Epitome Transform Coding

：Towards Joint Compression of a Set of Images

Changjiang Cai

https://www.changjiangcai.com/

Nov 3, 2016

Introduction

- What do traditional compression techniques do?

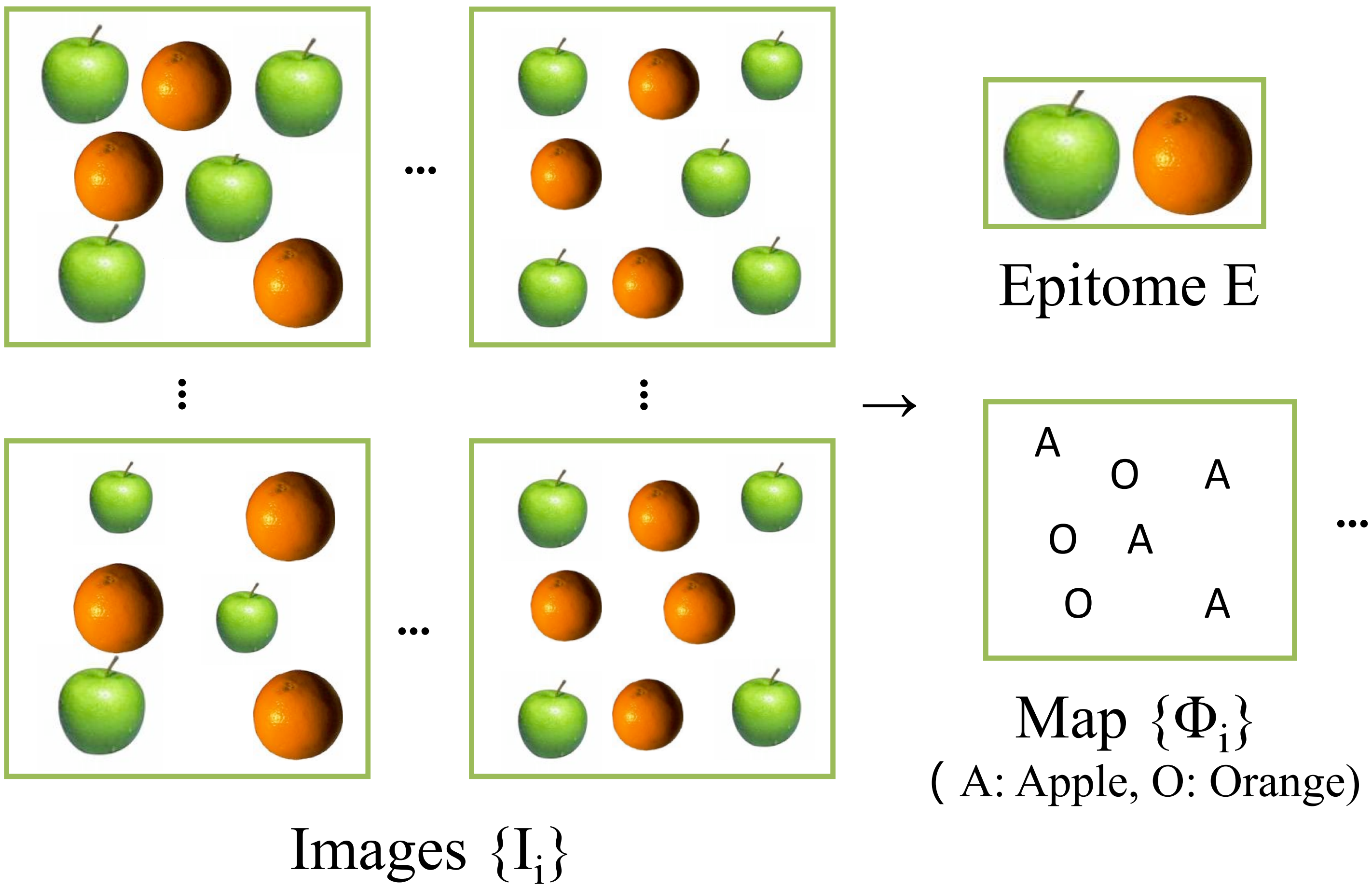
- JPEG and JPEG2000: compress a single image, by exploiting the local self-similarity within one image.

