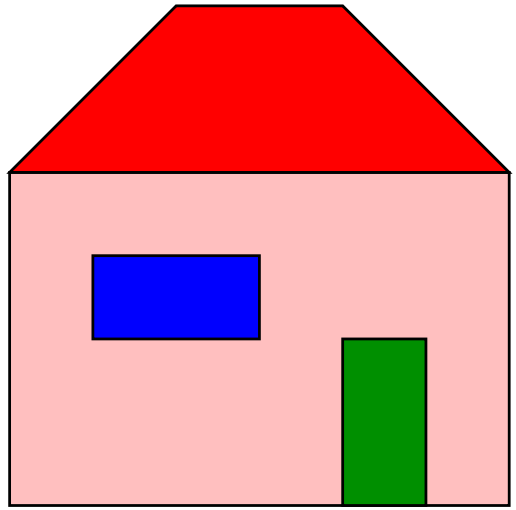
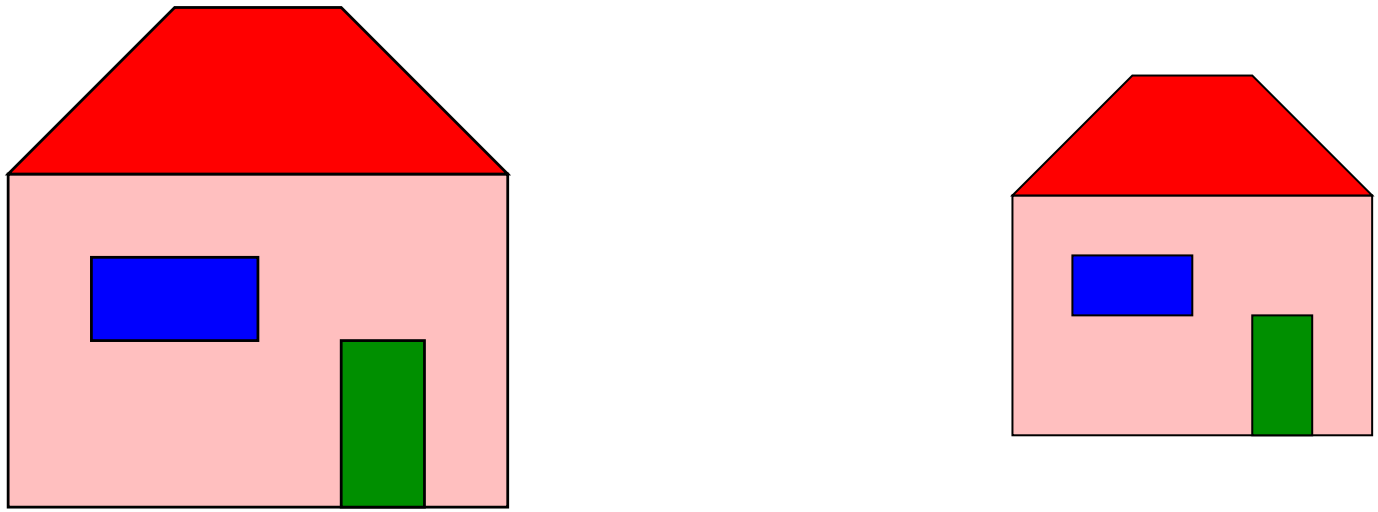
Plan

Les opérations de base

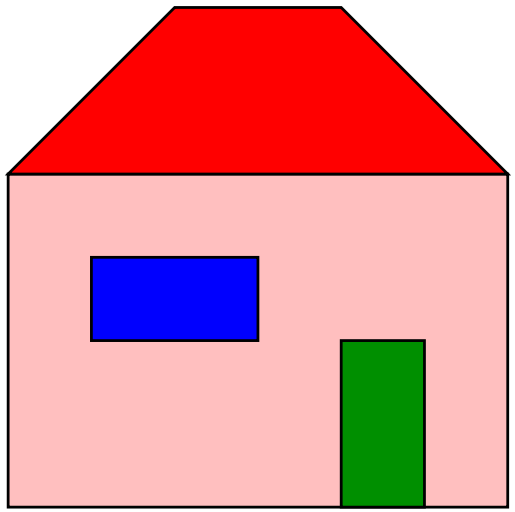


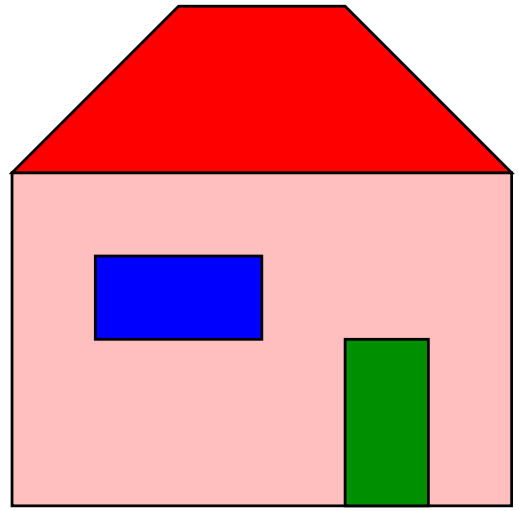
Les opérations de base



Homothetie

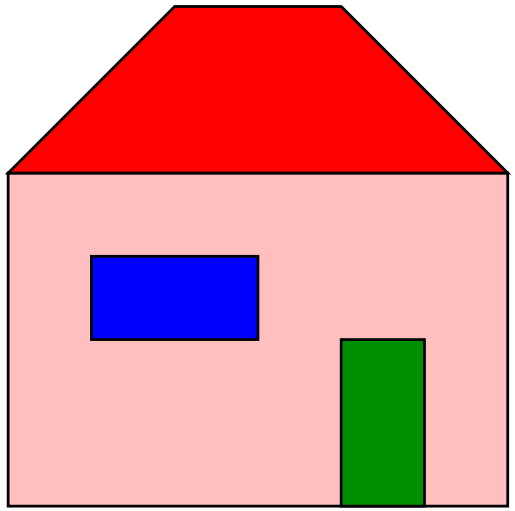
Les opérations de base

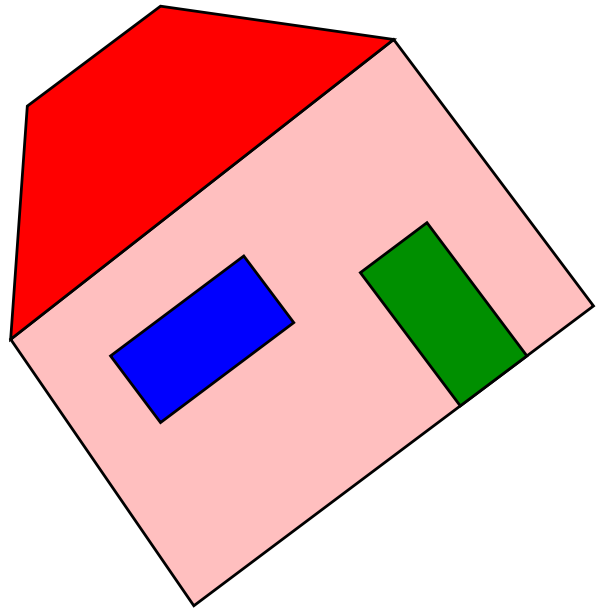




Translation

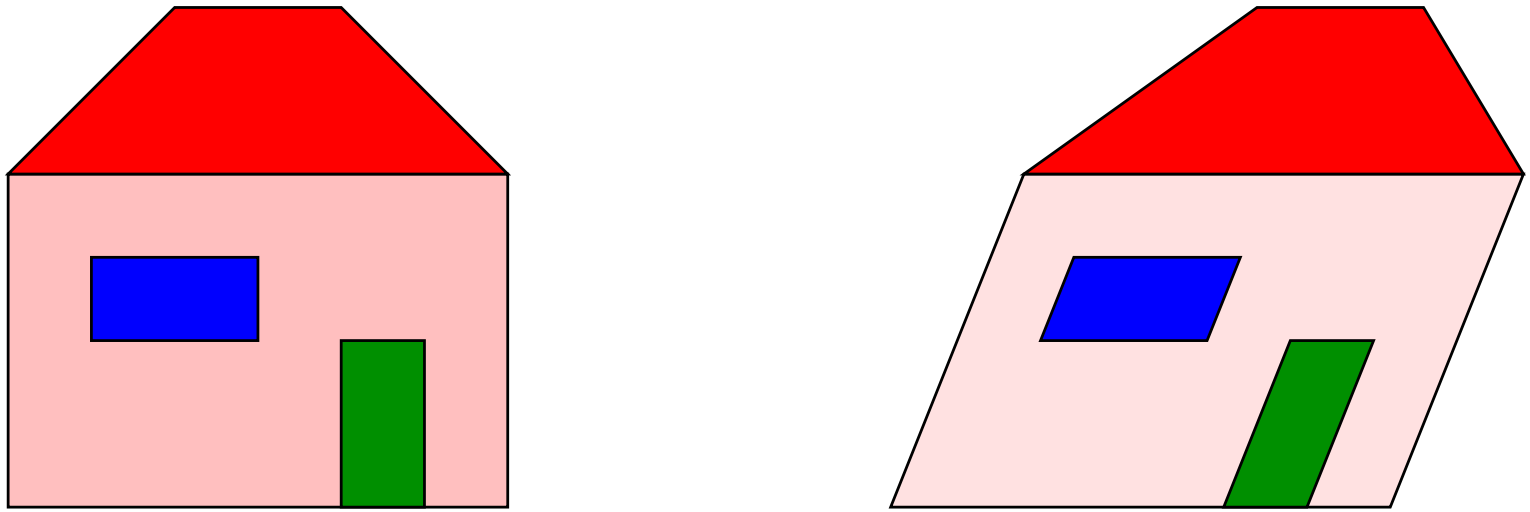
Les opérations de base





Rotation

Les opérations de base



Cisaillement

Cas d'une image vectorielle

Opération de rotation

Cas d'une image vectorielle

Opération de rotation

Exemple : rotation du segment $S = [P_1 = (0, 0), P_2 = (10, 0)]$ avec

$$\text{matrice de rotation } M = \begin{pmatrix} 0,8 & -0,6 \\ 0,6 & 0,8 \end{pmatrix}$$

