

RAPPORT DE TP - SY26

---

## TP05 - La compression vidéo

---

Rémi BURTIN

Cyril FOUGERAY

16 juin 2014



UNIVERSITÉ DE TECHNOLOGIE DE  
COMPIÈGNE

# 1 Introduction

Le but de ce TP est de mettre en oeuvre la technique du block matching utilisée dans certains algorithmes de compression vidéo.

## 2 Mise en oeuvre du block matching

### 2.1 Padding de l'image

La division de l'image en bloc implique que la largeur et la hauteur de l'image soient respectivement multiples de la largeur et de la hauteur des blocs. Si ce n'est pas le cas, nous complétons avec des zéros (pixels noirs) grâce à la fonction `padarray`. Pour calculer le nombre de pixels à rajouter en largeur on utilise la formule suivante :

$$(M - (\text{largeur}(\text{image}) \bmod M)) \bmod M$$

avec M largeur d'un bloc.



FIGURE 1 – Image issue de la video garden qu'on a voulu diviser en bloc de 7x7. L'image faisant 352x240, il a fallu rajouter 5 pixels noirs en largeur et 5 en hauteur.

### 2.2 Calcul de la fenêtre de recherche

### 2.3 Recherche du meilleur bloc

## 3 Résultats

## 4 Conclusion

## A Codes source MATLAB

### A.1 Calcul du MSD

```
1 function msd = compute_msd( current_block, block )
2     M = size(block,1);
3     N = size(block,2);
4     msd = 1/(M*N);
5     somme = 0;
6
7     for i=1:M,
8         for j=1:N,
9             somme = somme + (current_block(i,j)-block(i,j))^2;
10        end;
11    end;
12    msd = msd * somme;
13 end
```

## A.2 Calcul de la fenêtre de recherche

```
1 function [ window, orig_x, orig_y ] = search_window( img, W, M, i, j )
2     %i,j : indices dans l'image, du pixel en haut a gauche du bloc
3     %M : largeur/hauteur du bloc
4     %W : taille zone de recherche
5     y1 = i-W;
6     if y1 <= 0
7         y1 = 1;
8     end;
9     y2 = i + (M-1) + W;
10    if y2 > size(img,1)
11        y2 = size(img,1);
12    end;
13
14    x1 = j-W;
15    if x1 <= 0
16        x1 = 1;
17    end;
18    x2 = j + (M-1) + W;
19    if x2 > size(img,2)
20        x2 = size(img,2);
21    end;
22
23    %indices du bloc courant dans la fenetre de recherche (utile pour le
24    %calcul du vecteur de mouvement)
25    orig_y = (i - y1)+1;
26    orig_x = (j - x1)+1;
27
28    %recuperation de la fenetre de recherche
29    window = img(y1:y2,x1:x2);
30 end
```

### A.3 Recherche d'un bloc dans l'image de référence

```
1 function [delta_x,delta_y,error] = block_matching(current_block,search_window,orig_i,orig_j)
2     %initialisation du MSD minimum a Infini
3     min_msd = inf;
4
5     M = size(current_block,1);
6
7     for i=1:(size(search_window,1)- M) + 1
8         for j=1:(size(search_window,2)- M) +1
9             %recuperation d'un bloc dans la fenetre de recherche
10            block = search_window(i:(i+M)-1,j:(j+M)-1);
11
12            %calcul du msd entre le bloc courant et le bloc que nous venons
13            %de recuperer dans la fenetre de recherche
14            msd = compute_msd(current_block, block);
15
16            if msd < min_msd
17                %sauvegarde du msd
18                min_msd = msd;
19
20                %calcul du vecteur de mouvement
21                delta_y = i - orig_i;
22                delta_x = j - orig_j;
23
24                %calcul de l'erreur de prediction
25                error = current_block - block;
26            end;
27        end;
28    end;
29 end
```

## A.4 Block Matching

```
1 function [ time, output_image, error, msd ] = block_matching_encode( img_ref, img, N, W )
2     M = 2*N+1;
3
4     %Conversion en niveaux de gris
5     img_ref = rgb2gray(img_ref);
6     img = rgb2gray(img);
7
8     %padding de l'image (on ajoute des 0 pour que la largeur/hauteur soit
9     %multiple de M
10    pad_y = mod(size(img,1),M);
11    if pad_y > 0
12        pad_y = M - pad_y;
13    end;
14
15    pad_x = mod(size(img,2),M);
16    if pad_x > 0
17        pad_x = M - pad_x;
18    end;
19    imwrite(img,'garden.jpg','jpg');
20    img = double(padarray(img,[pad_y pad_x],0,'post'));
21    imwrite(img/255,'garden_padding.jpg','jpg');
22    img_ref = double(padarray(img_ref,[pad_y pad_x],0,'post'));
23
24    %end
```