CMPT365 Multimedia – Moving Sprite

Due: 11:59 pm, Mar 15(Wed), 2017

Note: You may complete this assignment with a 2-person team or by yourself. We will use code scan tools to ensure that the programs are written by yourself.

Create two programs that one can read two image files, merge them and save them with compression applied. Another program should read your saved file and display it.

Your first program should first show an open file dialog box for picking up the two image files (one scene, one sprite). Then you can design your code to merge the images into one as the attached Sprite Animation showing, then display it.

After you created this image, your program should be able to compress and save the image into *.mrg format. The compression steps could follow Jpeg compression, you can use third-party libraries individually (e.g. DCT, Run-Length coding, but can't use the whole Jpeg encoder).

Your another program needs to read *.mrg file, decompress it, and display it.

For summary report, you should explicitly list out all the steps you have done, the third-party modules you have used, as well as your observation or thoughts through the assignment.

Regarding the submission, you should submit a zip file to course system (https://courses.cs.sfu.ca/) including your source code, an executable file of your program, a short manual of your program (with screen samples), and a summary report. If the file size is too large, please contact with TA

Grades:

1. File input (2 marks)
2. Display the images (original and merged) (4 marks)
3. Compression and decompression (9 marks) (Minimum Requirements)

Minimum Requirements:

1. Transform from RGB to YUV and do 4:2:0 chroma subsampling
2. Do 2D DCT/IDCT transformation
3. Do quantization with the quantization matrix
4. Scan the output matrix of quantization to make it into 1-D array, concatenate it with quantization matrix (scan it the same way), and save it into *.mrg file format
5. Read the *.mrg file, do all the reverse process, and display it

Optionally:

1. Have a scale value Q that can be used to generate different quantization matrix
2. Implement Zig-Zag scan
3. Do DPCM and RLC for DC and AC separately

4. Do the entropy coding

Recommendation:

You are recommended to use git to control your whole project, it will allow you to collaborate with your classmates easier, and can save you a lot of time when you messed up your codes. (tutorial: https://try.github.io/levels/1/challenges/1)

	Basic UI	Matrix Computation
C/C++	MFC/Opencv/QT	Opencv/Eigen
Java	AWT	Comman Math
Python	Opencv/TkInter	Numpy
Matlab	included	included

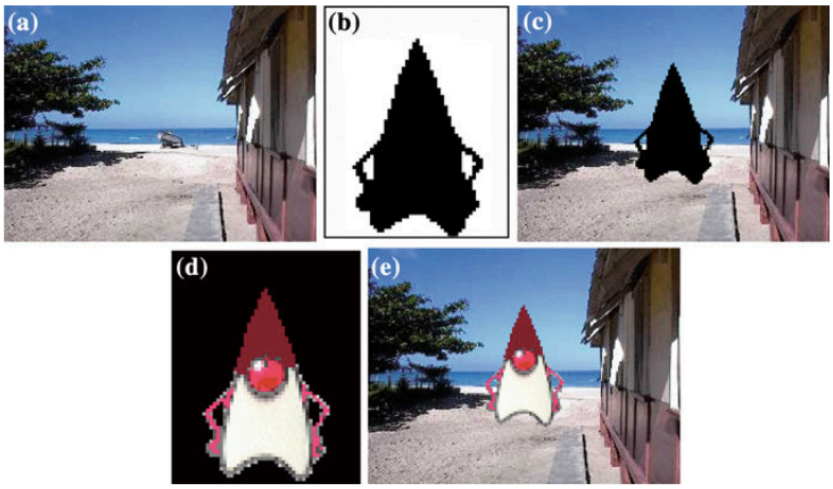

Fig. 2.5 Sprite animation: **a** Background B; **b** Mask M; **c** B and M; **d** Sprite S; **e** B and M or S

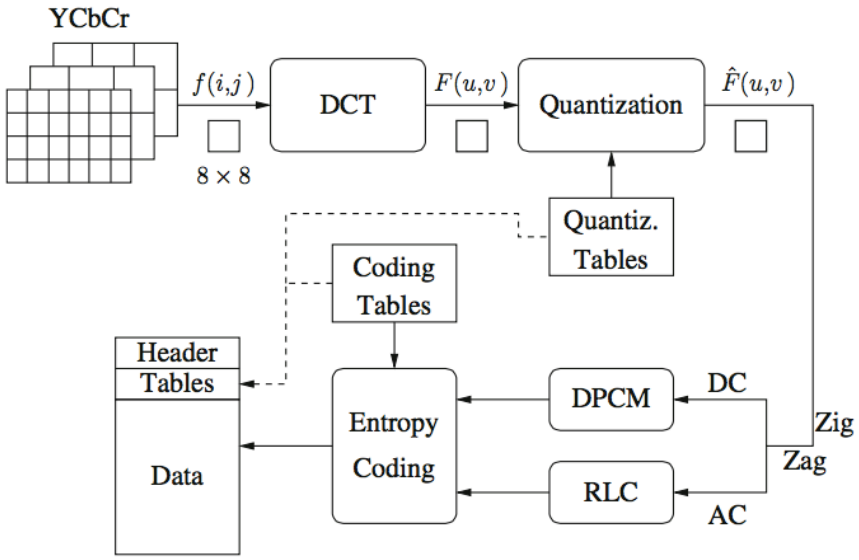

Fig. 9.1 Block diagram for JPEG encoder