Introduction

- What does our approach do?

- Ours: compresses a set of images, by using the similarity and repetition ***within*** and ***among*** them.

Introduction

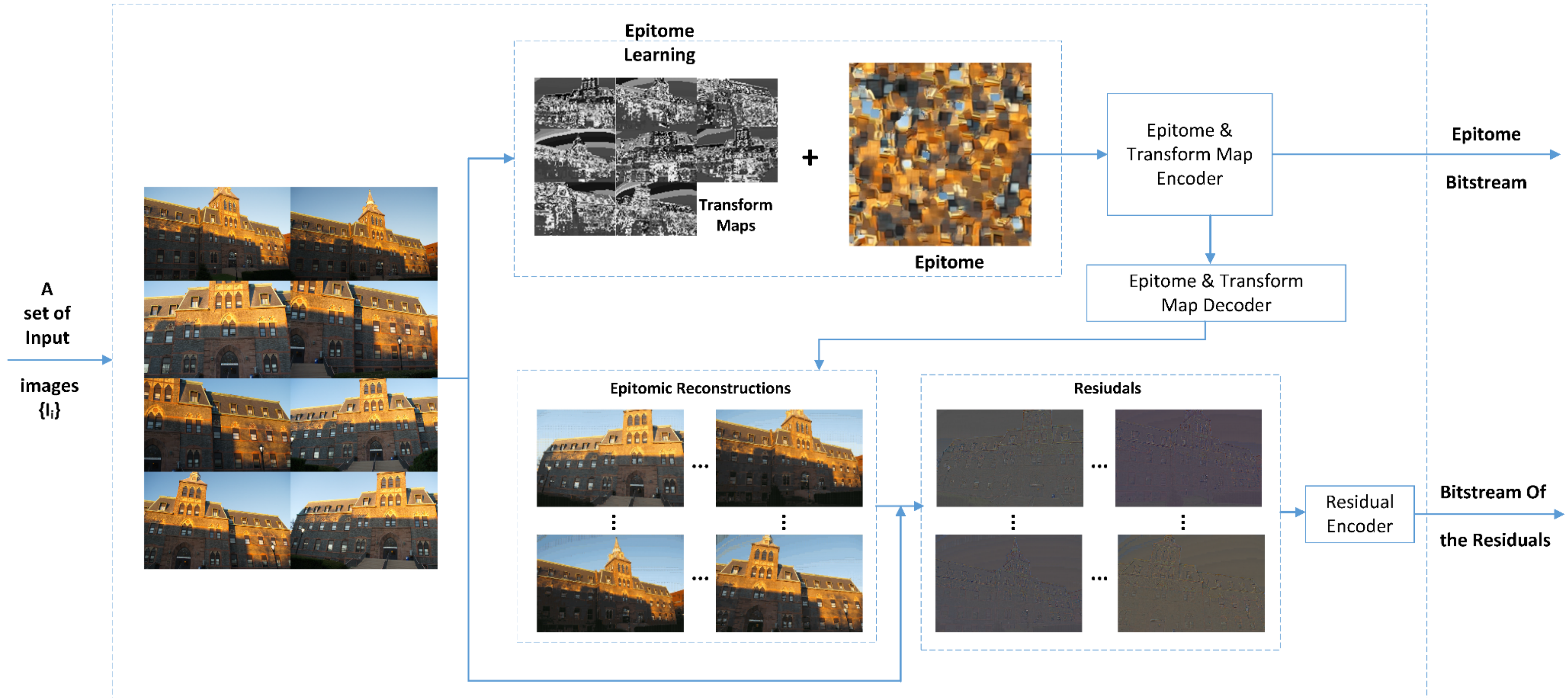


Courtesy of Huamin Wang

Our Approach: Epitome Transform Coding

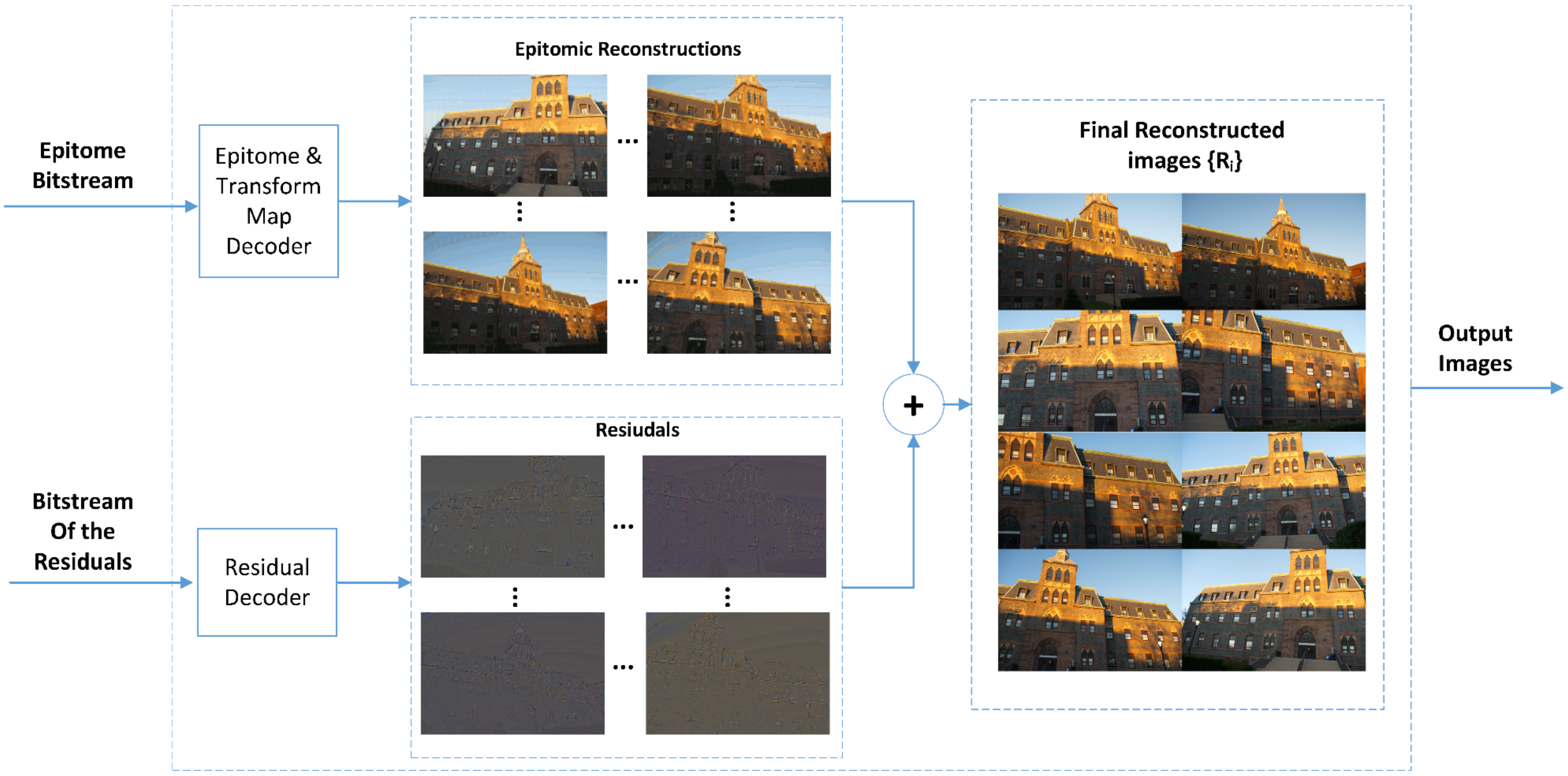
- Overall Framework of Our Approach
- Epitome Learning and Reconstruction
- Epitome Encoding
- Transform Map Encoding
- Residual Processing and Encoding

Overall Workflow of Our Approach (1)



(1)

Overall Workflow of Our Approach (2)



(2)

Overall Workflow of Our Approach (3)

- Learning the epitome E of a collection of images $\{I_i\}$, and doing the reconstruction via E and the associated transform map $\{\Phi_i\}$.
- Then the bitstream of entropy-encoded epitome, transform maps, and residuals, can be transmitted with bandwidth saving and economic storage.
- The transmitted bitstream will be decoded for the final rendering.

