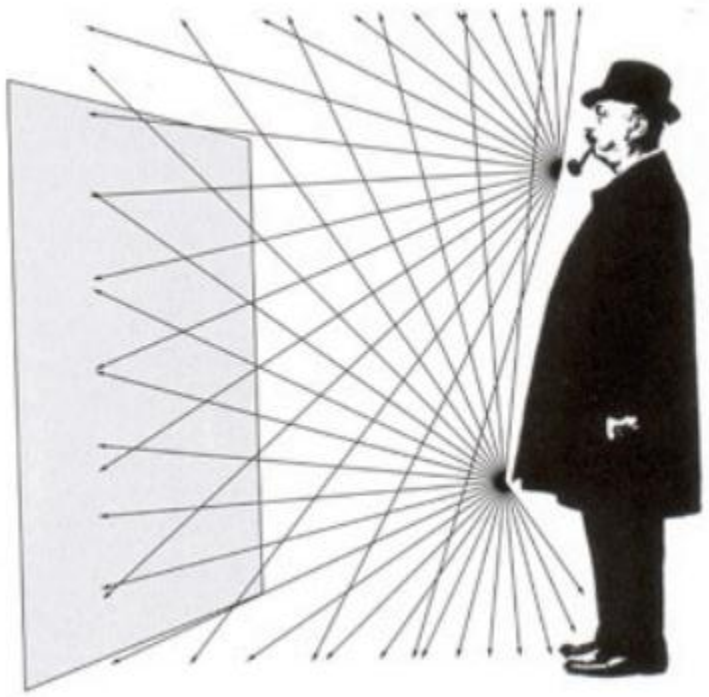


Image capture

- Un sténopé *(pinehole)* permet de sélectionner certains rayons

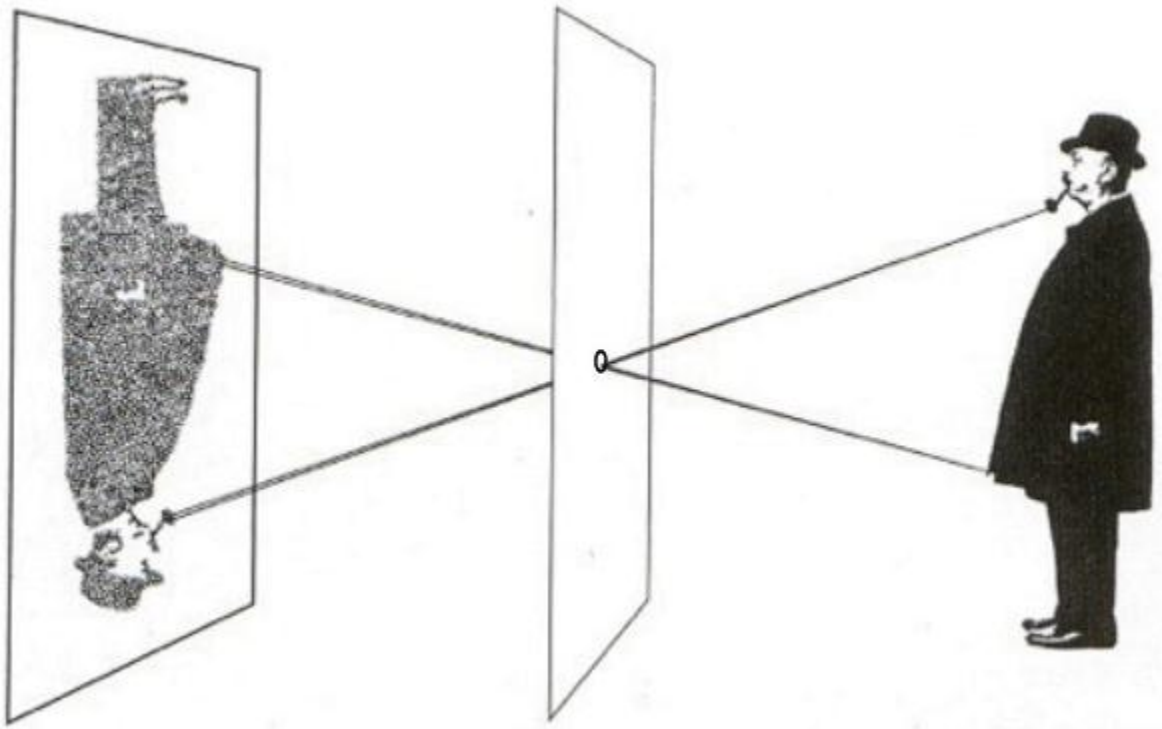


Image formation: optics

- Les optiques sélectionnent et intègrent les rayons lumineux ⇨ **forment** une image

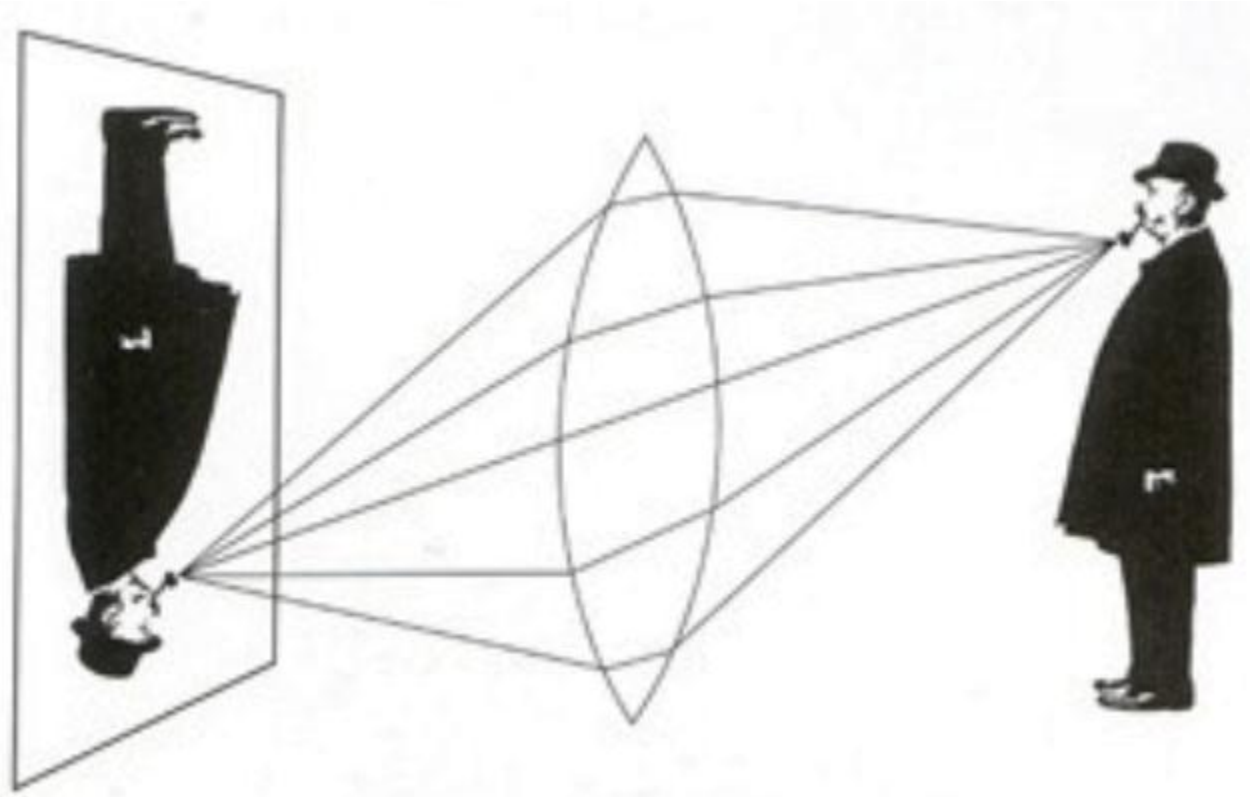


Image formation: computation

- La combinaison optiques & algorithmes forme une image

Computation

Generalized optics

Image finale

Image intermédiaire

Computational imaging goals

- Meilleur capture de l'information
- Former un image en post-process

Computation

Generalized optics

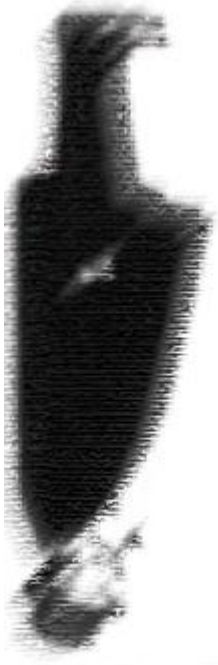
Image finale

Image intermédiaire

Better capture information

- Les optiques encodent, l'algorithme décode
- L'encodage vise à minimiser les distorsions

Computation



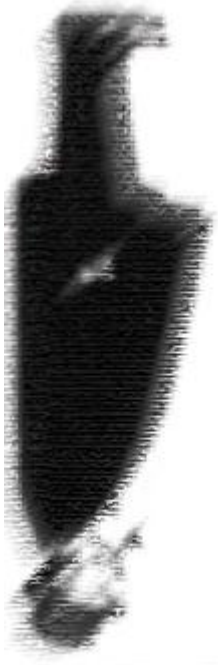
Generalized optics

Image finale **Image intermédiaire**

Form images as a post-process

- Peut être exécuté plus tard
- Répété plusieurs fois avec ≠ paramètres

Computation



Generalized optics

Image finale **Image intermédiaire**

Plan

- Computational processing
- Computational illumination
- **Computational optics**
  - **Digital Refocusing**
  - Coded Aperture

