Computational Photography

Photographie Algorithmique

Crédits :

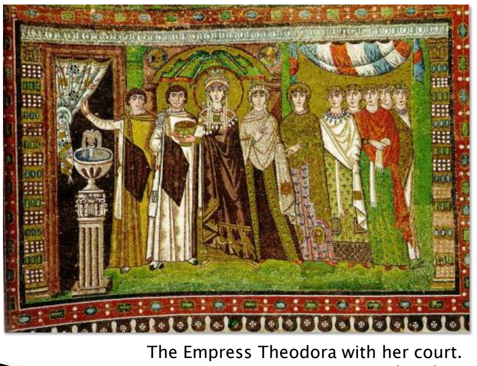
Alexei Efros, James Hays, Frédo Durand, Steve Seitz, Rick Szeliski, Paul Debevec, Stephen Palmer, Paul Heckbert, David Forsyth, Steve Marschner, Shai Avidan

Un peu d'histoire: les débuts

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

2

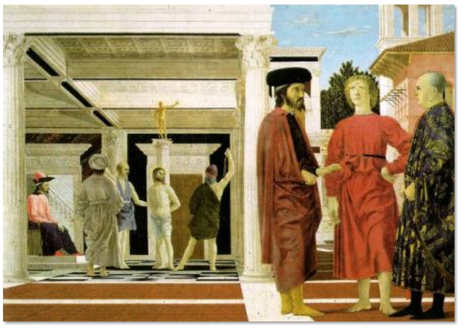
Le Moyen-Age



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

3

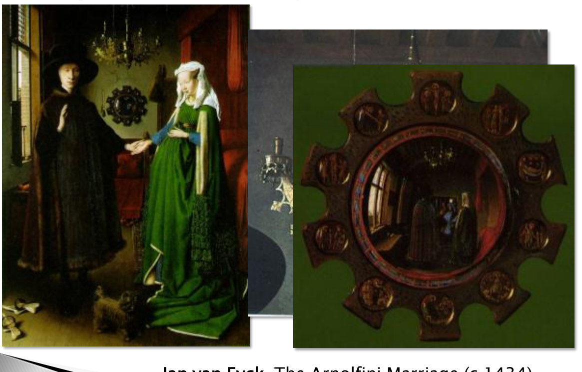
Renaissance



Piero della Francesca, The Flagellation (c.1469)

4

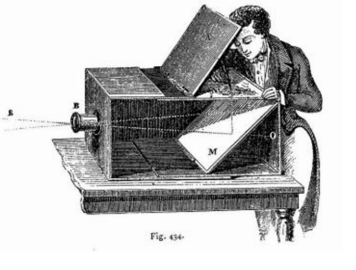
La quête de la « perfection »



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

5

La quête de la « perfection »



Lens Based Camera Obscura, 1568

6

La « perfection »

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

7

Perfection = réalité ?

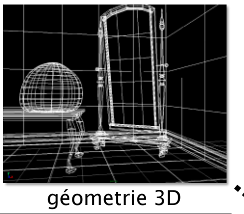
Multiple viewpoints | **Single viewpoint**

David Hockney,
Place Furstenberg, 1985

Alyosha Efros
Place Furstenberg, 2009

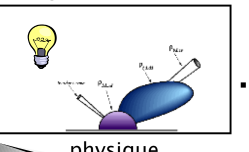
Arrivée de la synthèse d'image

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 llaguno

11

La richesse du monde réel

Photo de Svetlana Lazebnik

12

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

19

« Retouche » d'images


1860

d'après **Photo Fakery**, D. Brugioni

20

« Retouche » d'images

1942

d'après **Photo Fakery**, D. Brugioni

21

« Retouche » d'images

1942

d'après **Photo Fakery**, D. Brugioni

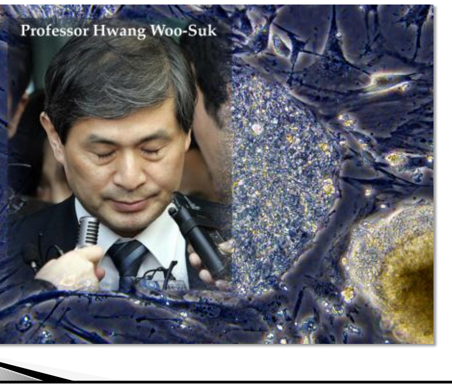
22

« Retouche » d'images


Dans la presse, juillet 2008

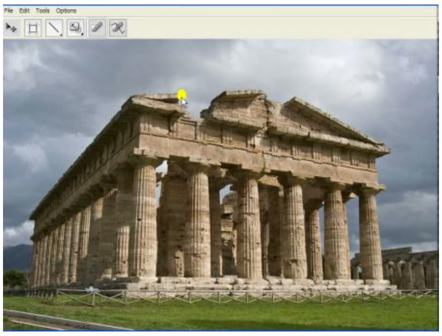
23

« Retouche » d'images



24

PatchMatch [2009]



25

Plan

- Computational processing
- Computational illumination
- Computational optics

26

Plan

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

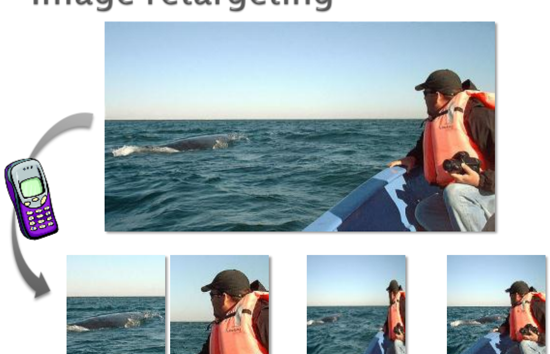
27

Plan

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

28

Image retargeting

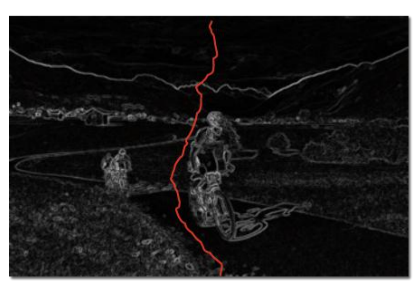


Cropping Scaling Retargeting

29

Seam carving

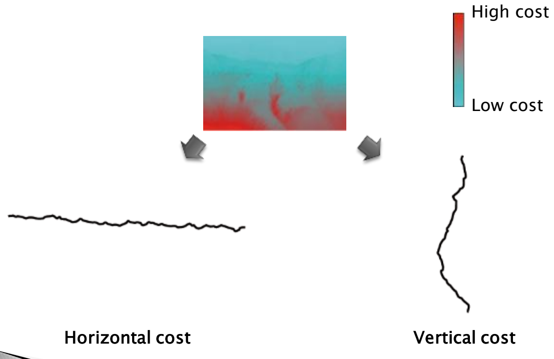
[Avidan et al. 2007]