Cas d'une image vectorielle

Opération de rotation

Exemple : rotation du segment $S = [P_1 = (0,0), P_2 = (10,0)]$ avec

$$\text{matrice de rotation } M = \begin{pmatrix} 0,8 & -0,6 \\ 0,6 & 0,8 \end{pmatrix}$$

$$\Rightarrow M(S) = [M(P_1), M(P_2)] = [(0,0), (8,6)]$$

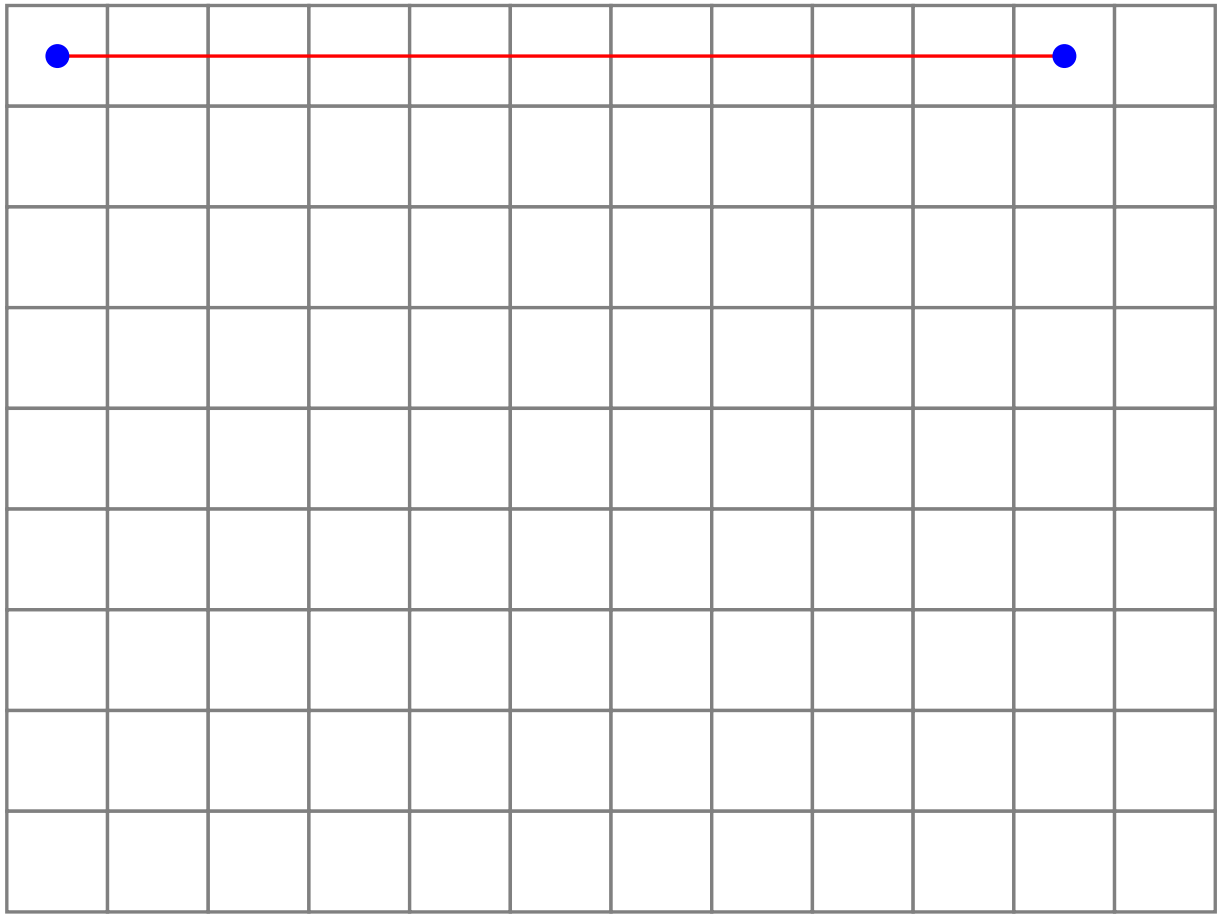
Cas d'une image vectorielle

Opération de rotation

Segment $S = [P_1 = (0,0)\ ,\ P_2 = (10,0)]$

Cas d'une image vectorielle

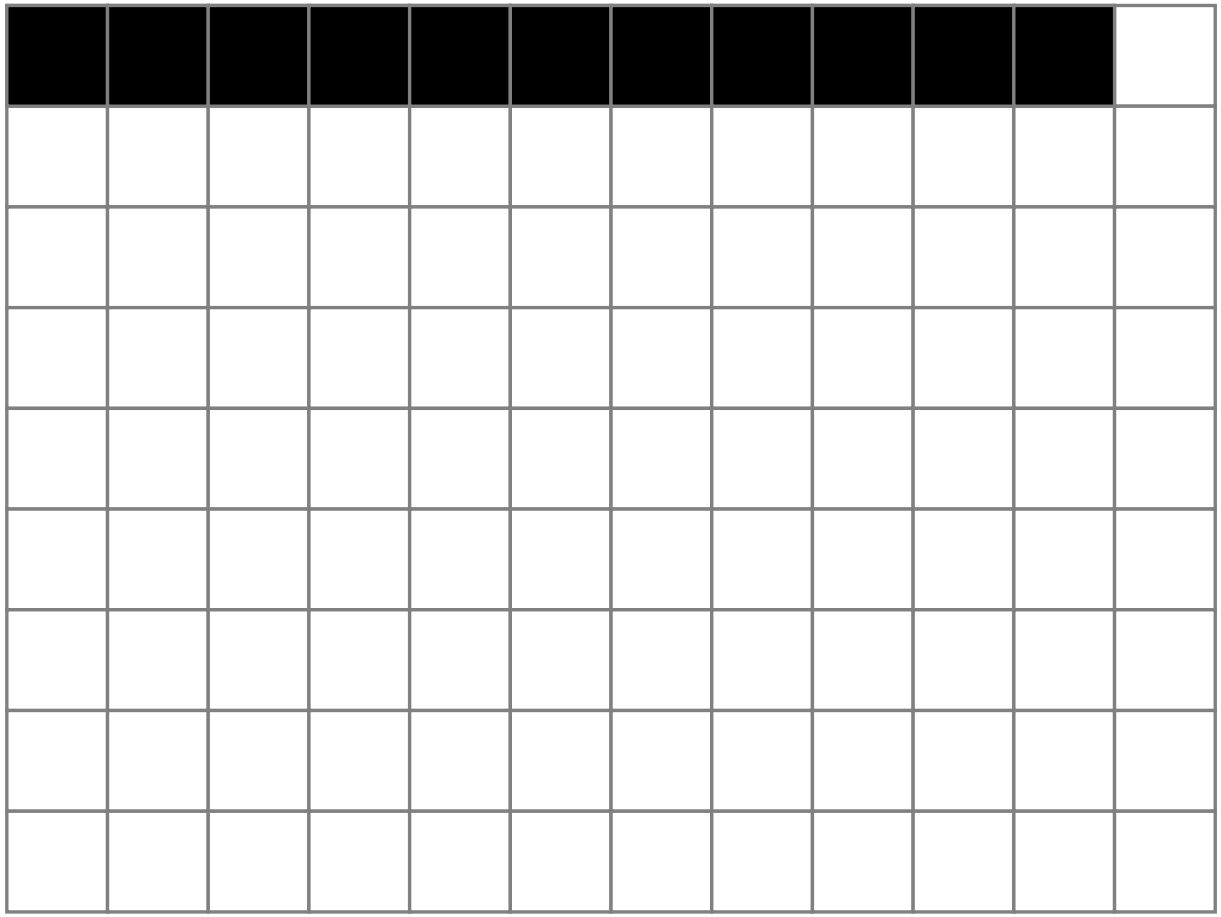
Opération de rotation



Affichage dans une grille pixel : algorithme de Bresenham

Cas d'une image vectorielle

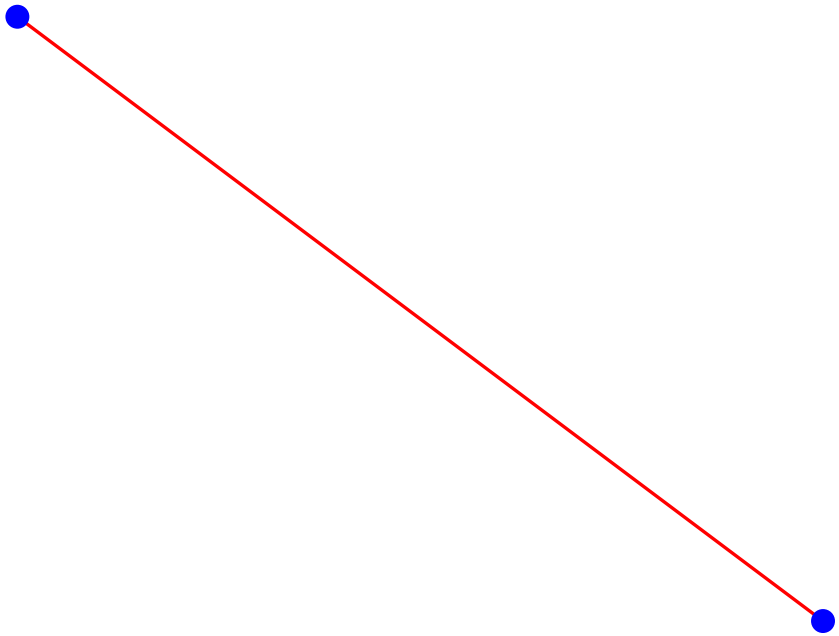
Opération de rotation



Affichage dans une grille pixel : algorithme de Bresenham

Cas d'une image vectorielle

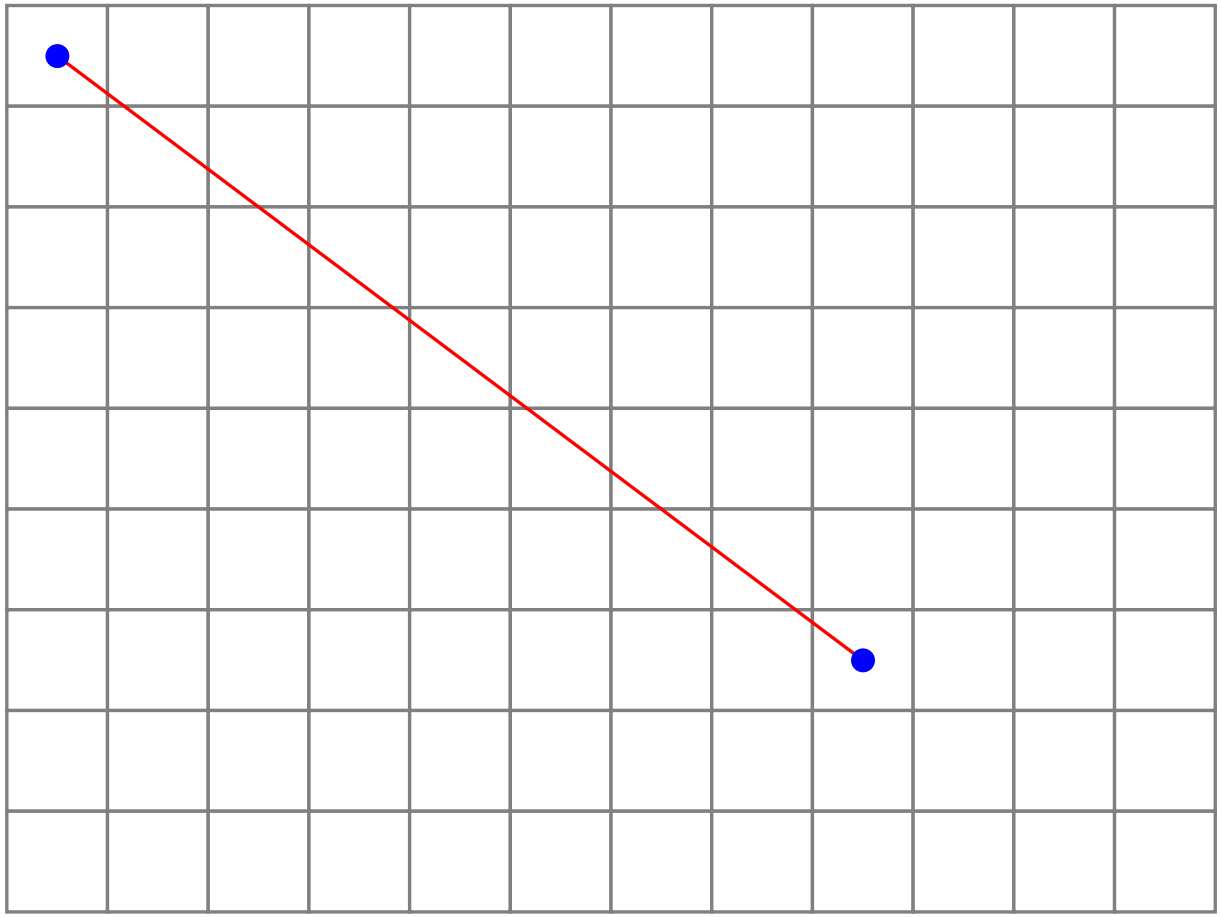
Opération de rotation



Segment $f(S) = [f(P_1) = (0,0)\ ,\ f(P_2) = (8,6)]$

Cas d'une image vectorielle

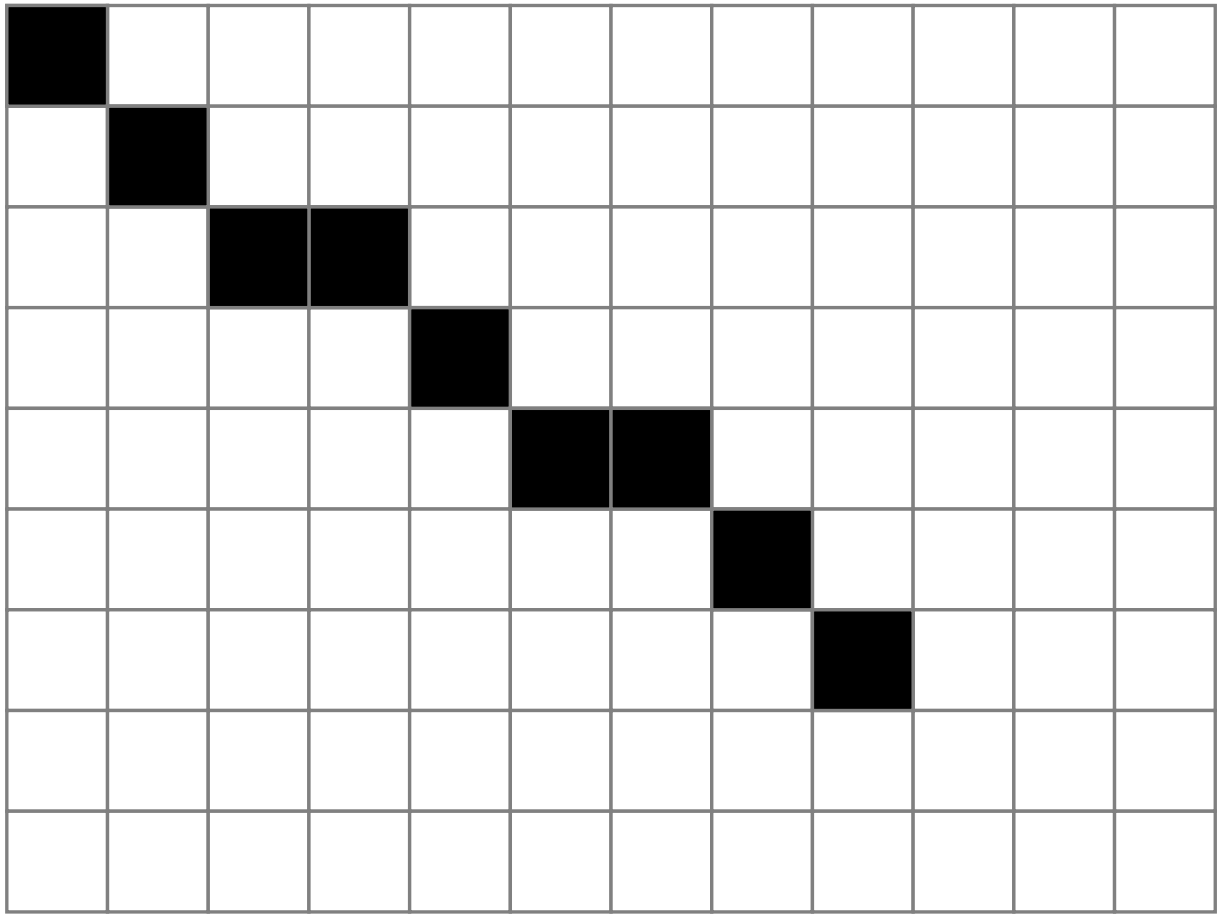
Opération de rotation



Affichage dans une grille pixel : algorithme de Bresenham

Cas d'une image vectorielle

Opération de rotation



Affichage dans une grille pixel : algorithme de Bresenham

Cas d'une image bitmap

Opération d'agrandissement

Cas d'une image bitmap

Opération d'agrandissement

Image I de dimensions $L \times H$

Facteur d'agrandissement n (n entier ≥ 2)

Cas d'une image bitmap

Opération d'agrandissement