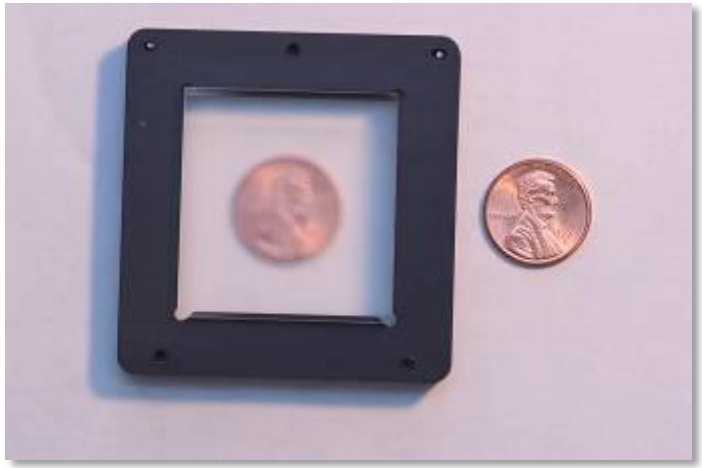
Digital refocusing

[Ng et al. 2005]

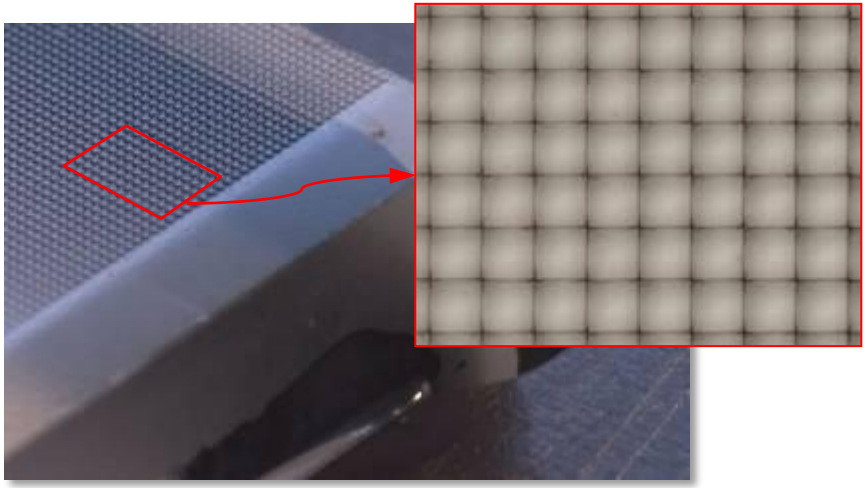
Prototype camera

Contax medium format camera

Kodak 16-megapixel sensor



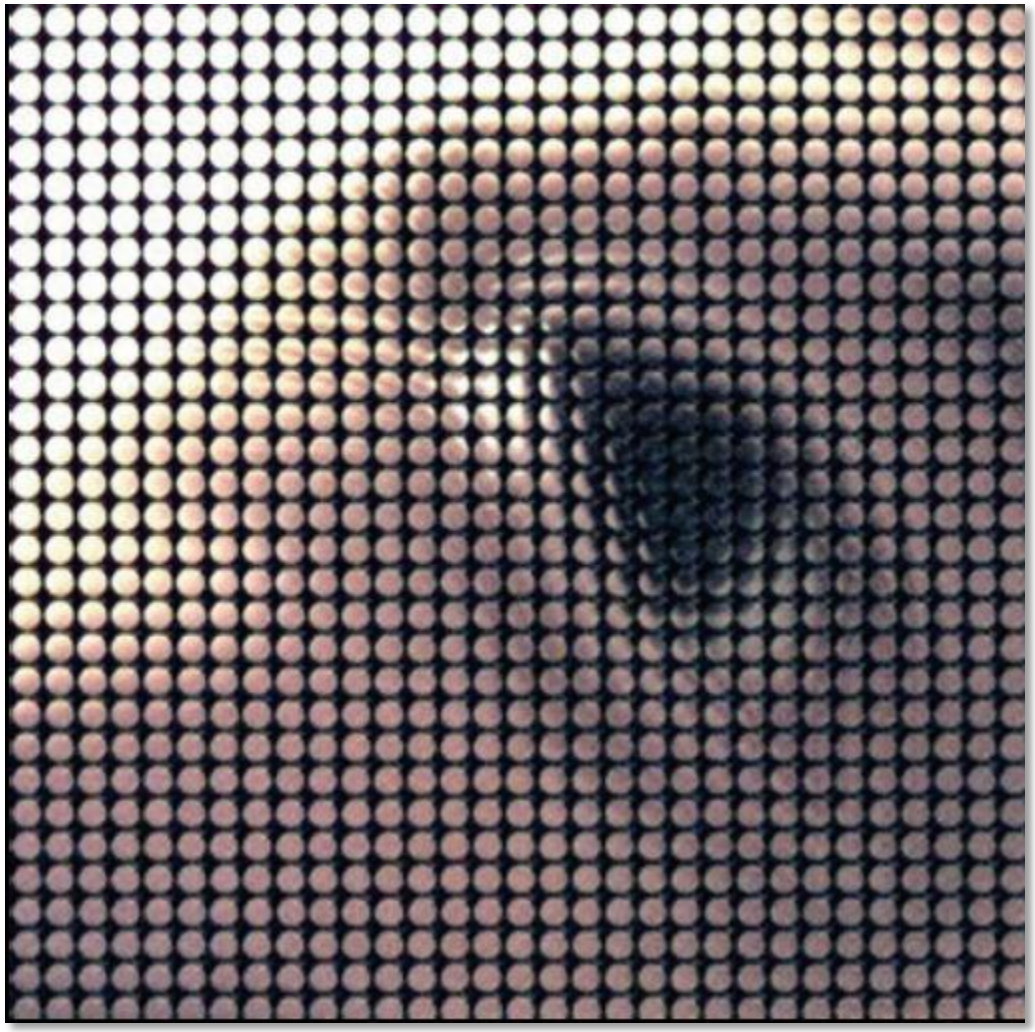
Adaptive Optics microlens array



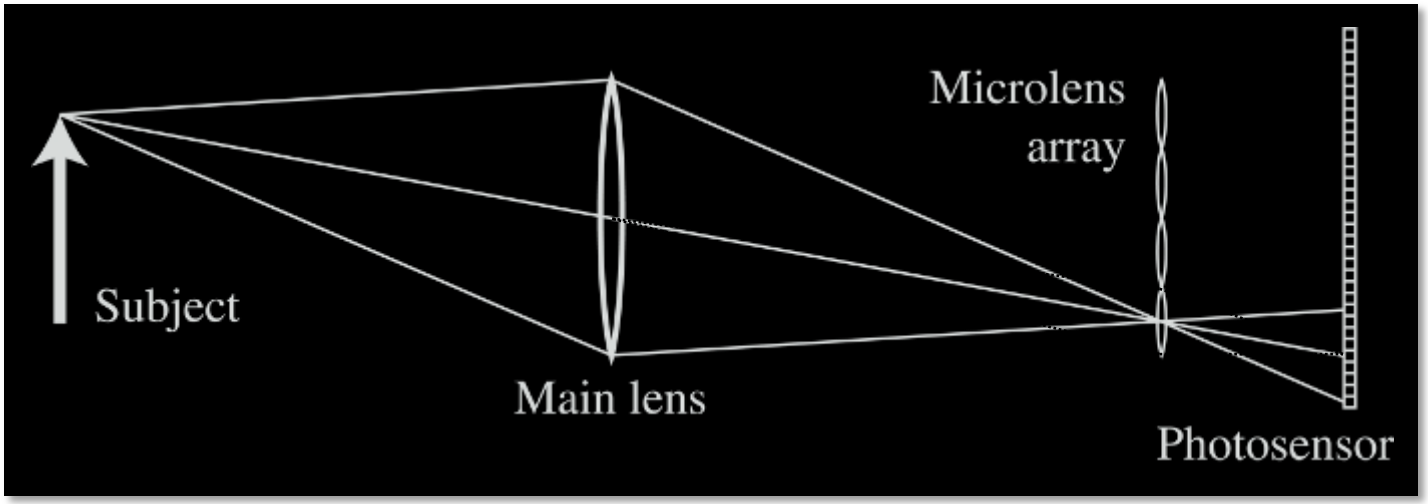
