

TP 2: Atténuation du phénomène de moiré version 2012

François LEPAN

5 mars 2013

1 Manipulations

Tout d'abord on charge l'image "256a.jpg" (*cf.* Fig. 1)

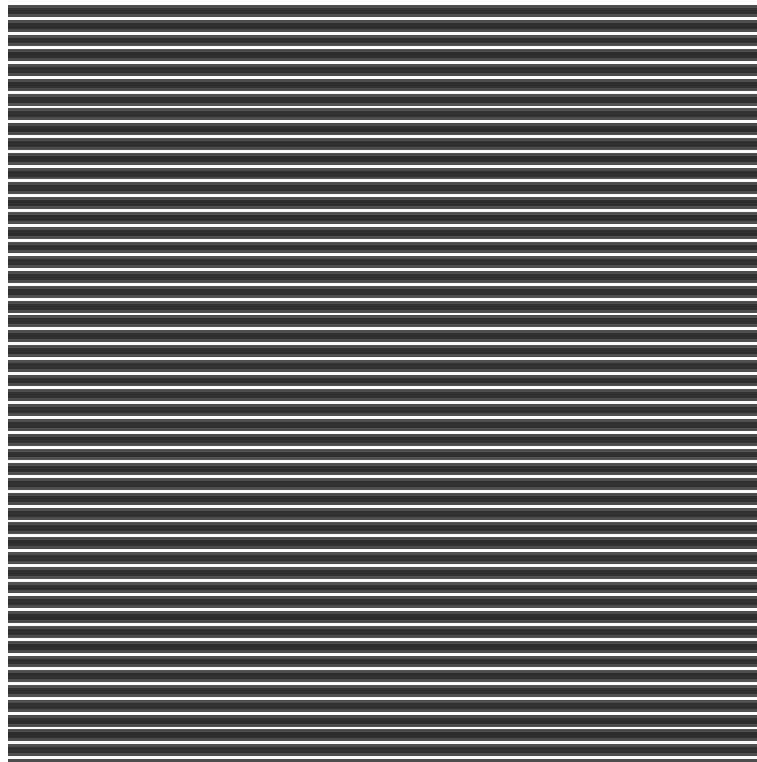


FIGURE 1 – Image 256_a.jpg

1.1 Fréquence spatiale ω (cycle/pixel) du motif cyclique de la Fig. 1

Grâce à l'outil *scrolling tool* on peut récupérer la période spatiale T qui est de 5 pixels.
Or la fréquence spatiale F est l'inverse de la période spatiale.
Donc $F = \frac{1}{5} = 0.2 \text{ cycle/pixel}$

1.2 Coordonnées de la raie maximale

Voici le code correspondant à l'affichage des coordonnées :

```
max_1 = 0;
i_max_1 = 0;
j_max_1 = 0;

// Get the primary and secondary ray position
for (j=0; j<H; j++)
{
    for (i=0; i<W; i++)
    {
        p = getPixel(i,j);

        // get the primary ray
        if ( max_1 < p)
        {
            max_1 = p;
            i_max_1 = i;
            j_max_1 =j;
        }
    }
}

// Printing the value of the points
print ("i_max_1 = ", i_max_1, ", j_max_1 = ", j_max_1);
```

Les valeurs de *i_max_1* et *j_max_1* correspondants aux coordonnées de la raie maximale dans le plan de Fourier valent après exécution du code précédent : (128,128)

1.3 Affichage de la raie secondaire ainsi que la distance la séparant de la raie principale

Voici le code correspondant :

```
max_1 = 0;
i_max_1 = 0;
j_max_1 = 0;

max_2 = 0;
i_max_2 = 0;
j_max_2 = 0;

// Get the primary and secondary ray position
for (j=0; j<H; j++)
{
    for (i=0; i<W; i++)
    {
        p = getPixel(i,j);
```

```

        // get the primary ray
        if ( max_1 < p)
        {
            max_1 = p;
            i_max_1 = i;
            j_max_1 =j;
        }

        // get the secondary ray
        if ( max_2 < p && max_1 > p)
        {
            max_2 =p;
            i_max_2 = i;
            j_max_2 =j;
        }
    }
}

// Calculating the distance between the two ray
distance = sqrt(pow(i_max_2 - i_max_1, 2) + pow(j_max_2 - j_max_1, 2));

// Printing the value of the points and the distance between them
print ("i_max_1 = ", i_max_1, ", j_max_1 = ", j_max_1);
print ("i_max_2 = ", i_max_2, ", j_max_2 = ", j_max_2);
print ("D = ", distance);

```

1.4 Tableau regroupant les informations (Direction, ω , P_x , etc) des images de taille 128x128

Image	Direction (x,y,xy)	T (pixel dans l'image)	ω (cycle/pixel)	P_x (dans fourier)	P_y (dans fourier)	D (dans fourier)
128a	verticale	5	0.2	64	90	26
128b	diagonale	1.77	0.57	115	115	72
128c	horizontale	5	0.2	90	64	26
128d	verticale	10	0.1	64	77	13
128e	diagonale	7.14	0.14	77	77	18
128f	horizontale	10	0.1	77	64	13

Pour ces images 128x128 on a :

$$\omega_a = 26 / 128 = 0.2$$

$$\omega_c = 26 / 128 = 0.2$$

$$\omega_d = 13 / 128 = 0.1$$

$$\omega_f = 13 / 128 = 0.1$$

1.5 Tableau regroupant les informations (Direction, ω , P_x , etc) des images de taille 256x256

Image	Direction (x,y,xy)	T (pixel dans l'image)	ω (cycle/pixel)	P_x (dans fourier)	P_y (dans fourier)	D (dans fourier)
256a	verticale	5	0.2	128	179	51
256b	diagonale	3.6	0.28	179	179	72
256c	horizontale	5	0.2	179	128	51
256d	verticale	10	0.1	128	154	26
256e	diagonale	2.33	0.43	205	205	108.9
256f	horizontale	10	0.1	154	128	26

