

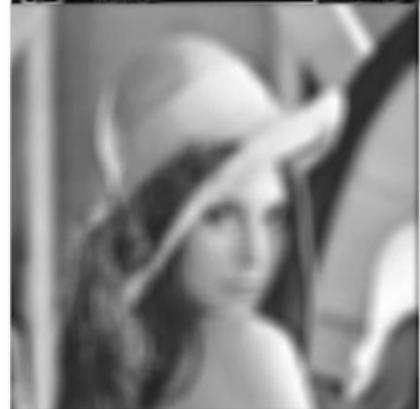
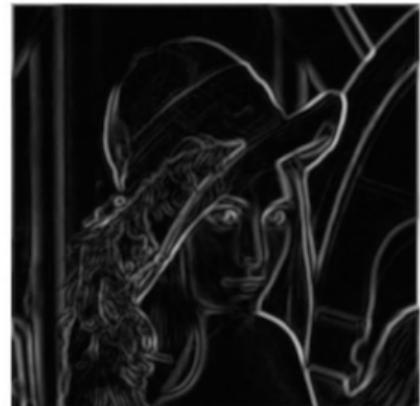
Transformations ?

Traitement d'images = **Ensemble d'opérations** qui transforme une image numérique en une autre image numérique.

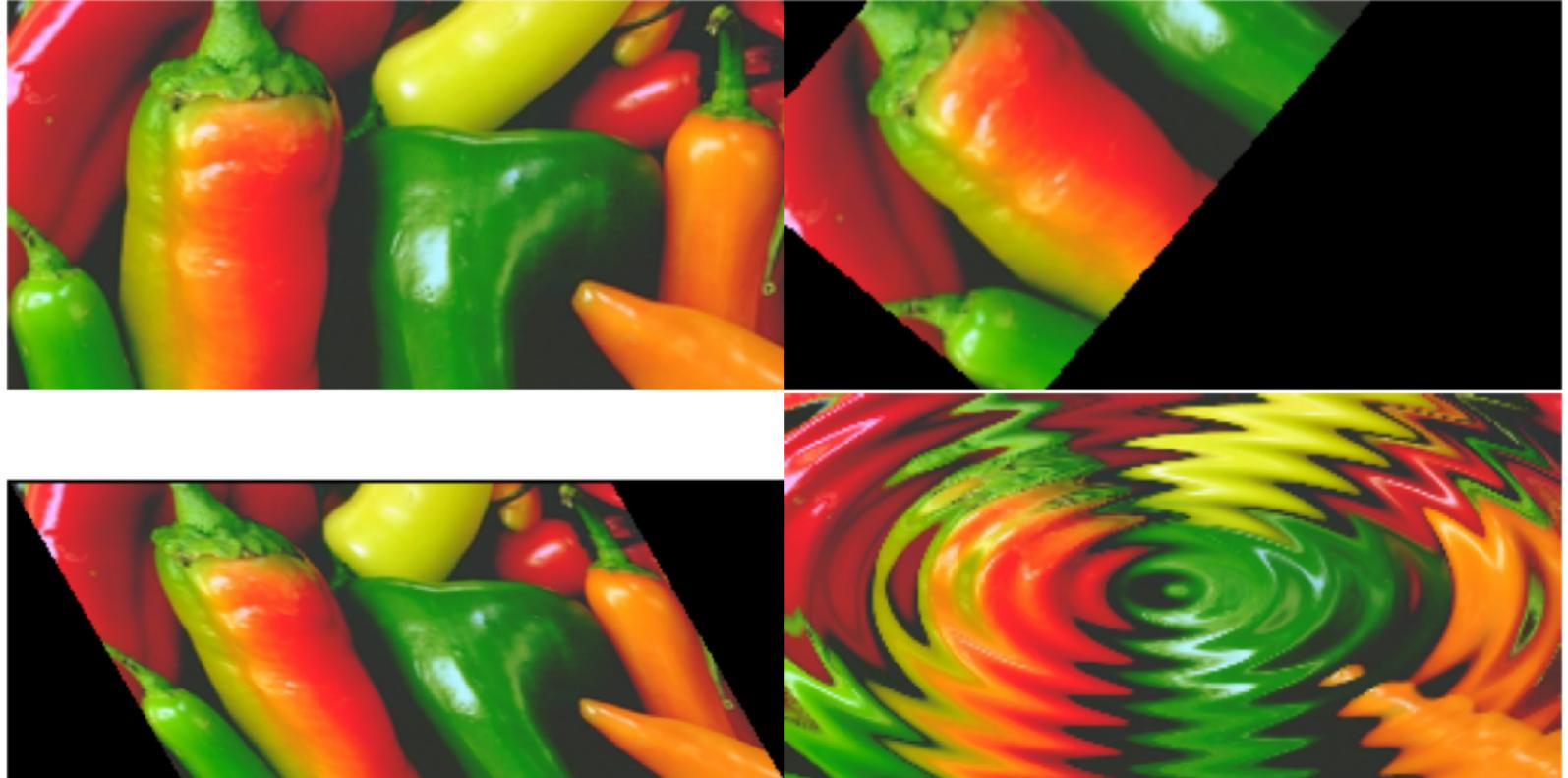
Quels types de transformations ?

- **colorimétriques** : modifier la valeur du pixel (les canaux C)
- **géométriques** : modifier la position du pixel (le domaine D)

Transformation colorimétrique – Modifier la **valeur** du pixel



Transformation géométrique – Modifier la **position** du pixel



Transformations géométriques

Transformation géométrique

Modifier la **position** du pixel sans (trop) modifier sa valeur

image initiale $I : D \rightarrow C$

une transformation $h : D \rightarrow D'$

une image transformée $I' : D' \rightarrow C$ (C est \pm conservé) t.q. $I' = h(I)$

réciproquement $I = h^{-1}(I')$: **transformation inverse** (si h est une fonction inversible)

Ex. de transformation h : translation, rotation, cisaillement, ...