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

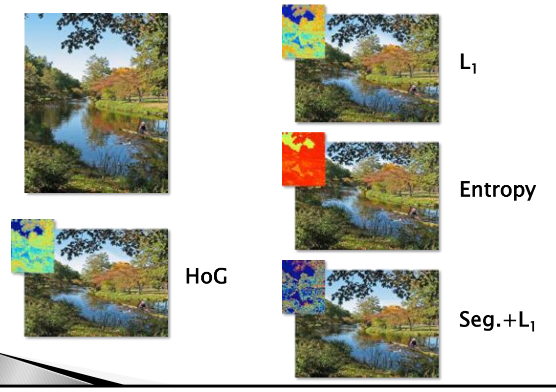
30

Dynamic Programming



31

Energy / Error functions

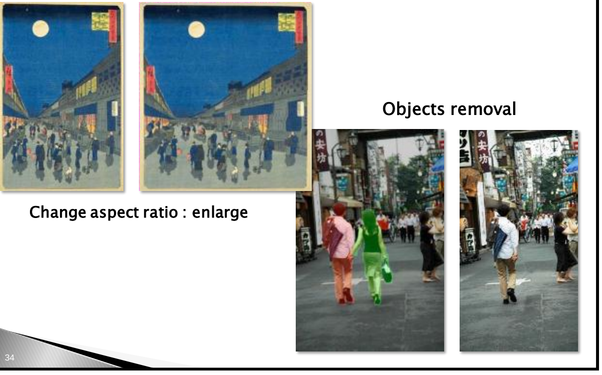


32

Results

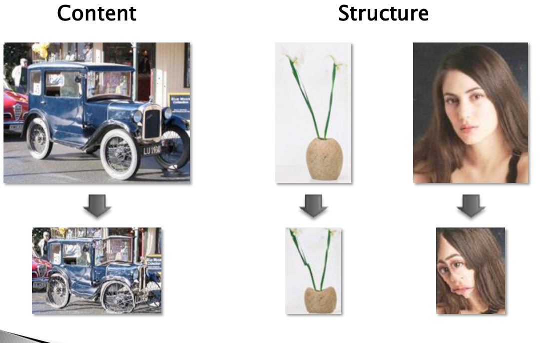
33

Results



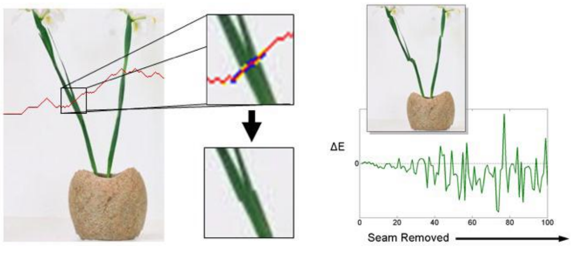
34

Limitations



35

Limitations: why?



Energy inserted into the retargeted image **ignored**

36

Forward energy

Seam carving

+ forward energy

37

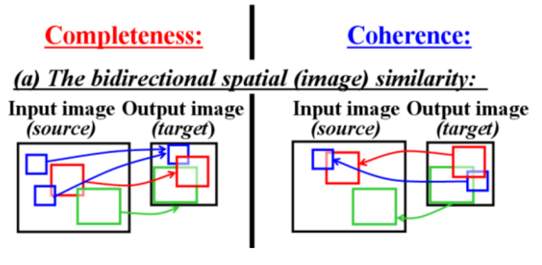
"Improved Seam Carving for Video Retargeting"

[Rubinstein et al. 2008]

38

Bidirectional similarity

[Simakov et al. 2008]



39

Plan

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

40

Filtrage

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

41

Convolution

image filtrée — noyau — image d'entrée

$$J = f * I$$

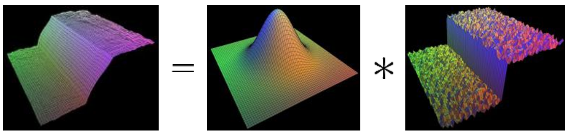
- Applications :
  - Rendre flou (blur)
  - Rendre plus net (sharpen)
  - Débruiter (denoising)
  - …

42

Filtre gaussien

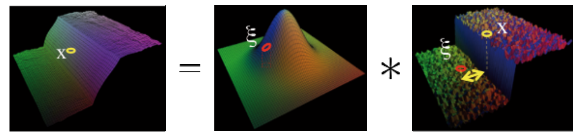
gaussienne marche + bruit

$$J = f * I$$



Filtre gaussien : somme pondérée

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



Filtre gaussien

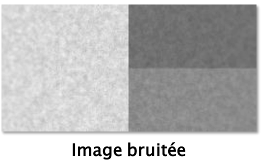
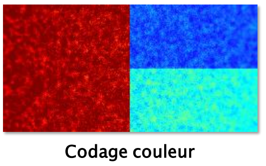
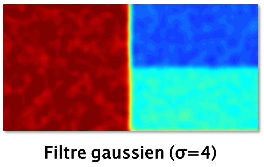


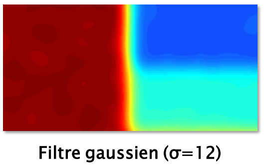
Image bruitée



Codage couleur



Filtre gaussien (σ=4)



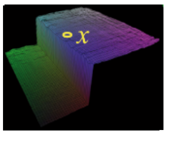
Filtre gaussien (σ=12)

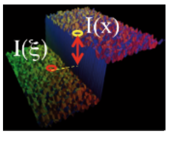
Filtre bilatéral

Tomasi and Manduci 1998

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

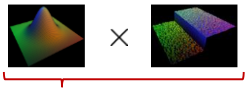
$$J(x) = \frac{1}{k(x)} \sum_{\xi} f(x,\xi)\ g(I(\xi) - I(x))\ I(\xi)$$

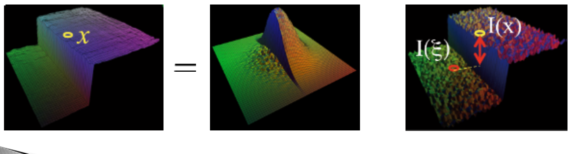




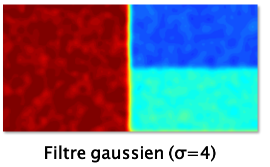
Filtre bilatéral

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

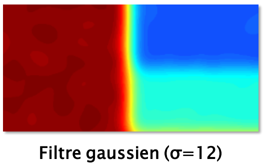




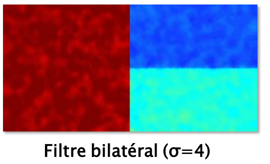
Filtre gaussien



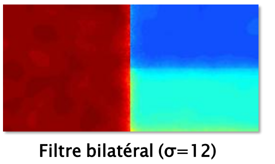
Filtre gaussien (σ=4)



Filtre gaussien (σ=12)



Filtre bilatéral (σ=4)



Filtre bilatéral (σ=12)

Débruitage

Image bruitée

Après filtrage gaussien

Débruitage

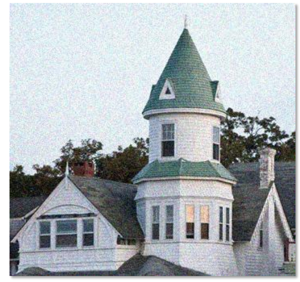
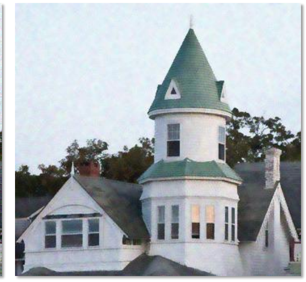


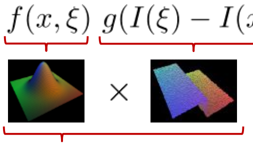
Image bruitée

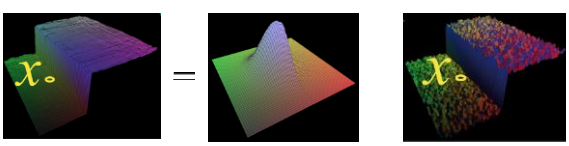


Après filtrage bilatéral

Filtre non-linéaire

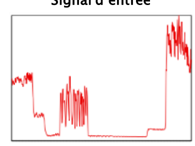
$$J(x) = \frac{1}{k(x)} \sum_{\xi} f(x,\xi)\; g(I(\xi) - I(x))\; I(\xi)$$

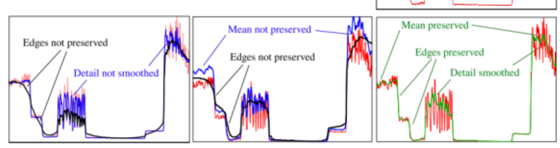




Limitation

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

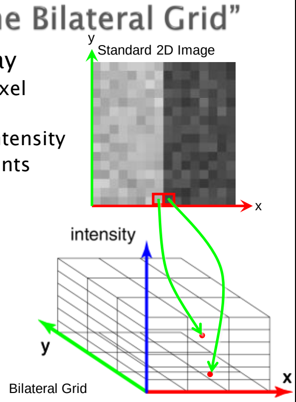




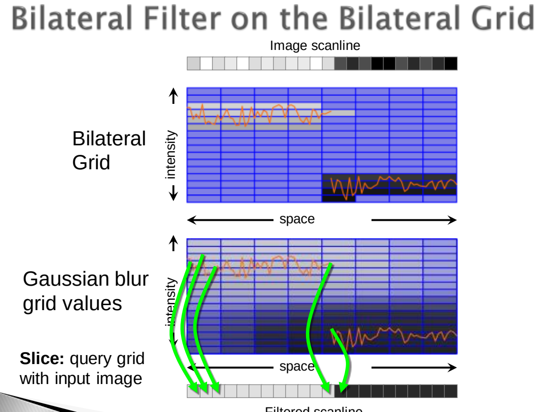
Filtre bilatéral — [Farbman et al. 2008] — [Subr et al. 2009]

