

Informatique Graphique Warping et Morphing

Jean-Baptiste.Yunes@univ-paris-diderot.fr

9/2018

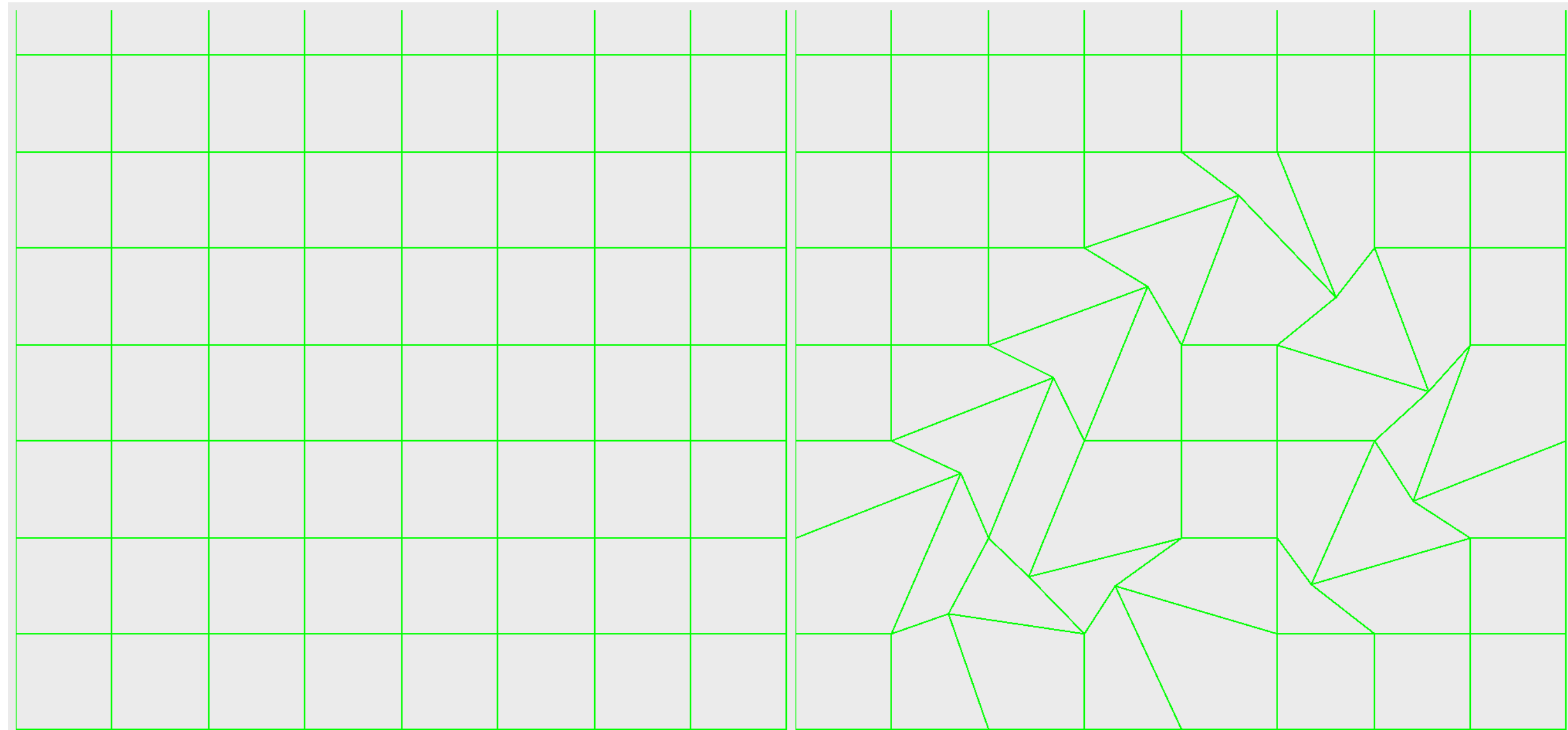
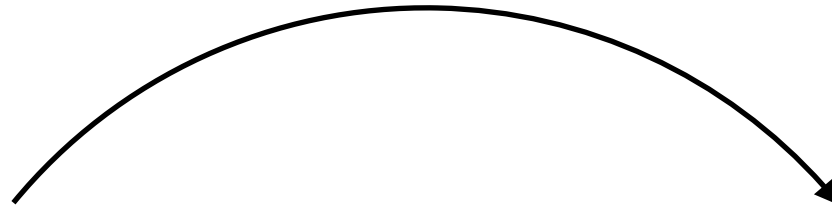
Gauchissement d'image (Image Warping)

Procédé de déformation d'image utilisé pour la correction d'image ou la création d'effets

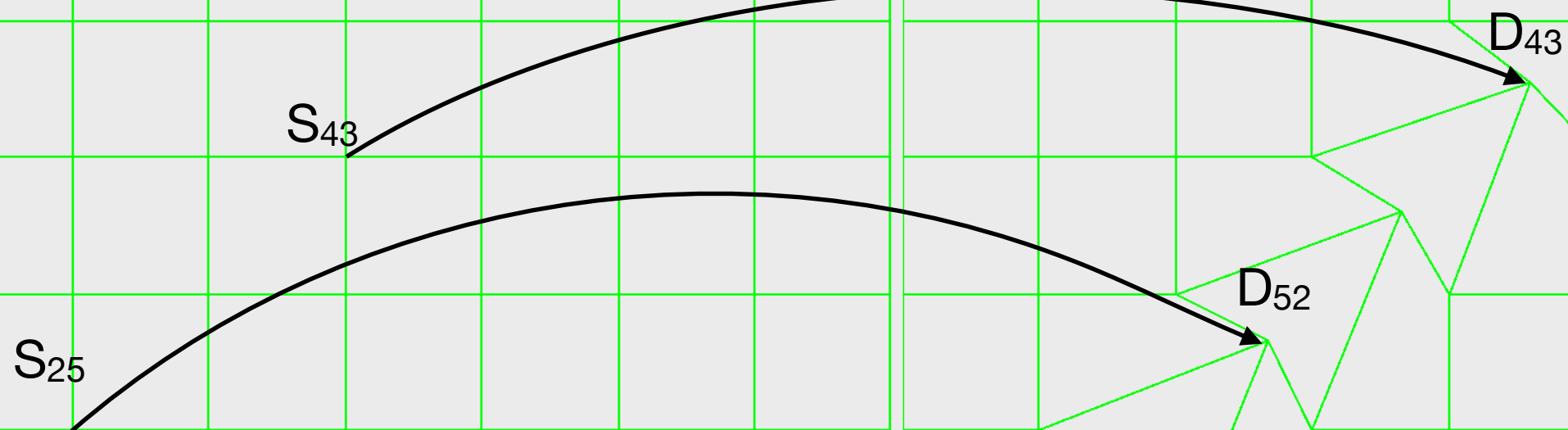
Morphing

Gauchissement animé utilisé pour transformer une forme en une autre (souvent des visages)

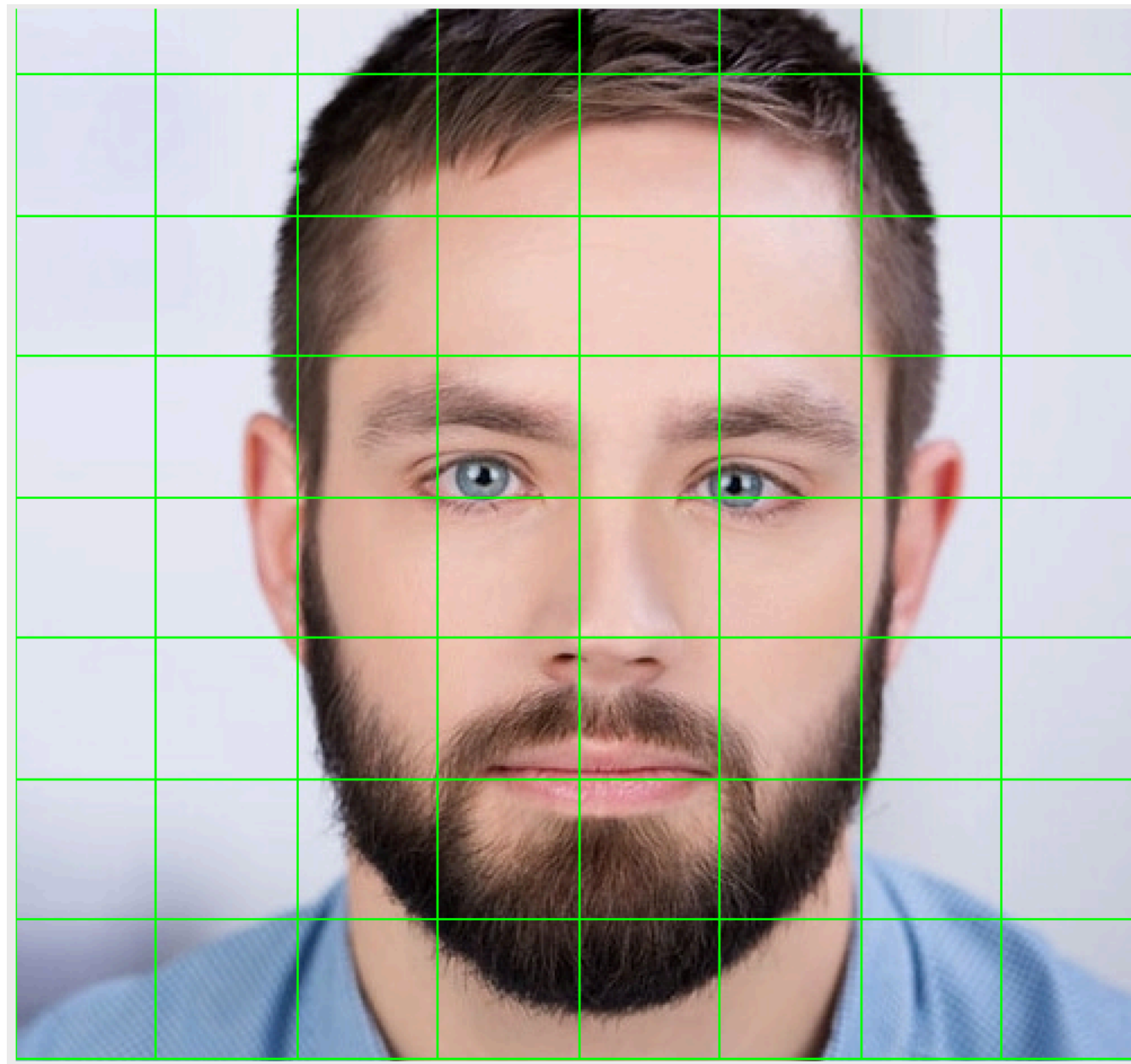
Gauchissement par déformation d'un filet (mesh)