What is the Epitome? (1)

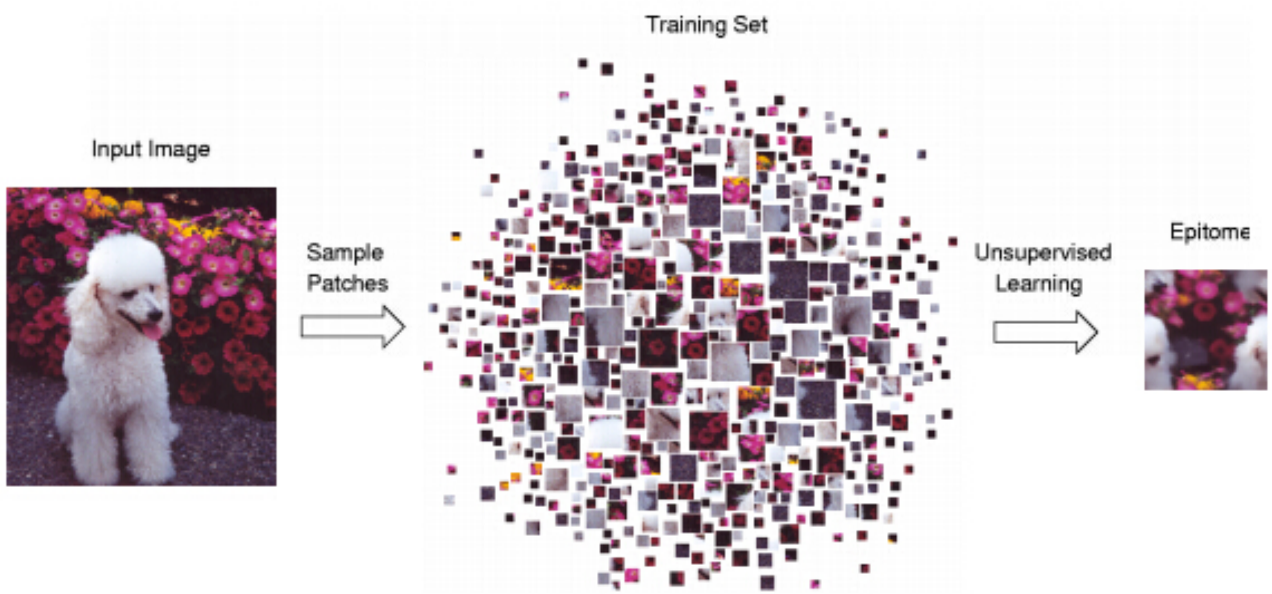
- An epitome is a condensed image of parameters that specifies a generative model of patches taken from input images.

- It contains high-order statistics of the texture and shape properties of the input images.

- A $W_e \times H_e$ epitome can be viewed as a mixture of $W_e \times H_e$ Gaussians.

What is the Epitome? (2)

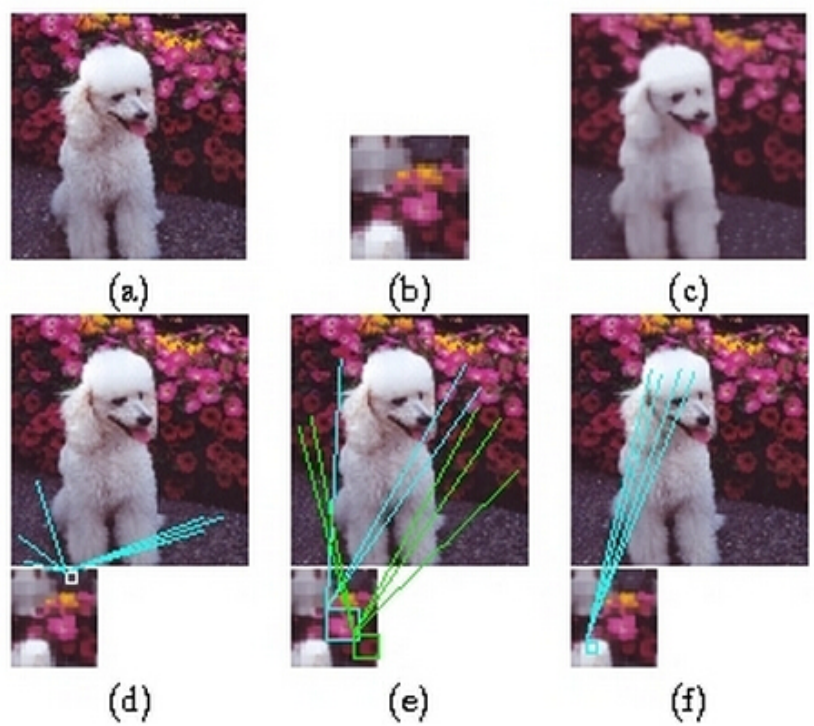
- The epitome is learned so that if small patches are sampled in an unordered fashion from it, they will have nearly the same appearance as patches sampled from the original input image.