"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



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

[Durand 02]

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

56

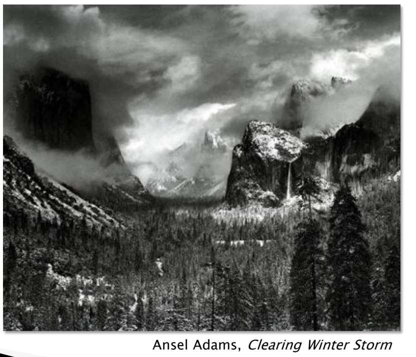
"Real-time video abstraction"

[Willmöller et al. 2002]

57

"Two-scale Tone Management for Photographic Look"

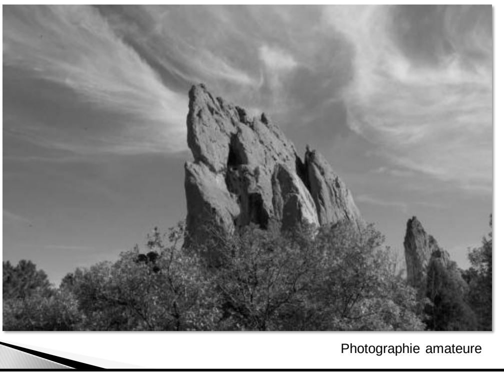
[Bae et al. 2006]



Ansel Adams, *Clearing Winter Storm*

58

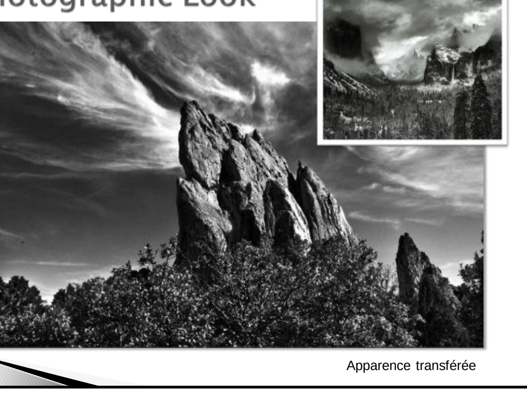
"Two-scale Tone Management for Photographic Look"



Photographie amateure

59

"Two-scale Tone Management for Photographic Look"



Apparence transférée

60

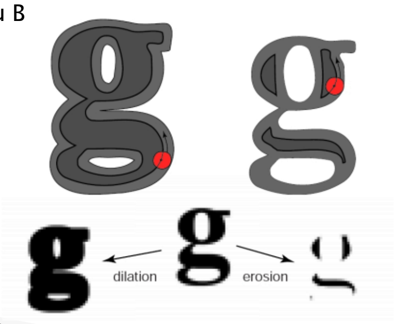
Opérateurs morphologiques

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

61

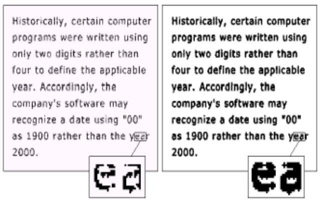
Dilatation / érosion

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



62

Dilatation


Historically, certain computer programs were written using only two digits rather than four to define the applicable year. Accordingly, the company's software may recognize a date using "00" as 1900 rather than the year 2000.

Historically, certain computer programs were written using only two digits rather than four to define the applicable year. Accordingly, the company's software may recognize a date using "00" as 1900 rather than the year 2000.

« élément de structure » ↔ noyau :

0	1	0
1	1	1
0	1	0

63

Érosion

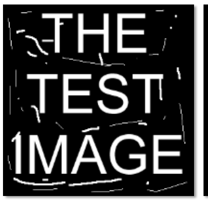


Image d'origine

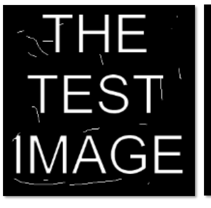


Image érodée 1x

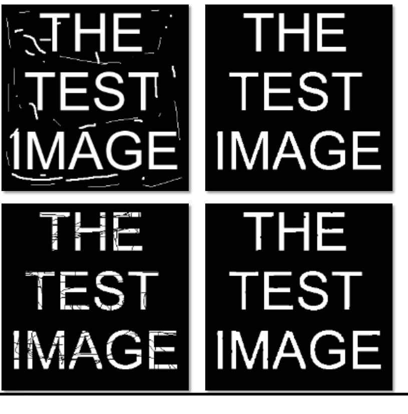
Image érodée 2x

64

Ouverture et fermeture

Ouverture :
2 érosions puis 2 dilatations.
⇨ Suppression du bruit

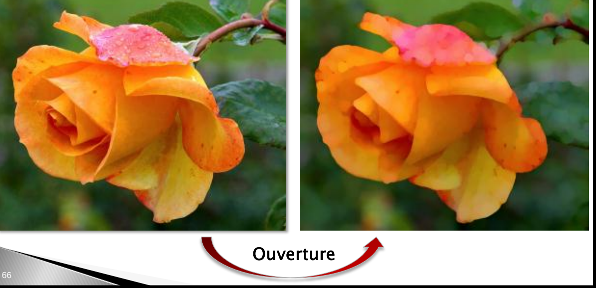
Fermeture:
dilatation puis érosion.
⇨ Remplissage des trous.



65

Application au NPR

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

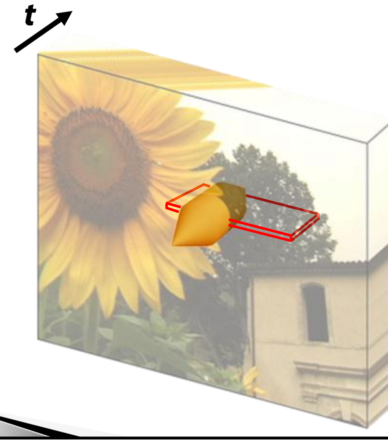


66

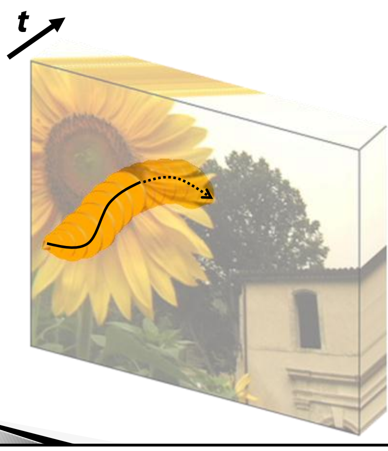
“Video Watercolorization using Bidirectional Texture Advection”

Filtrage spatial

67



68



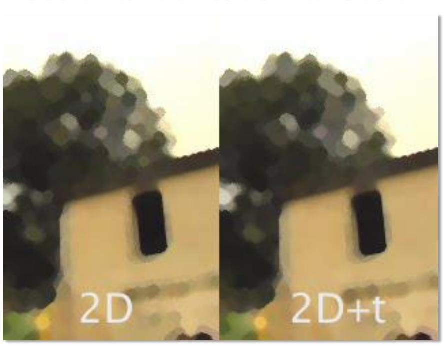
69

“Video Watercolorization using Bidirectional Texture Advection”

Filtrage spatio-temporel

70

“Video Watercolorization using Bidirectional Texture Advection”



Comparaison

71

“Video Watercolorization using Bidirectional Texture Advection”