Chaque point du filet source S_{ij} est translaté vers le point correspondant du filet destination D_{ij}



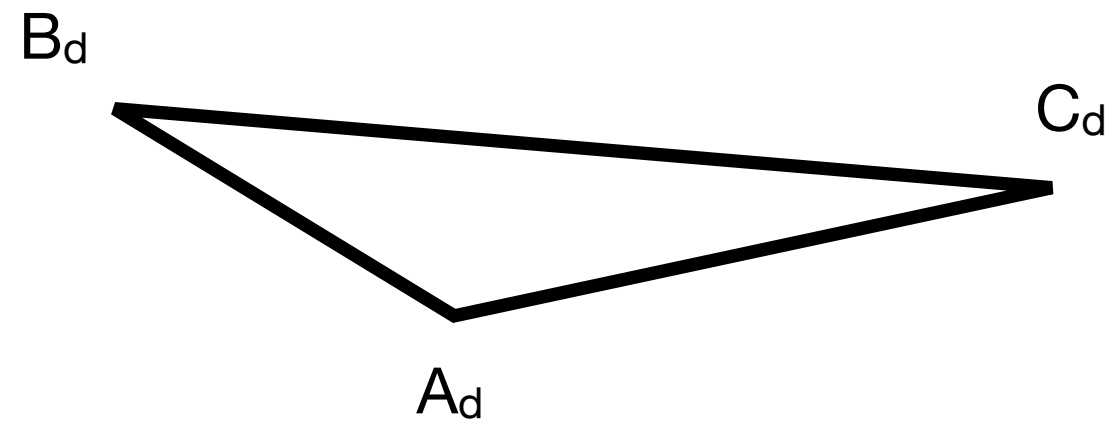
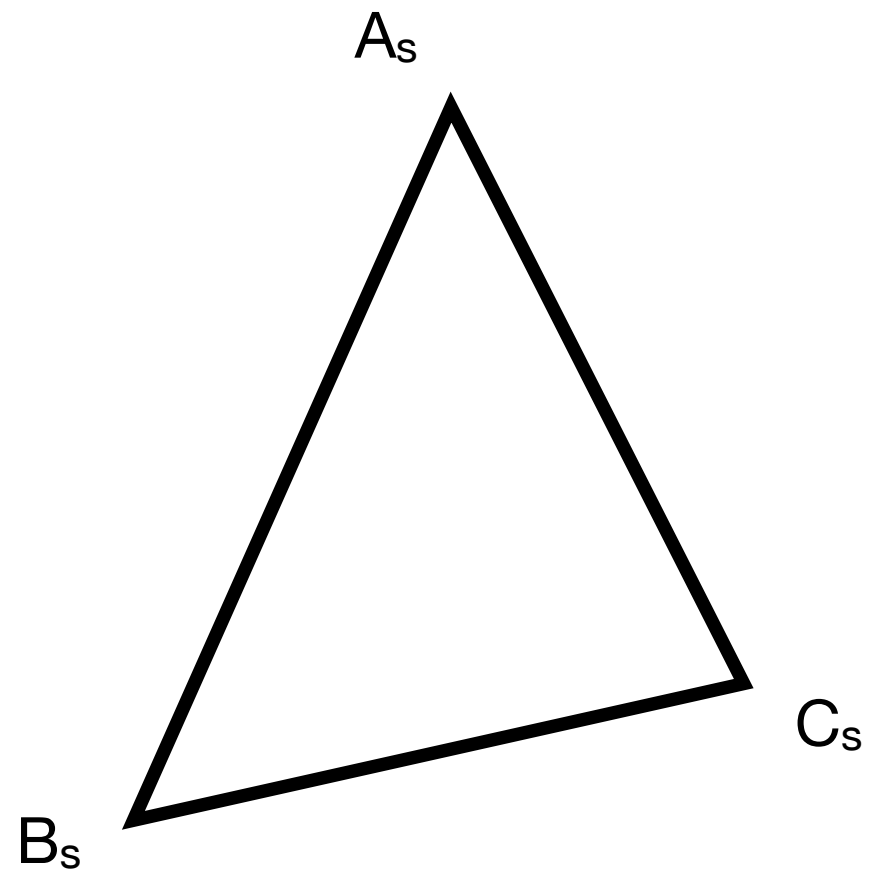
L'image pouvant être plus fine que le filet il faut donc interpoler entre les points du filet



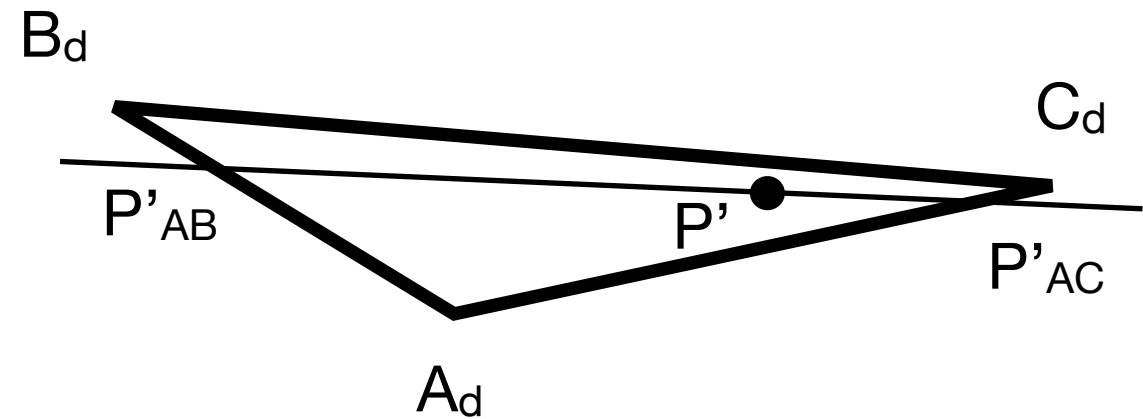
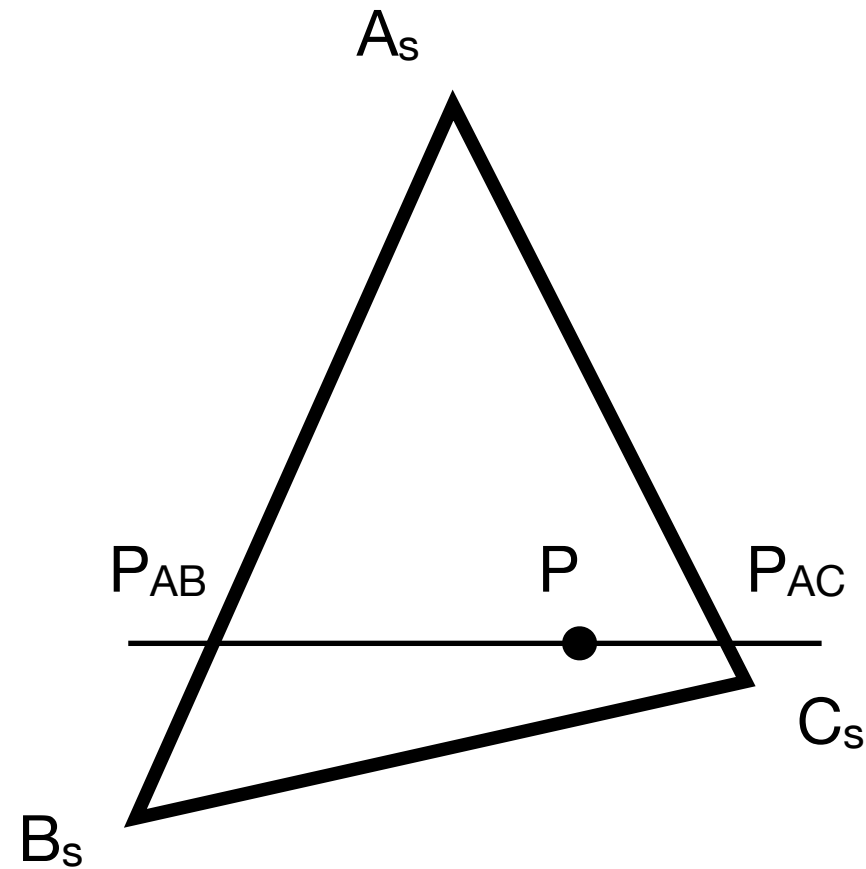
Afin de simplifier les calculs, on va trianguler le filet (c'est trivial)



Le problème se réduit donc à transformer un triangle en un autre source $\rightarrow$ destination