Courtesy of Brendan J. Frey and Nebojsa Jojic

What is the Epitome? (3)

- An image (a) is epitomized in the texture (b). (c) is the reconstructed image using the mappings that map a patch in (a) to a patch in (b). In (d) to (f), some of the learned mappings are shown.



Courtesy of Brendan J. Frey and Nebojsa Jojic

Epitome Learning and Reconstruction (1)

- Given a set of image patches $\{X_i\}$, EM algorithm iteratively maximizes the log-probability:

$$L(E) = \sum_{i=1}^{P} \log\left(\sum_{\Phi_j \in \Phi} \rho(\Phi_j) p(X_i | \Phi_j, E) \right)$$

- It describes that $\{X_i\}$ were generated from the epitome E, based on the posterior distribution of transform mapping Φ_j

$$p(\Phi_j | X_i, E) = \frac{\rho(\Phi_j) p(X_i | \Phi_j, E)}{\sum_{\Phi_j \in \Phi} \rho(\Phi_j) p(X_i | \Phi_j, E)}$$

- The posterior will be calculated in the current E step of EM, and be used for the next EM iteration, illustrated in the following table.

Epitome Learning and Reconstruction (2)

EM algorithm for epitome learning and image reconstruction.

Input: a number of patches $\{X_i\}_{i=1}^{P}$ extracted out of the image set $\{I_n\}_{n=1}^{N}$

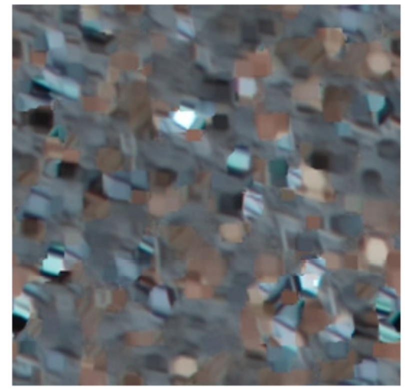
Output: a condensed $H_e \times W_e$ epitome E, composed of mixture of $H_e W_e$ Gaussian components, and the associated transform map $\Phi = \{\Phi_i\}_{i=1}^{P}$.

01: **Initialization:** set the variances as 1s', and the means was randomized with Gaussian noise, whose mean and variance is determined by the mean and the standard deviation of all pixel values in the same channel of the images.

02: **for** 10 times EM iteration
03:     **for** n = 1 : N, i.e., each image I_n
04:         **for** i = 1 : P, i.e., each image patch X_i
05:             calculate posterior for each channel
06:             find Φ_i based on maximal posterior
07:         **end for**
08:         to sum up posterior through 3 channels
09:         to normalize posterior
10:         to accumulate posterior information
11:     **end for**
12:     update epitome E for next EM iteration
13: **end EM**
14: Image reconstruction based on E and Φ

Epitome Learning and Reconstruction (3)

Input Image, 1024 X 768



Epitome, 256 X 256

Epitome Reconstruction-4

Epitome Reconstruction-6

Epitome Reconstruction-8

Epitome Encoding (1)

- For accuracy calculation, all parameters are represented as 64-bit floating numbers.

- In order to reduce the overhead of our approach, the learned epitome is encoded as 8-bit integers and saved as *JPEG and/or PNG* images.

