

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

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
20    img = double(padarray(img,[pad_y pad_x],0,'post'));
21    img_ref = double(padarray(img_ref,[pad_y pad_x],0,'post'));
22
23    %affichage de la difference en tre l'image courante et l'image de
24    %reference
25    figure('name','Diff current vs ref');
26    imshow(((img-img_ref)+128)/255);
27
28    %Initialisation matrice contenant les composantes x et y des vecteurs
29    %de mouvement
30    matx = zeros(size(img,1)/M, size(img,2)/M);
31    maty = zeros(size(img,1)/M, size(img,2)/M);
32
33    %timer
34    tic
35
36    k=1;
37    for i=1:M:size(img,1),
38        l=1;
39        for j=1:M:size(img,2),
40            %extraction bloc courant
41            block = img(i:(i+M)-1,j:(j+M)-1);
42
43            %calcul de la fenetre de recherche
44            [window, orig_x, orig_y] = search_window(img_ref,W,M,i,j);
45
46            %recherche du bloc qui se rapproche le plus du bloc courant
47            %dans la fenetre de recherche
48            [delta_x, delta_y, error] = block_matching(block>window,orig_y,orig_x);
49
50            %on met le vecteur de mouvement dans des matrices (une pour la
51            %composante x et une autre pour la y)
52            matx(k,l) = delta_x;
53            maty(k,l) = delta_y;
```

```

54
55         l = l + 1;
56
57         %on recupere le bloc que nous avons trouve dans l'image de ref
58         %grace au vecteur de mouvement et le mettons a la place qu'il
59         %devrait avoir dans l'image courante
60         ref_block = img_ref(i+delta_y : i+M-1+delta_y , j+delta_x : j+M-1+delta_x);
61         output_image(i:(i+M)-1,j:(j+M)-1) = ref_block;
62
63         end;
64         k = k + 1;
65     end;
66     time = toc;
67
68     %calcul et affichage de l'erreur de prediction
69     error = (img - output_image)+128;
70     figure('name',strcat('Erreur de prediction n=',num2str(N),' w=', num2str(W)));
71     imshow(error/255);
72
73     msd = compute_msd(img, output_image);
74
75     %champ de vecteur
76     figure('name',strcat('Vecteurs de mouvement n=',num2str(N),' w=', num2str(W)));
77     quiver(matx/M,maty/M);
78     %inversion de l'axe des y
79     axis ij;
80
81
82     %affichage de l'image predite
83     figure('name',strcat('Image predite n=',num2str(N),' w=', num2str(W)));
84     imshow(output_image/255);
85 end

```