Vidéo d’origine

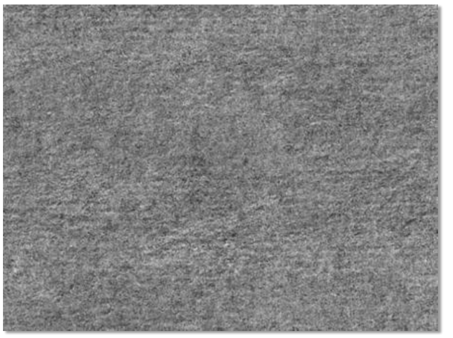
72

"Video Watercolorization using Bidirectional Texture Advection"

Filtrage spatio-temporel

73

"Video Watercolorization using Bidirectional Texture Advection"



Advection de texture

74

"Video Watercolorization using Bidirectional Texture Advection"

Stylisation aquarelle

75

Plan

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

76

Image Warping & Morphing

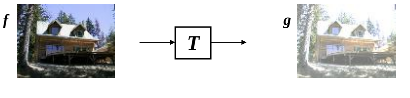
http://youtube.com/watch?v=nUDIoN-_Hxs

77

Image Warping

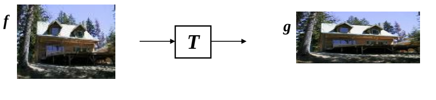
- **Filtrage** : modifie la plage de valeur

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



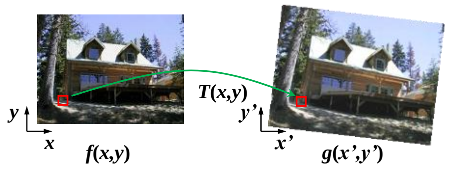
- **Warping** : modifie le domaine de l'image

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



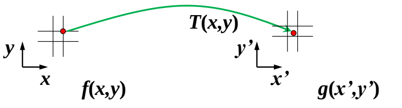
78

Image Warping



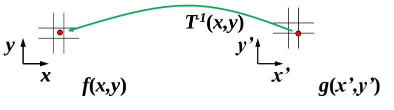
Connaissant la transformation (x',y')=T(x,y) et l'image source f(x,y), comment calculer l'image transformée g(x',y')=f(T(x,t)) ?

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 ?

85

Alignement puis fondu

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

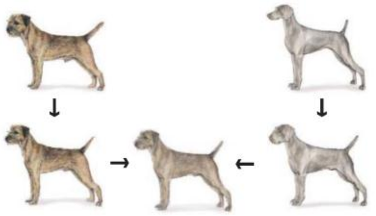
86

Et pour le chien ?

- Fondu ne marche pas
- Alignement global ne marche pas

87

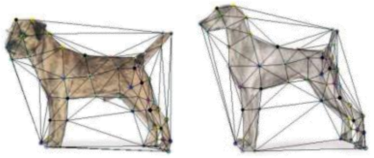
Warping local, puis fondu



Comment spécifier la transformation ?

88

Solution 1 : Maillage



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

89

Solution 2 : autres coordonnées

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

90

Solution 2 : autres coordonnées

91

Solution 2 : autres coordonnées

92

Plan

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

93

Comment volent les super-héros ?

Superpouvoirs ?
ou
Image matting ?

94

Image matting – Compositing

Cinefex

95

Compositing

Digital Domain

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

96

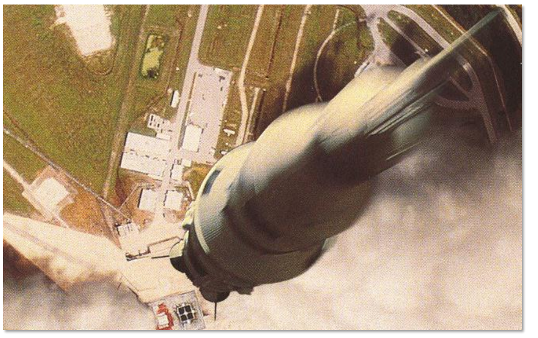
Alpha binaire

Digital Domain

97

Alpha continue

Digital Domain



98

L'écran bleu

99

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 : $\alpha = \mathbf{1 - p_1(b - p_2\, g)}$
  - $b<p_2\, g$, avec a_2 entre 0.5 et 1.5
  - p_1 et p_2 définis par l'utilisateur

100

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

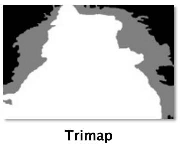
101

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

102

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

103

Bayesian matting

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

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

le paramètre à estimer | ce qu'on observe | fonction de vraisemblance | probabilité a priori | constante vis-à-vis de x

104

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

105

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

106

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

107

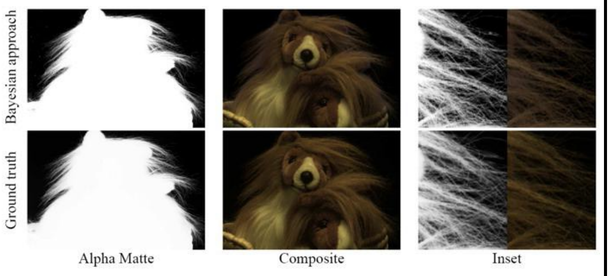
Bayesian matting

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

108

Bayesian matting

Chuang et al. 2001



Bayesian matting

Chuang et al. 2001

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

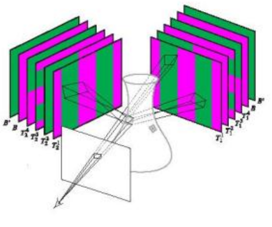
$$C = F + (1-\alpha)B + \Phi$$

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

Explanation of Φ

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

R – reflectance image
T – Texture image



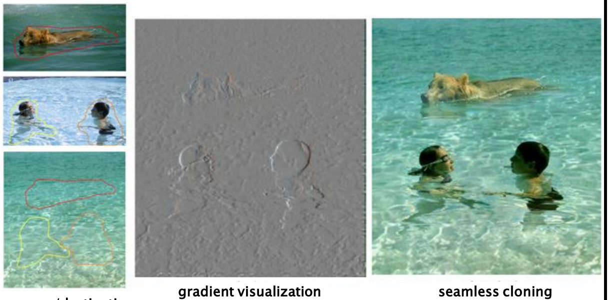
Environment matting

Plan

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

116

Gradient Domain Blending

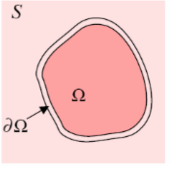


[Perez et al. 2003]

117

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



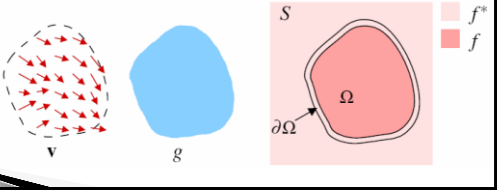
118

Gradient Domain Blending

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

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

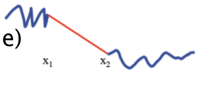
⇨ équation de Poisson avec conditions au bord de Dirichlet



119

Gradient Domain Blending

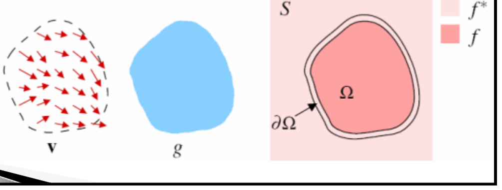
- Si **v** est nul
- Equation de Laplace (membrane)
- En 1D = interpolation linéaire
- En 2D :