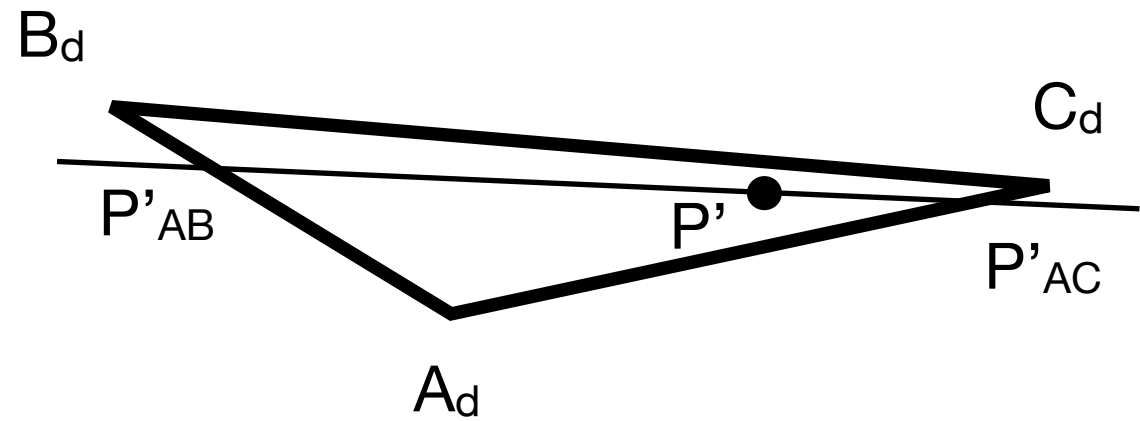
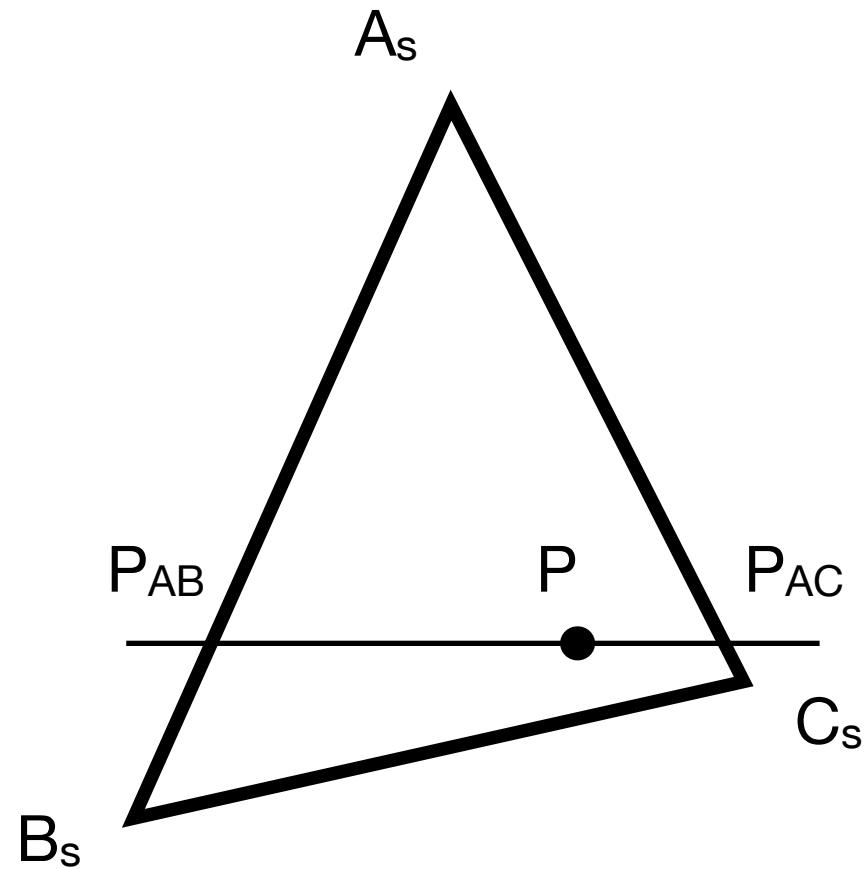
Le problème se réduit donc à transformer un triangle en un autre et chaque point à l'intérieur de l'un en un point équivalent à l'intérieur de l'autre



$$P_{AB}A_s/B_sA_s = P'_{AB}A_d/B_dA_d$$

$$P_{AC}A_s/C_sA_s = P'_{AC}A_d/C_dA_d$$

$$PP_{AB}/P_{AC}P_{AB} = P'P'_{AB}/P'_{AC}P'_{AB}$$



Pour éviter les «trous» on procède à rebours :

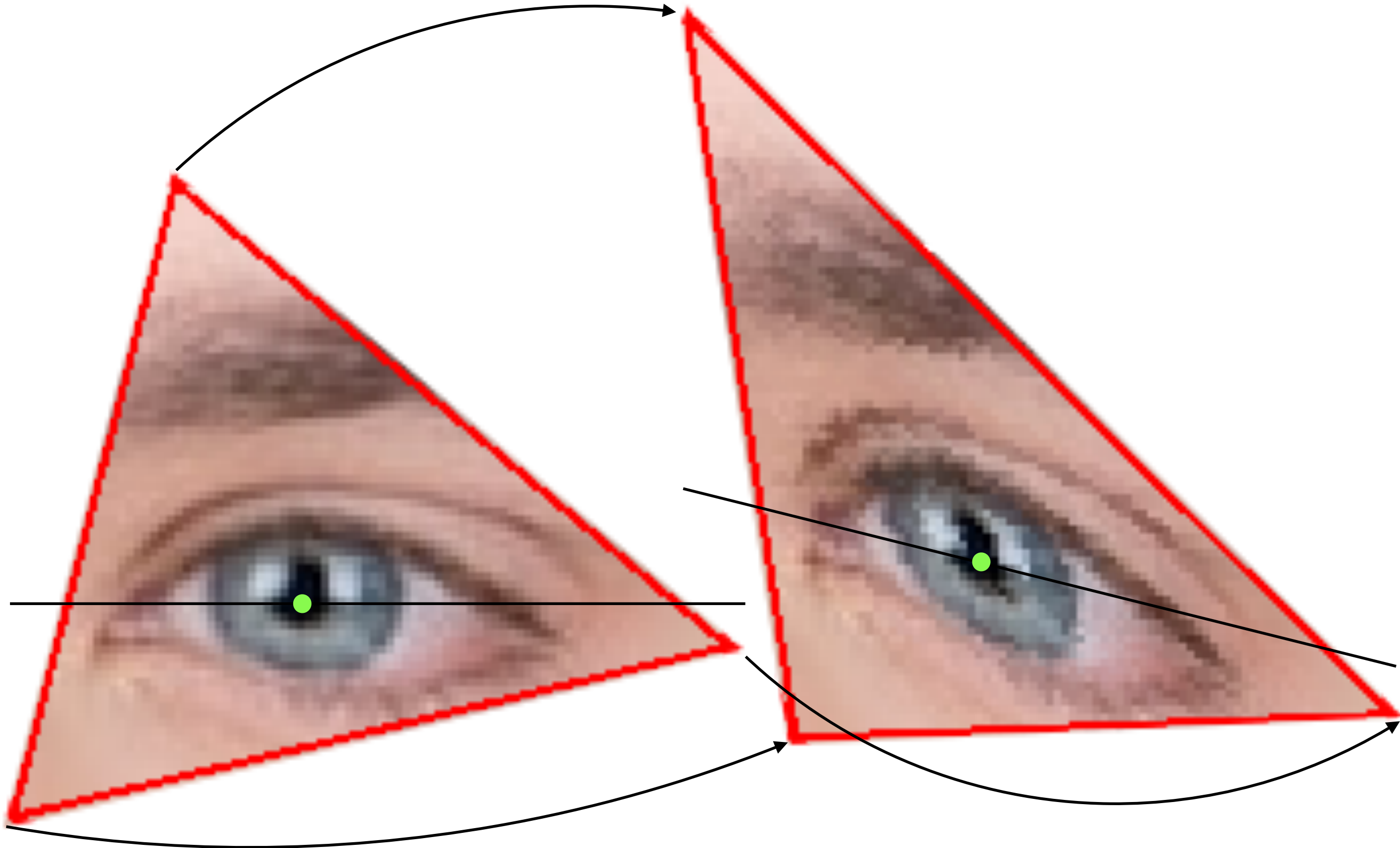
on passe par tous les pixels du triangle de destination et
on recherche la position dans le triangle source

attention ceci ne donne pas nécessairement la position
d'un pixel mais une position entre pixels