Translation



$$x' = x + d_x \text{ et } y' = y + d_y$$

ou

$$\begin{pmatrix} x' \\ y' \end{pmatrix} = \begin{pmatrix} x \\ y \end{pmatrix} + \begin{pmatrix} d_x \\ d_y \end{pmatrix}$$

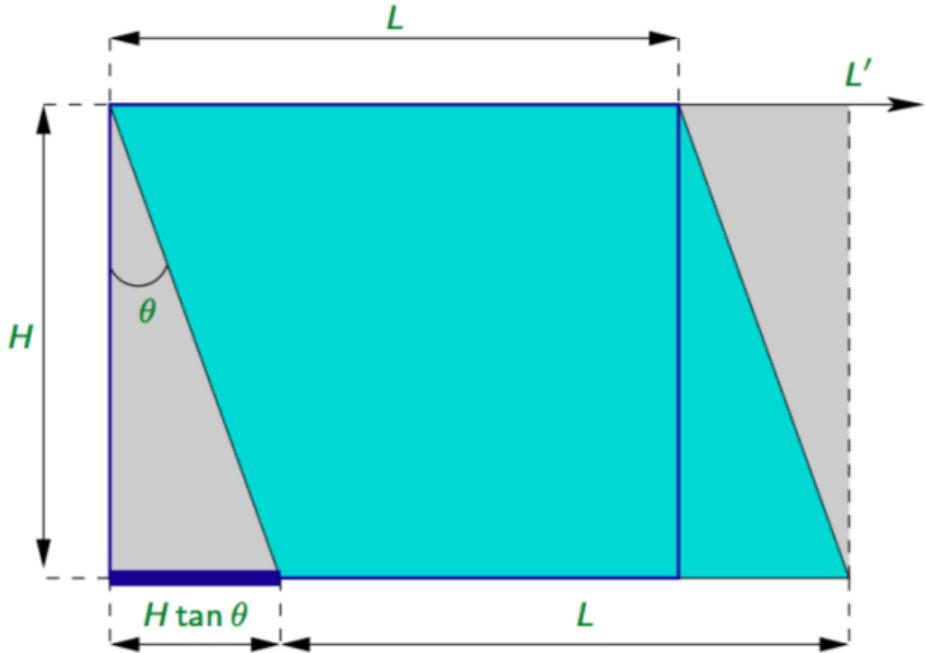
avec $(x', y') \in I'$ et $(x, y) \in I$

Cisaillement (a.k.a shear)



Cisaillement **horizontal** ou **vertical** d'angle θ

Cisaillement horizontal



$$(x', y') = (x + y \tan(\theta), y)$$

ou

$$\begin{pmatrix} x' \\ y' \end{pmatrix} = \begin{pmatrix} 1 & a \\ 0 & 1 \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix}$$

et verticalement

$$\begin{pmatrix} x' \\ y' \end{pmatrix} = \begin{pmatrix} 1 & 0 \\ b & 1 \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix}$$

Rotation d'angle θ



$$x' = x \cos \theta - y \sin \theta \text{ et } y' = x \sin \theta + y \cos \theta$$

ou

$$\begin{pmatrix} x' \\ y' \end{pmatrix} = \begin{pmatrix} \cos \theta & -\sin \theta \\ \sin \theta & \cos \theta \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix}$$

avec $(x', y') \in I'$ et $(x, y) \in I$ et centrée en $(0, 0)$

Rotation avec cisaillement d'angle θ

$$\begin{pmatrix} \cos \theta & -\sin \theta \\ \sin \theta & \cos \theta \end{pmatrix} = A \cdot B \cdot C = \begin{pmatrix} 1 & a \\ 0 & 1 \end{pmatrix} \begin{pmatrix} 1 & 0 \\ b & 1 \end{pmatrix} \begin{pmatrix} 1 & c \\ 0 & 1 \end{pmatrix}$$

A, B, C sont **3 matrices de cisaillement**

Une rotation d'angle θ peut se faire par

- A, C : 2 cisaillements horizontaux d'angle $-\theta/2$
- B : un cisaillement vertical d'angle $\alpha = \arctan(\sin(\theta))$.

Rotation multi-passe



Rotation de 30° :

- cisaillement horizontal -15°
- cisaillement vertical 26.57°
- cisaillement horizontal -15°

Déformations non linéaires

h peut être une fonction quelconque : permet de faire des effets sur les images

Par exemple :

- *tourbillon*
- *vague*
- *sphérique*
- ...

