

TP2 : Traitement images en VHDL

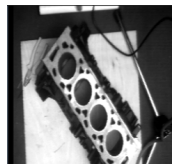
Les objectifs de ce TP sont :

- L'utilisation des sous programmes (fonctions et procédures),
- Les tableaux à 2 dimensions non contraints,
- L'utilisation du package Textio,
- L'utilisation des entrées sorties avec des fichiers.

TP2 : Traitement images en VHDL

Environnement :

3 images (format gdr)



1 package (entree_sortie_pkg.vhd → Moodle):

- Type de données (tableau 2D)
- 2 sous programmes
 - Getpixels : extraction des pixels d'un fichier image → tableau 2D
 - Putpixels : Tableau 2D (pixels traités) → fichier image

3 utilitaires :

- gdr2txt (conversion d'un fichier image gdr → fichier image txt)
- txt2gdr (conversion d'un fichier image txt → fichier image gdr)
- XV visualisation des images gdr

→ Fournis sur Moodle

Package : entree_sortie_pkg

```
package entree_sortie_pkg is

type IMAGE2D is array (natural range <>, natural range <>) of integer;

function get_pixels(entree : string; N_col : integer; N_ligne : integer) return IMAGE2D;

procedure put_pixels(sortie : string; IM2D : IMAGE2D; n_pix_x : integer; n_pix_y : integer;
                     n_bit_pixel : integer);

end entree_sortie_pkg;
```

La fonction Put_Pixel

```
function get_pixels(entree : string; N_col : integer; N_ligne : integer) return IMAGE2D is
variable ligne : LINE;
file FI       : TEXT is in entree;
variable IM2D : IMAGE2D(1 to N_ligne, 1 to N_col);
variable I    : natural:=0;
begin
-- enlever l'entête
READLINE(FI,ligne); READLINE(FI,ligne); READLINE(FI,ligne); READLINE(FI,ligne);
loop
    if not ENDFILE(FI) then
        READLINE(FI,ligne);
        I := I + 1;
        else return IM2D;

    end if;

    for j in 1 to N_col loop
        READ(ligne, IM2D(I,J));
    end loop;

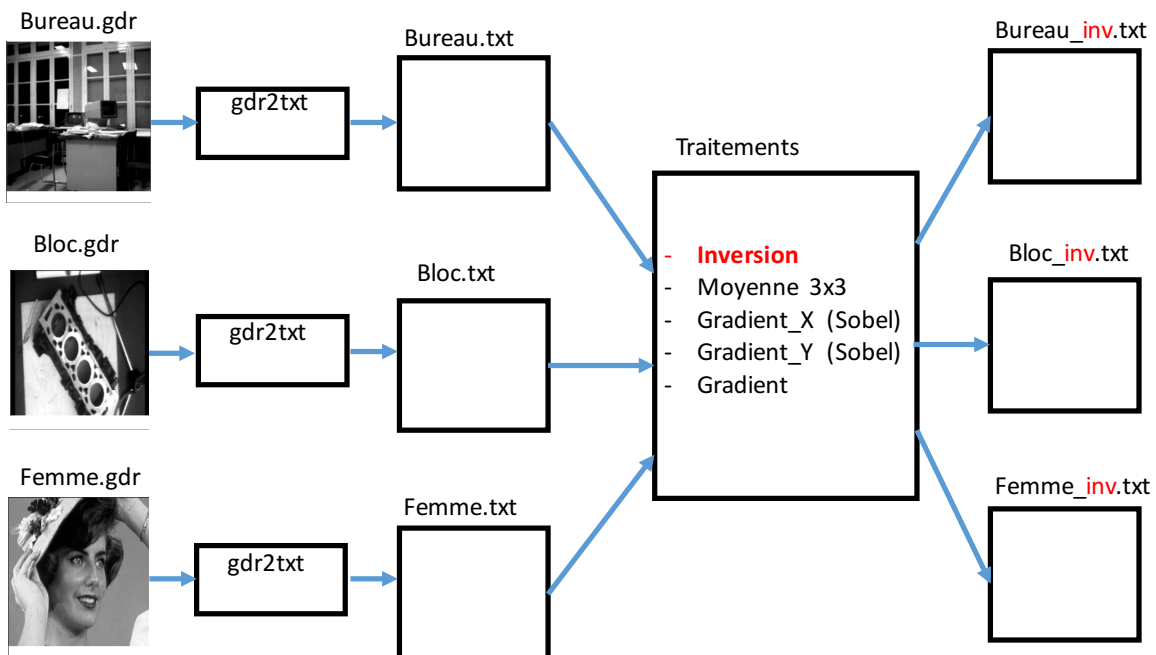
end loop;
end get_pixels;
```