125µ square-sided microlenses

Light Field in a Single Exposure

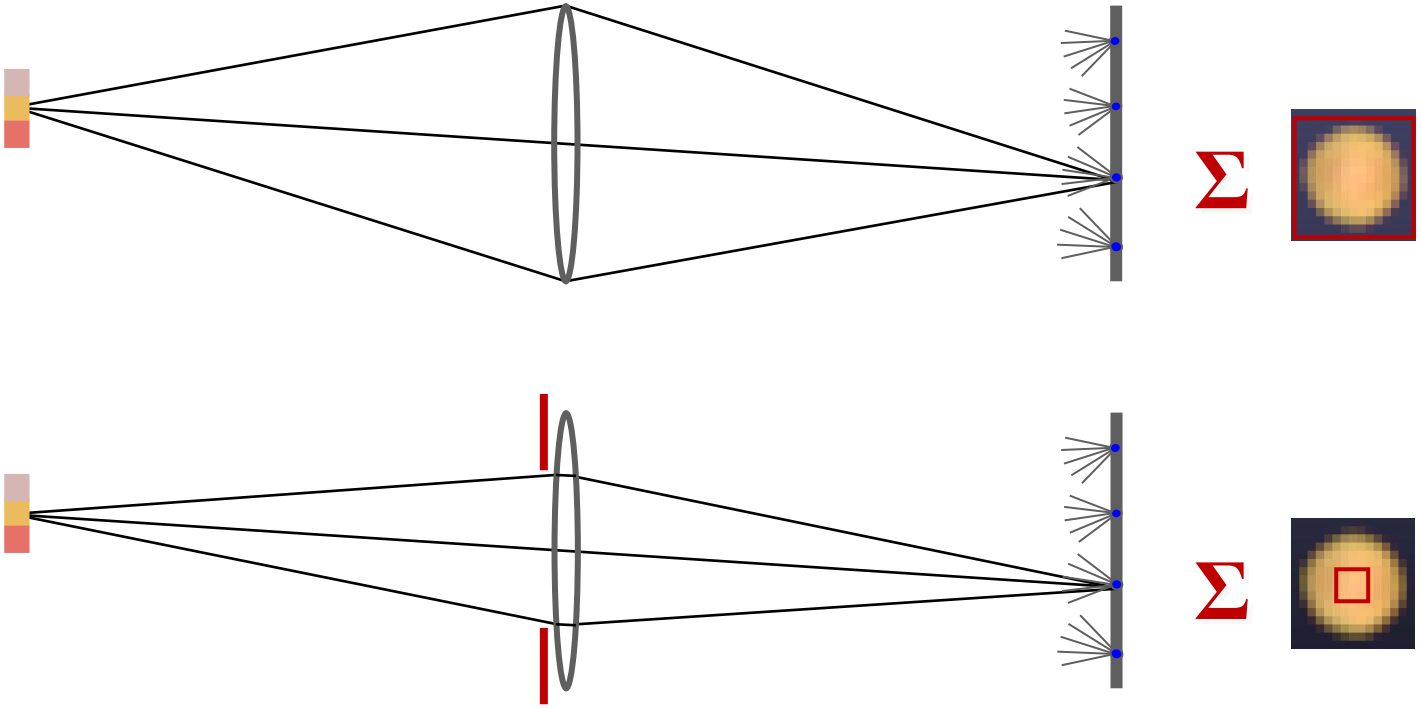
Light Field in a Single Exposure



Light field camera

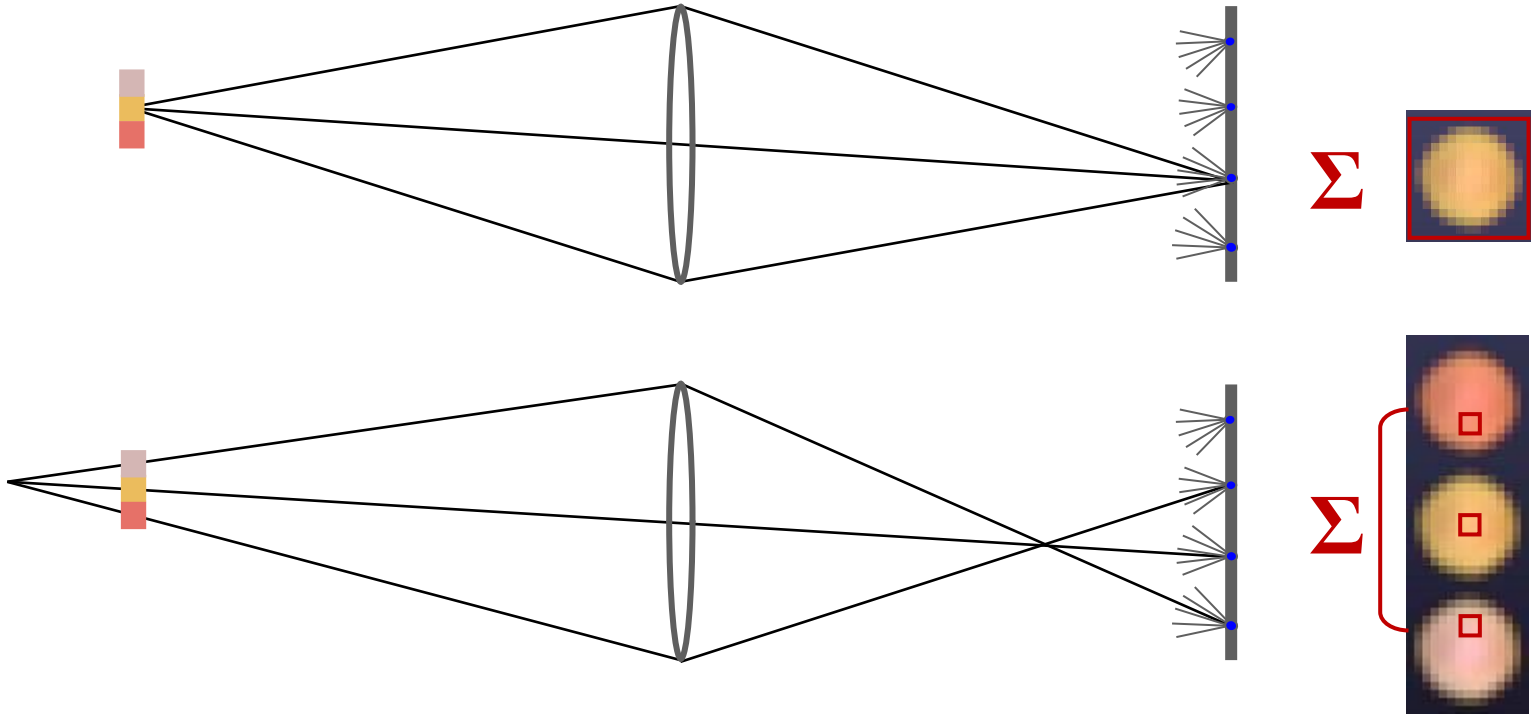


Digitally stopping-down



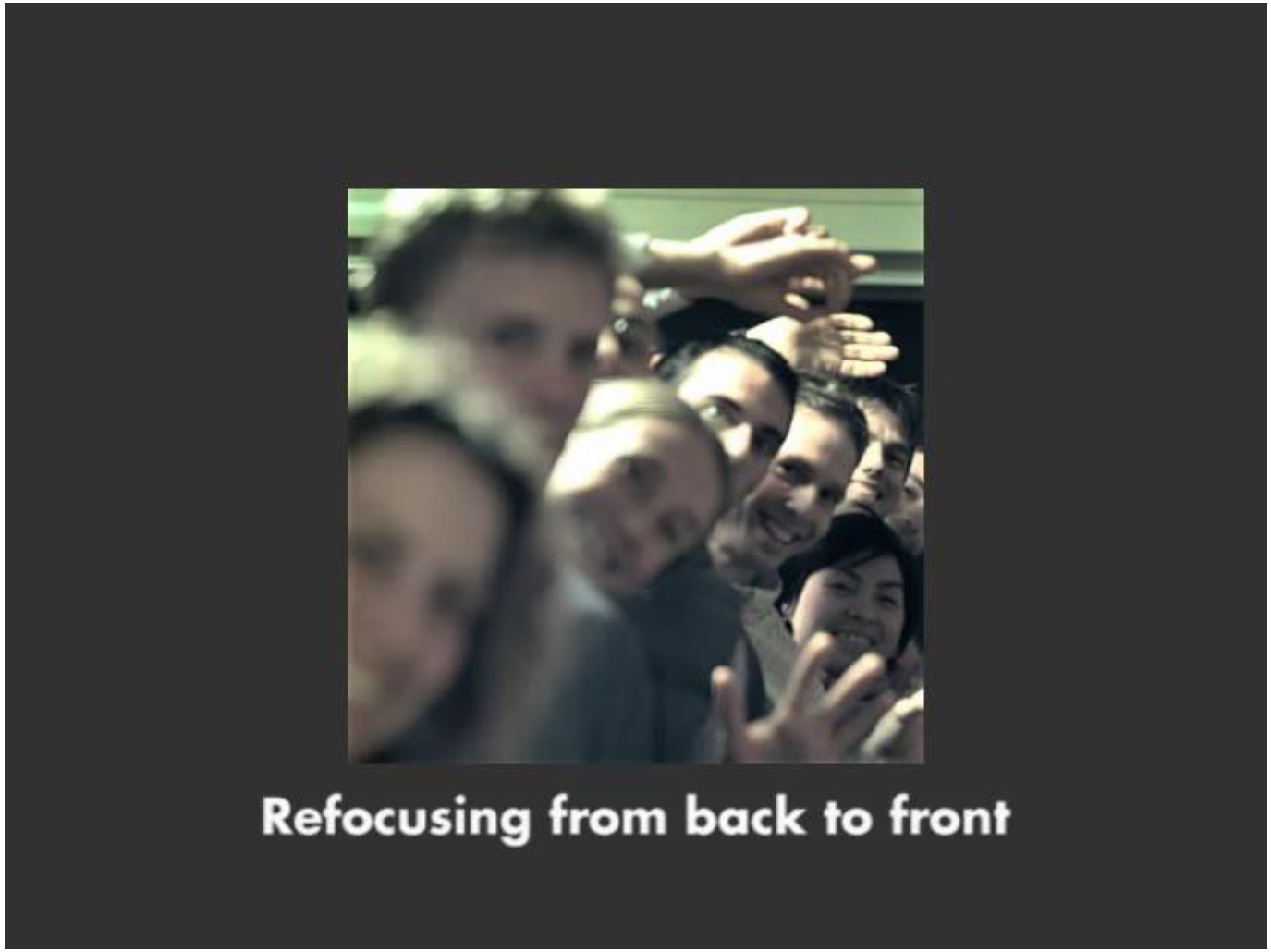
stopping down = summing only the central portion of each microlens