Effet tourbillon



Effet vague



Effet *sphere*



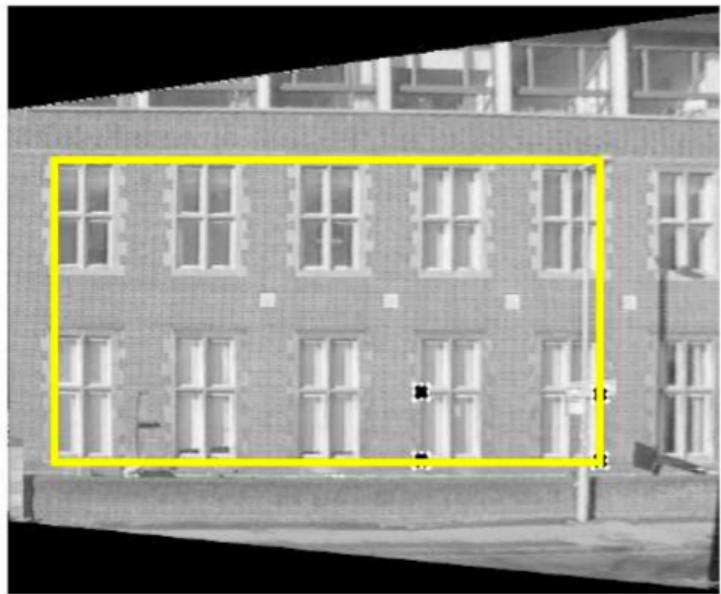
Exemple d'application : recollement d'image



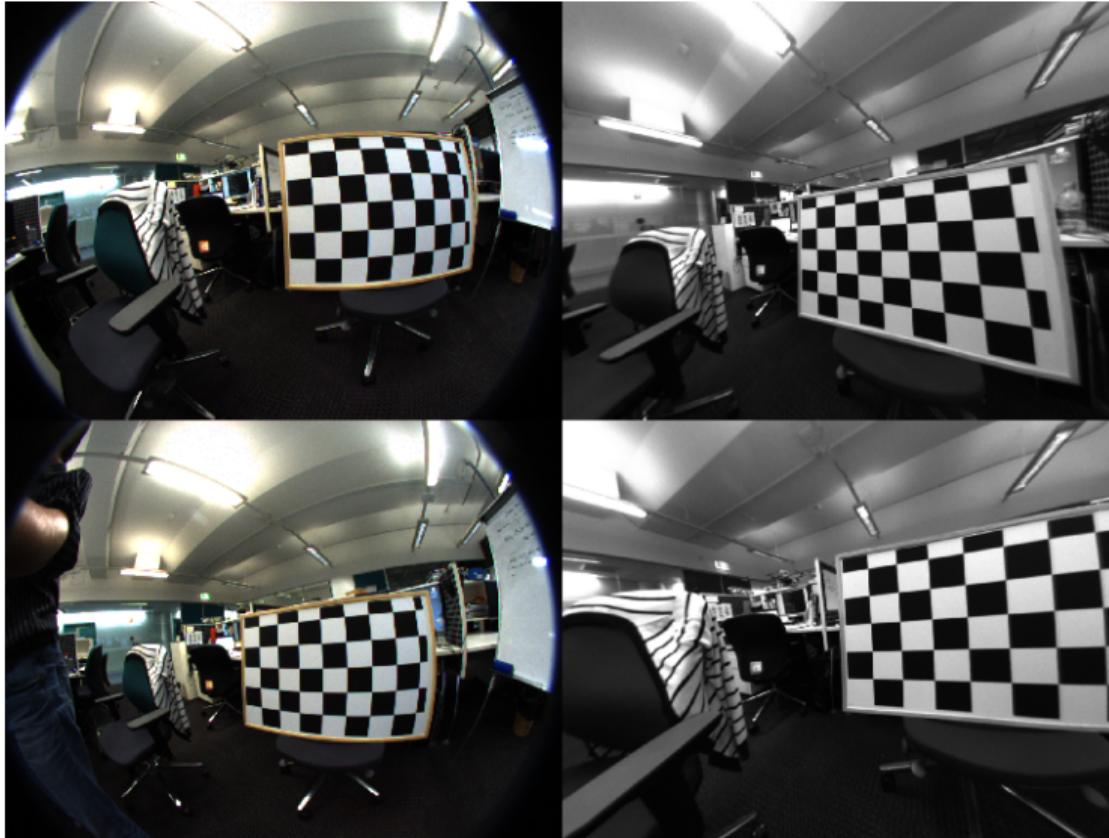
Exemple d'application : panorama



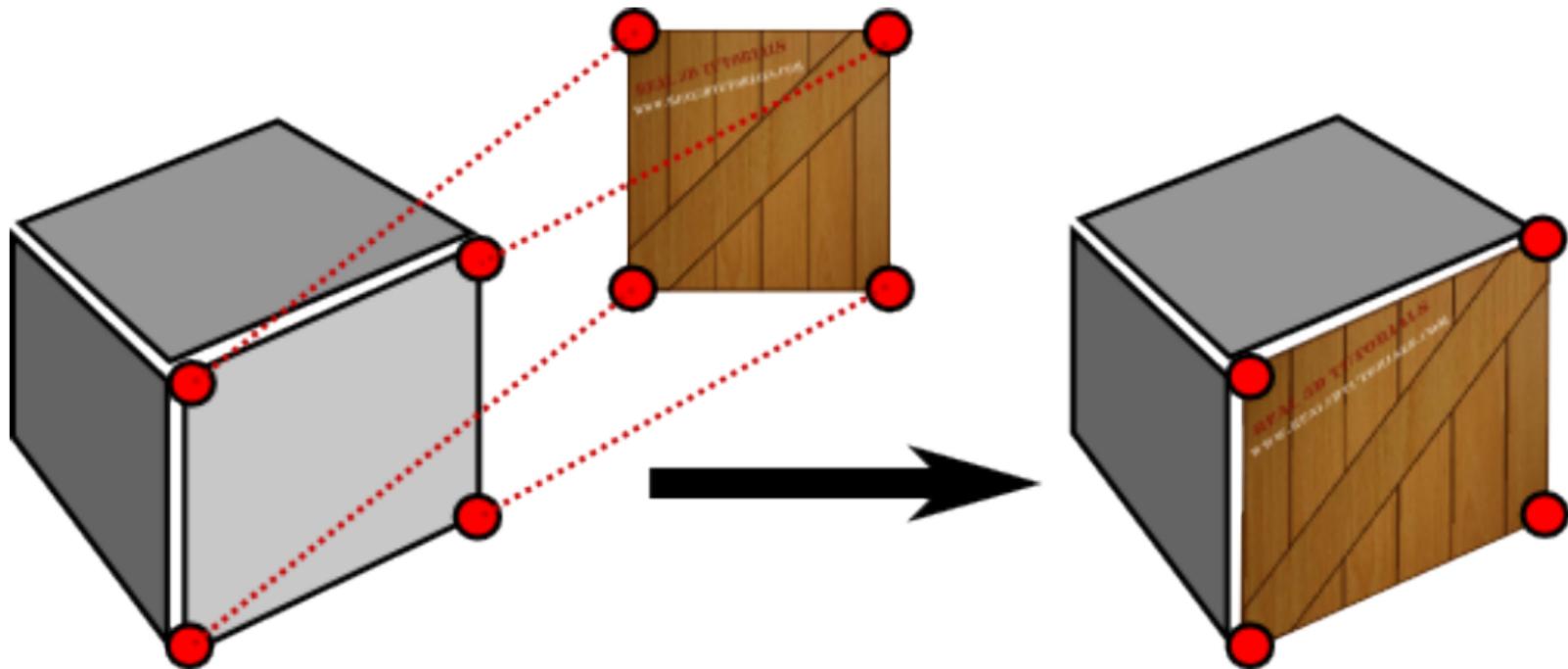
Exemple d'application : rectification de vue