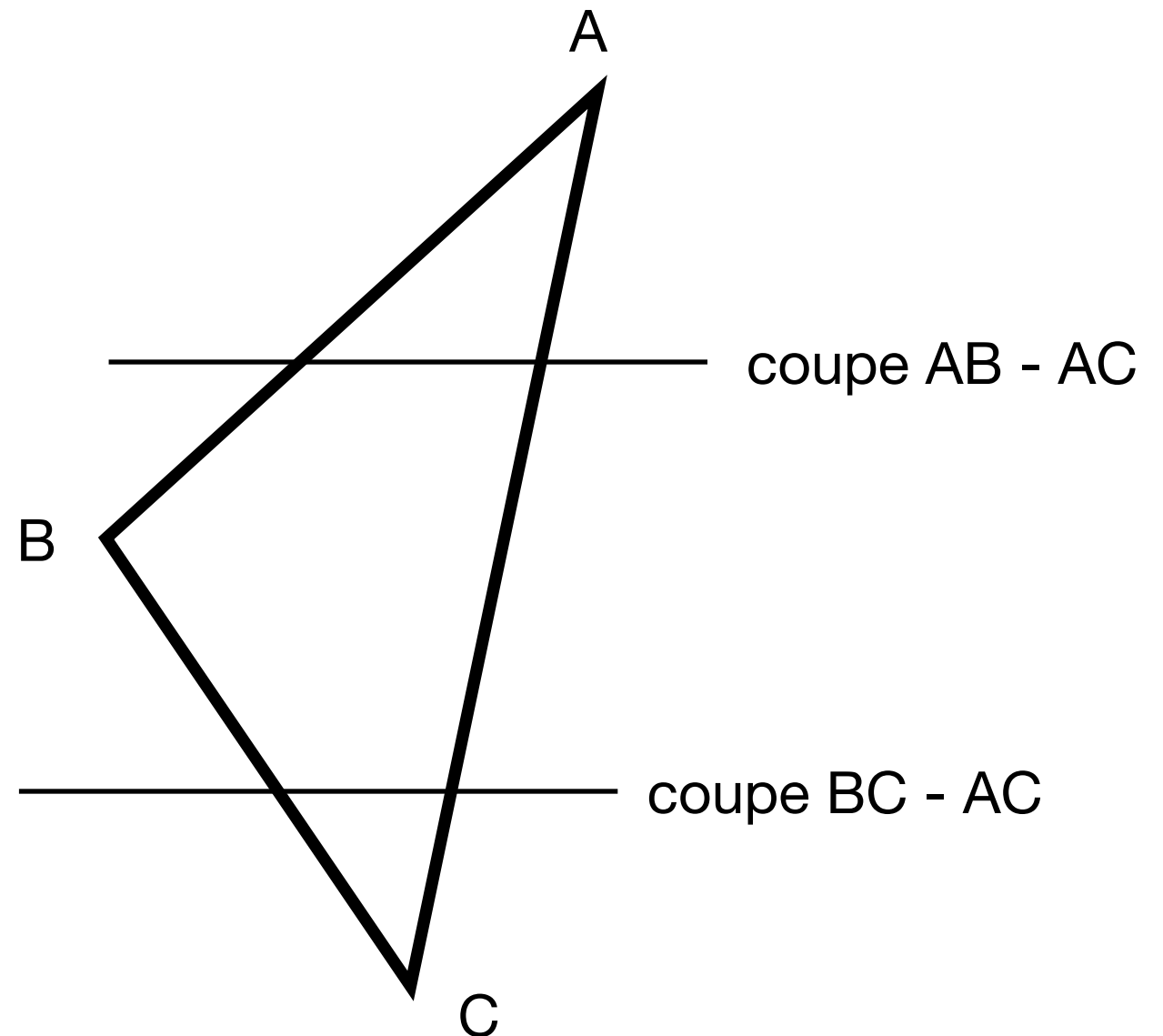
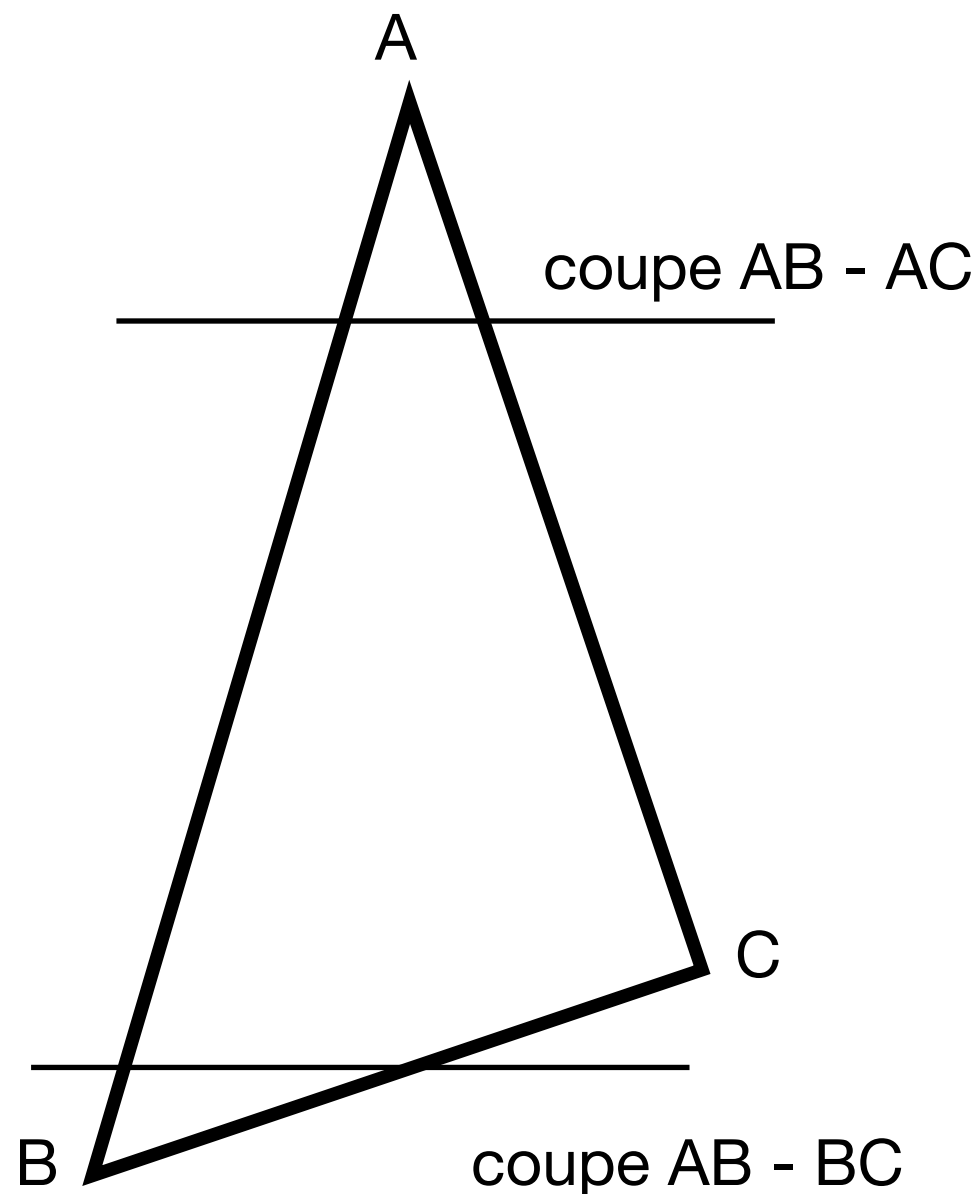
il faut donc une fonction permettant de construire une
valeur correspondant à la valeur d'une image idéale
sous-jacente

pixel le plus proche, approximation bilinéaire,
bicubique, Lanczos, etc.

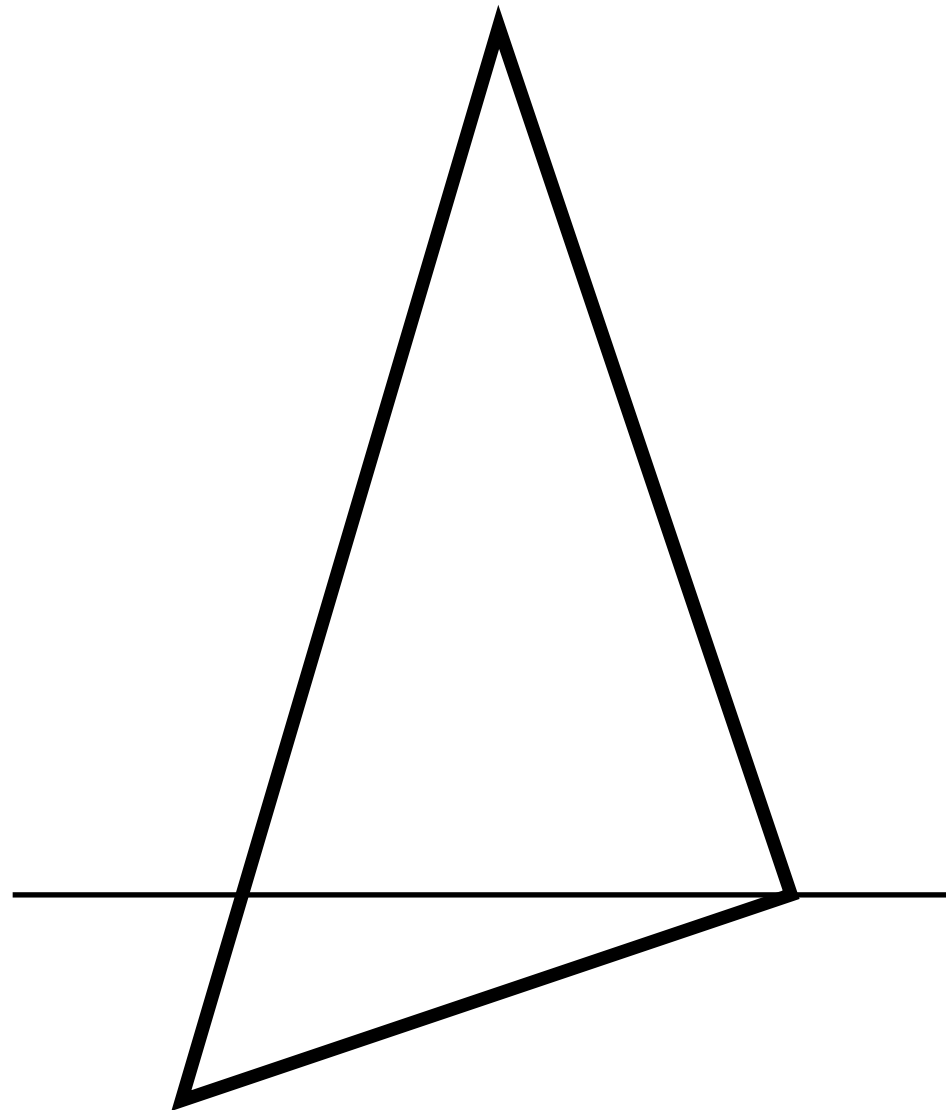
Pixel le plus proche...