Image I de dimensions $L \times H$
Facteur d'agrandissement n (n entier ≥ 2)

$\rightarrow$ Image agrandie $\bar{I}$ de dimensions $nL \times nH$

Cas d'une image bitmap

Opération d'agrandissement

Image I de dimensions $L \times H$
Facteur d'agrandissement n (n entier ≥ 2)

$\rightarrow$ Image agrandie $\bar{I}$ de dimensions $nL \times nH$

Pixel (x, y) de I	$0 \leq x \leq L-1$	$0 \leq y \leq H-1$
Pixel $(\bar{x}, \bar{y})$ de $\bar{I}$	$0 \leq \bar{x} \leq nL-1$	$0 \leq \bar{y} \leq nH-1$

Cas d'une image bitmap

Opération d'agrandissement

Image I de dimensions $L \times H$
Facteur d'agrandissement n (n entier ≥ 2)

$\rightarrow$ Image agrandie $\bar{I}$ de dimensions $nL \times nH$

Correspondance entre x et $\bar{x}$	$\bar{x} = \frac{x(nL-1)}{L-1}$	$x = \frac{\bar{x}(L-1)}{nL-1}$
Correspondance entre y et $\bar{y}$	$\bar{y} = \frac{y(nH-1)}{H-1}$	$y = \frac{\bar{y}(H-1)}{nH-1}$

Cas d'une image bitmap

Opération d'agrandissement

Calcul de l'image $\bar{I}$

Algorithme 1 : interpolation *au plus proche*

pour $\bar{x}$ **de** 0 **a** $nL-1$ **faire**
  pour $\bar{y}$ **de** 0 **a** $nH-1$ **faire**
    // partie entière la plus proche
    $x \leftarrow \texttt{arrondi}\left(\frac{\bar{x}(L-1)}{nL-1}\right)$, $y \leftarrow \texttt{arrondi}\left(\frac{\bar{y}(H-1)}{nH-1}\right)$
    $\bar{I}(\bar{x},\bar{y}) \leftarrow I(x,y)$
  fin_pour
fin_pour