Digital refocusing



refocusing = summing windows extracted from several microlenses

Résultats

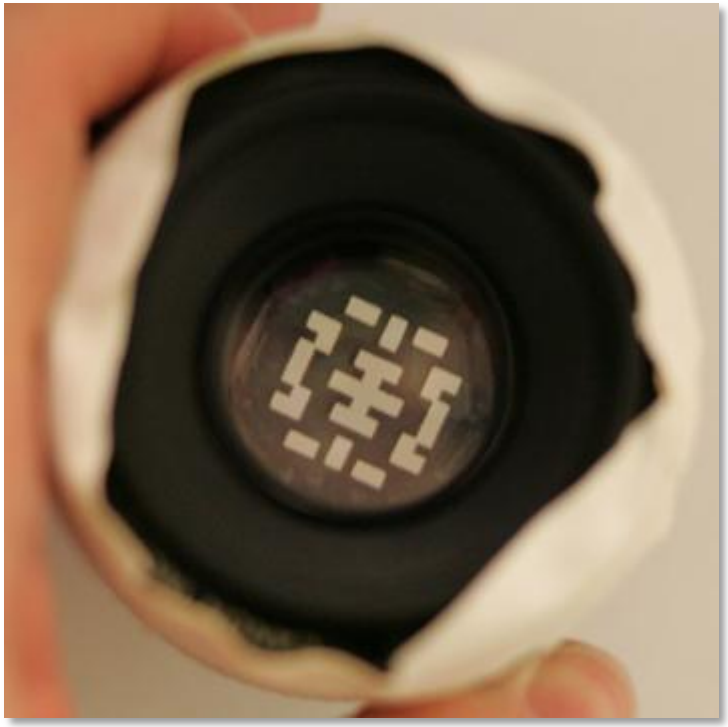


Plan

- Computational processing
- Computational illumination
- **Computational optics**
  - Digital Refocusing
  - **Coded Aperture**

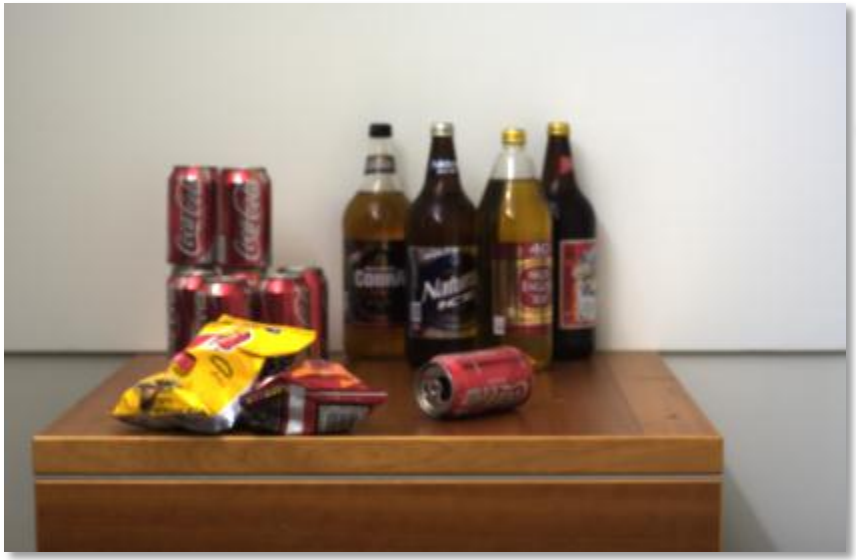
“Image and Depth from a Conventional Camera with a Coded Aperture”

[Levin et al. 2007]