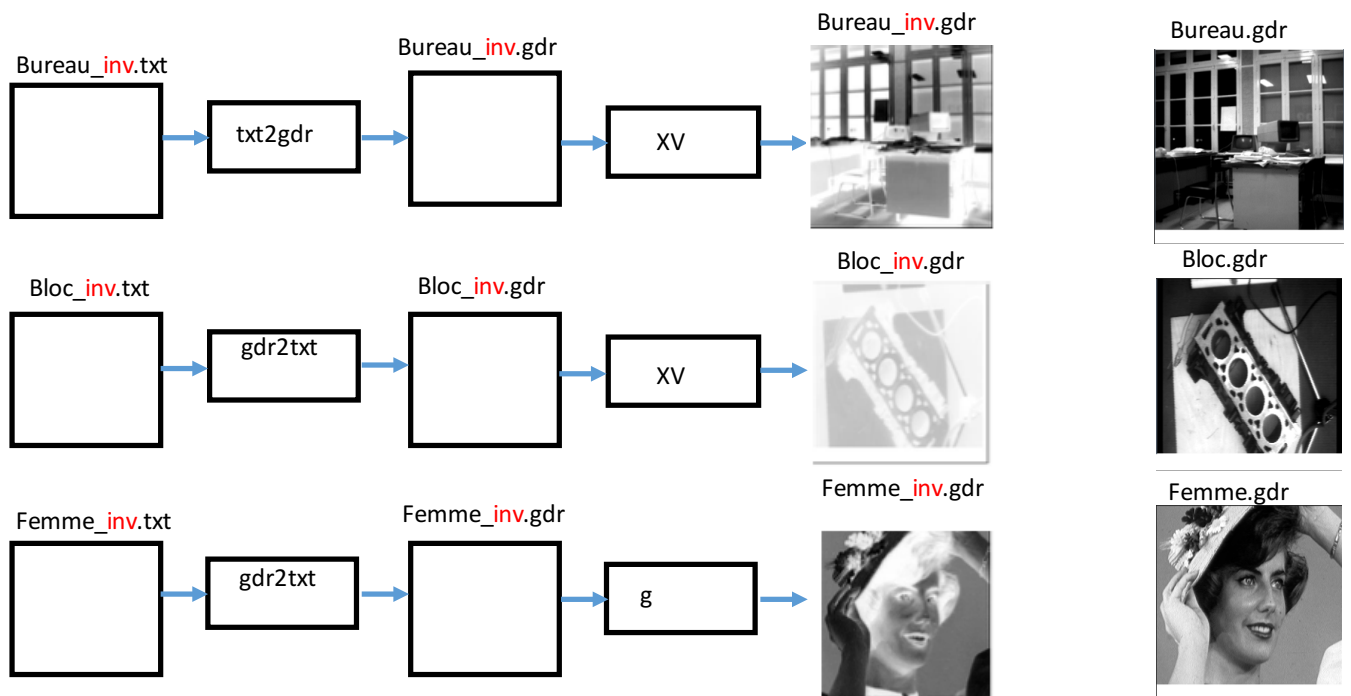
La procédure Put_Pixel

```
procedure put_pixels(sortie : string; IM2D : IMAGE2D; n_pix_x : integer;
                    n_pix_y : integer; n_bit_pixel : integer) is
variable ligne : LINE;
file      FO    : TEXT is out sortie;
Begin

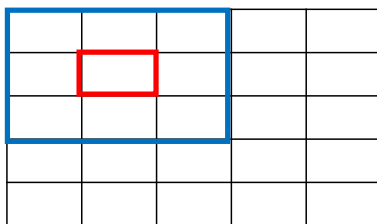
WRITE (LIGNE, STRING'("Type=GDR 2.0"));WRITELINE(FO,LIGNE);
WRITE (LIGNE,  STRING'("Nb.Pixel/Ligne="));WRITE (LIGNE,n_pix_x);WRITELINE(FO,LIGNE);
WRITE (LIGNE,STRING'("Nb.Lignes="));WRITE (LIGNE,n_pix_x);WRITELINE(FO,LIGNE);
WRITE (LIGNE,  STRING'("Nb.Bits="));WRITE (LIGNE,n_bit_pixel);WRITELINE(FO,LIGNE);
WRITE(LIGNE,  STRING'(" "));WRITELINE(FO,LIGNE);

for I in 1 to n_pix_y loop
  for J in 1 to n_pix_x loop
    WRITE(LIGNE, IM2D(I,J));
    WRITE(LIGNE, STRING'(" ")); -- Separateur
  end loop;
  WRITELINE(FO,LIGNE);
end loop;
end put_pixels;
```