Cas d'une image bitmap

Opération d'agrandissement

Calcul de l'image $\bar{I}$

Algorithme 2 : interpolation *bilinéaire*

pour $\bar{x}$ **de** 0 **a** $nL-1$ **faire**
  pour $\bar{y}$ **de** 0 **a** $nH-1$ **faire**
    $xr \leftarrow \dfrac{\bar{x}(L-1)}{nL-1}$, $yr \leftarrow \dfrac{\bar{y}(H-1)}{nH-1}$
    // *partie entière infèrieure*
    $x \leftarrow$ `arrondi_inf`(xr) , $y \leftarrow$ `arrondi_inf`(yr)
    $dx \leftarrow xr - x$, $dy \leftarrow yr - y$ // *partie décimale*
    $\bar{I}(\bar{x},\bar{y}) \leftarrow (1-dx)(1-dy)\, I(x,y) + dx\,(1-dy)\, I(x+1,y)$
    $+ (1-dx)\, dy\, I(x,y+1) + dx\, dy\, I(x+1,y+1)$
  fin_pour
fin_pour

Cas d'une image bitmap

Opération d'agrandissement - exemple 1

Cas d'une image bitmap

Opération d'agrandissement - exemple 1

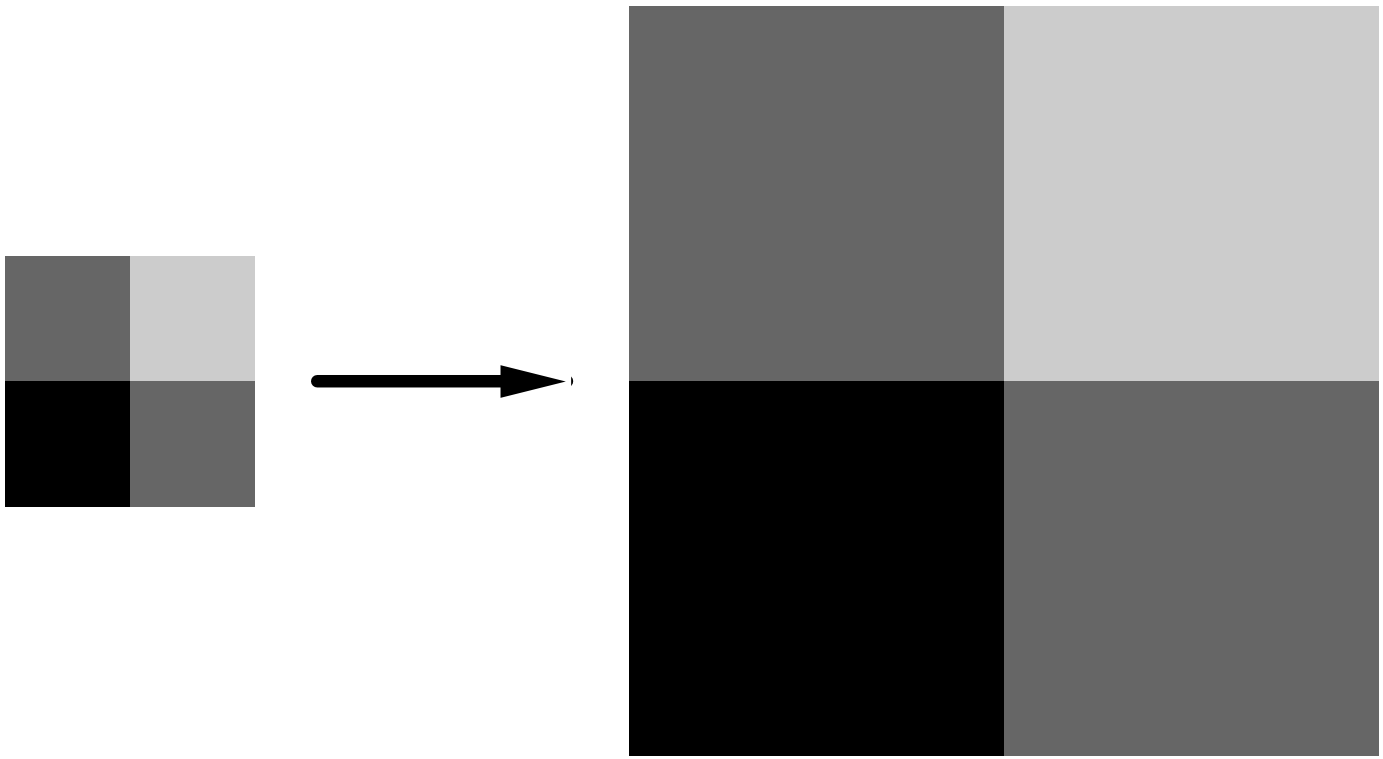
100	200
0	100

100	100	100	200	200	200
100	100	100	200	200	200
100	100	100	200	200	200
0	0	0	100	100	100
0	0	0	100	100	100
0	0	0	100	100	100

Interpolation *au plus proche*

Cas d'une image bitmap

Opération d'agrandissement - exemple 1



Interpolation *au plus proche*

Cas d'une image bitmap

Opération d'agrandissement - exemple 1

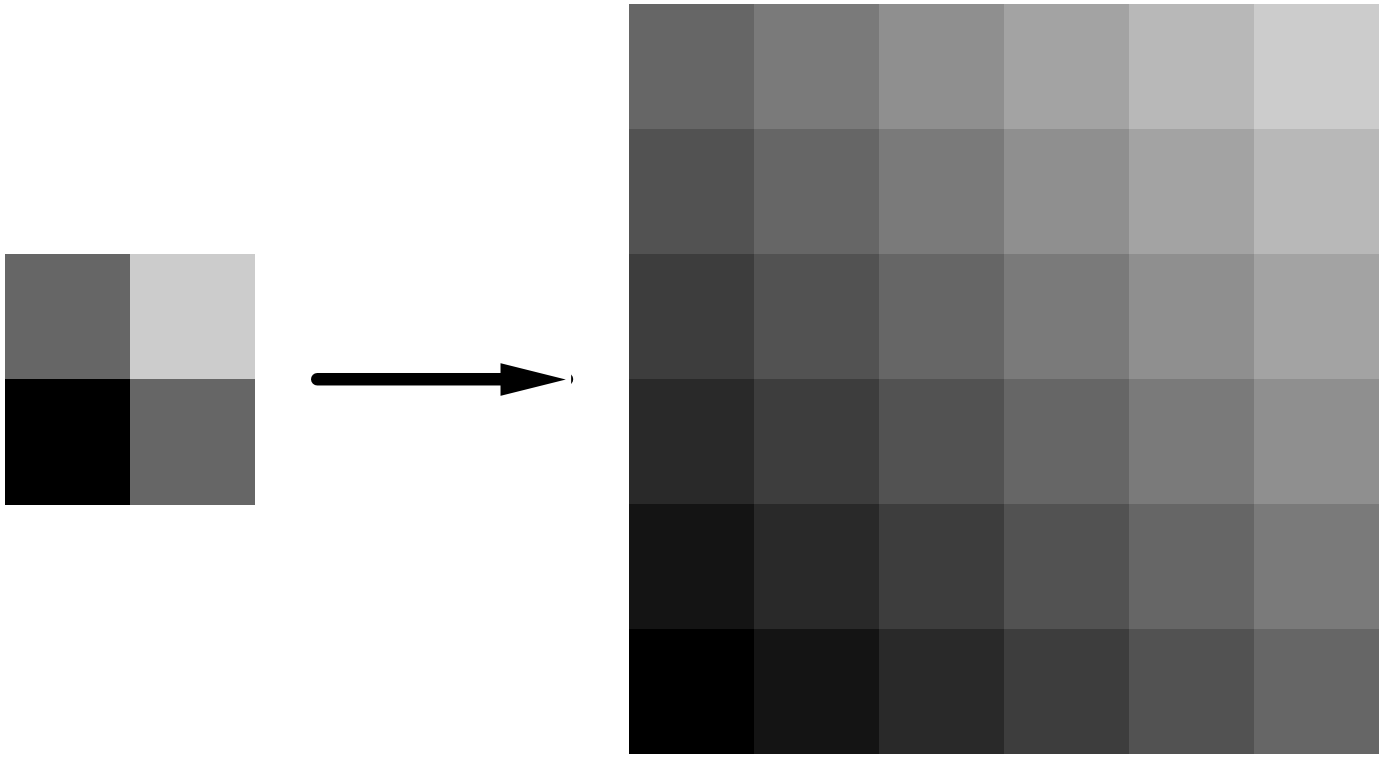
100	200
0	100

100	120	140	160	180	200
80	100	120	140	160	180
60	80	100	120	140	160
40	60	80	100	120	140
20	40	60	80	100	120
0	20	40	60	80	100

Interpolation *bilinéaire*

Cas d'une image bitmap

Opération d'agrandissement - exemple 1



Interpolation *bilinéaire*

Cas d'une image bitmap

Opération d'agrandissement - exemple 2

Image initiale

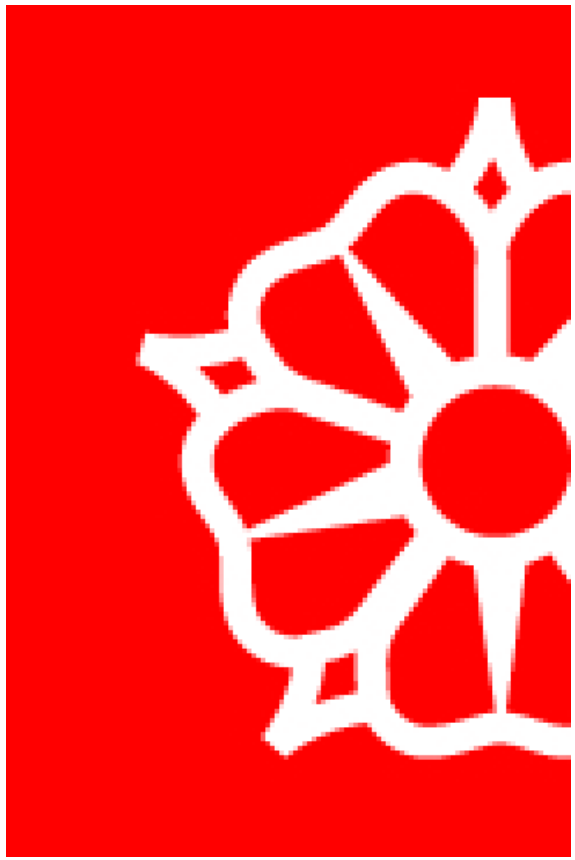


Image agrandie $\times 3$ (au plus proche)