120

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



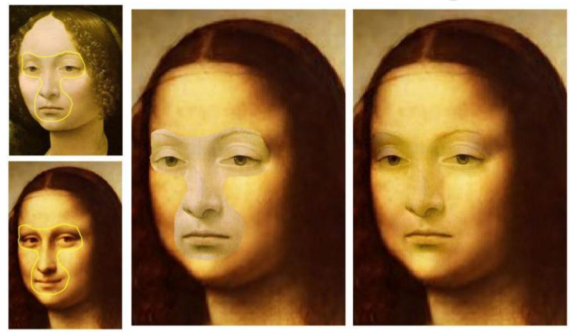
121

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…

122

Gradient Domain Blending



sources/destination — cloning — seamless cloning

123

Local color changes

124

Limitations

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

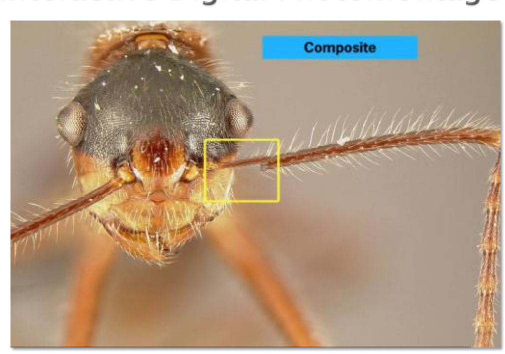
125

Applications

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

126

"Interactive Digital Photomontage"



[Agarwala et al. 2004]

127

Applications

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

128

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]

[Orzan et al. 2008]

[McCann et al. 2008]

129

Plan

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

130

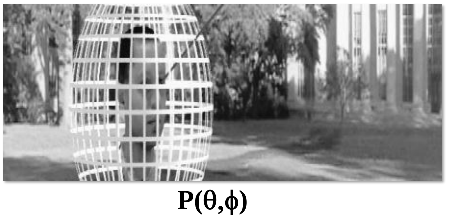
The Plenoptic Function [Adelson & Bergen 91]

Crédit: Leonard McMillan

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

131

Grayscale snapshot



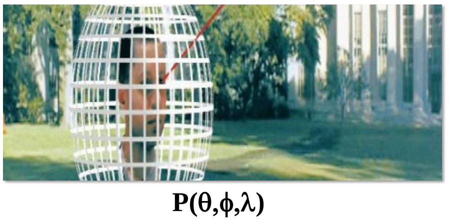
$$P(\theta,\phi)$$

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)

132

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**

133

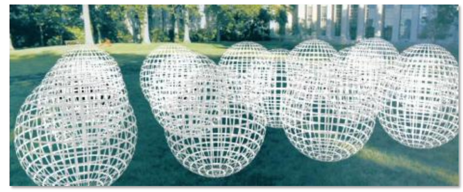
A movie

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

is intensity of light
- seen from a single view point
- over time
- as a function of wavelength

134

Holographic movie

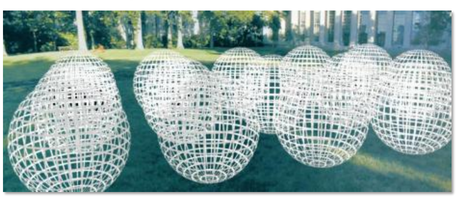


$P(\theta,\phi,\lambda,t,V_X,V_Y,V_Z)$

is intensity of light
- seen from ANY viewpoint
- over time
- as a function of wavelength

135

The Plenoptic Function



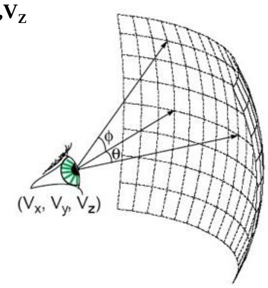
$P(\theta,\phi,\lambda,t,V_X,V_Y,V_Z)$

- 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…

136

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

137

Lumigraph / Light fields

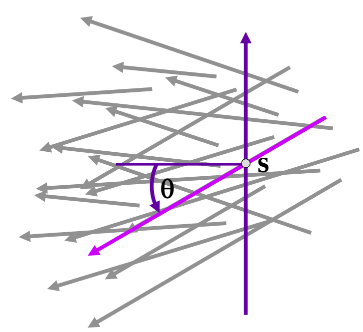
- Extérieur = enveloppe convexe de la scène
- Pour chaque rayon dans cet espace, capturer et stocker la radiance (couleur RGB)
- Rendu = lookup
- 2 publications en1996
  - "Light field rendering" [Levoy & Hanrahan]
  - "The Lumigraph" [Gortler et al.]

138

Lumigraph - Organization

2D position: s

2D direction: θ



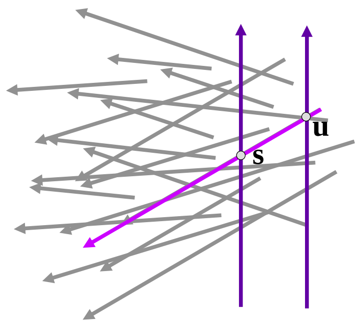
Slide by Rick Szeliski and Michael Cohen

139

Lumigraph – Organization

2D position: **s**

2D position: **u**



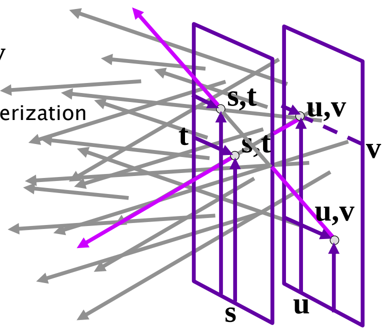
Slide by Rick Szeliski and Michael Cohen

Lumigraph – Organization

2D position: **s,t**

2D position: **u,v**

2 plane parameterization



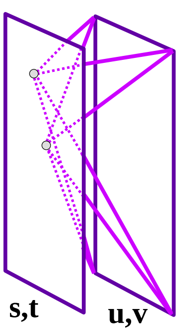
Slide by Rick Szeliski and Michael Cohen

Lumigraph – Organization

Hold **s,t** constant

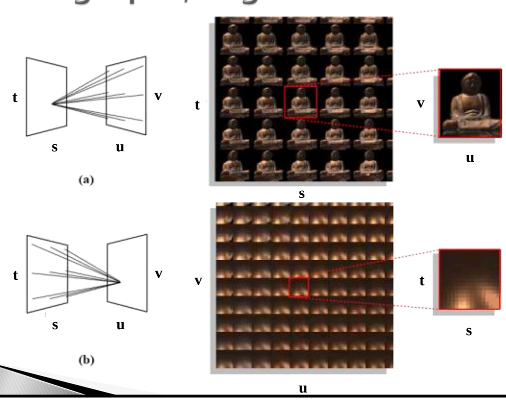
Let **u,v** vary

⇨ an image



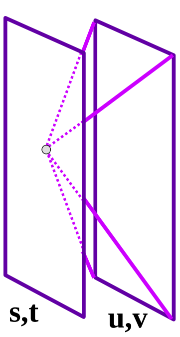
Slide by Rick Szeliski and Michael Cohen

Lumigraph / Light field



Lumigraph – Capture

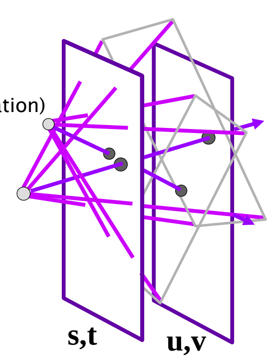
- Idea 1
  - Move camera carefully over s,t plane



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/

146

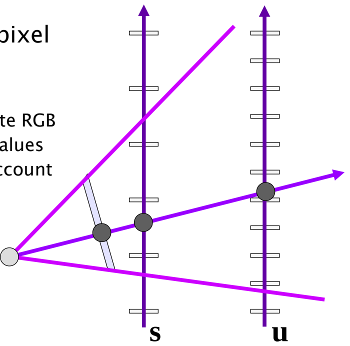
Time splice

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

147

Lumigraph – Rendering

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



148

Slide by Rick Szeliski and Michael Cohen

Démo lightfield viewer

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

149

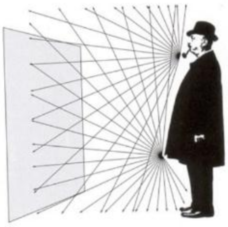
Plan

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

150

Image capture

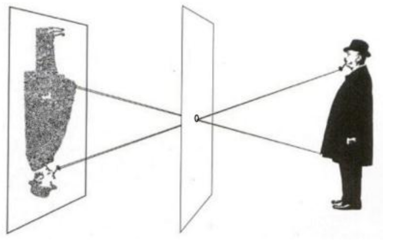
- Un capteur seul ne capture pas une image