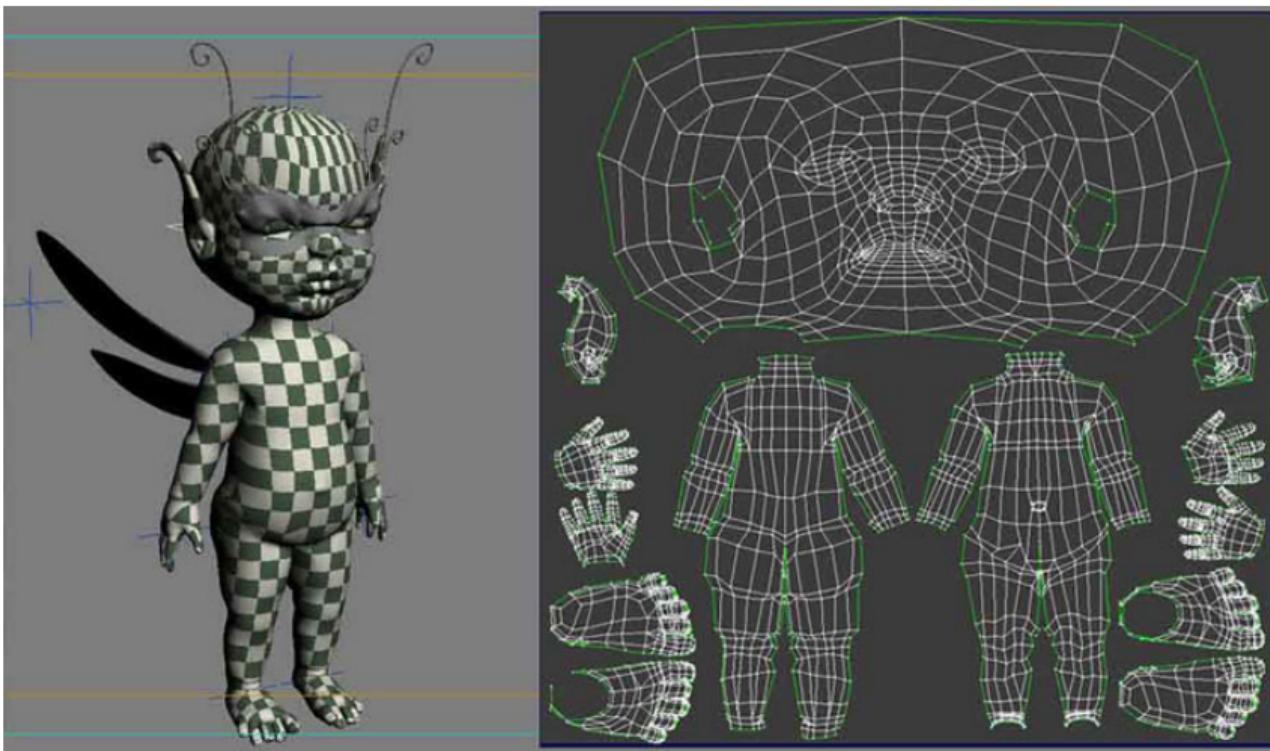
Exemple d'application : calibration de caméra