Image agrandie $\times 3$ (bilinéaire)

Cas d'une image bitmap

Opération de rotation

Cas d'une image bitmap

Opération de rotation

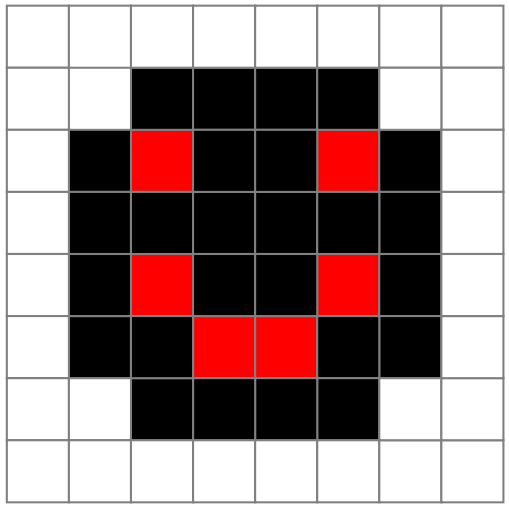
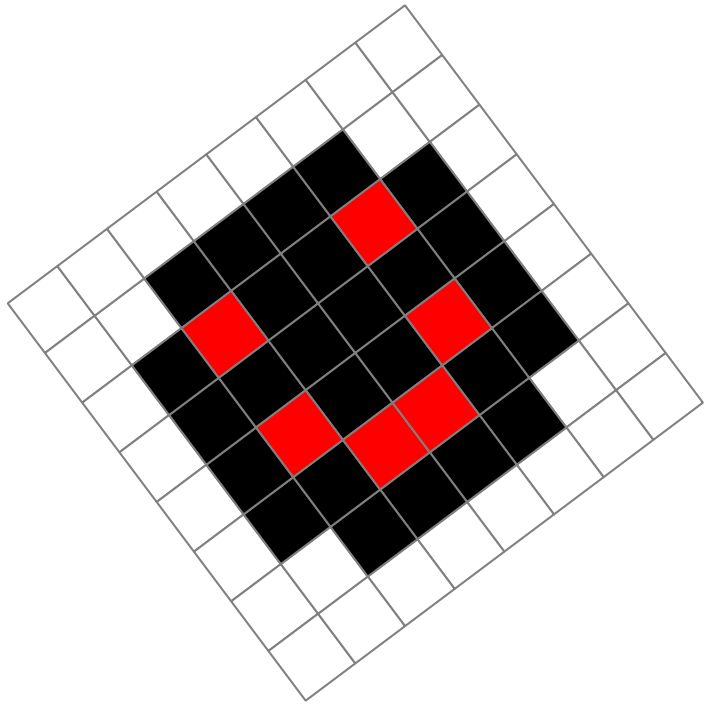