151

Image capture

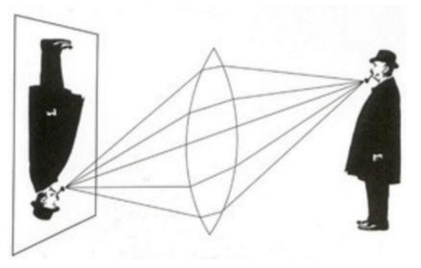
- Un sténopé *(pinehole)* permet de sélectionner certains rayons



152

Image formation: optics

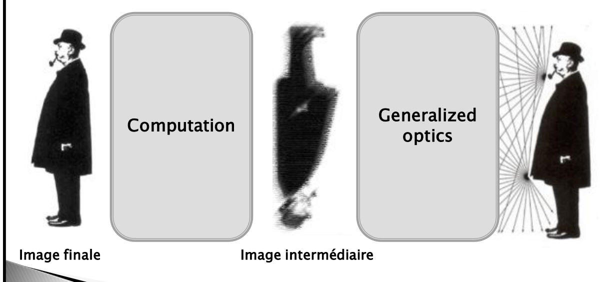
- Les optiques sélectionnent et intègrent les rayons lumineux ⇨ **forment** une image



153

Image formation: computation

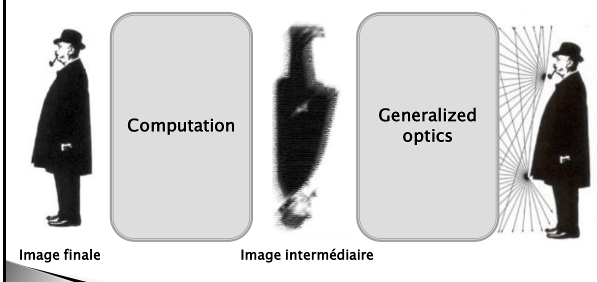
- La combinaison optiques & algorithmes forme une image



154

Computational imaging goals

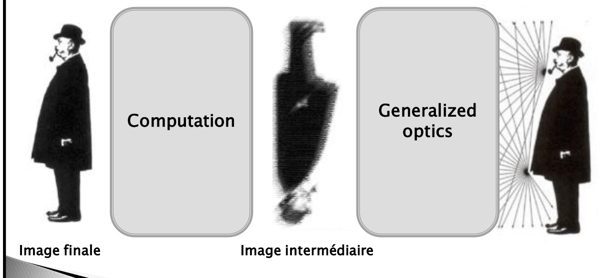
- Meilleur capture de l'information
- Former un image en post-process



155

Better capture information

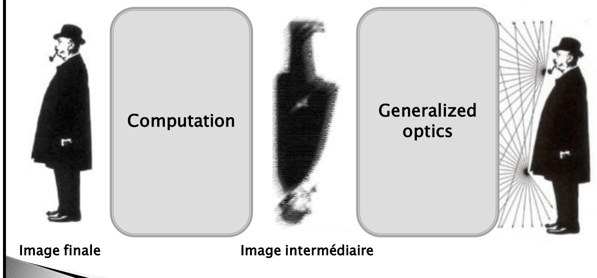
- Les optiques encodent, l'algorithme décode
- L'encodage vise à minimiser les distorsions



156

Form images as a post-process

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



157

Plan

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

Digital refocusing

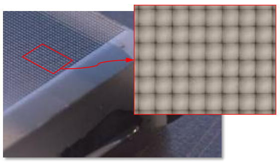
[Ng et al. 2005]

Prototype camera

Contax medium format camera

Kodak 16-megapixel sensor

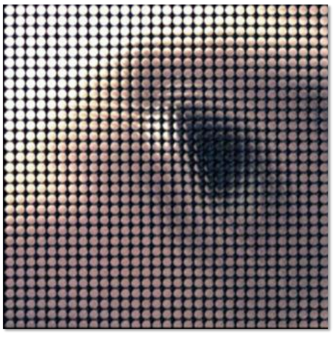
Adaptive Optics microlens array



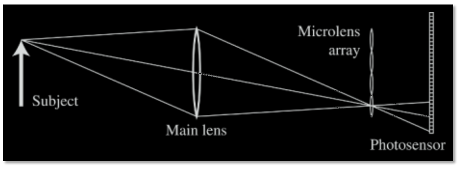
125μ square-sided microlenses

Light Field in a Single Exposure

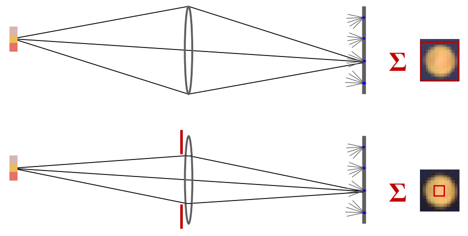
Light Field in a Single Exposure



Light field camera



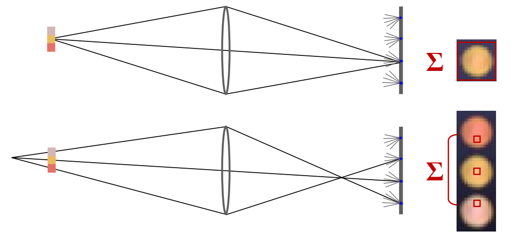
Digitally stopping-down



stopping down = summing only the central portion of each microlens

164

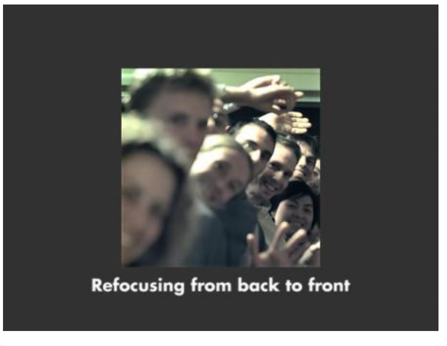
Digital refocusing



refocusing = summing windows extracted from several microlenses

165

Résultats



166

Plan

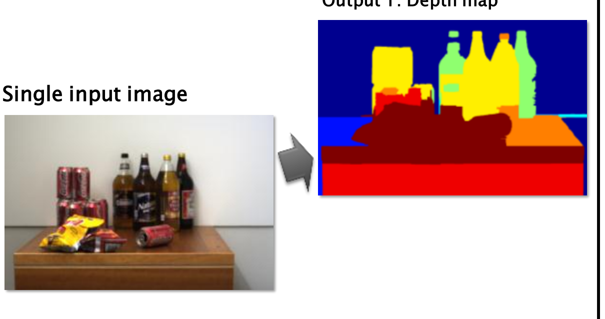
- Computational processing
- Computational illumination
- **Computational optics**
  - Digital Refocusing
  - Coded Aperture

167

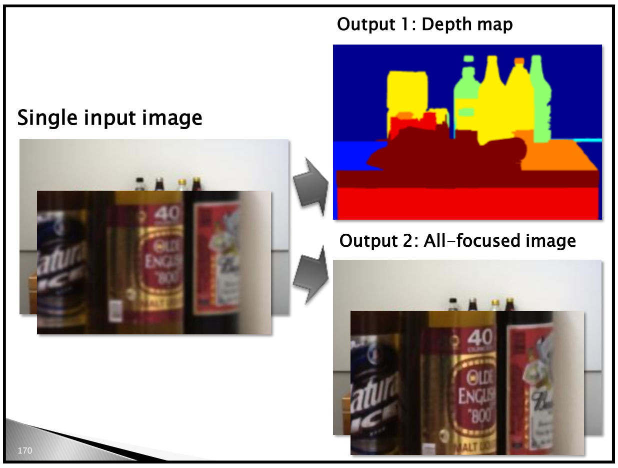
"Image and Depth from a Conventional Camera with a Coded Aperture"