Pour ces images 256x256 on a :

$$\omega_a = 51 / 256 = 0.2$$

$$\omega_c = 51 / 256 = 0.2$$

$$\omega_d = 26 / 256 = 0.1$$

$$\omega_f = 26 / 256 = 0.1$$

1.6 Tableau regroupant les informations (Direction, ω , P_x , etc) des images de taille 512x512

Image	Direction (x,y,xy)	T (pixel dans l'image)	ω (cycle/pixel)	P_x (dans fourier)	P_y (dans fourier)	D (dans fourier)
512a	verticale	5	0.2	256	358	102
512b	diagonale	1.77	0.57	461	461	290
512c	horizontale	5	0.2	358	256	102
512d	verticale	10	0.1	256	307	51
512e	diagonale	7.14	0.14	307	307	72
512f	horizontale	10	0.1	307	256	51

Pour ces images 512x512 on a :

$$\omega_a = 102 / 512 = 0.2$$

$$\omega_c = 102 / 512 = 0.2$$

$$\omega_d = 51 / 512 = 0.1$$

$$\omega_f = 51 / 512 = 0.1$$

On se rend bien compte à la fin que la taille de l'image est en rapport avec ω . On a la rapport suivant :

$$\omega = \frac{D}{\sqrt{\text{taille_image}}}$$

1.7 Classification de toutes les images en trois classes : horizontale, verticale, diagonale et calcule de la fréquence spatiale pour chaque image

Afin de classer ces images il suffit de regarder les différence sur la position de la raie principale et la raie secondaire. Si pour une image I c'est le x qui varie alors il faut placé I dans la classe horizontale. Si c'est le y qui varie alors il faut placé I dans la classe verticale. Et si c'est x et y qui varie alors il faut placé I dans la classe diagonale.

Et ensuite pour calculer la fréquence spatiale il suffit d'utiliser la formule $\omega = \frac{D}{\sqrt{\text{taille_image}}}$.

Voici le code correspondant :

```
macro "direction FFT" {

    // The arrays to classify the texture of the images
    classe_horizontale = newArray();
    classe_verticale = newArray();
    classe_diagonale = newArray();

    // The arrays containing the spatial frequencies
    classe_horizontale_freq = newArray();
    classe_verticale_freq = newArray();
    classe_diagonale_freq = newArray();

    // The current image name construction
    image_name_letter = "";
    image_name_size = ""
    image_name = "";

    // For all of the images
    for (k=1; k<19; k++)
    {
        // Getting the size_name
        if (k <= 6 )      { image_name_size = "128_"; }
        else if (k<=12) { image_name_size = "256_"; }
        else              { image_name_size = "512_"; }

        // Getting the letter_name
        if (k % 6 == 0)    { image_name_letter = "f.jpg"; }
        else if (k % 6 == 5) { image_name_letter = "e.jpg"; }
        else if (k % 6 == 4) { image_name_letter = "d.jpg"; }
        else if (k % 6 == 3) { image_name_letter = "c.jpg"; }
        else if (k % 6 == 2) { image_name_letter = "b.jpg"; }
        else if (k % 6 == 1) { image_name_letter = "a.jpg"; }

        // Opening the image
        image_name = image_name_size + image_name_letter;
        open (image_name_size + image_name_letter);

        // récupération du ID de l'image
```

```

image = getImageID();

// application de la FFT
run("FFT", "title='FFT de l'image courante'");

// récupération de ID de la FFT : du plan de fourier
fourier = getImageID();

// Getting the size of the fourier plan
W = getWidth();
H = getHeight();

max_1 = 0;
i_max_1 = 0;
j_max_1 = 0;

max_2 = 0;
i_max_2 = 0;
j_max_2 = 0;

// Get the primary and secondary ray position
for (j=0; j<H; j++)
{
    for (i=0; i<W; i++)
    {
        p = getPixel(i,j);

        // get the primary ray
        if ( max_1 < p)
        {
            max_1 =p;
            i_max_1 = i;
            j_max_1 =j;
        }

        // get the secondary ray
        if ( max_2 < p && max_1 > p)
        {
            max_2 =p;
            i_max_2 = i;
            j_max_2 =j;
        }
    }
}

// Calculating the distance between the primary ray and the second one
distance = sqrt(pow(i_max_2 - i_max_1, 2) + pow(j_max_2 - j_max_1, 2));

// Calculating the spatial frequency of the current image

```

```

        omega = distance / sqrt(W*H);

        // store the name of the image into the right array for classification
        // and the spacial frequencies
        if (i_max_1 != i_max_2 && j_max_1 != j_max_2)
        {
            classe_diagonale = Array.concat(classe_diagonale, image_name);
            classe_diagonale_freq = Array.concat(classe_diagonale_freq, omega);
        }
        else if (i_max_1 != i_max_2)
        {
            classe_horizontale = Array.concat(classe_horizontale, image_name);
            classe_horizontale_freq = Array.concat(classe_horizontale_freq, omega);
        }
        else if (j_max_1 != j_max_2)
        {
            classe_verticale = Array.concat(classe_verticale, image_name);
            classe_verticale_freq = Array.concat(classe_verticale_freq, omega);
        }
    }

    // print the elements of the different classes
    print("classe_diagonale : ");
    Array.print(classe_diagonale);
    Array.print(classe_diagonale_freq);

    print("classe_horizontale : ");
    Array.print(classe_horizontale);
    Array.print(classe_horizontale_freq);

    print("classe_verticale : ");
    Array.print(classe_verticale);
    Array.print(classe_verticale_freq);
}

```