Image bitmap

Cas d'une image bitmap

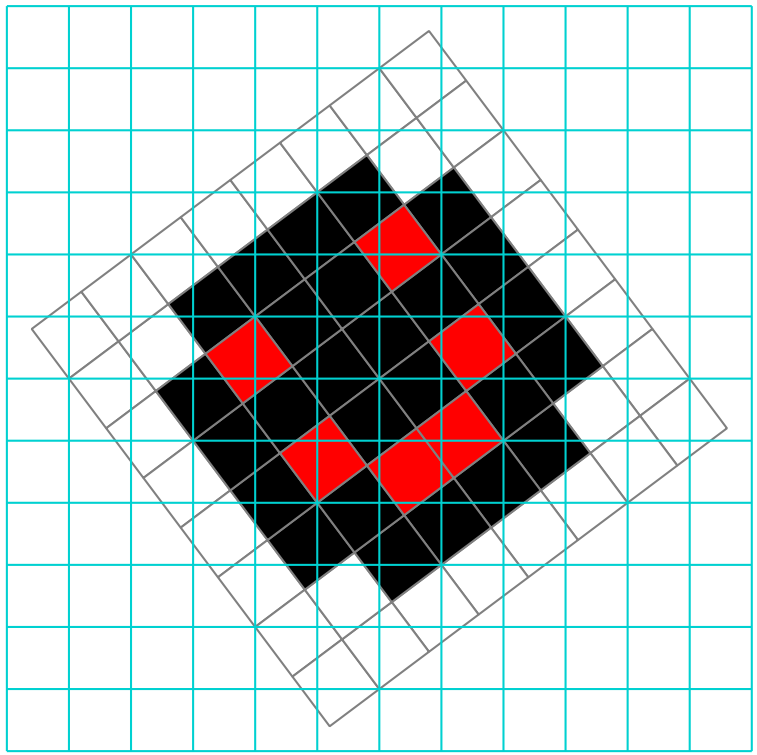
Opération de rotation



Rotation de l'image bitmap

Cas d'une image bitmap

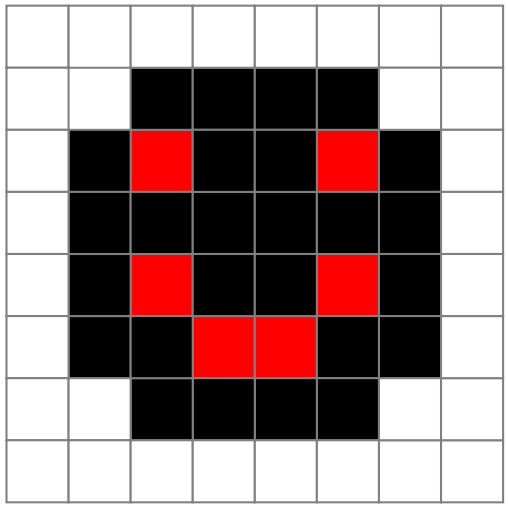
Opération de rotation

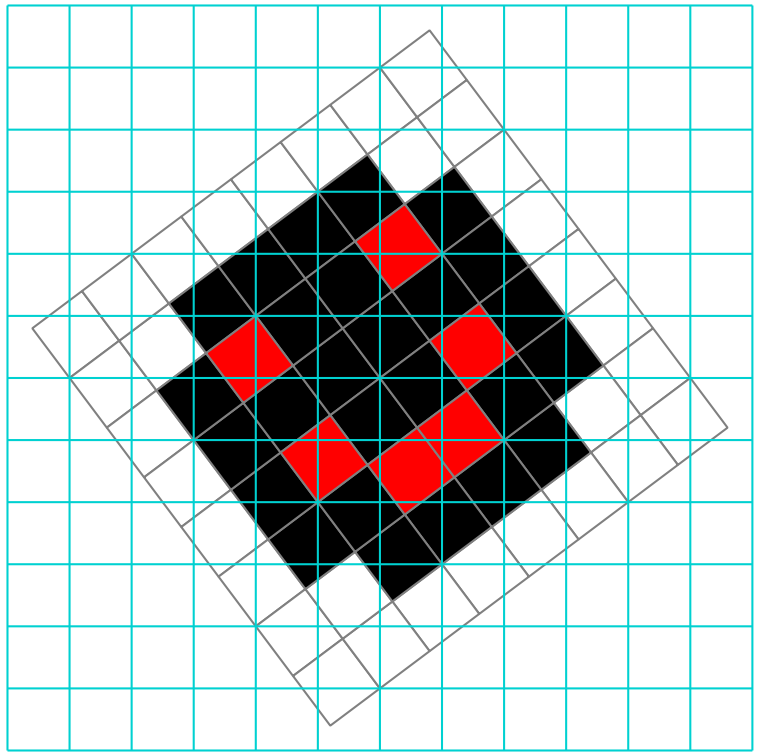


RESPECTER UNE GRILLE HORIZONTALE-VERTICALE

Cas d'une image bitmap

Opération de rotation

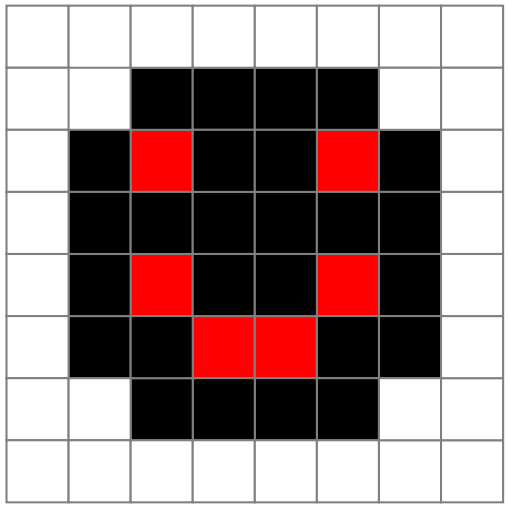


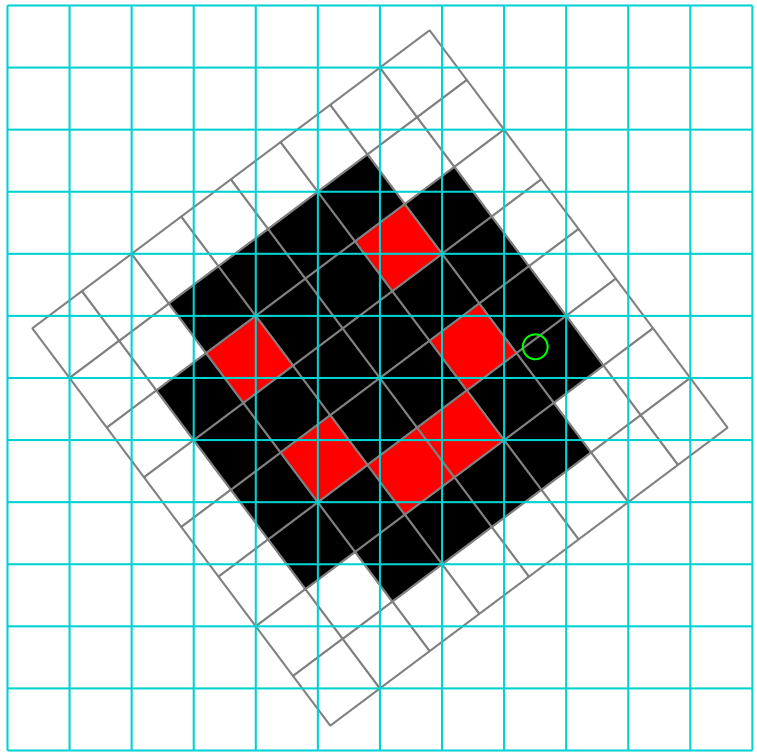


Calcul de l'image tournée par la fonction réciproque

Cas d'une image bitmap

Opération de rotation

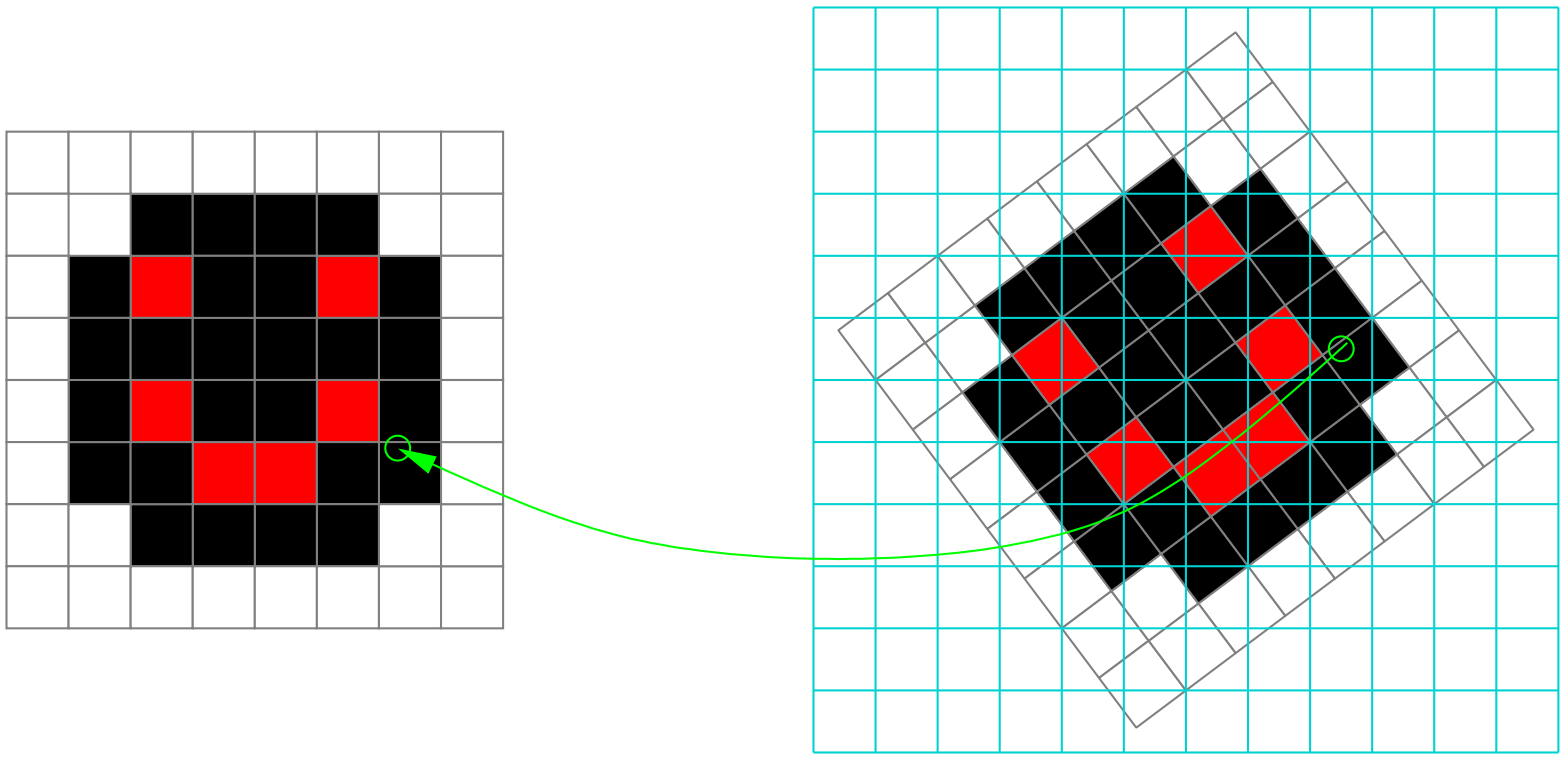




Calcul de l'image tournée par la fonction réciproque

Cas d'une image bitmap

Opération de rotation



Calcul de l'image tournée par la fonction réciproque

Image bitmap

Calcul de l'image finale

Image bitmap

Calcul de l'image finale

Pour chaque pixel (x', y') de l'image transformée I',

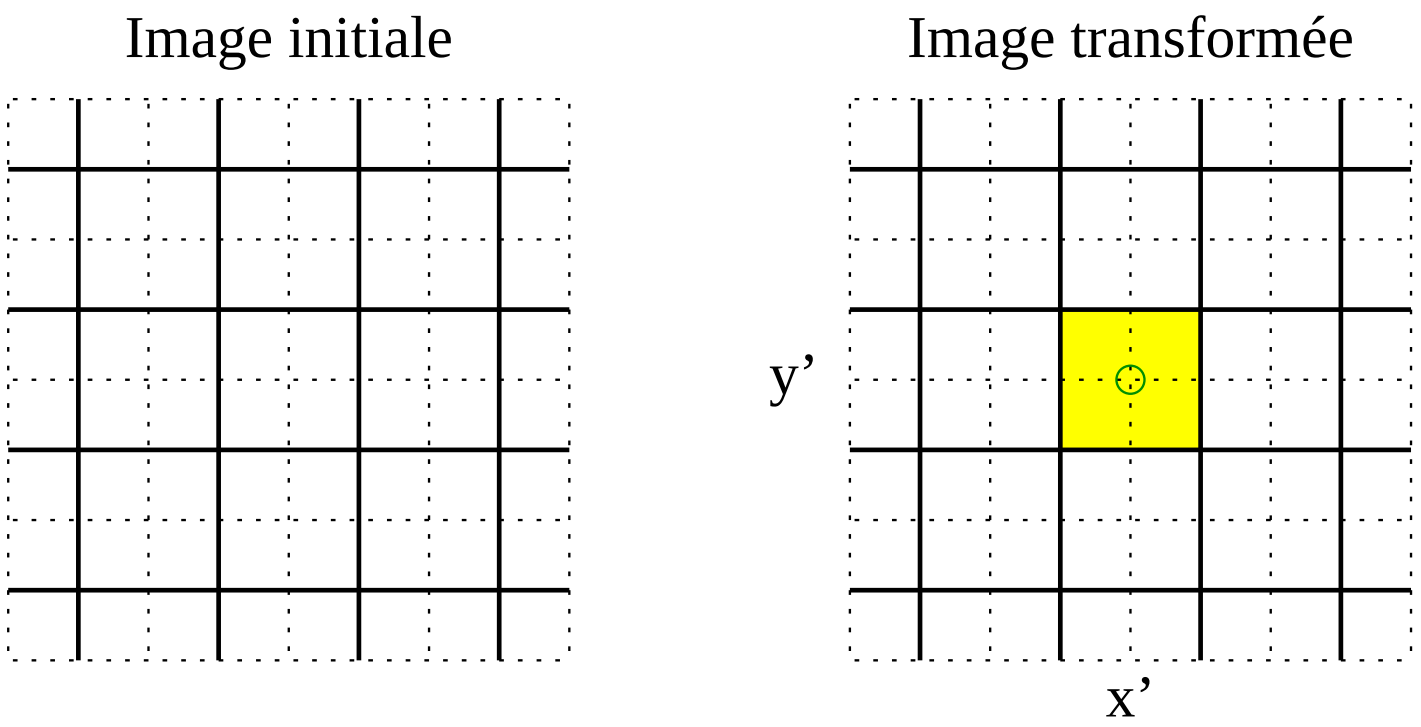


Image bitmap

Calcul de l'image finale

Pour chaque pixel (x', y') de l'image transformée I',
calculer $(x, y) = f^{-1}(x', y')$