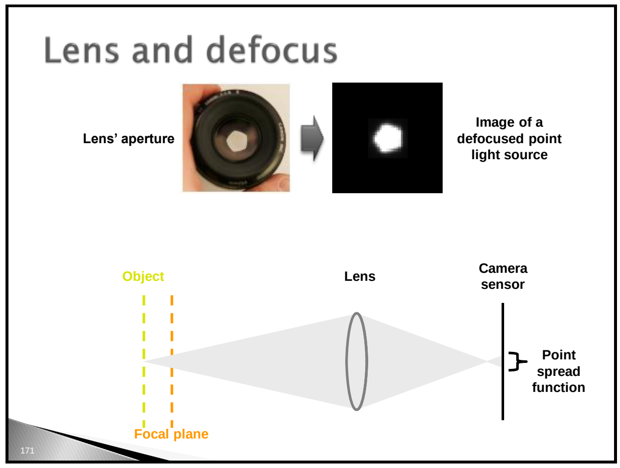
[Levin et al. 2007]

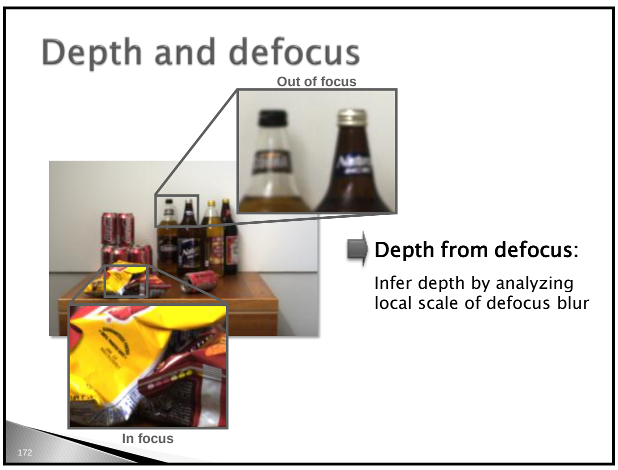
168

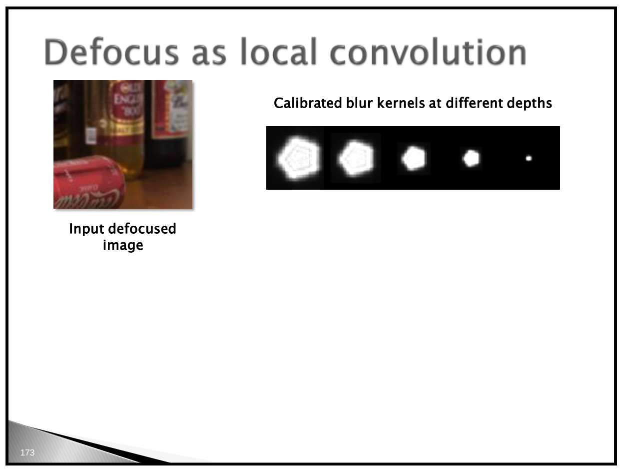


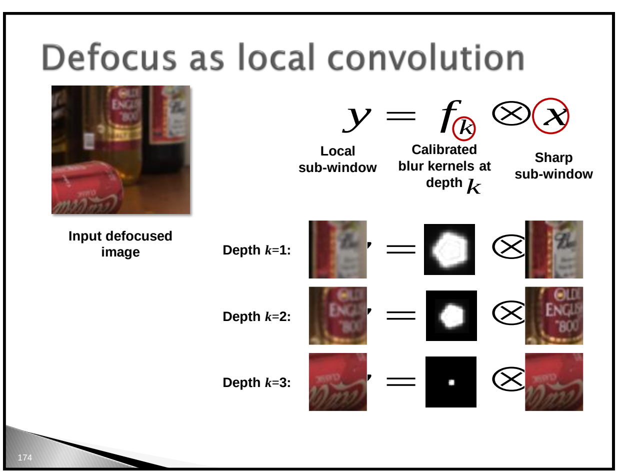
169

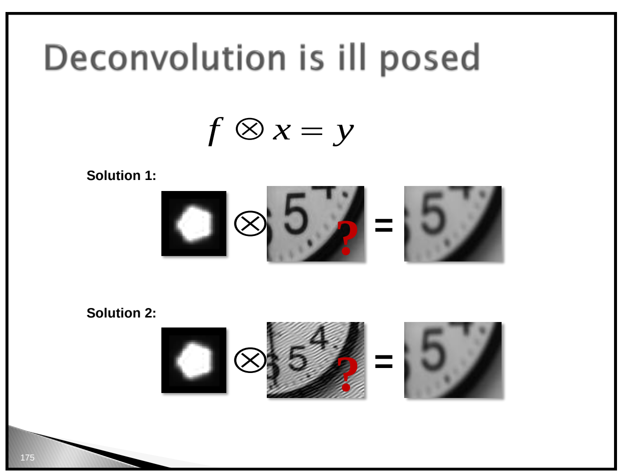

Output 1: Depth map
Single input image
Output 2: All-focused image
170


Lens and defocus
Lens' aperture
Image of a defocused point light source
Object
Lens
Camera sensor
Point spread function
Focal plane
171


Depth and defocus
Out of focus
Depth from defocus:
Infer depth by analyzing local scale of defocus blur
In focus
172

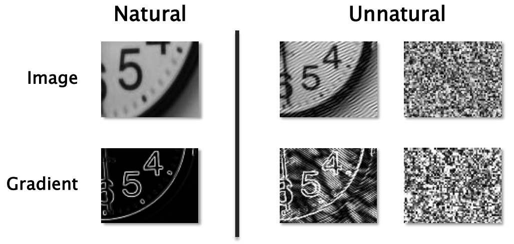

Defocus as local convolution
Calibrated blur kernels at different depths
Input defocused image
173


Defocus as local convolution
$y = f_k \otimes x$
Local sub-window
Calibrated blur kernels at depth k
Sharp sub-window
Input defocused image
Depth k=1:
Depth k=2:
Depth k=3:
174


Deconvolution is ill posed
$f \otimes x = y$
Solution 1:
Solution 2:
175

Idea 1: Natural images prior

What makes images special?



