Video Signal Processing Assignment

LI WEI ZHEN G1701058D

1.

First apply row transform:

Since each row is the same:

$$\begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & -1 & 1 & -1 & 1 & -1 & 1 & -1 \\ 1 & 1 & -1 & -1 & 1 & 1 & -1 & -1 \\ 1 & -1 & -1 & 1 & 1 & -1 & -1 & 1 \\ 1 & 1 & 1 & 1 & -1 & -1 & -1 & -1 \\ 1 & -1 & 1 & -1 & -1 & 1 & -1 & 1 \\ 1 & 1 & -1 & -1 & -1 & -1 & 1 & 1 \\ 1 & -1 & -1 & 1 & -1 & 1 & 1 & -1 \end{bmatrix} \begin{bmatrix} 50 \\ 40 \\ 25 \\ 20 \\ 10 \\ 10 \\ 5 \\ 1 \end{bmatrix} = \begin{bmatrix} 161 \\ 19 \\ 59 \\ 1 \\ 109 \\ 11 \\ 31 \\ 9 \end{bmatrix}$$

We derive:

$$\begin{bmatrix} 161 & 19 & 59 & 1 & 109 & 11 & 31 & 9 \\ 161 & 19 & 59 & 1 & 109 & 11 & 31 & 9 \\ 161 & 19 & 59 & 1 & 109 & 11 & 31 & 9 \\ 161 & 19 & 59 & 1 & 109 & 11 & 31 & 9 \\ 161 & 19 & 59 & 1 & 109 & 11 & 31 & 9 \\ 161 & 19 & 59 & 1 & 109 & 11 & 31 & 9 \\ 161 & 19 & 59 & 1 & 109 & 11 & 31 & 9 \\ 161 & 19 & 59 & 1 & 109 & 11 & 31 & 9 \end{bmatrix}$$

Then column transform:

$$\begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & -1 & 1 & -1 & 1 & -1 & 1 & -1 \\ 1 & 1 & -1 & -1 & 1 & 1 & -1 & -1 \\ 1 & -1 & -1 & 1 & 1 & -1 & -1 & 1 \\ 1 & 1 & 1 & 1 & -1 & -1 & -1 & -1 \\ 1 & -1 & 1 & -1 & -1 & 1 & -1 & 1 \\ 1 & 1 & -1 & -1 & -1 & -1 & 1 & 1 \\ 1 & -1 & -1 & 1 & -1 & 1 & 1 & -1 \end{bmatrix} \begin{bmatrix} 161 \\ 161 \\ 161 \\ 161 \\ 161 \\ 161 \\ 161 \\ 161 \end{bmatrix} = \begin{bmatrix} 1288 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}$$

And implement such transform to each column:

We get :

$$\begin{bmatrix} 1288 & 152 & 472 & 8 & 872 & 88 & 248 & 72 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

After applying the quantization, we derive:

$$\begin{bmatrix} 81 & 14 & 47 & 1 & 36 & 2 & 5 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

After zig-zag scanning, the result is:

[81,14,0,0,0,47,1,0,0,0,0,0,0,0,36,2,0,0,0,0,0,0,0,0,0,0,0,5,1,0,0,0,0,0,0,0,

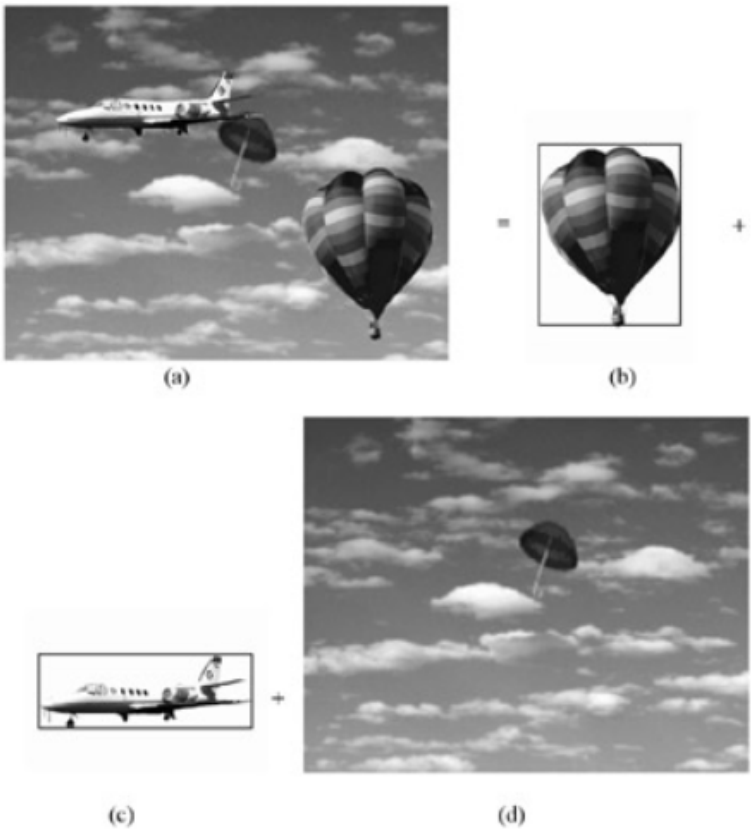
0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0]

We can use the run-length coded:

(0,81),(0,14),(3,47),(0,1),(7,36),(0,2),(11,5),(0,1),(0,0).