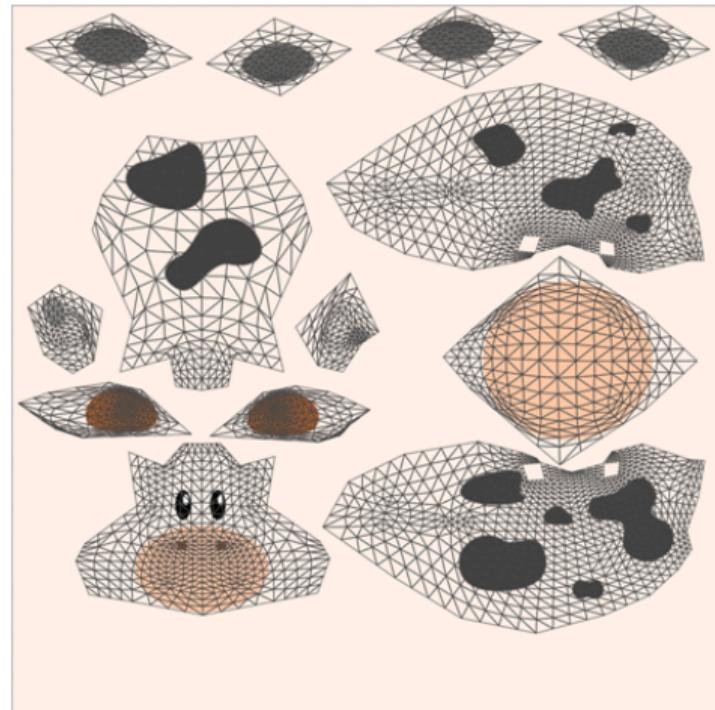
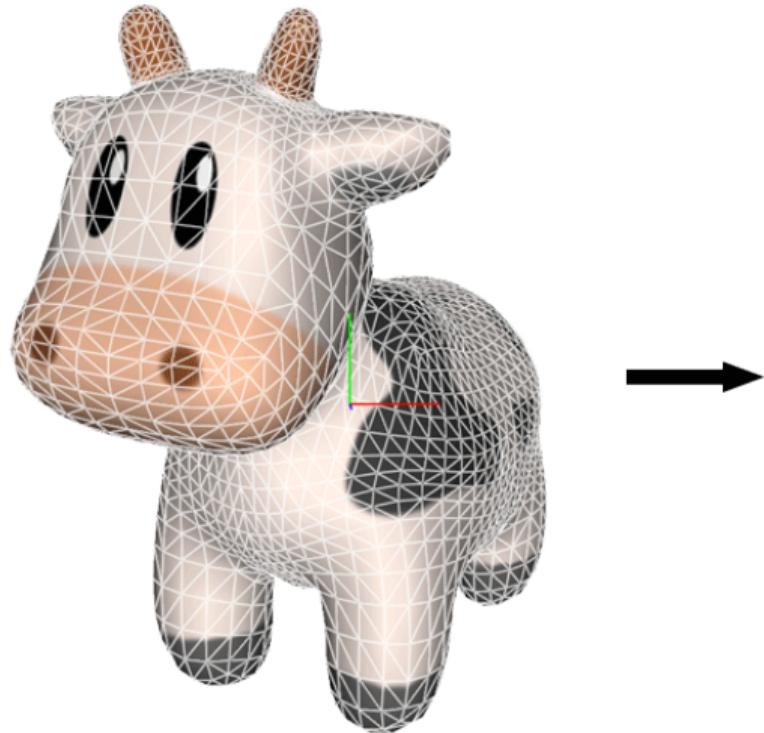
Exemple d'application : plaquage de texture



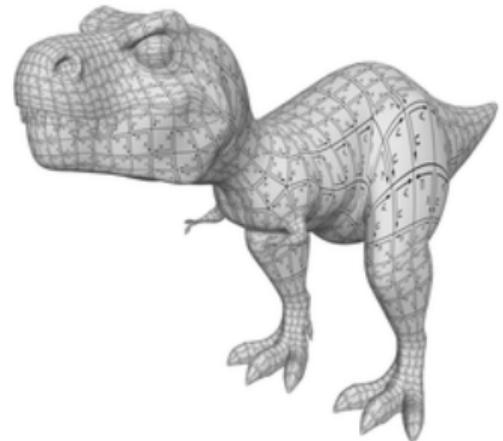
Exemple d'application : dépliement de surface