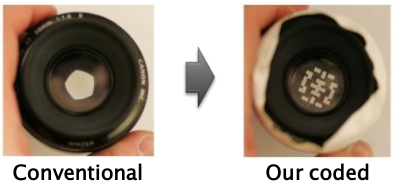
Natural images have sparse gradients

put a penalty on gradients

176

Idea 2: Coded Aperture

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

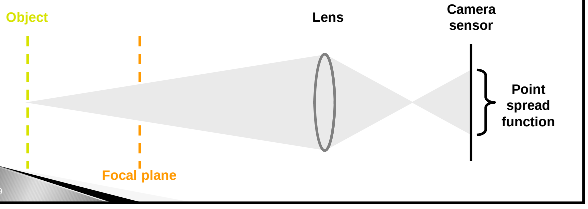


177

Build your own coded aperture

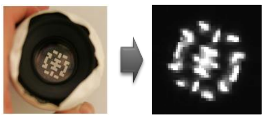
178

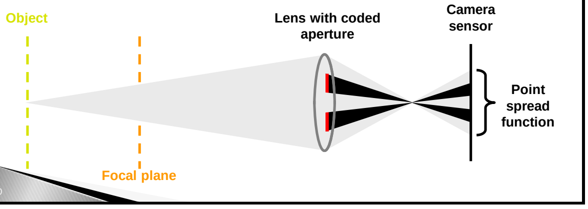
Solution: lens with occluder



179

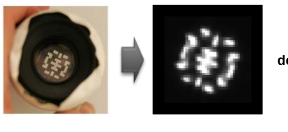
Solution: lens with occluder

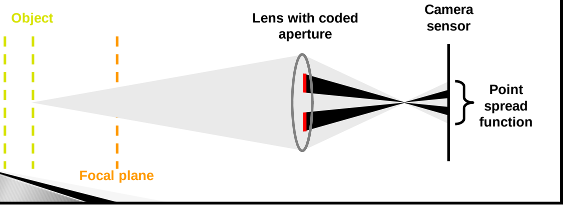




180

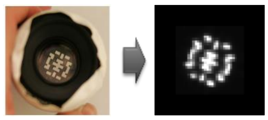
Solution: lens with occluder

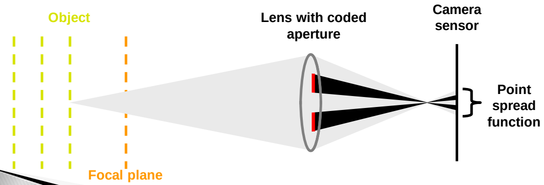




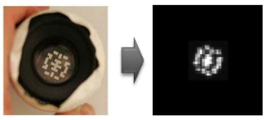
181

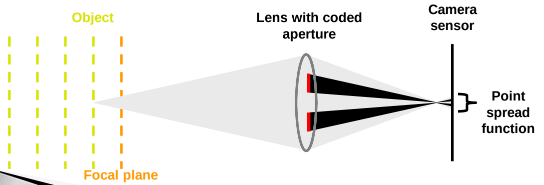
Solution: lens with occluder



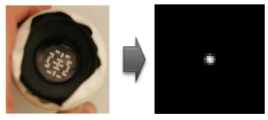


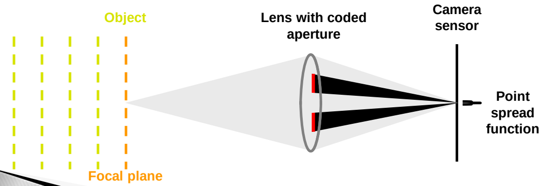
Solution: lens with occluder





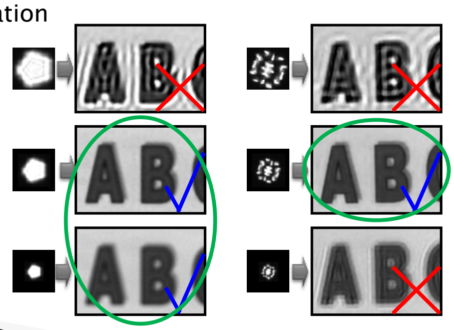
Solution: lens with occluder





Why coded?

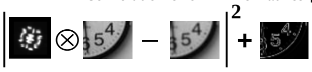
Coded aperture reduces uncertainty in scale identification



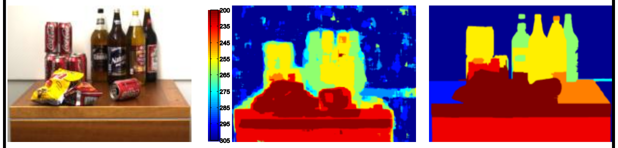
Regularizing depth estimation

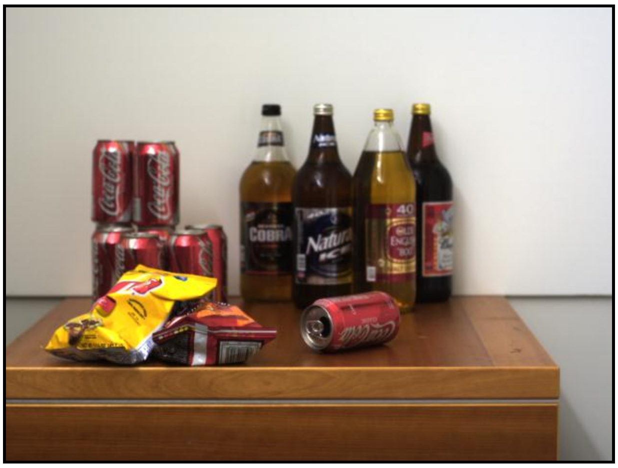
Try deblurring with 10 different aperture scales

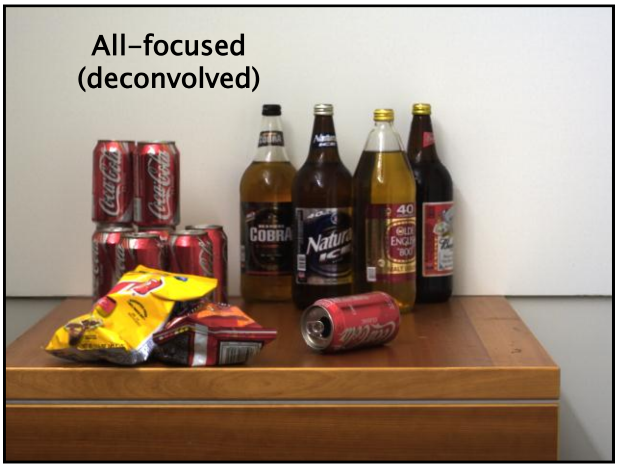
$$x = \arg\min \underbrace{|f \otimes x - y|^2}_{\text{Convolution error}} + \underbrace{\lambda \sum_i \rho(\nabla x_i)}_{\text{Derivatives prior}}$$



Keep minimal error scale in each local window + regularization







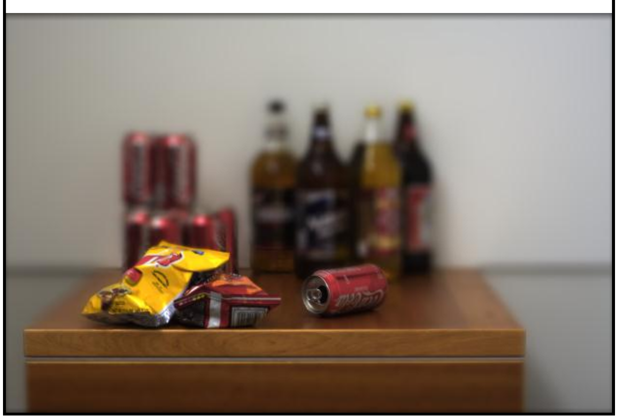
Close-up

Original image

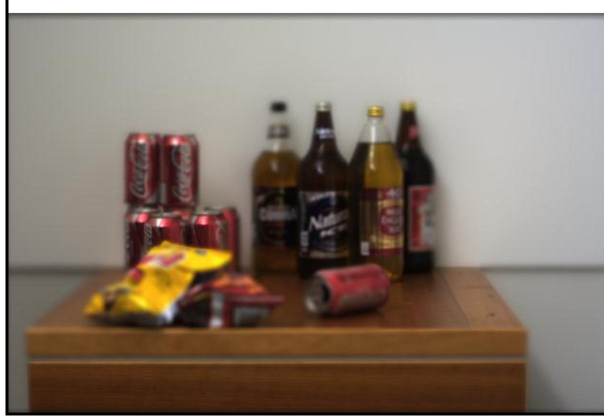
All-focus image

189

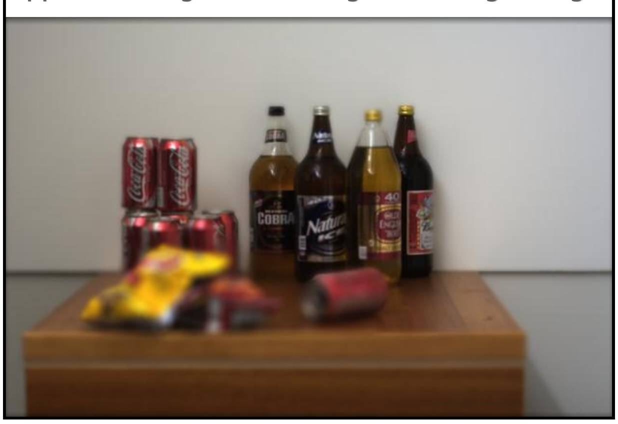
Application: Digital refocusing from a single image



Application: Digital refocusing from a single image



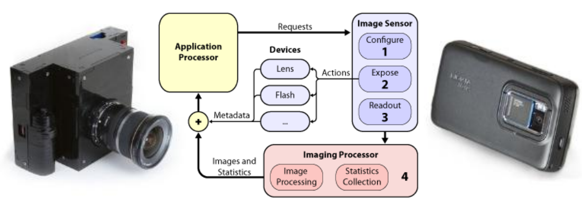
Application: Digital refocusing from a single image



The "Frankencamera"

[Adams et al. 2010]

- Programmable camera
- C++ API



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

193

The "Frankencamera"

[Adams et al. 2010]

194

Conclusion

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

195