JPEG Code Design

//read image file

1. readImage(filename , &imgHigh, &imgWidth)

if ( filename != ‘ \*.bat ’ )

error(we can not deal with file except .bat now.)

else

//.bat file do not include image’s size information. Set high and width to zeros as a flag.

//TODO : Risk , need main function to do some work about image file opening.

imgHigh =0 ; imgWidth=0;

datafile = openFile(filename)

\*\*imgData = readData(datafile)

Return imgData

1. cutDataIntoBlock(imgDdata, block, imgHigh, imgWidth, ithBlock, jthBlock)

assert ithBlock<=imgHigh/BLKSIZE

assert jthBlock<=imgWidth/BLKSIZE

for I from ithBlock\*BLKSIZE to ithBlock\*BLKSIZE + BLKSIZE

for j from jthBlock\*BLKSIZE to jthBlock\*BLKSIZE + BLKSIZE

if(i>imgHigh or j > imgWidth )

block[I - ithBlock\*BLKSIZE][ j - jthBlock\*BLKSIZE] = 0

else

block[I - ithBlock\*BLKSIZE][ j - jthBlock\*BLKSIZE] = imgData[i][j]

1. DCT 使用已有的代码
2. Quantizaion(DCTBlock)

//quantization coefficient should be specified explicitly in code.

Coefficient = [ … ]

For I,j in everypoint:

DCTBlock[i][j] = DCTBlock[i][j]/coefficient[i][j] + 0.5

1. Zigzag(DCTBlock)

//zigzag order can store as two vectors.

zigzagX = { … }

zigzagY = { … }

for I in BLKSIZE\*BLKSIZE:

zigzagList.push\_back(zigzagX[i],zigzagY[i])

//it point to last nonzero number’s next in the list.

if DCTBlock[zigzagX[i],zigzagY[i]] ! = 0 :

it = zigzagList.end()

zigzagList.delete(it)

return zigzagList

1. RLE(zigzagList,prvsDC)

It = zigzagList.begin()

//deal with DC

sizeDC = (int)(log(abs(\*it - prvsDC))/log(2) )+ 1

//-1 indicate DC

RLEList.push\_back(RLEClass(-1,sizeDC,\*it-prvsDC))

it ++;

//deal with AC

for it iterate:

if \*it==0:

ZeroCount++

//we should cutoff zeros list if it is too long.

If zeroCount>=zeroMaxLength:

FLEList.push\_back(RLEClass(zeroMaxlength,0,0)

zeroCount=0

Break;

Else

sizeAC = (int)(log(abs(\*it))/log2) + 1

FLEList.push\_back(RLEClass(zeroCount,sizeAC,\*it)

zeroCount =0;

//indicate the end of list

RLEList.push\_back(RLEClass(0,0,0));

Return RLEList

1. enCode(RLEList)

//deal with DC

It = RLEList.begin()

S += DCCodeTable[\*it.second]

S += valueCode(\*it.third)

For it iterate:

S += ACTable[\*it.first][\*it.second]

S += valueCode(\*it.third)

Return s

1. ConvertBit(s)

//TODO : 每8个连续的1要插入八个0

//每8个字符转化为一个byte，输入文件