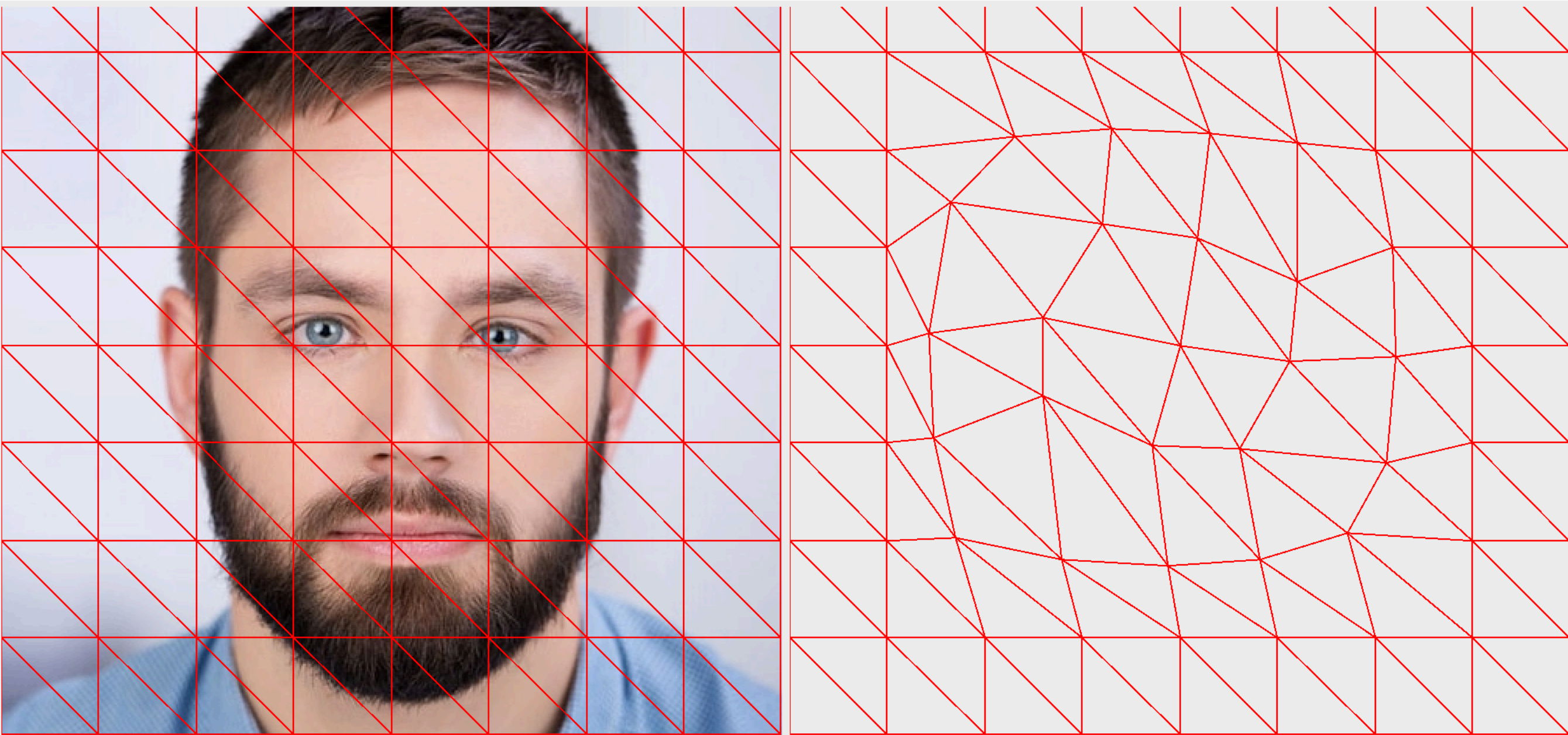
Attention, il y a quelques difficultés
le balayage de ligne nécessite de prendre en compte
différents cas
par exemple :



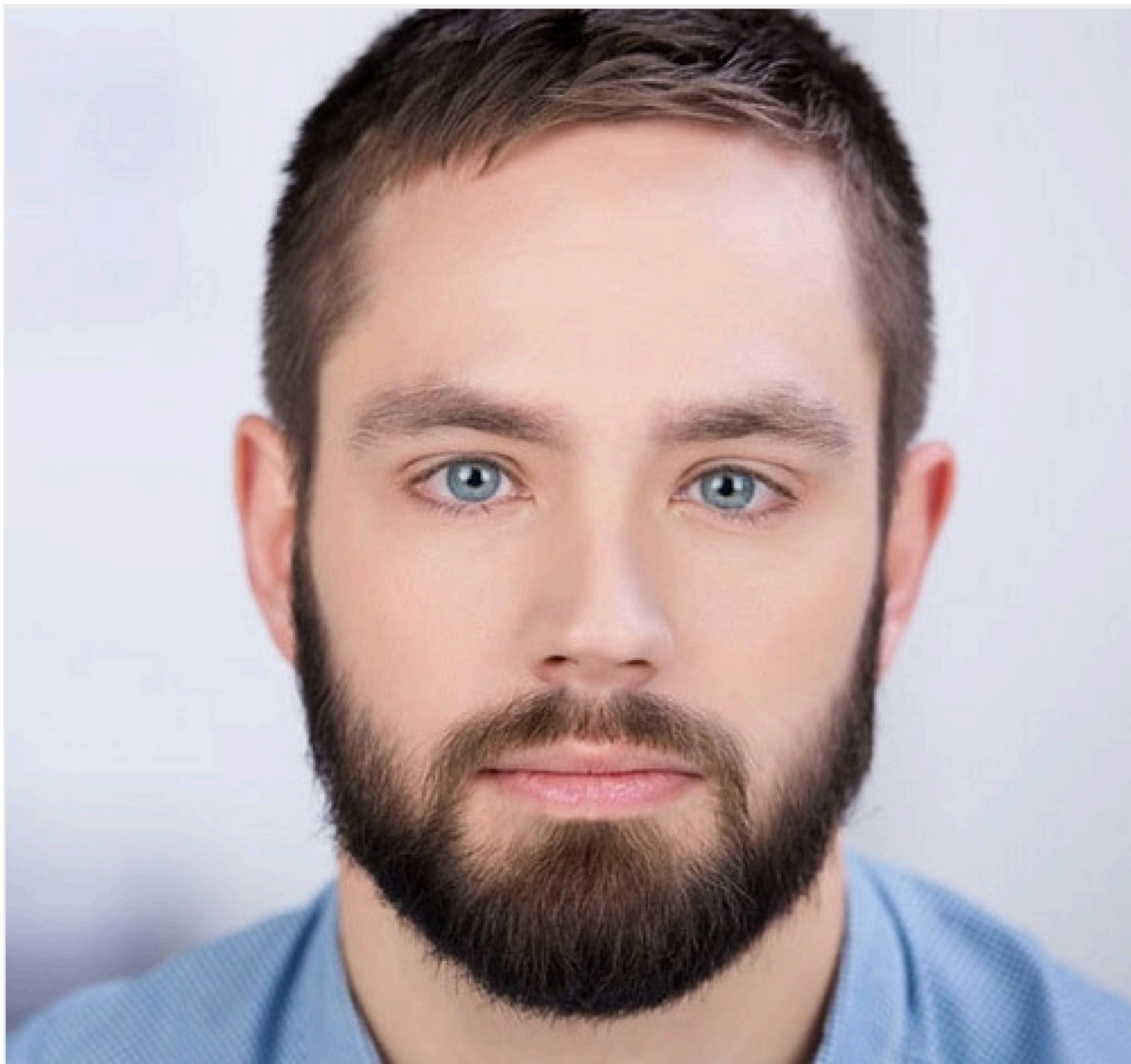
une technique possible consiste à découper le triangle en deux triangles joints par une horizontale



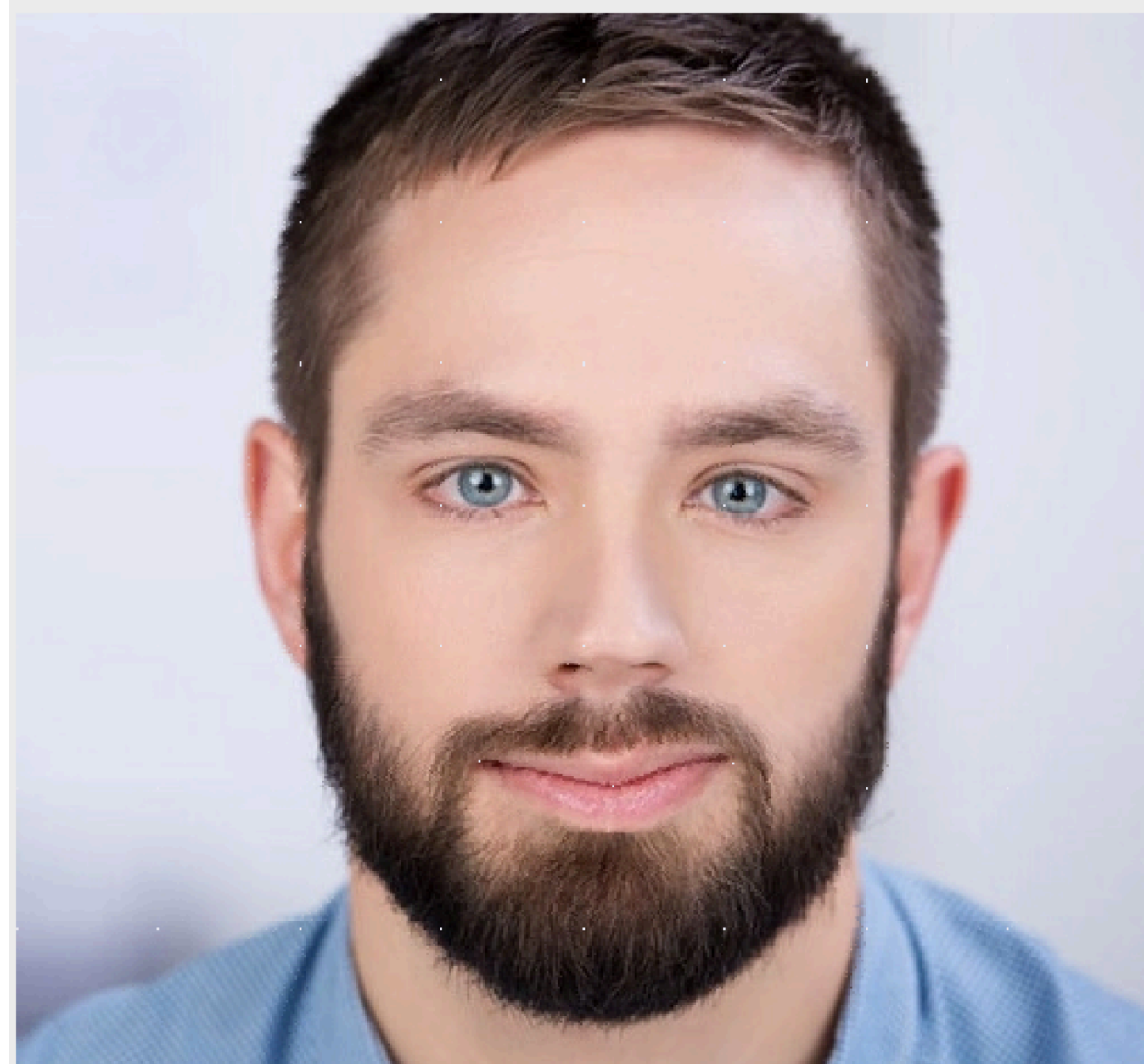
La transformation finale donne par exemple :