Exemple d'application : dépliement et plaquage



Exemple d'application : plaquage de texture

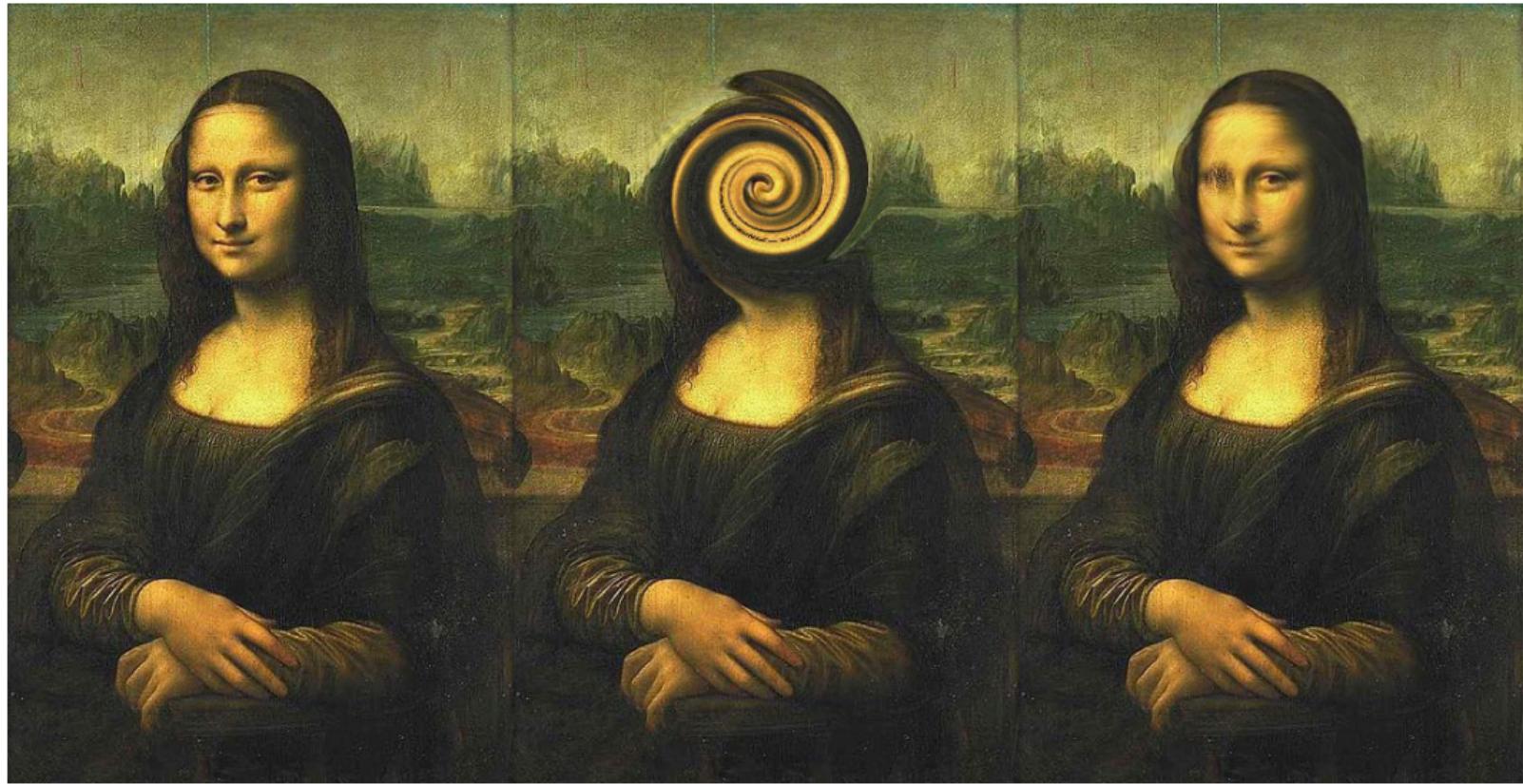


Exemple d'application : les experts ?



(c)

Exemple d'application : les experts ?



Quizz !



► Comment obtenir l'image du bas à partir des trois images du haut ?



- A** trois cisaillements
 - B** deux cisaillements + une rotation
 - C** deux cisaillements + une rotation + un zoom
 - D** trois cisaillements + un zoom
- (on ne compte pas les éventuelles translations)



► Comment obtenir l'image du bas à partir des trois images du haut ?



- A trois cisaillements
- B deux cisaillements + une rotation
- C deux cisaillements + une rotation + un zoom
- D trois cisaillements + un zoom
(on ne compte pas les éventuelles translations)

Cisaillement vertical



Cisaillement vertical



Rotation + zoom horizontal



Cisaillement vertical



Cisaillement vertical



Rotation + zoom horizontal



Cisaillement vertical



Cisaillement vertical



Rotation + zoom horizontal



Cisaillement vertical



Cisaillement vertical



Rotation + zoom horizontal



Cisaillement vertical



Cisaillement vertical



Rotation + zoom horizontal



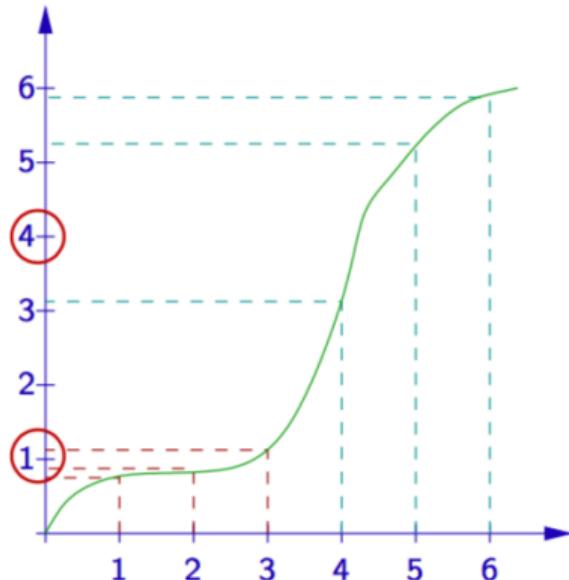


Formalisons 30"

Transformation h

En général h est continue ($h : Z^2 \rightarrow R^2$)

Problème de discréétisation : les coordonnées des pixels sont entières dans l'image

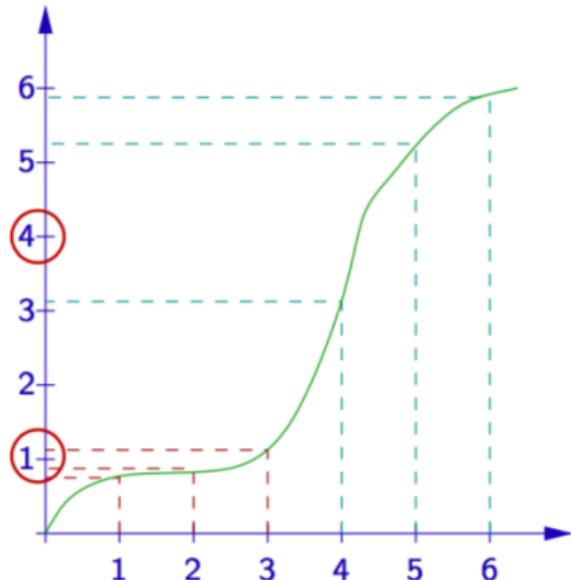


- effet de contraction des valeurs $\{1, 2, 3\} \rightarrow 1$
- trous en 2, 4 → **pixels noirs** dans I'

Quelle couleur donner aux pixels de I' ?