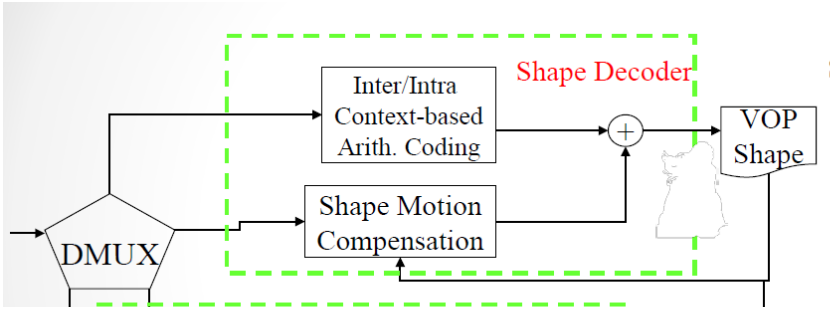
2.

In object-based coding the video frames are defined in terms of layers of video object planes. As is shown in the figure, it is assumed that the foreground object (the airplane and the balloon) can be segmented from the background and it can be extracted from the sequence prior to coding.



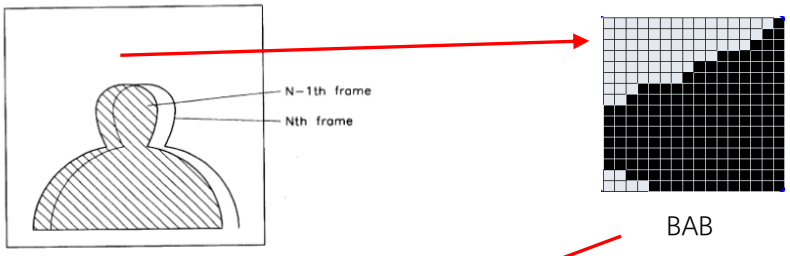
The receiver in fact receives 3 classes of information, i.e. shape texture and motion information. The shape decoder need to combine the data into the VOP shape. From its coding process we know that For the intra frame and inter frames, the encoder divides them into 16*16 binary alpha blocks (BAB)where pixel value is either 0 or 255 and applies context-based arithmetic encoding(CAE) to each of them.

So when the decoder receives the first intra frame, it decodes the CAE and combine all the blocks to derive the target VOP shape. When the following inter frames come, it will not only decode CAE to derive the new shape but also use the previous reference frame as well as received shape motion vectors to do the shape motion compensation, then the decoder adds them together to get the final new VOP shape, the flow diagram is as shown below:



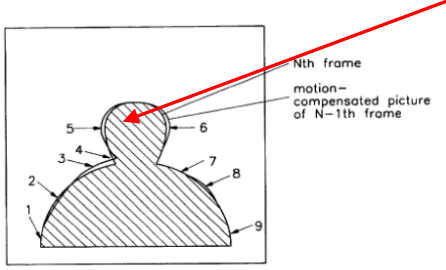
Take the following figure for an example: the shaded area is seen as the reference frame. The encoder will divide it into BABs and use CAE to code it. The decoder then uses inverse operation to reconstruct the VOP shape. For the following frame (the blank shape), it' s almost the same but extra use the reference frame and motion compensation to derive the new VOP shape.

FIG.5



BAB

FIG.6



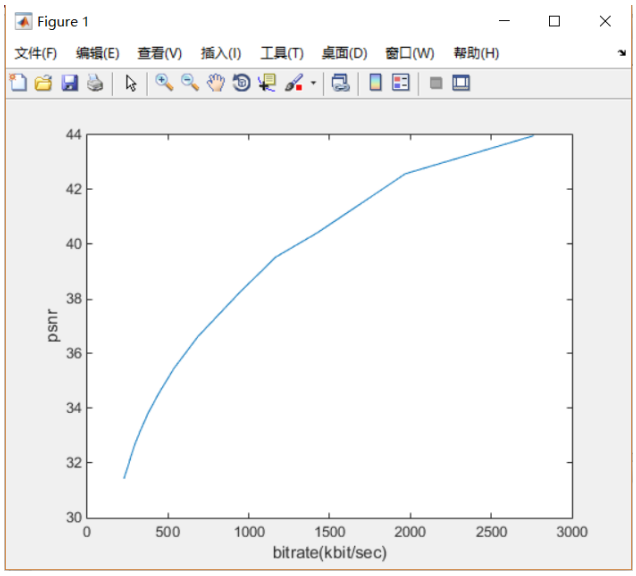
Combined with motion vectors (->)

3

a) Do 12 sets of experiments and the result is :