La transformation finale donne par exemple :



Une transformation finale donne par exemple :



Animation de la transition

on souhaite générer une séquence (paramétrée par le temps) d'images qui approxime le passage de l'image d'origine à l'image transformée

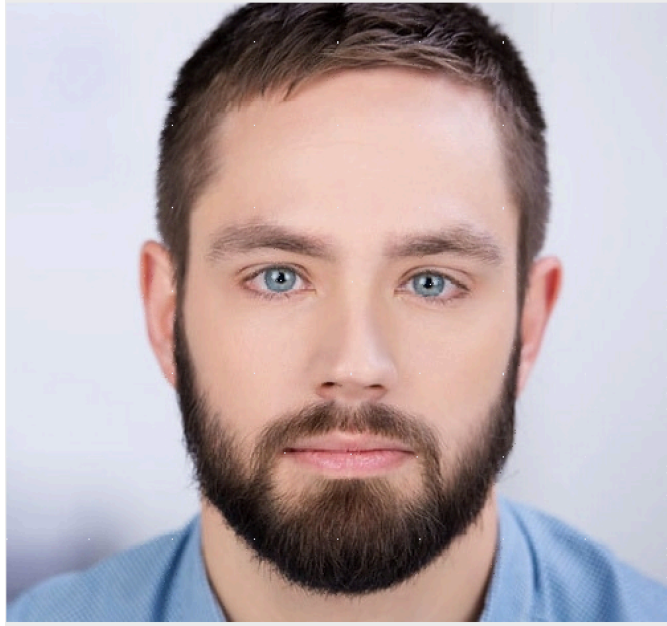
on souhaite $\text{filet}(0) = \text{source}$

on souhaite $\text{filet}(1) = \text{destination}$

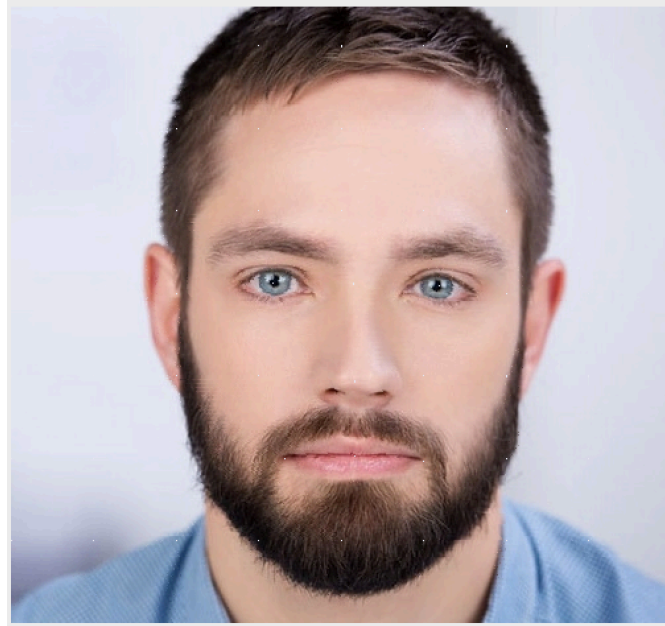
on va donc fabriquer $\text{filet}(t) = \text{source} * t + \text{destination} * (1 - t)$

on calcule des filets intermédiaires...