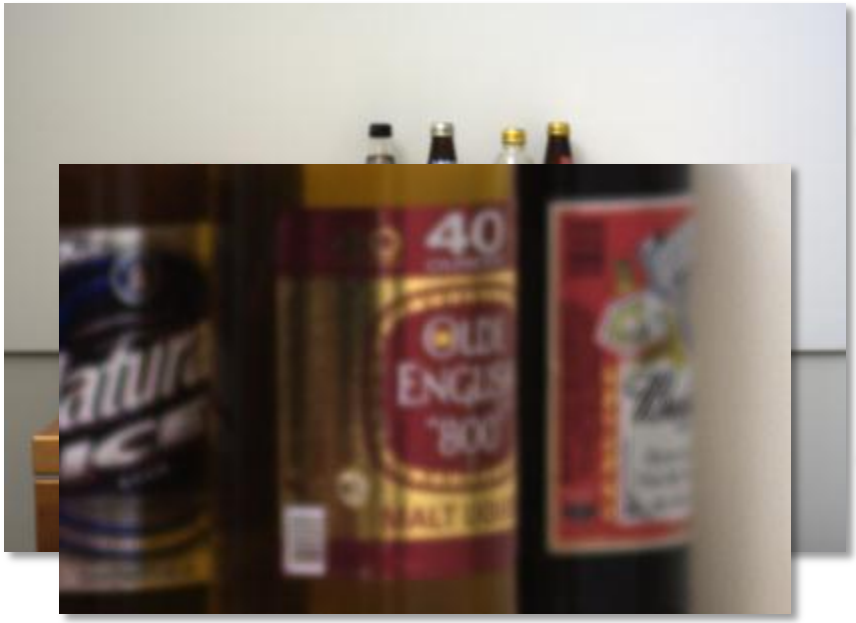
Output 1: Depth map

Single input image

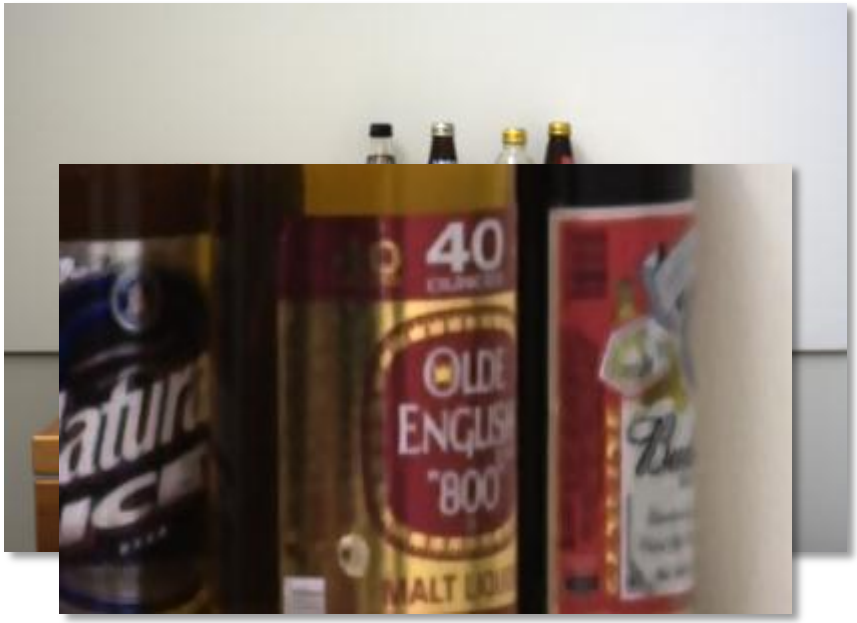


Output 1: Depth map

Single input image



Output 2: All-focused image



Lens and defocus

Lens' aperture

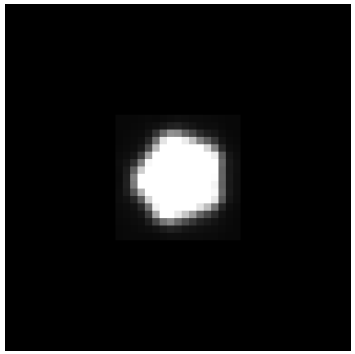
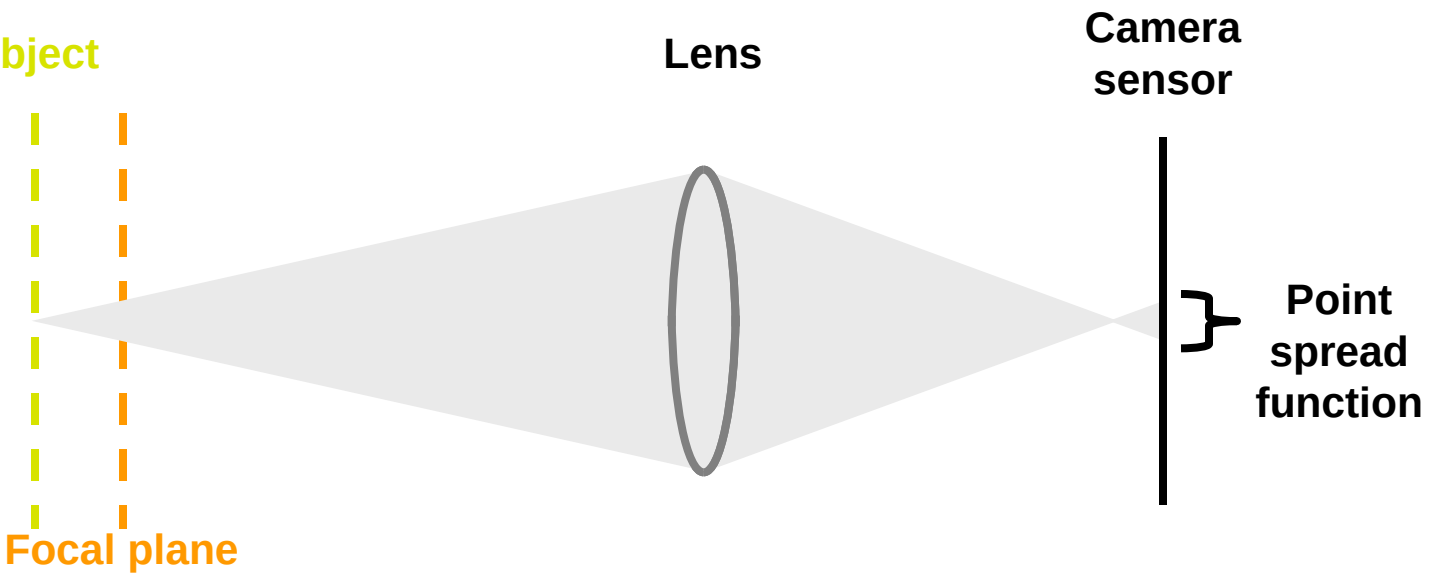


Image of a defocused point light source

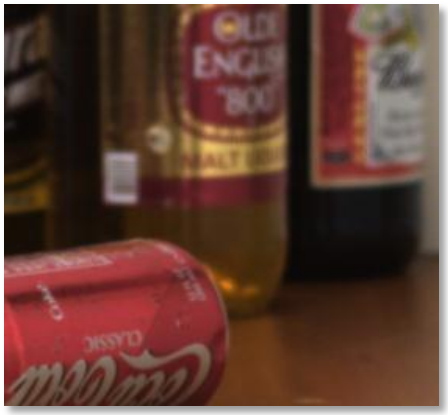


Depth and defocus

Depth from defocus:

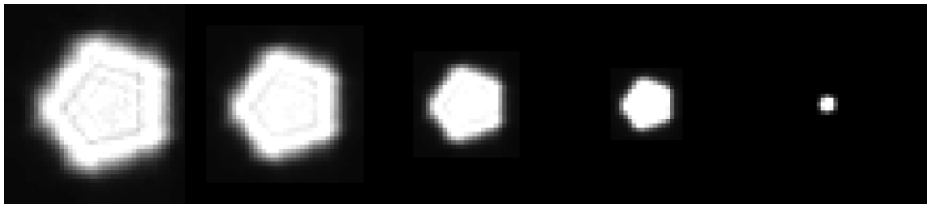
Infer depth by analyzing local scale of defocus blur

Defocus as local convolution

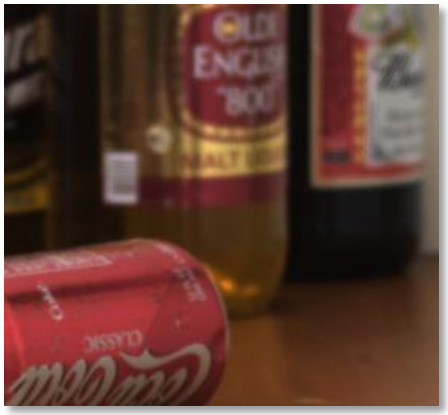


Input defocused image

Calibrated blur kernels at different depths



Defocus as local convolution



Input defocused image

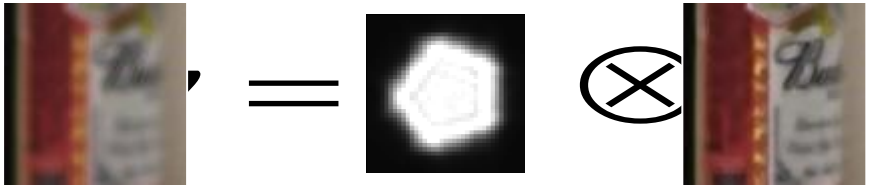
$$y = f_k \otimes x$$

Local sub-window

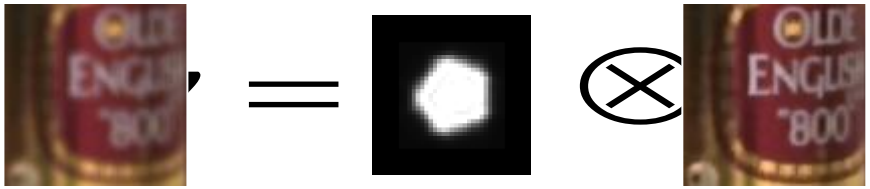
Calibrated blur kernels at depth k

Sharp sub-window

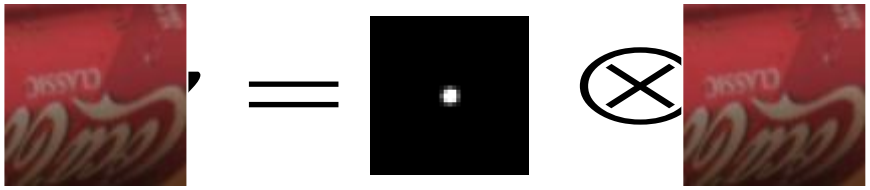
Depth k=1:



Depth k=2:



Depth k=3:



Deconvolution is ill posed

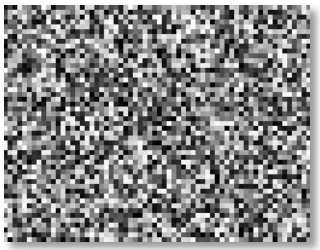
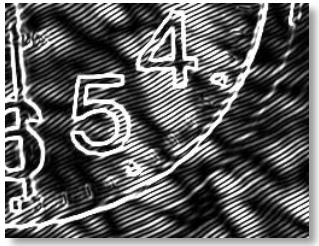
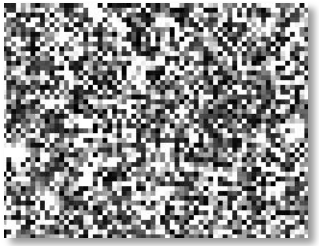
$$f \otimes x = y$$

Solution 1:

Solution 2:

Idea 1: Natural images prior

What makes images special?

	Natural	Unnatural	
Image			
Gradient			

Natural images have sparse gradients

➡ put a penalty on gradients

Idea 2: Coded Aperture

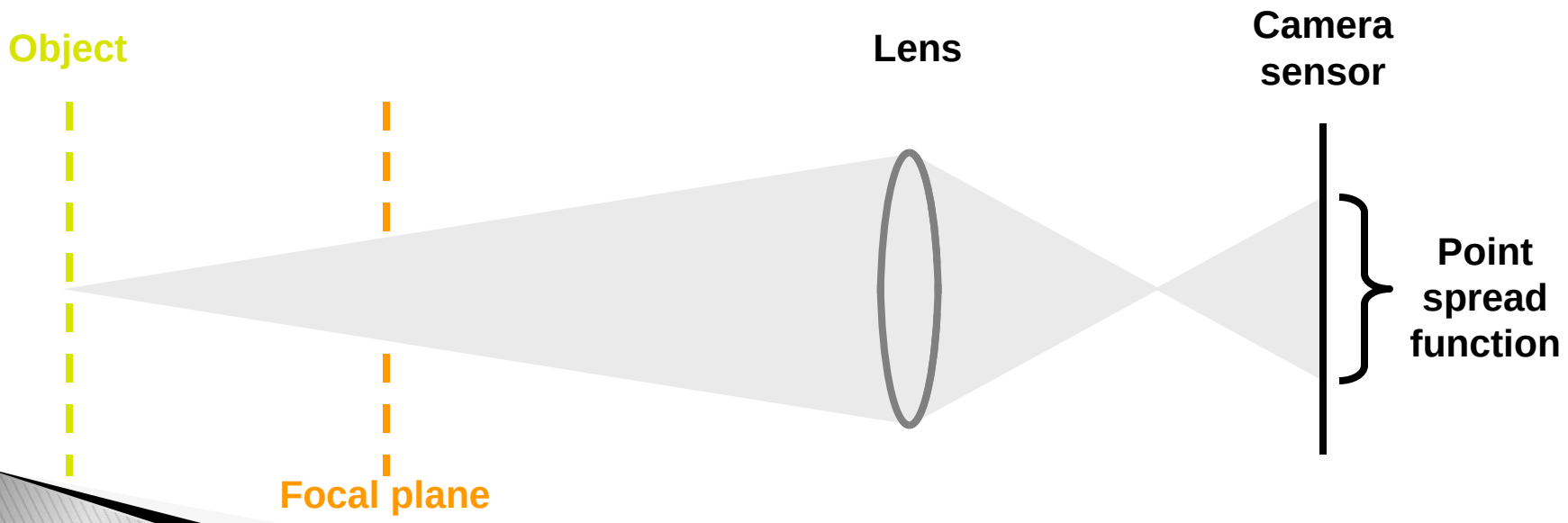
- Mask (code) in aperture plane
  - make defocus patterns different from natural images and easier to discriminate