Transformation continue

De manière générale les transformations h sont continues ($h : Z^2 \rightarrow R^2$) mais les coordonnées des pixels sont entières dans l'image → *problème de discréétisation*



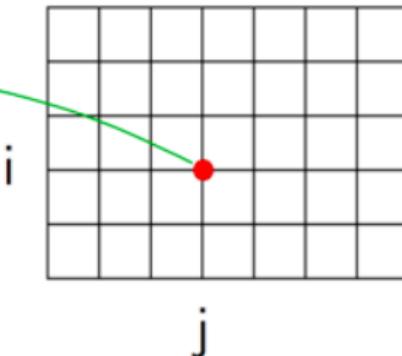
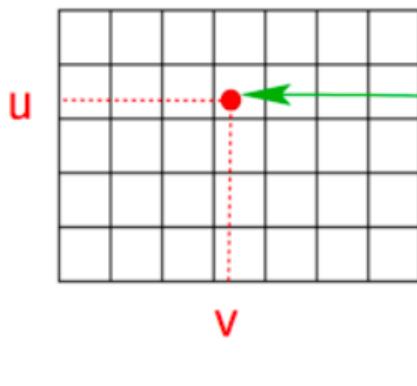
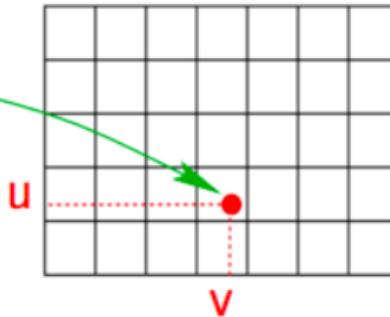
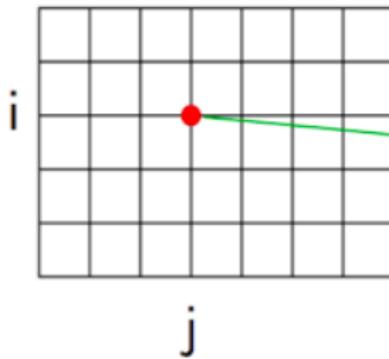
Transformation directe $I' = h(I)$

- effet de contraction des valeurs $\{1, 2, 3\} \rightarrow 1$
- trous en 2, 4 → **pixels noirs** dans I'
- phénomène **d'aliasing**

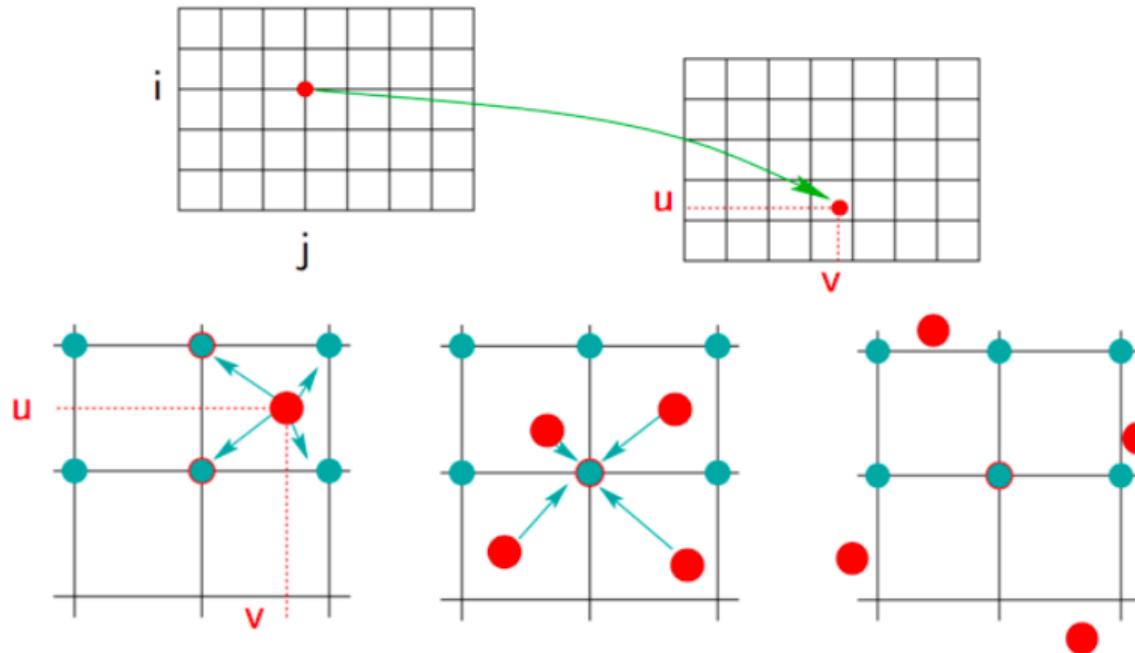
Quelle couleur donner aux pixels de I' ?

Une solution : considérer la transformation inverse :
 $I = h^{-1}(I')$ et interpoler les valeurs

Directe $I' = h(I)$? ou inverse $I = h^{-1}(I')$?