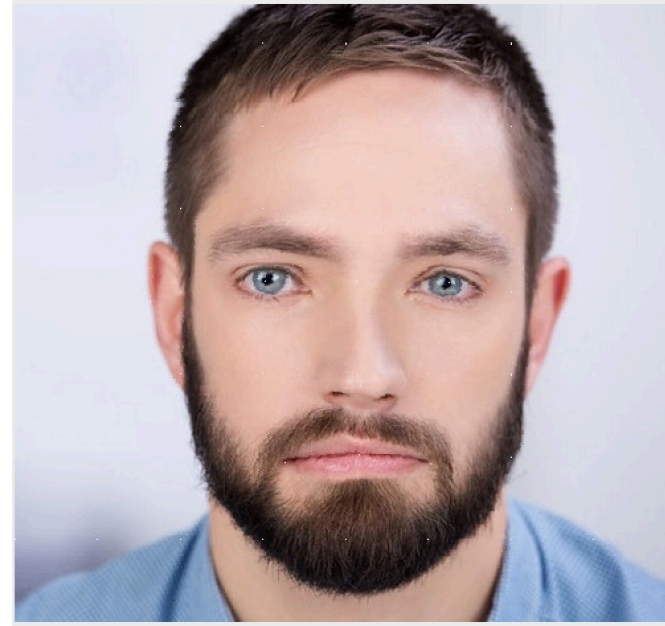
$t=0$



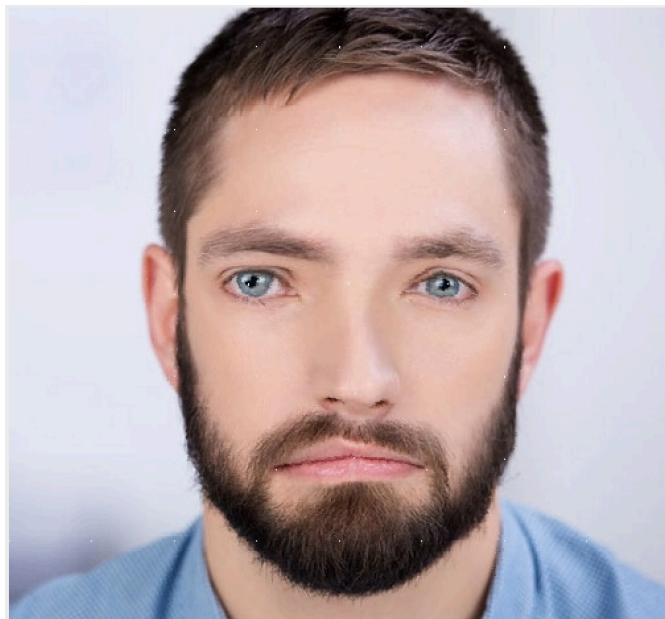
$t=0.2$



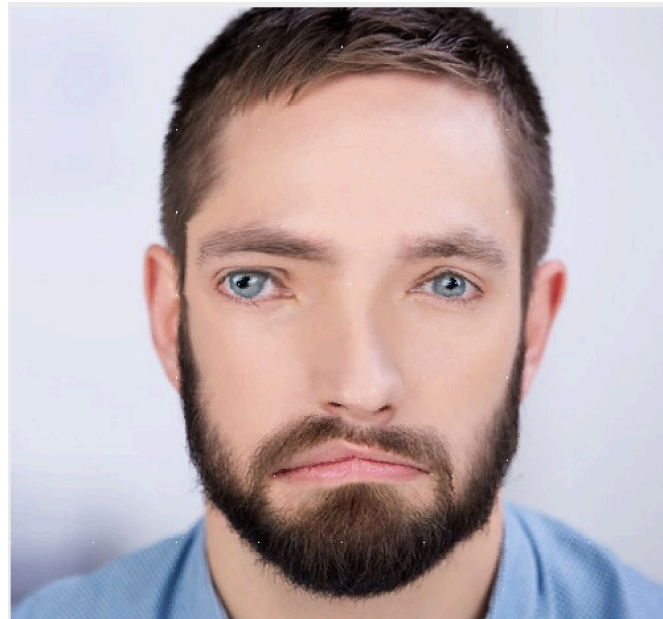
$t=0.4$



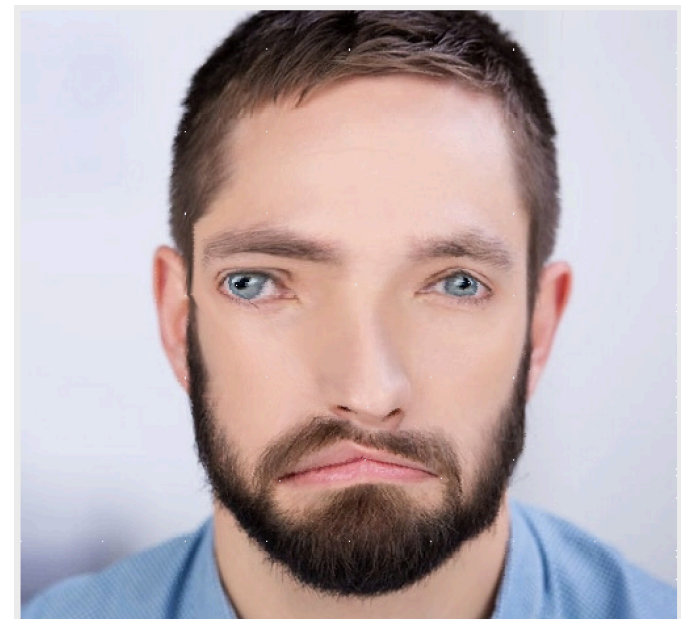
$t=0.6$



$t=0.8$



$t=1$



Le contrôle fin du résultat est assez difficile à obtenir

Il faut une grille relativement fine

Problème d'ergonomie

Gauchissement pour du morphing

On va simplement interpoler le filet

$$\text{filet}(t) = \text{filet}(0) * (1 - t) + \text{filet}(1) * t$$

Tout en mélangeant les images correspondantes

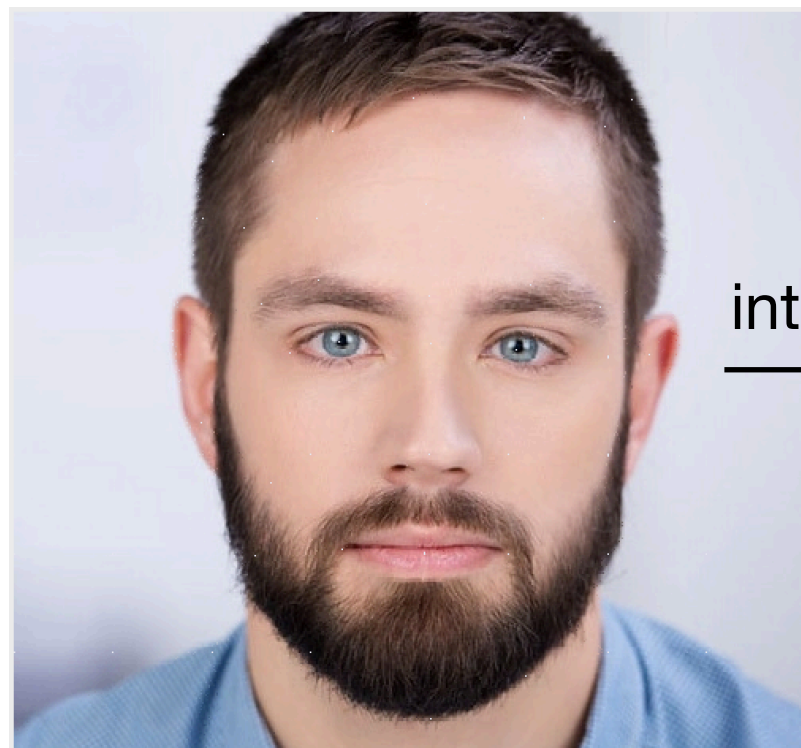
pour la source on gauchit avec $\text{filet}(0) \rightarrow \text{filet}(t)$

et pour la destination on gauchit avec $\text{filet}(1) \rightarrow \text{filet}(t)$

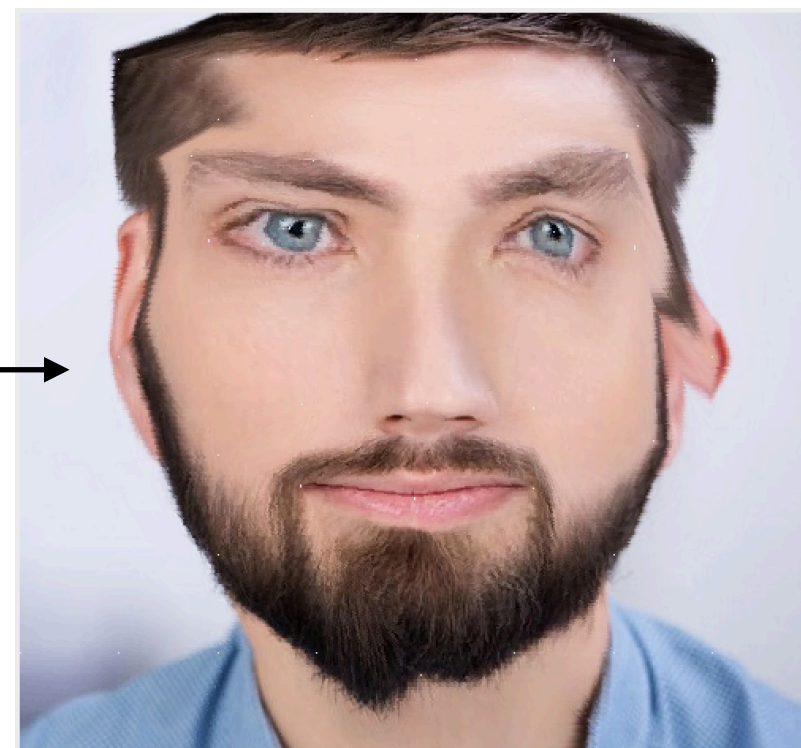
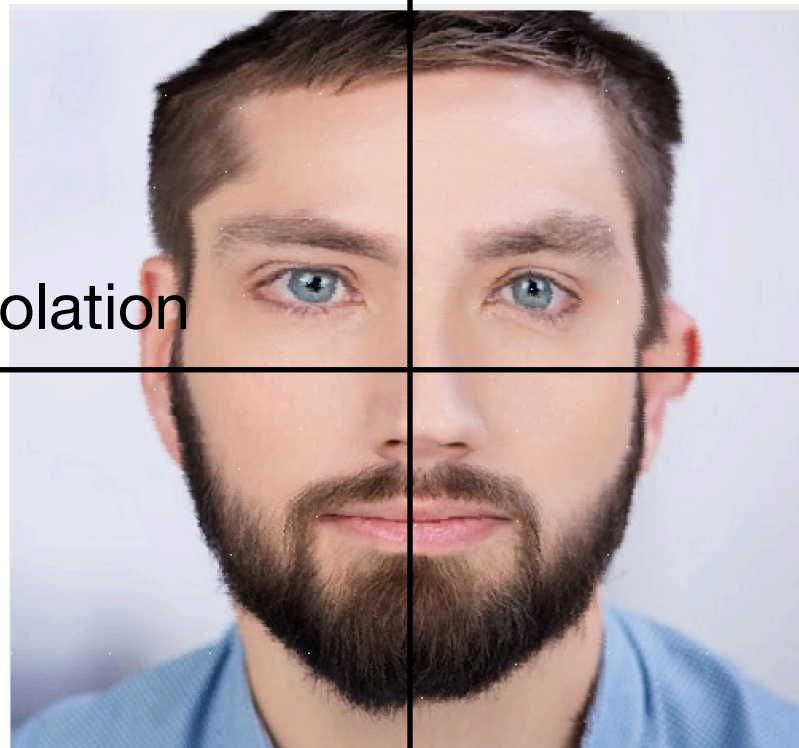
puis on mélange les résultats par un fondu (cross dissolve) de coefficient t

Tentative d'association de points caractéristiques d'une image à l'autre (œil pour œil, dent pour dent)