La moyenne : Masque de 3x3



Somme des 9 pixels / 9

Le gradient : Masque de 3x3

$G_x =$

+1	0	-1		
+2	0	-2		
+1	0	-1		



$G_y =$

+1	+2	+1		
0	0	0		
-1	-2	-1		



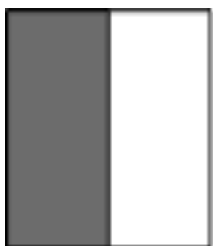
→ Utiliser valeur absolue

$$\text{Le gradient } G = \sqrt{G_x^2 + G_y^2}$$

(Utiliser la fonction sqrt présente dans le package math_real de la bibliothèque iee (voir prototype dans C:\PROGRAMS\altera\13.0sp1\quartus\libraries\vhdl\ieee))

Synthétiser les images :

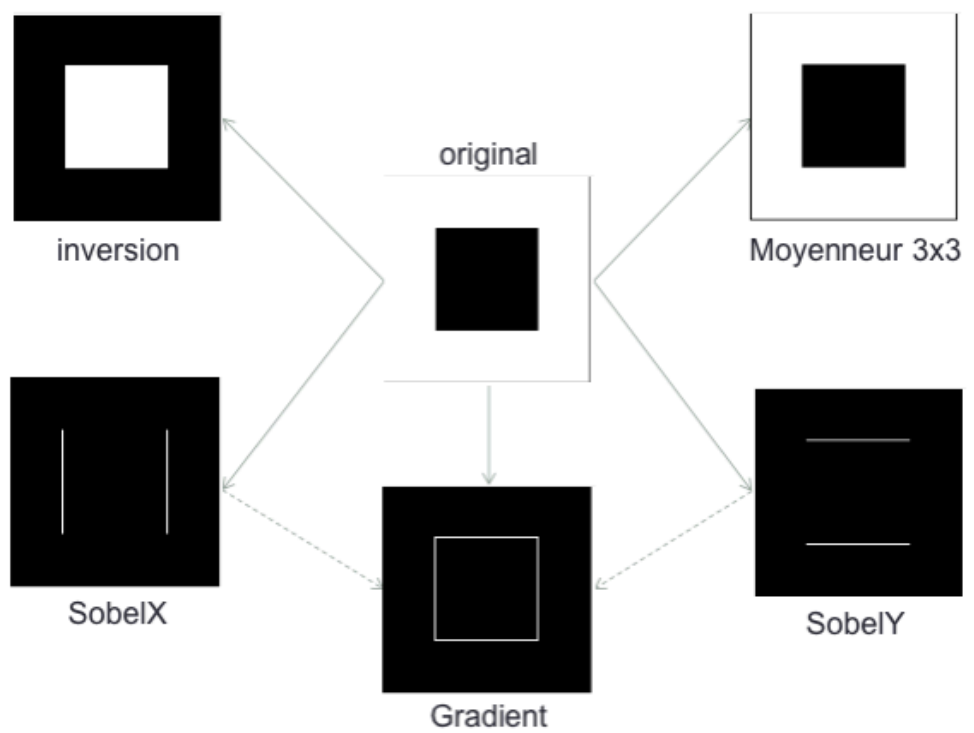
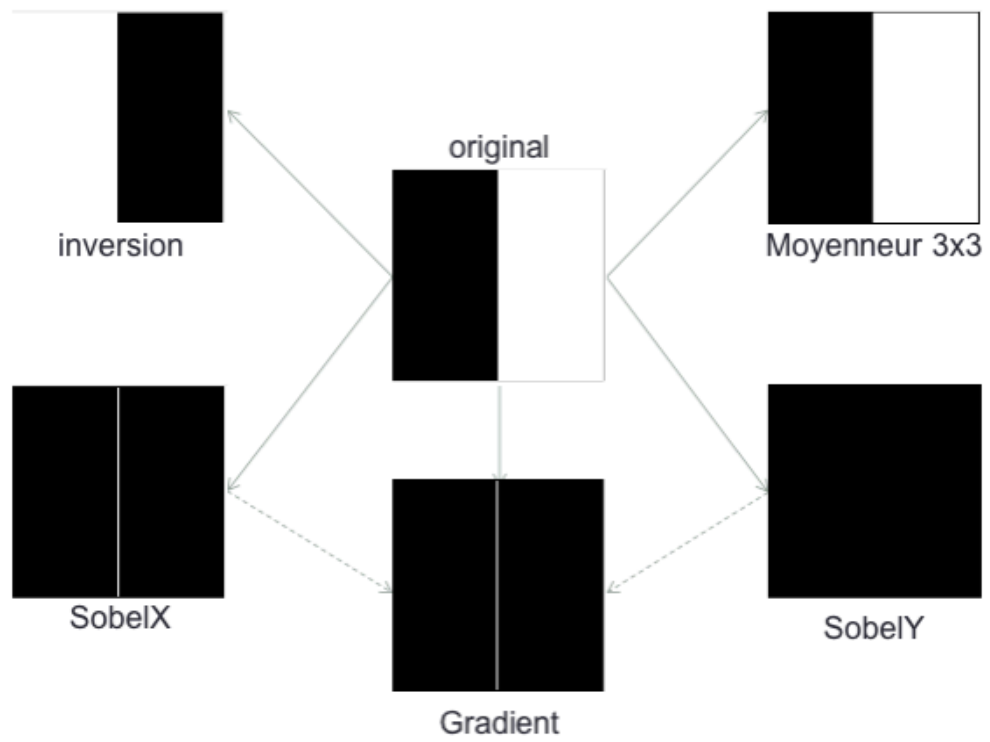
A partir d'un tableau 2D vide → remplir le tableau