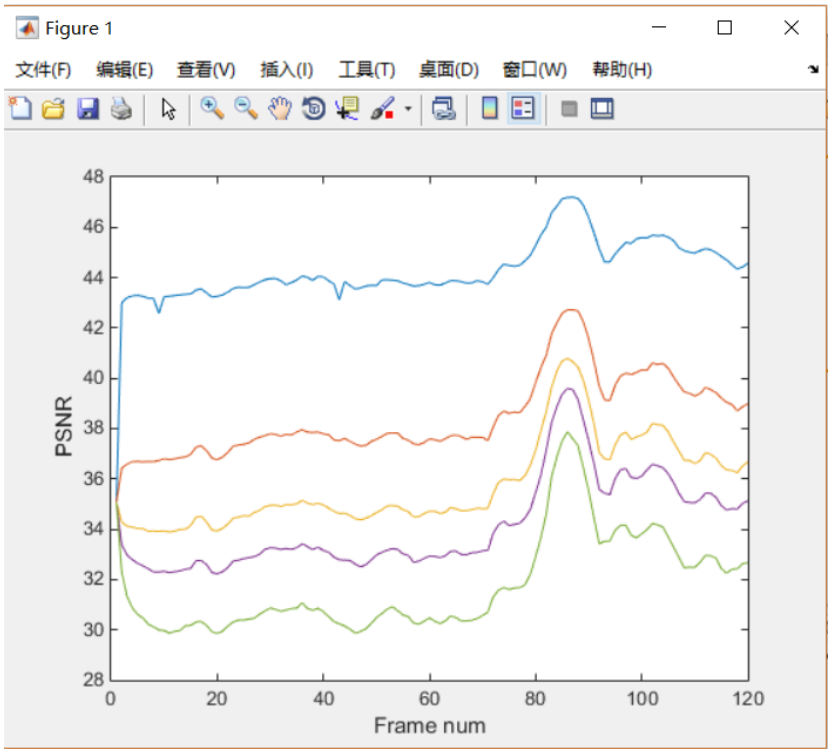
Q	Bitrate	PSNR
2	2762.68	43.9711
3	1968.28	42.5799
4	1425.20	40.4281
5	1164.47	39.5284
6	933.93	38.1855
8	685.21	36.6329
10	537.35	35.4744
12	442.30	34.5508
14	376.27	33.8281
16	328.31	33.1851
18	292.74	32.6693
24	227.08	227.08

Then I use it to plot the PSNR against various bitrate:



It can be clearly seen from the figure that when bitrate becomes larger, the psnr will get larger accordingly. It is reasonable because when quantization parameter becomes bigger, the original video was deeper compressed and the bitrate accordingly becomes small because it represents the amount of data in every second of compressed video. However, it leads to the loss of information and enhance the noise. As a result, the psnr gets smaller.

b)



The bitrate(kbit/sec) of the five curve are namely：blue: 2762.68; orange:933.93; yellow:537.35; purple:376.27; green:227.08.

We can see from the figure that high bitrate always achieves high PSNR in all the frames and they in general share the same trend. We can also see that there is a peak in all the curves. I think the reason is that the two adjacent frames are similar and the motion vectors needed to be encoded is much fewer than other frames which leads to less noise introduced . As a result, the reconstructed frame achieves a higher PSNR.

4.

Since my name is “LI WEI ZHEN”, I need to encode” LI WEI ZHE”

So I list the table as follows:

Character	Probability	Interval(range)
space	2/10	[0.00-0.20)

L	1/10	[0.20-0.30)
I	2/10	[0.30-0.50)
W	1/10	[0.50-0.60)
E	2/10	[0.60-0.80)
Z	1/10	[0.80-0.90)
H	1/10	[0.90-1.00)

Encode:

New character	Low value	High value
	0	1
L	0.2	0.3
I	0.23	0.25
SPACE	0.230	0.234
W	0.2320	0.2324
E	0.23224	0.23232
I	0.232264	0.232280
SPACE	0.2322640	0.2322672
Z	0.23226656	0.23226688
H	0.232266848	0.232266880
E	0.2322668672	0.2322668736

Thus it has been coded as: 0.2322668672

5.

The target is " LI WEI ZHEN G1701058D" and its ASCII codes for the first 16 values are

"76,73,32,87,69,73,32,90,72,69,78,32,71,49,55,48"

76	73	32	87
69	73	32	90
72	69	78	32
71	49	55	48

a) First stage:

2D Haar Wavelet transform can be divided into row transform and column transform separately：

Firstly row transform:

75	60	2	-28
71	61	-2	-29
71	55	2	23
60	52	11	4

Then column transform:

73	61	0	-29

66	54	7	14
2	-1	2	1
6	2	-5	10

So the output is as shown above.

b) Second stage:

Still, we use row transform first to the left-top 4 elements:

67	6	0	-29
60	6	7	14
2	-1	2	1
6	2	-5	10

And then column transform:

64	6	0	-29
4	0	7	14
2	-1	2	1
6	2	-5	10

So the output of second stage is shown above.