interpolation

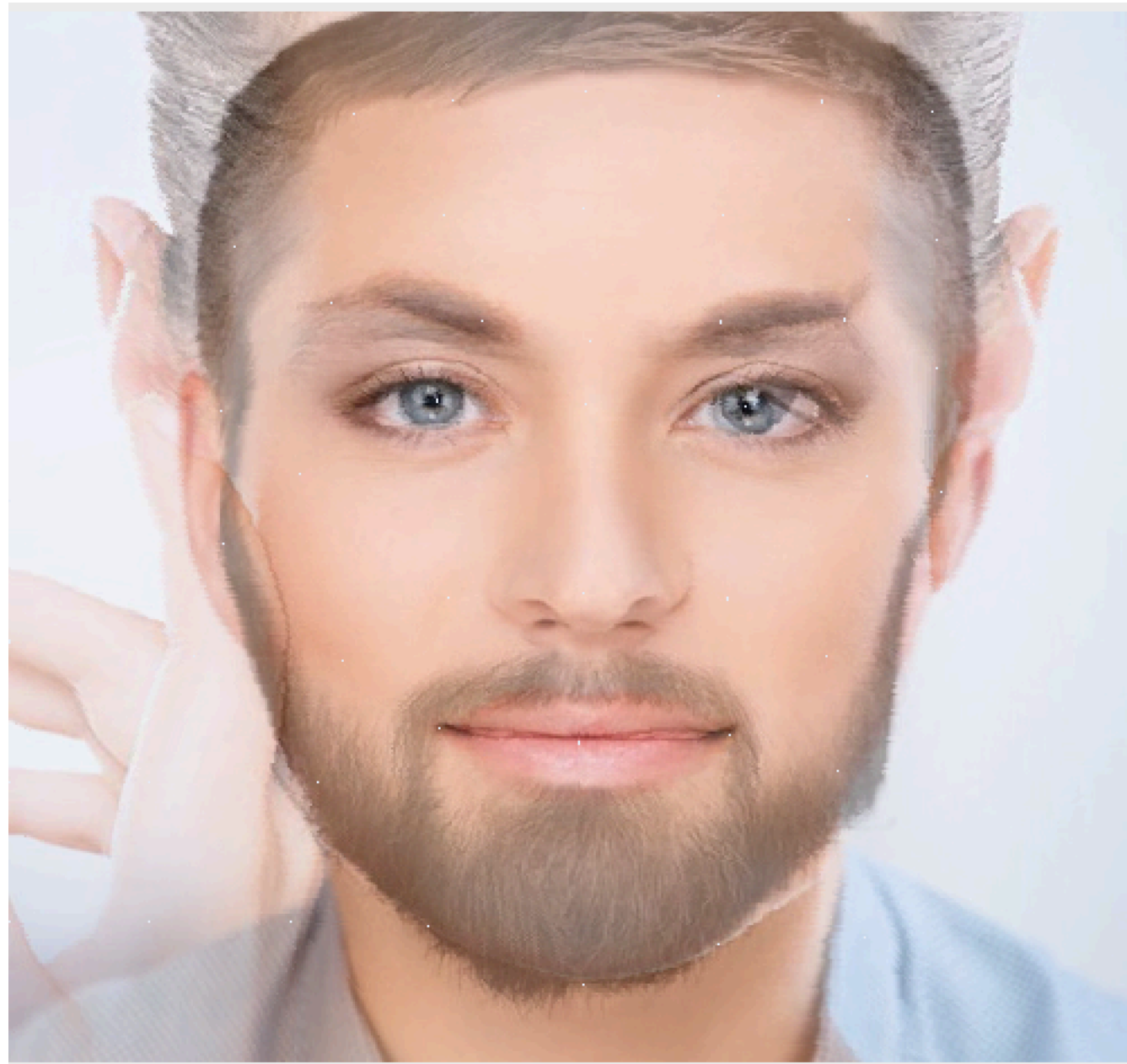


interpolation

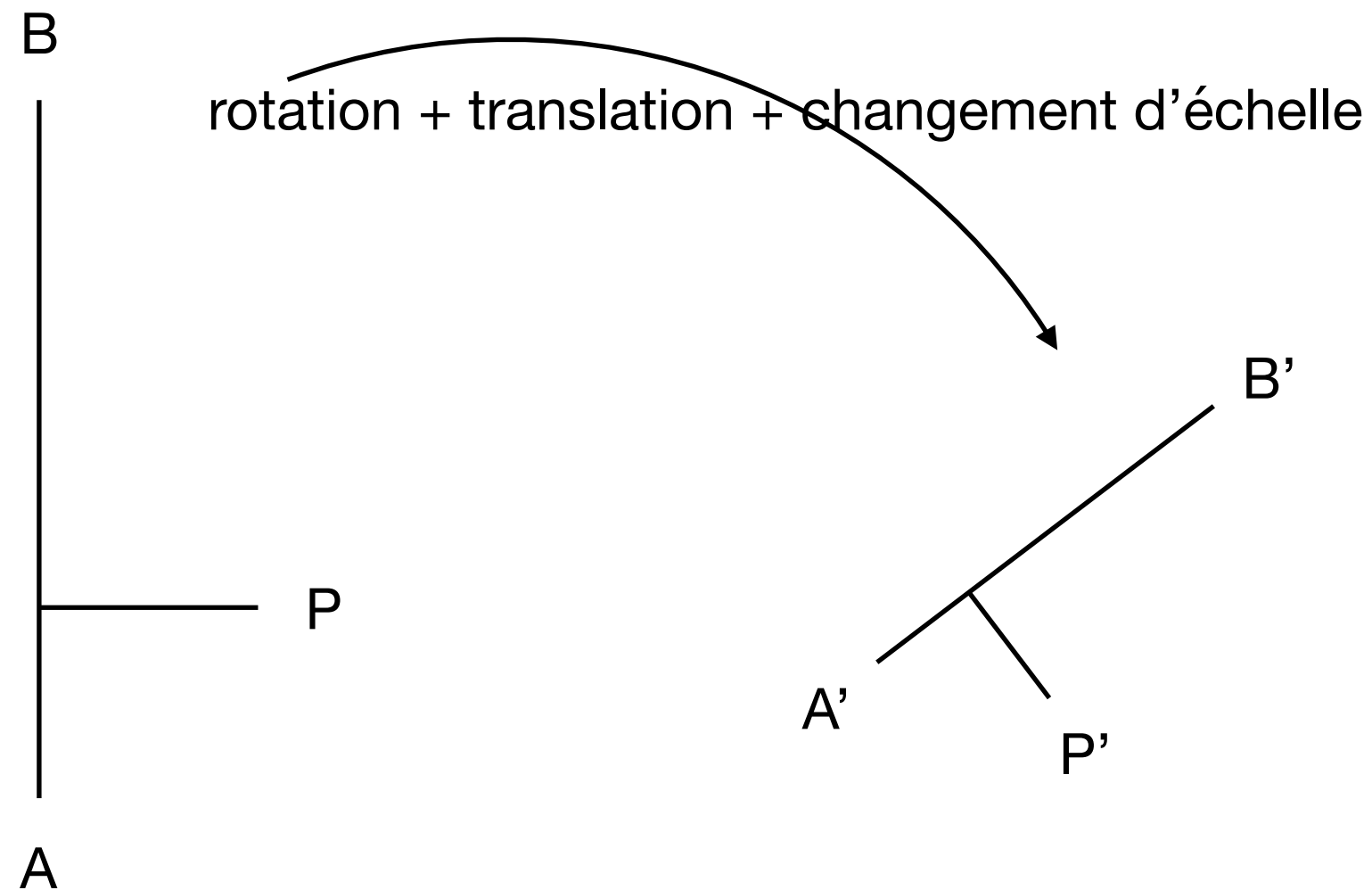


fondue

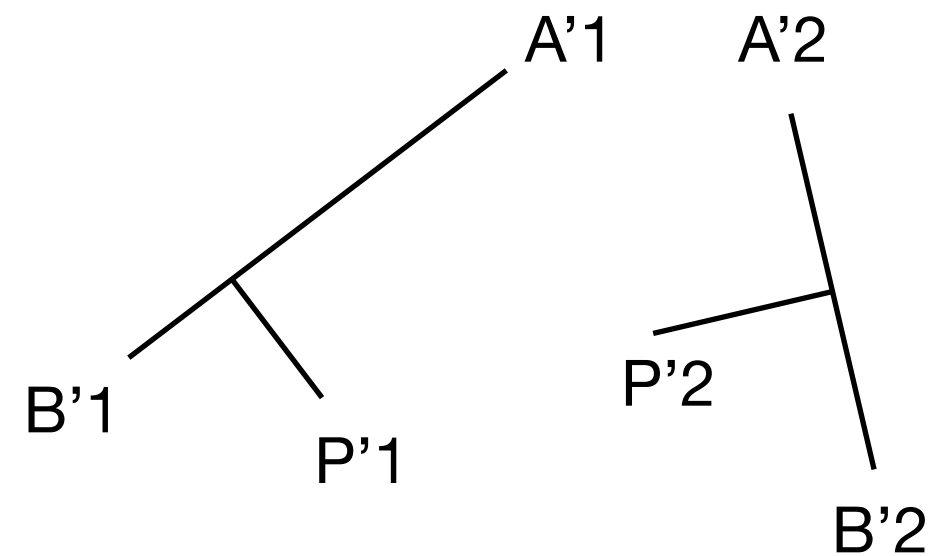
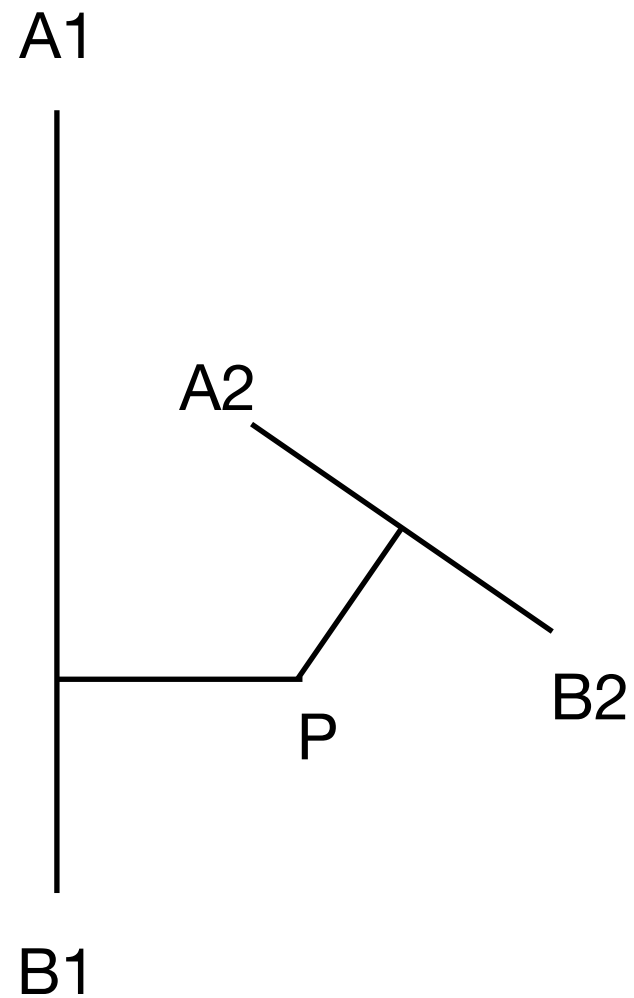
On obtient un morphing



- Des contrôles plus simples, l'algorithme de Beier-Neely
- idée de base un changement de base!



- Le problème est la recomposition de plusieurs de ces transformations
- Mélanger P'i ?



- La formule générique des poids pour le mélange est

$$w = \left(\frac{l^p}{a + d} \right)^b$$

- w : le poids du point
- l la longueur du segment
- d la distance au segment
- a, p, b des paramètres à ajuster selon les cas
 - valeurs possibles $p=0, b=2, a=0.001$

- Références
 - Feature-based image metamorphosis
Thaddeus Beier & Shawn Neely
1992, Computer Graphics, Vol. 26 (2)