Conventional aperture

Our coded aperture

Build your own coded aperture

Solution: lens with occluder



Solution: lens with occluder

Aperture pattern

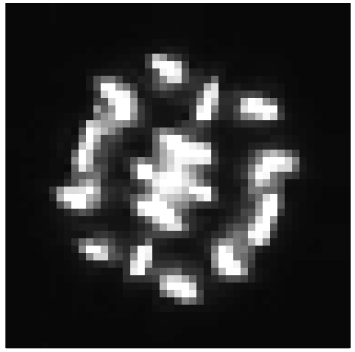
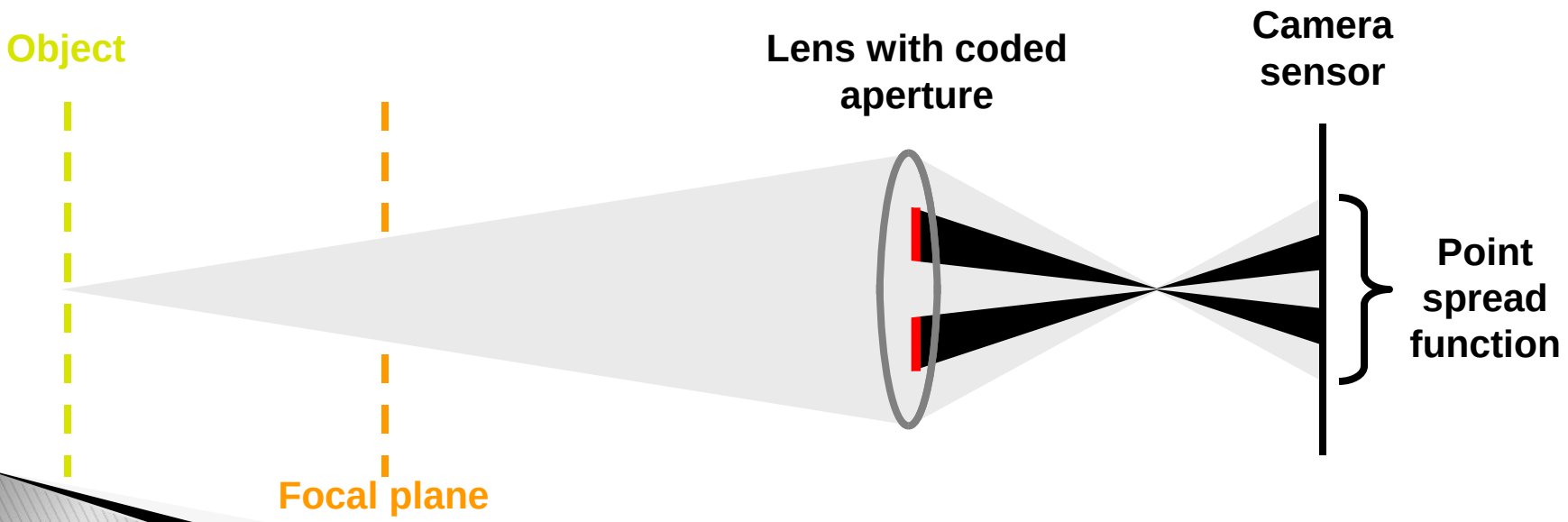


Image of a defocused point light source



Solution: lens with occluder

Aperture pattern

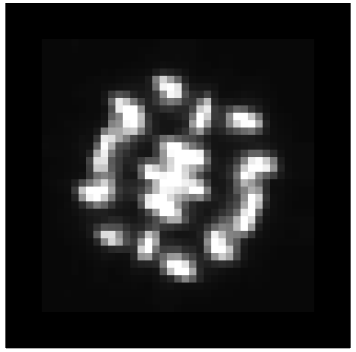
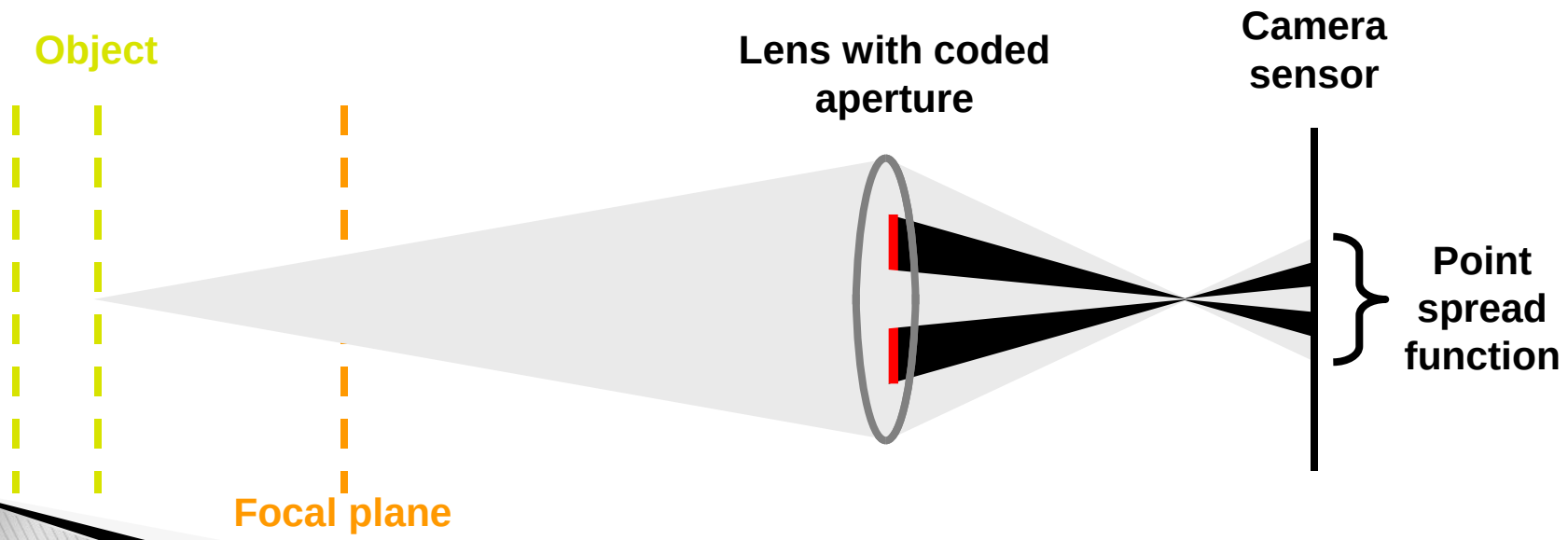


Image of a defocused point light source



Solution: lens with occluder

Aperture pattern

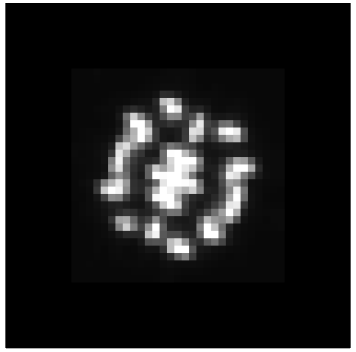
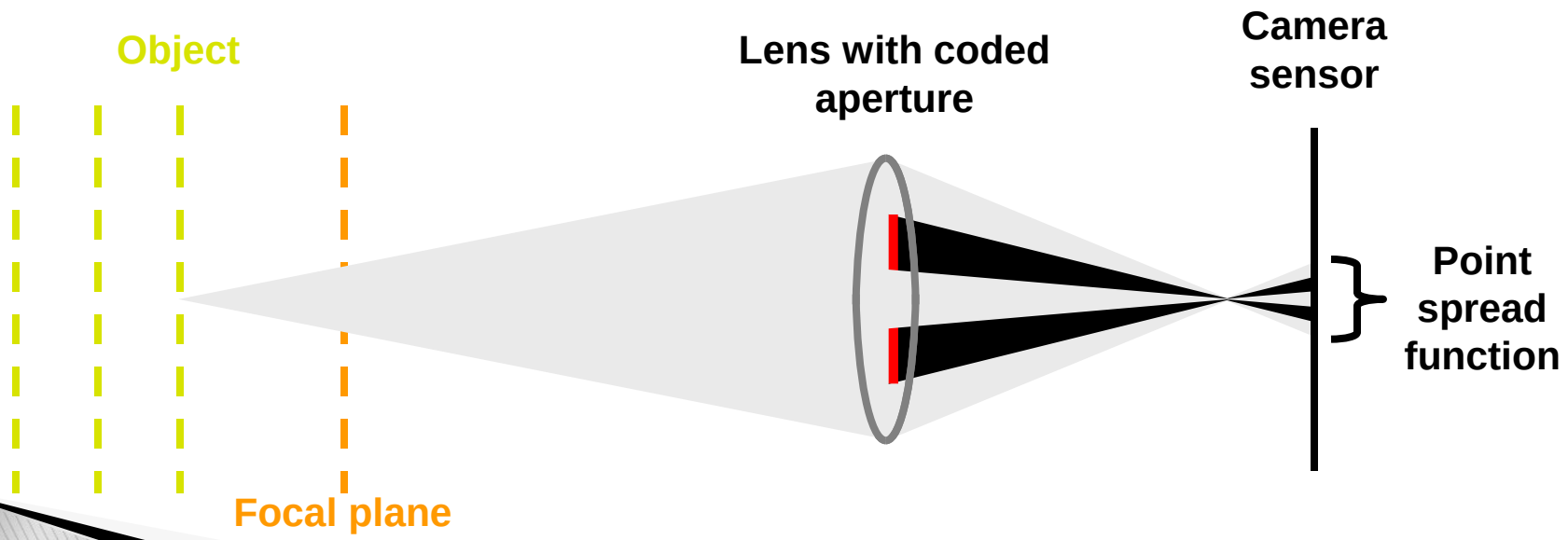