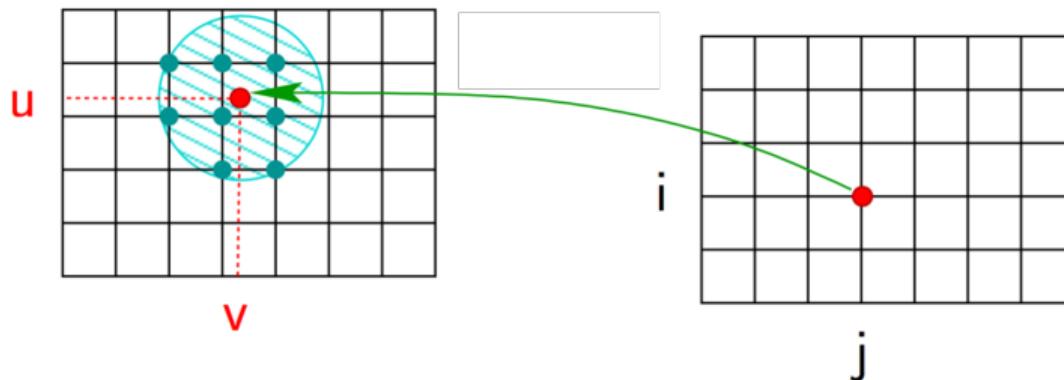


Transformation directe : $I' = h(I)$, quel I' ?



problème de trous

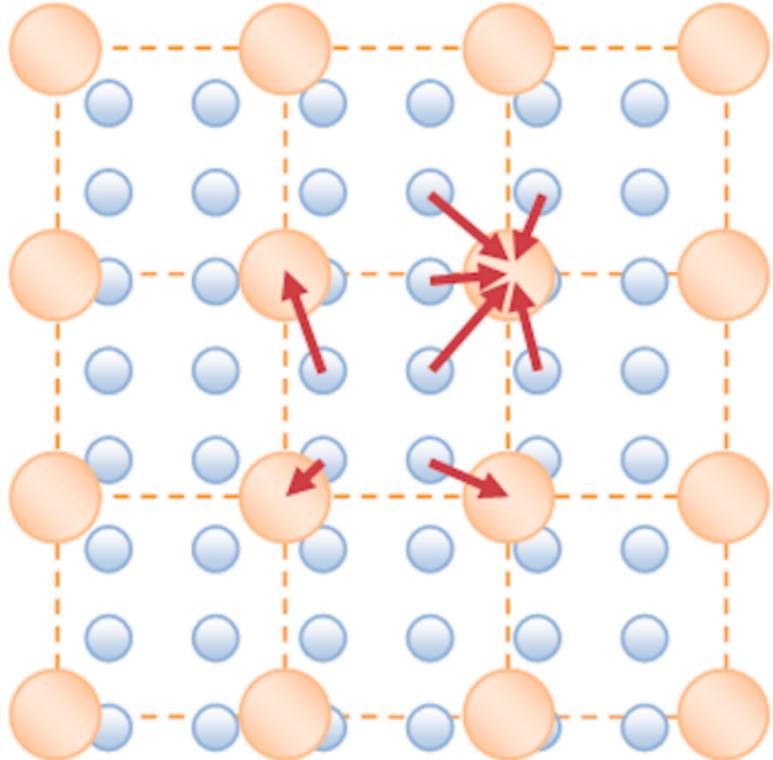
Transformation inverse : $I = h^{-1}(I')$, quel I' ?



- for int (x' , y') in I' :
- float (u , v) = $h^{-1}(x', y')$, (u, v) in I
- $I'(x', y') = \text{interpolation}(I, u, v)$

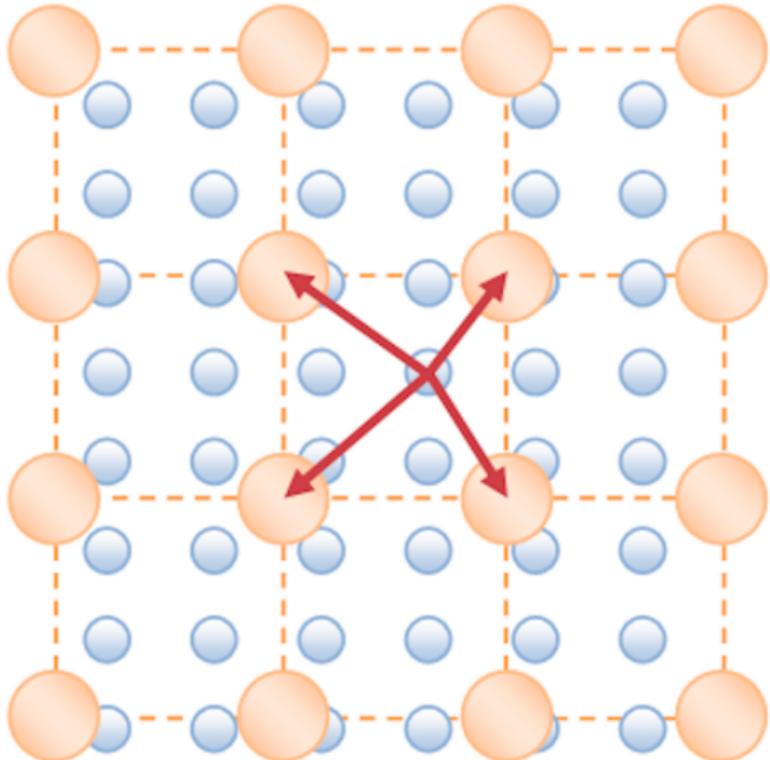
L'interpolation élimine l'effet d'aliasing et tous les pixels de I' sont traités (pas de pixels noirs)

Interpolation au plus proche



$I'(x', y') \leftarrow$ la valeur du pixel le plus proche dans I

Interpolation bi-linéaire



$I'(x', y') \leftarrow$ la contribution des 4 pixels les plus proches dans I
L'interpolation est une somme pondérée par les distances

Sans/avec interpolation



OK?

```
 wget http://www.labri.fr/perso/  
ta/tmp-files/eirb/domain.tar.gz
```