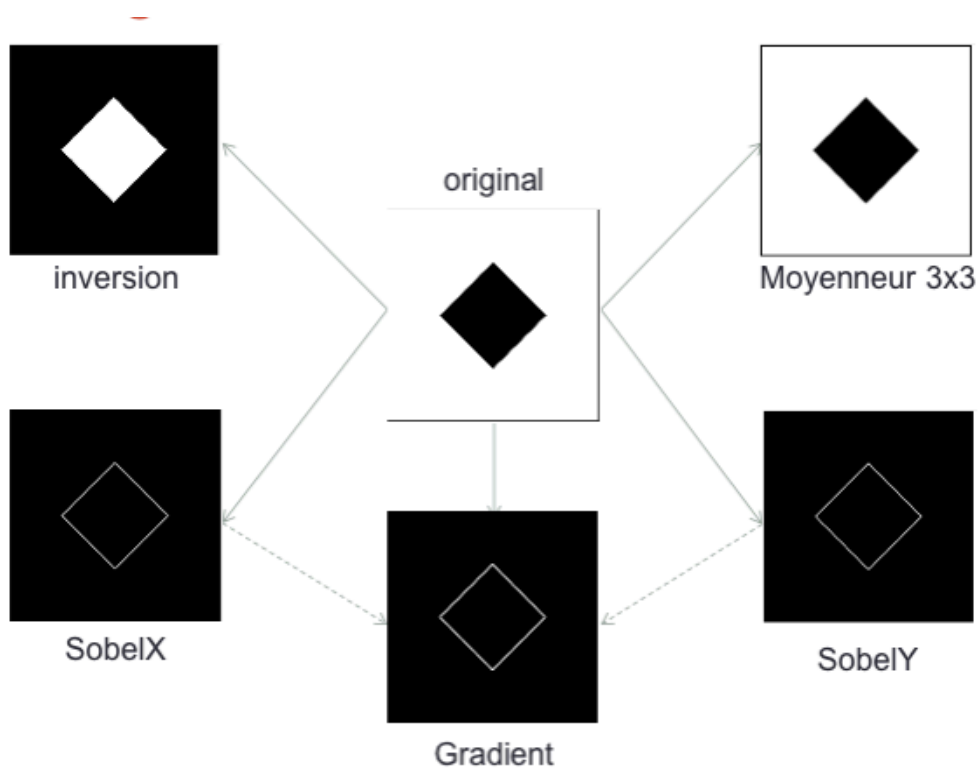
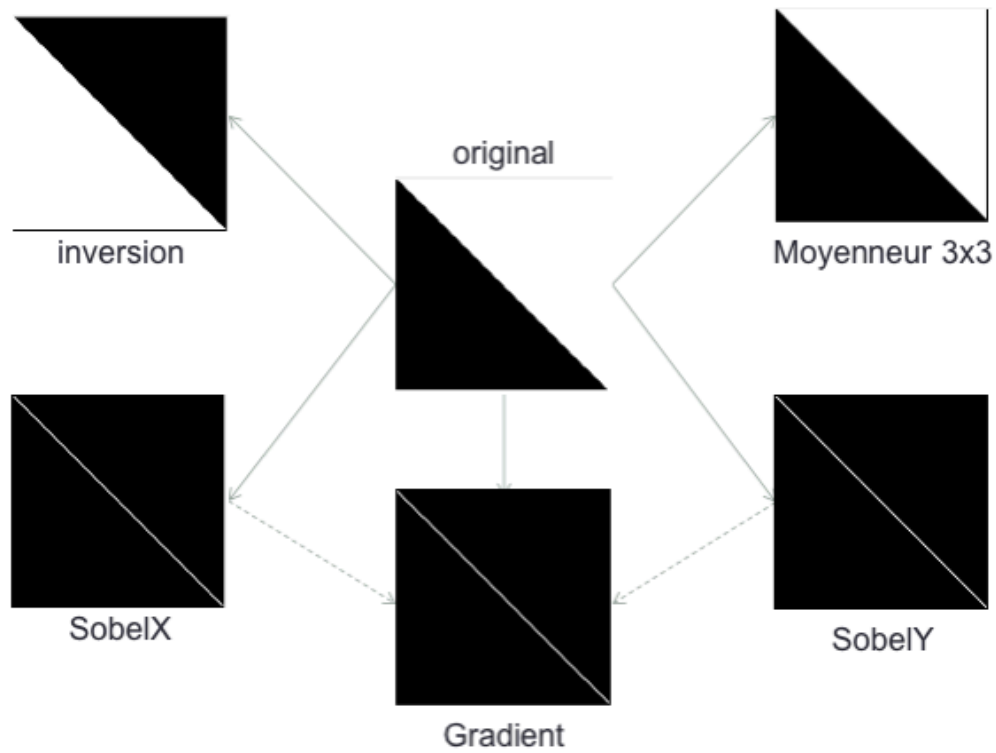