Image of a defocused point light source



Solution: lens with occluder

Aperture pattern

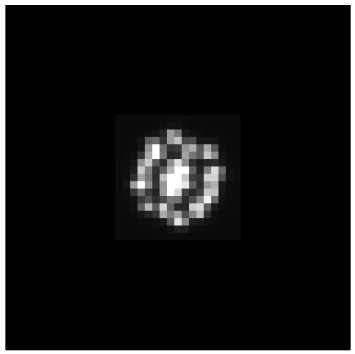
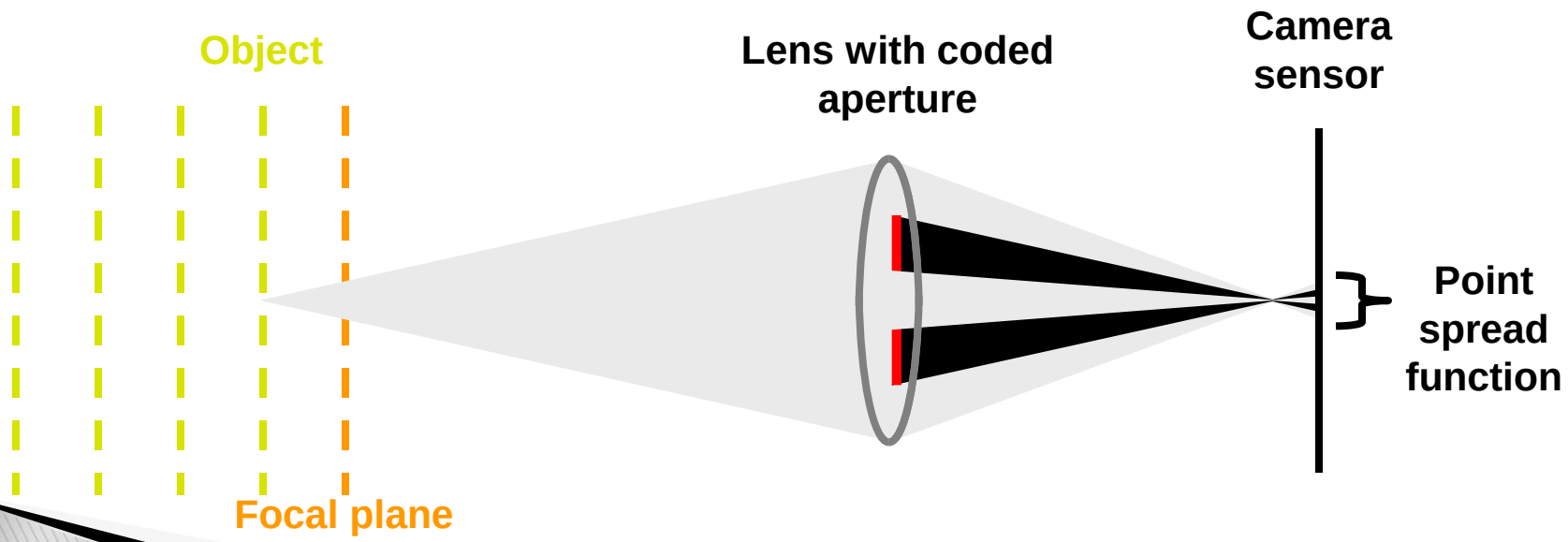


Image of a defocused point light source



Solution: lens with occluder

Aperture pattern

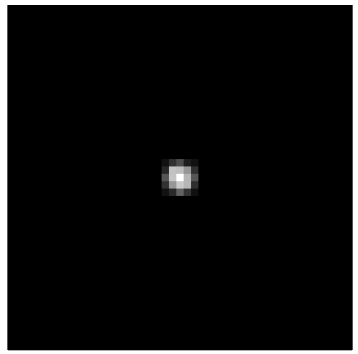
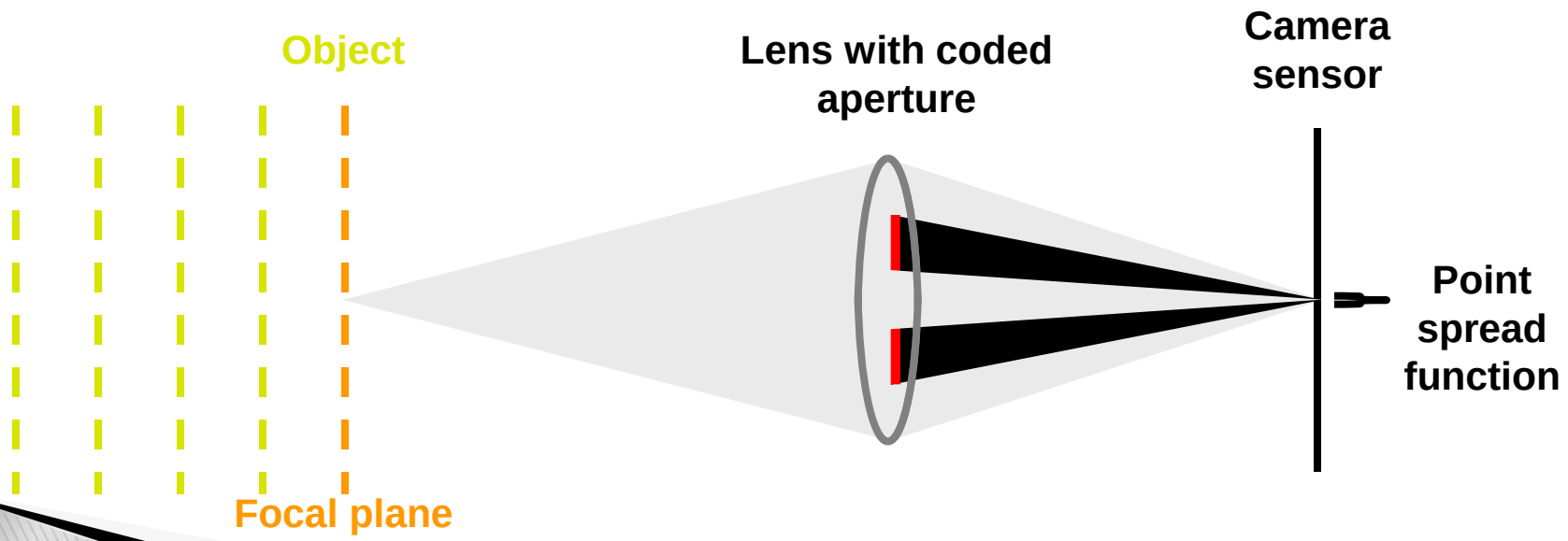
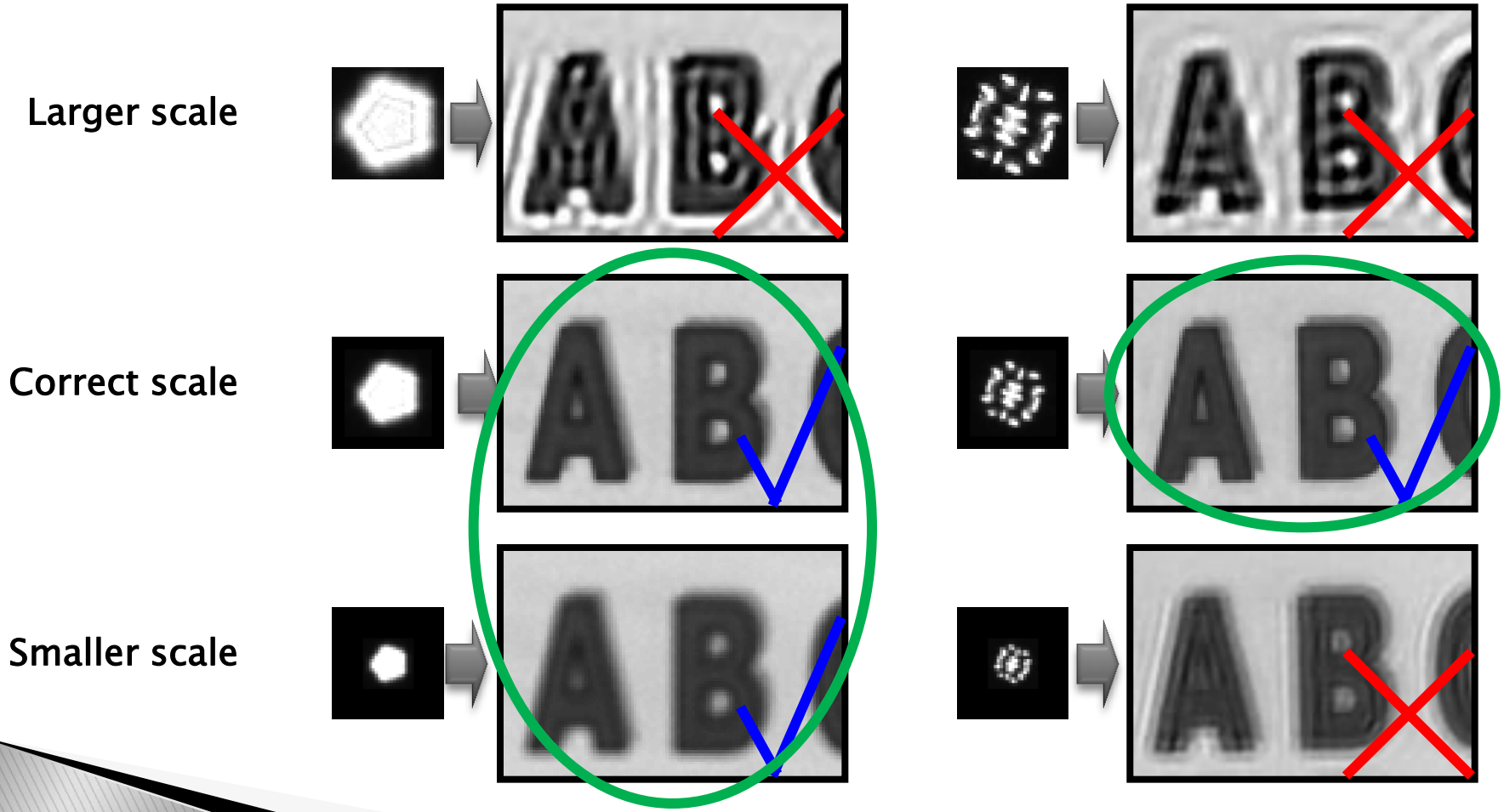