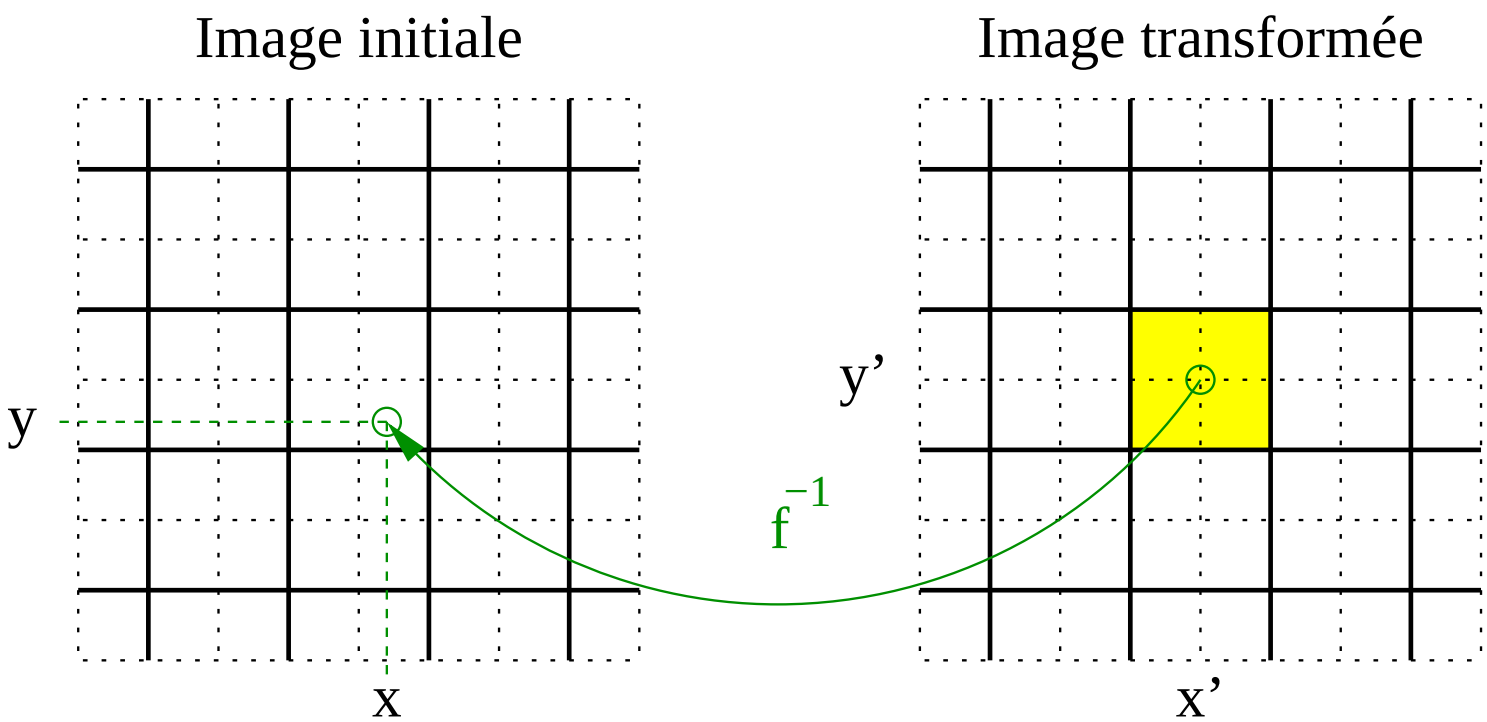


Image bitmap

Calcul de l'image finale

Pour chaque pixel (x', y') de l'image transformée I',
calculer $(x, y) = f^{-1}(x', y')$ en général (x, y) coordonnées non entières

Image bitmap

Calcul de l'image finale

Méthode 1 : *au plus proche*

choisir le pixel (X, Y) le plus proche de (x, y)

$\rightarrow I'(x', y') \leftarrow I(X, Y)$

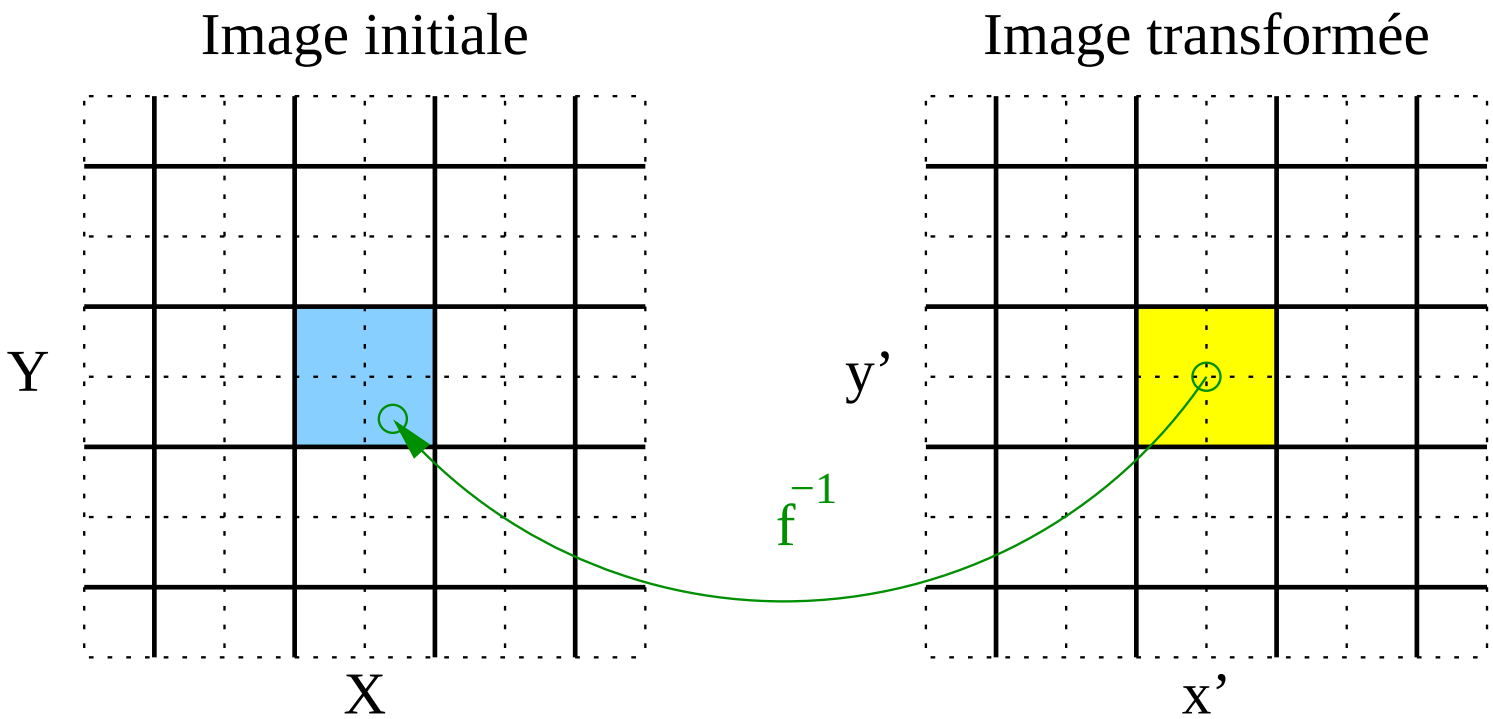


Image bitmap

Calcul de l'image finale

Méthode 2 : *interpolation bilinéaire*

choisir les 4 pixels (X, Y), $(X+1, Y)$, $(X, Y+1)$ et $(X+1, Y+1)$ autour de (x, y)

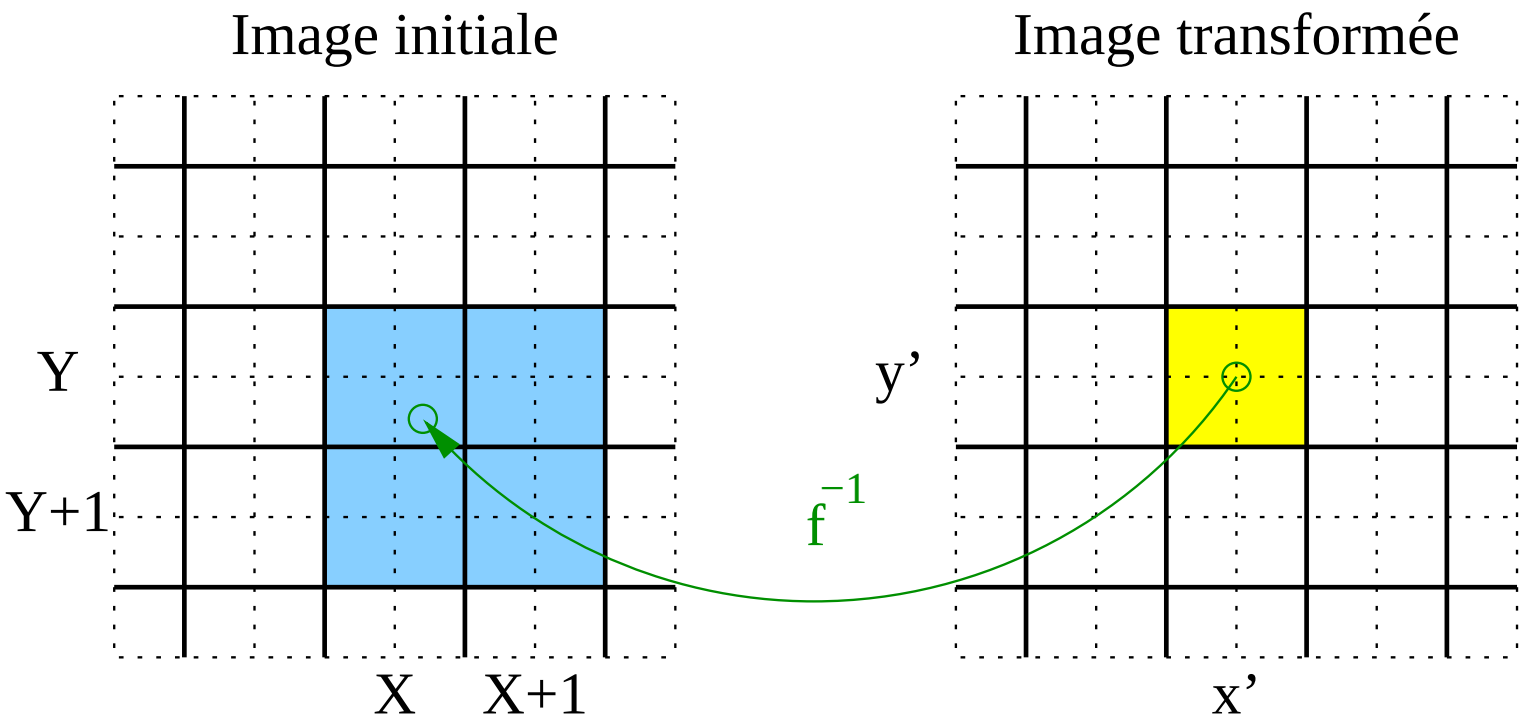


Image bitmap

Calcul de l'image finale

Méthode 2 : *interpolation bilinéaire*

Image bitmap

Calcul de l'image finale

Méthode 2 : *interpolation bilinéaire*

$$X = \lfloor x \rfloor = \text{floor}(x) \ , \quad Y = \lfloor y \rfloor = \text{floor}(y)$$
$$dx = x - X \ , \ dy = y - Y$$