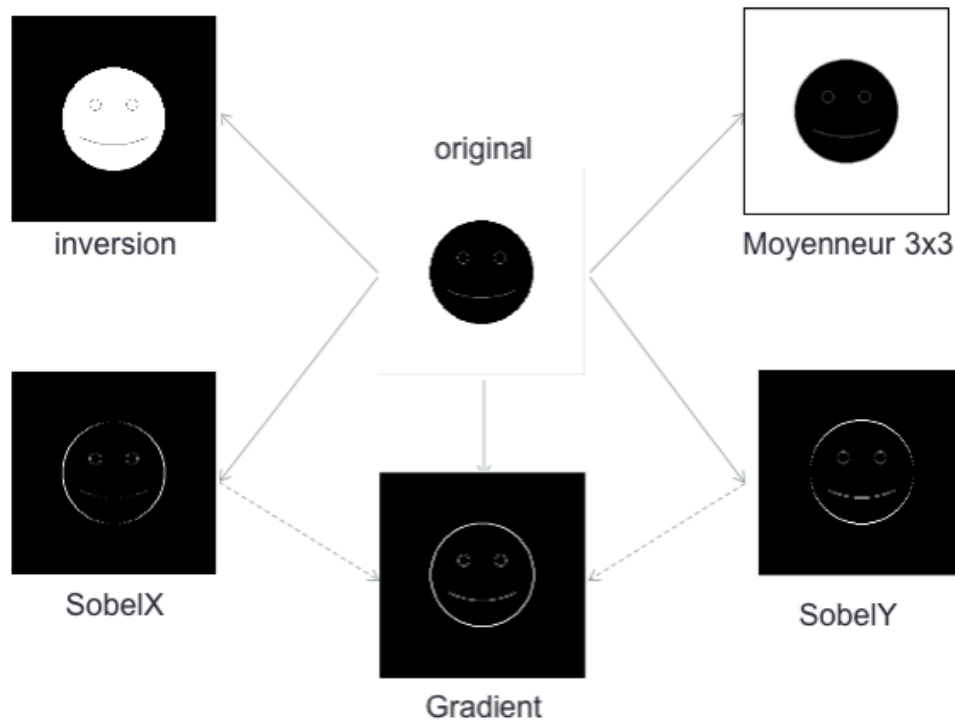
Putpixel → txt2gdr → xv : visualiser les images originales

Faire les mêmes traitements (inversion, moyenne, G_x , G_y , Gradient)

Putpixel → txt2gdr → xv : visualiser les images traitées







Exemple ENTREE-SORTIE FICHIER (5)

```

library ieee;
use work.LENA_PKG.all;
-----
entity Traitement is
generic( N_col : natural := 256; N_ligne: natural:= 256);
end Traitement ;
-----
architecture arch_Inv_niv of Traitement is
begin
process
variable tab_image: IMAGE2D(1 to N_col, 1 to N_ligne);
variable tab_image_traitee: IMAGE2D( 1 to N_col, 1 to N_ligne);
begin
tab_image := get_pixels("bureau.txt", N_col, N_ligne);
L1:for i in 1 to N_ligne-1 loop
L2: for j in 1 to N_col-1 loop
tab_image_traitee(i,j) := 255- tab_image(i,j); -- traitement
end loop L2;
end loop L1;
put_pixels("bureau_inv.txt",tab_image_traitee, N_col, N_ligne, 8);
wait;
end process;
end arch_Inv_niv;
  
```

Pas besoin de testbench !