Image of a defocused point light source



Why coded?

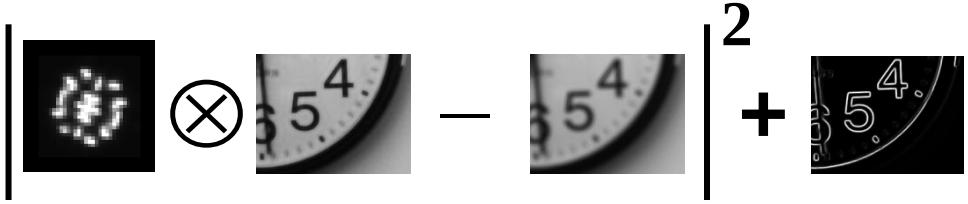
Coded aperture reduces uncertainty in scale identification



Regularizing depth estimation

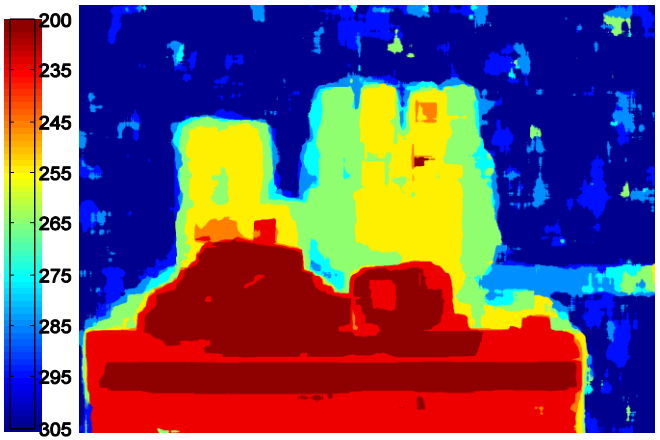
Try deblurring with 10 different aperture scales

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



Keep minimal error scale in each local window + regularization

Input



Local depth estimation

Regularized depth


Coca-Cola
COBRA
40

All-focused (deconvolved)

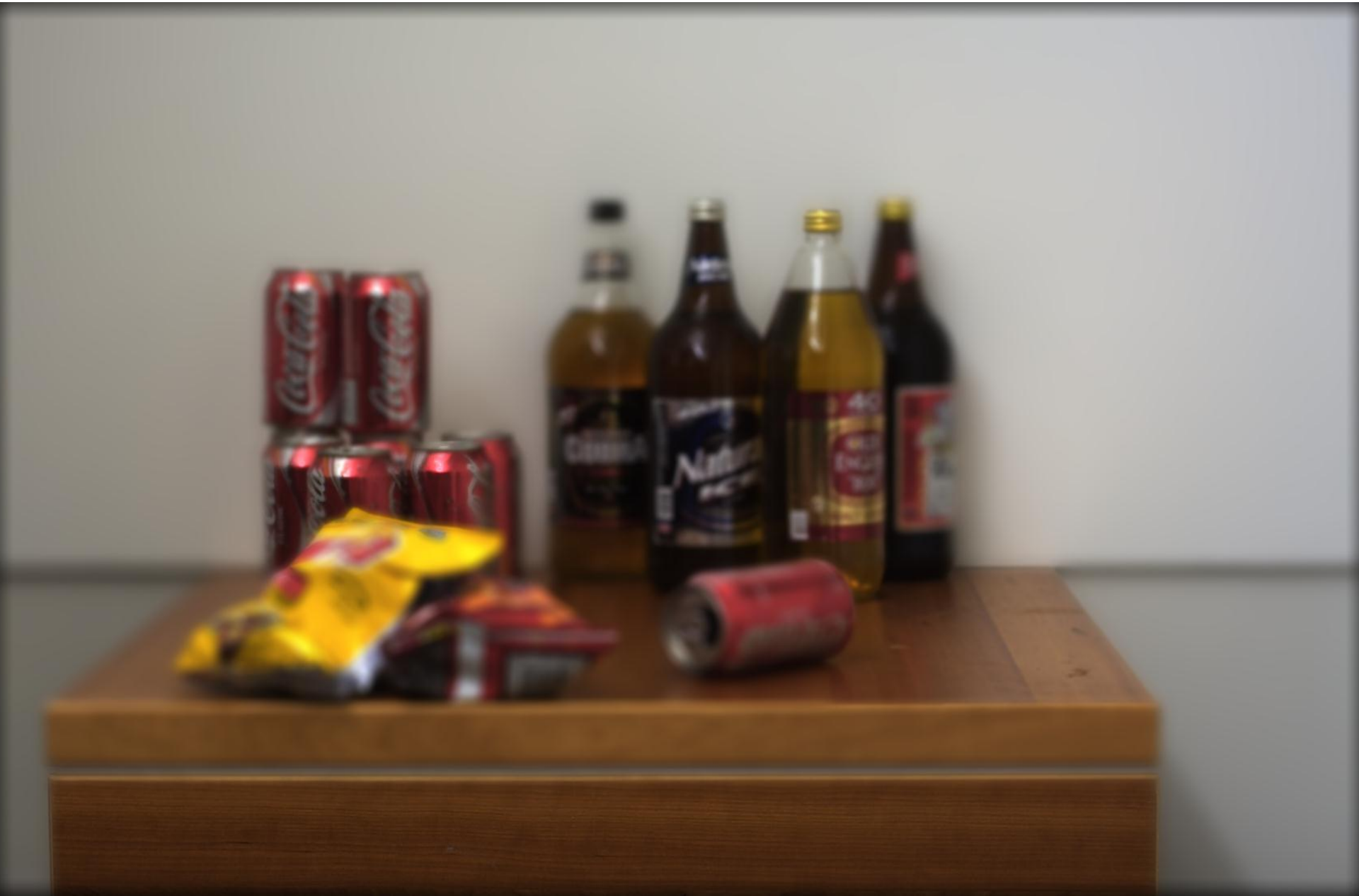
Close-up

Original image

All-focus image

Application: Digital refocusing from a single image

Application: Digital refocusing from a single image

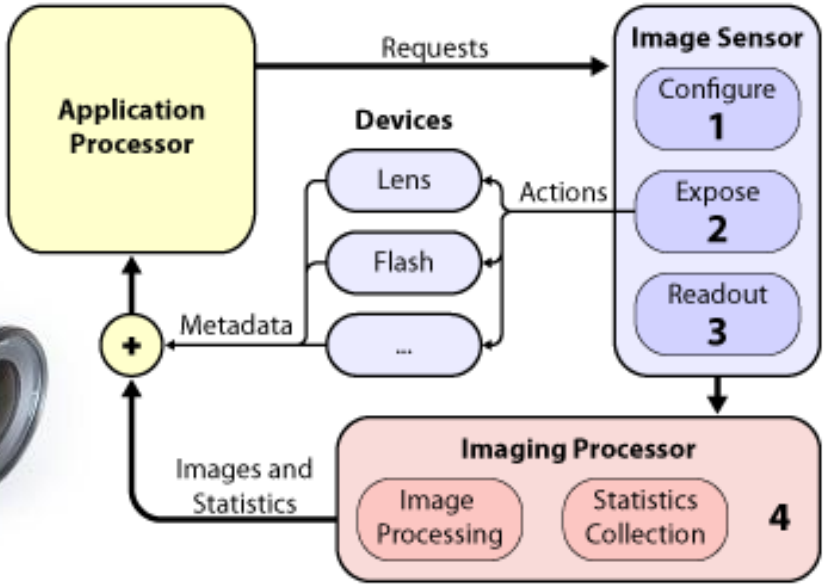


Application: Digital refocusing from a single image

The “Frankencamera”

[Adams et al. 2010]

- Programmable camera
- C++ API



http://graphics.stanford.edu/papers/fcam/

The “Frankencamera”

[Adams et al. 2010]

Conclusion

- Domaine très actif
- Pluridisciplinaire
- Interactions fortes avec l'industrie
  (Adobe, Microsoft, Disney, Mitsubichi, Google…)