Image bitmap

Calcul de l'image finale

Méthode 2 : *interpolation bilinéaire*

$$X = \lfloor x \rfloor = \text{floor}(x) \ , \quad Y = \lfloor y \rfloor = \text{floor}(y)$$
$$dx = x - X \ , \ dy = y - Y$$

$$I'(x', y')$$

Image bitmap

Calcul de l'image finale

Méthode 2 : *interpolation bilinéaire*

$$X = \lfloor x \rfloor = \text{floor}(x) \ , \quad Y = \lfloor y \rfloor = \text{floor}(y)$$
$$dx = x - X \ , \ dy = y - Y$$

$$I'(x', y')$$

$$= \begin{pmatrix} 1 - d_y & d_y \end{pmatrix} \begin{pmatrix} I(X, Y) & I(X+1, Y) \\ I(X, Y+1) & I(X+1, Y+1) \end{pmatrix} \begin{pmatrix} 1 - d_x \\ d_x \end{pmatrix}$$

Image bitmap

Calcul de l'image finale

Méthode 2 : *interpolation bilinéaire*

$$X = \lfloor x \rfloor = \text{floor}(x) \ , \quad Y = \lfloor y \rfloor = \text{floor}(y)$$
$$dx = x - X \ , \ dy = y - Y$$

$$I'(x', y')$$

$$= \begin{pmatrix} 1 - d_y & d_y \end{pmatrix} \begin{pmatrix} I(X, Y) & I(X+1, Y) \\ I(X, Y+1) & I(X+1, Y+1) \end{pmatrix} \begin{pmatrix} 1 - d_x \\ d_x \end{pmatrix}$$

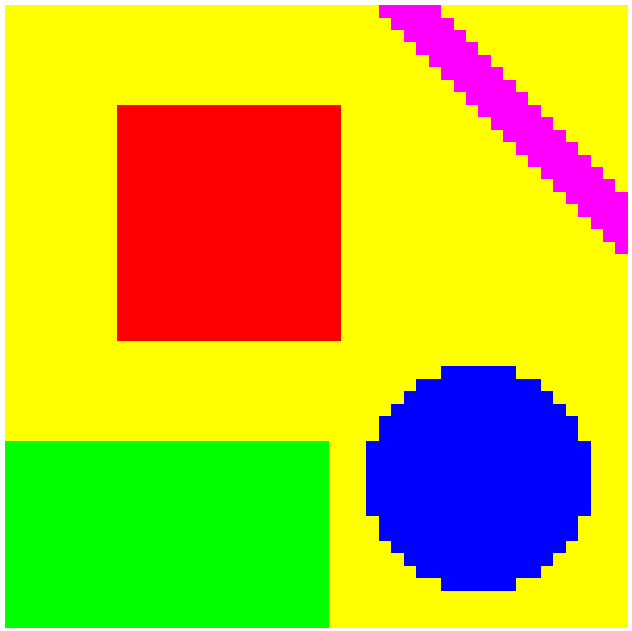
$$\begin{aligned} = & \ (1 - d_x)(1 - d_y) I(X, Y) + d_x (1 - d_y) I(X+1, Y) \\ + & \ (1 - d_x) d_y \ I(X, Y+1) + d_x \ d_y \ I(X+1, Y+1) \end{aligned}$$

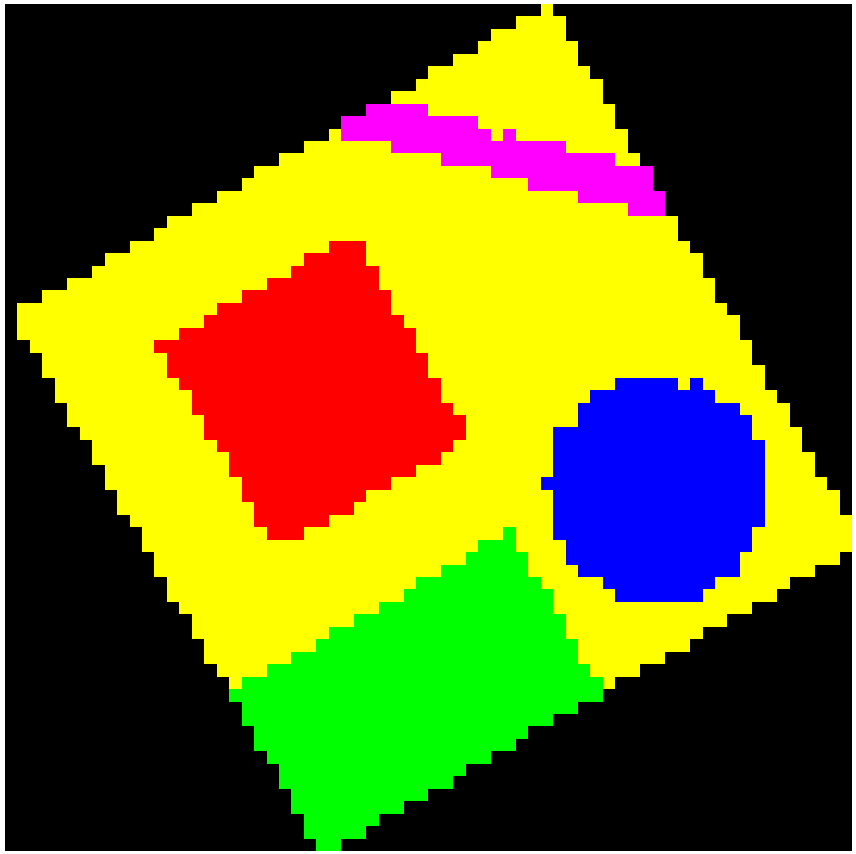
Image bitmap

Opération de rotation

Image bitmap

Opération de rotation

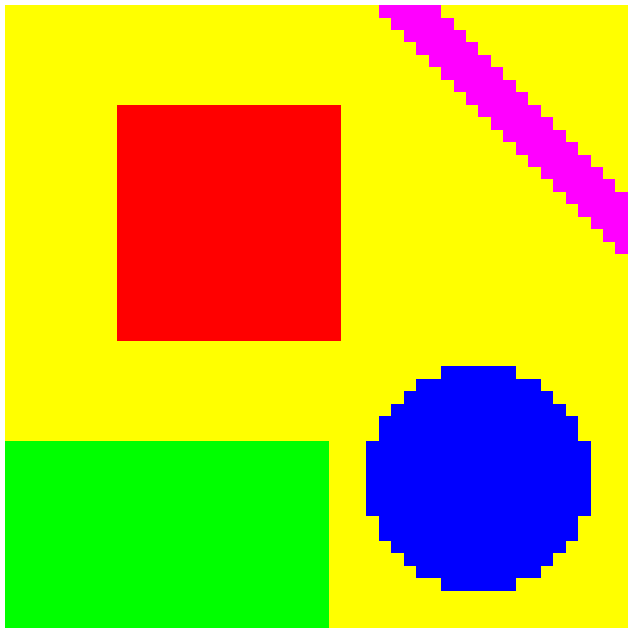


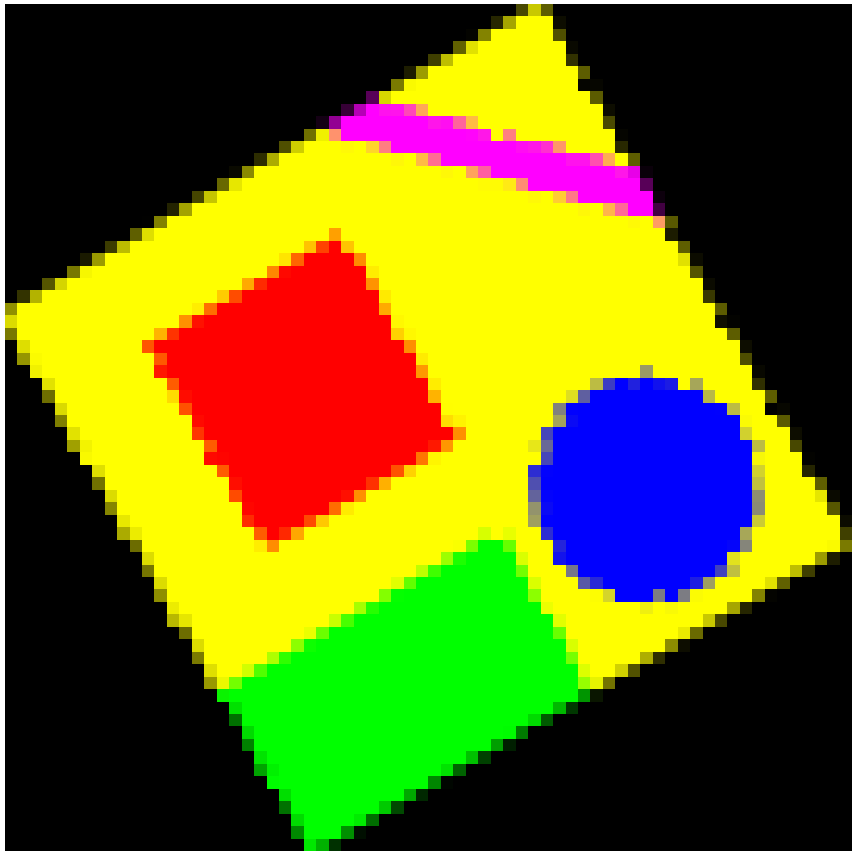


Calcul en utilisant le pixel le plus proche

Image bitmap

Opération de rotation





Calcul en utilisant l'interpolation bilinéaire