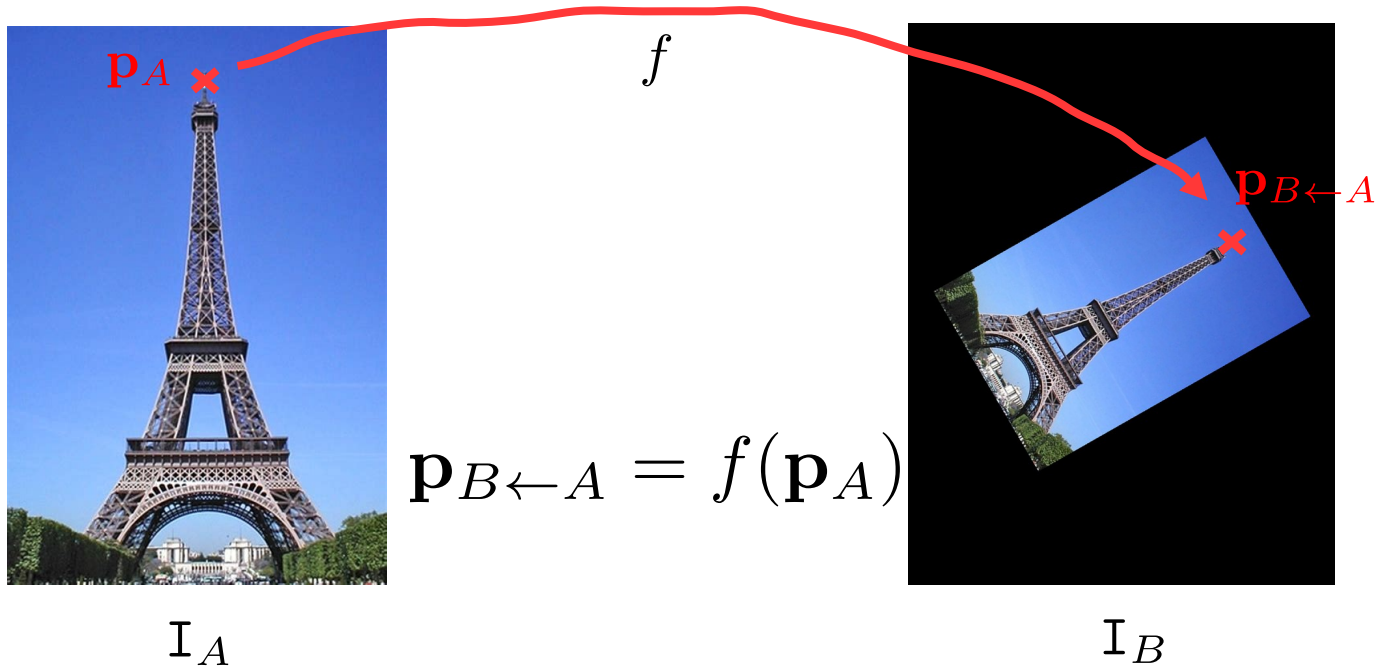


Appliquer une transformation à une image

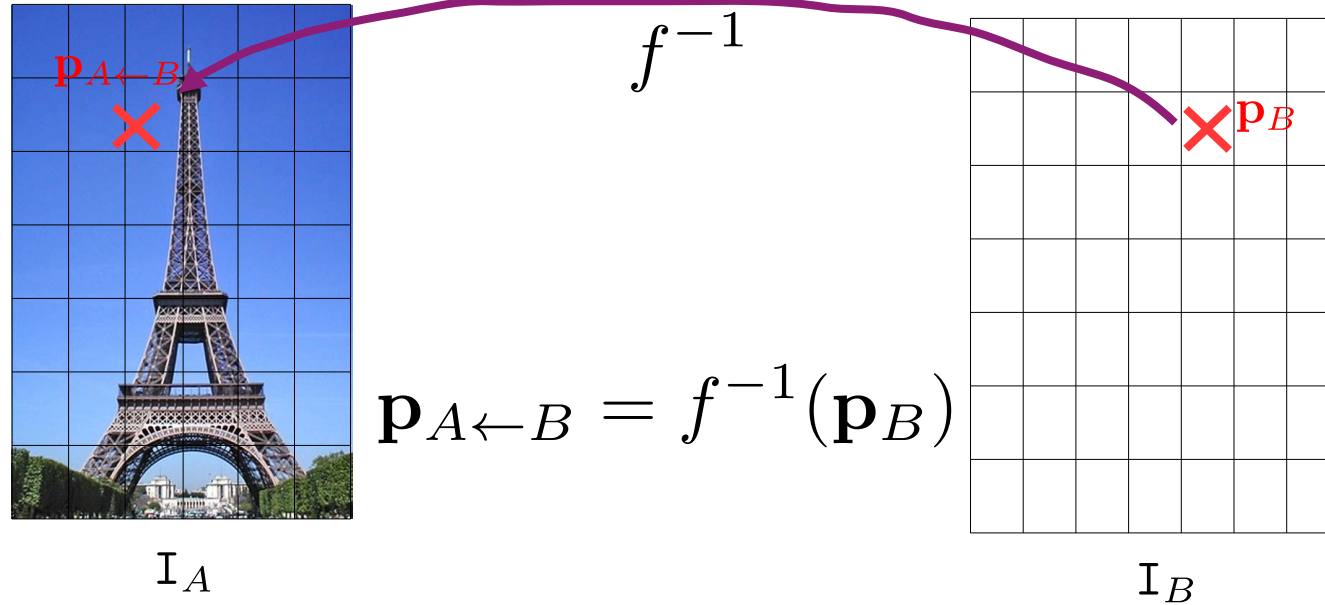
Guillaume Bourmaud

Transformation



Objectif : appliquer une transformation f à I_A pour obtenir une nouvelle image I_B

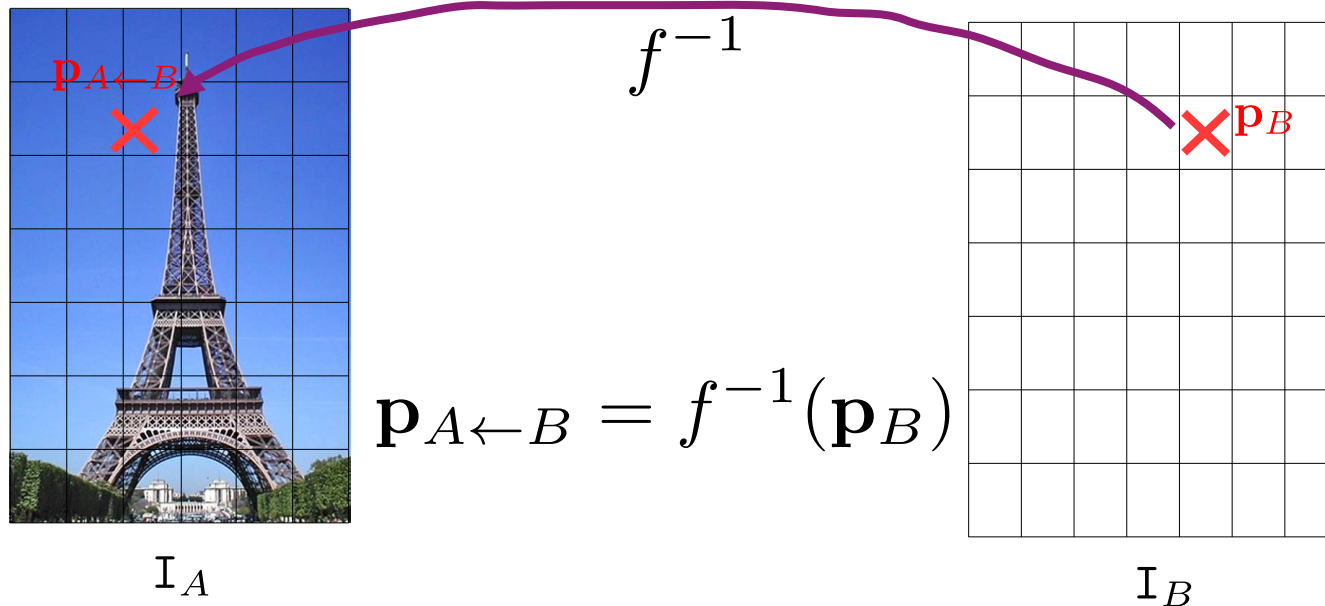
Principe



Pour chaque pixel de I_B , aller chercher la couleur dans I_A :

$$I_B(\mathbf{p}_B) = I_A(f^{-1}(\mathbf{p}_B))$$

Interpolation



Les pixels de \mathbf{I}_B ont des coordonnées \mathbf{p}_B entières,
 mais les coordonnées $\mathbf{p}_{A \leftarrow B} = f^{-1}(\mathbf{p}_B)$ sont **non-entières**.

→ **Nécessité d'interpoler**

~~$$\mathbf{I}_B(\mathbf{p}_B) = \mathbf{I}_A(f^{-1}(\mathbf{p}_B))$$~~

→ $\mathbf{I}_B(\mathbf{p}_B) = \text{interp}(\mathbf{I}_A, f^{-1}(\mathbf{p}_B))$ ₅

Implémentation en 3 étapes

1) Définir la taille $h_B \times w_B$ de I_B

2) Pour chaque pixel de I_B , calculer et stocker $\mathbf{p}_{A \leftarrow B} = f^{-1}(\mathbf{p}_B)$

Le résultat peut se stocker sous la forme d'un tableau de taille $h_B \times w_B \times 2$.
Par habitude, on stocke en général chaque canal indépendamment. On obtient alors deux tableaux XI et YI , appelés **grilles d'interpolation**, chacun de taille $h_B \times w_B$.

3) Utiliser d'une fonction d'interpolation $\mathbf{I}_B = \text{interp}(\mathbf{I}_A, XI, YI)$
(ex : fonction remap dans OpenCV)