Epitome Encoding (2)

- Obtains small file size and good reconstruction quality evaluated in terms of PSNR.

epitome size	64-bit double		8-bit integer	
	YML file size/KB	PSNR /dB	JPEG file size/KB	PSNR /dB
64*64	330	28.6783	4	28.5643
128*128	1320	29.9399	14	29.771
256*256	5000	31.3099	58	31.0858

Transform Map Encoding

- The transform maps, consisting of (b) column indices, and (c) row indices, are spatially redundant and similar to the input image (a).

(a) Input 616 X 408

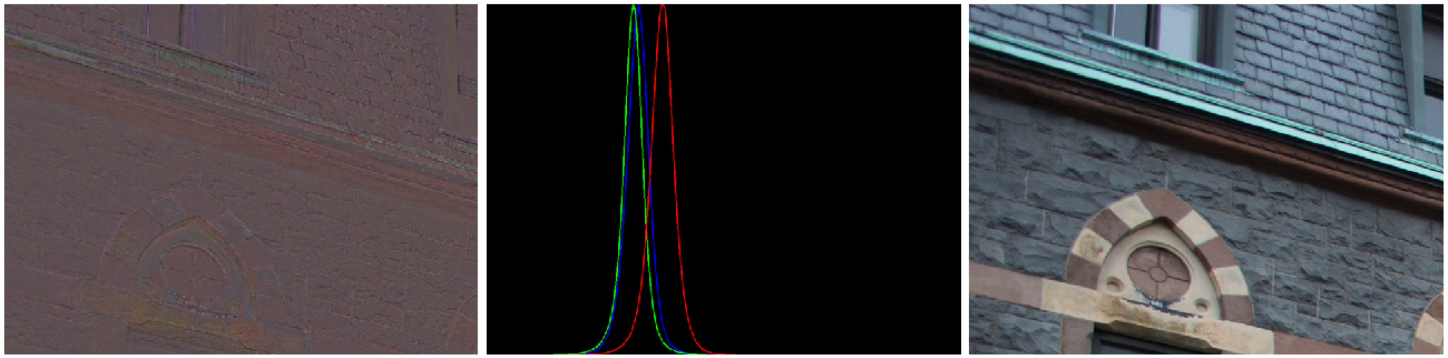
(b)77 X 51

(c)77 X 51

Residual Processing and Encoding (1)

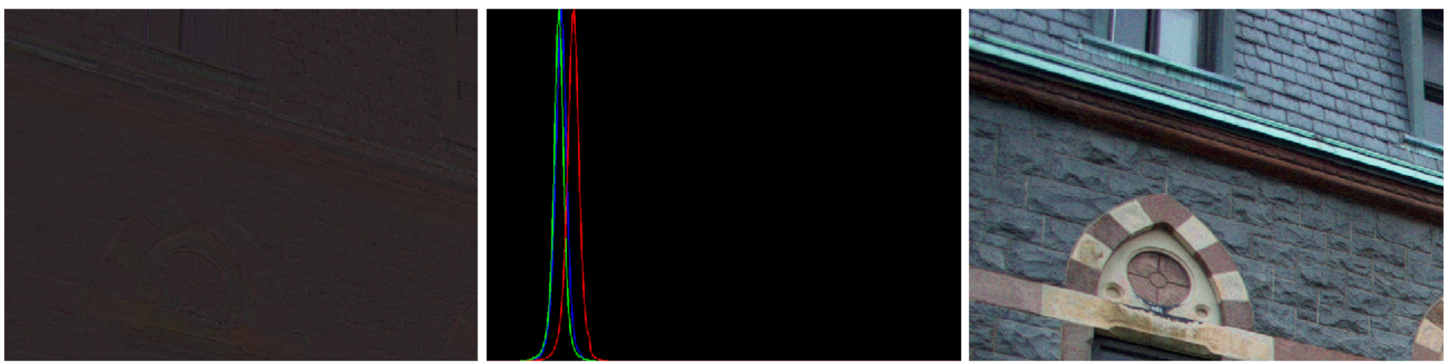
- Residual is the difference between the input images and the epitomic reconstructions.
- To encode and compress the residual is important to achieve large compression ration, as well as high reconstruction quality.
- Thresholding and quantizing are involved.

Residual Processing and Encoding (2)



Residual scaled into [0, 255] | Histogram ranged in [0, 255] | reconstruction

(a)



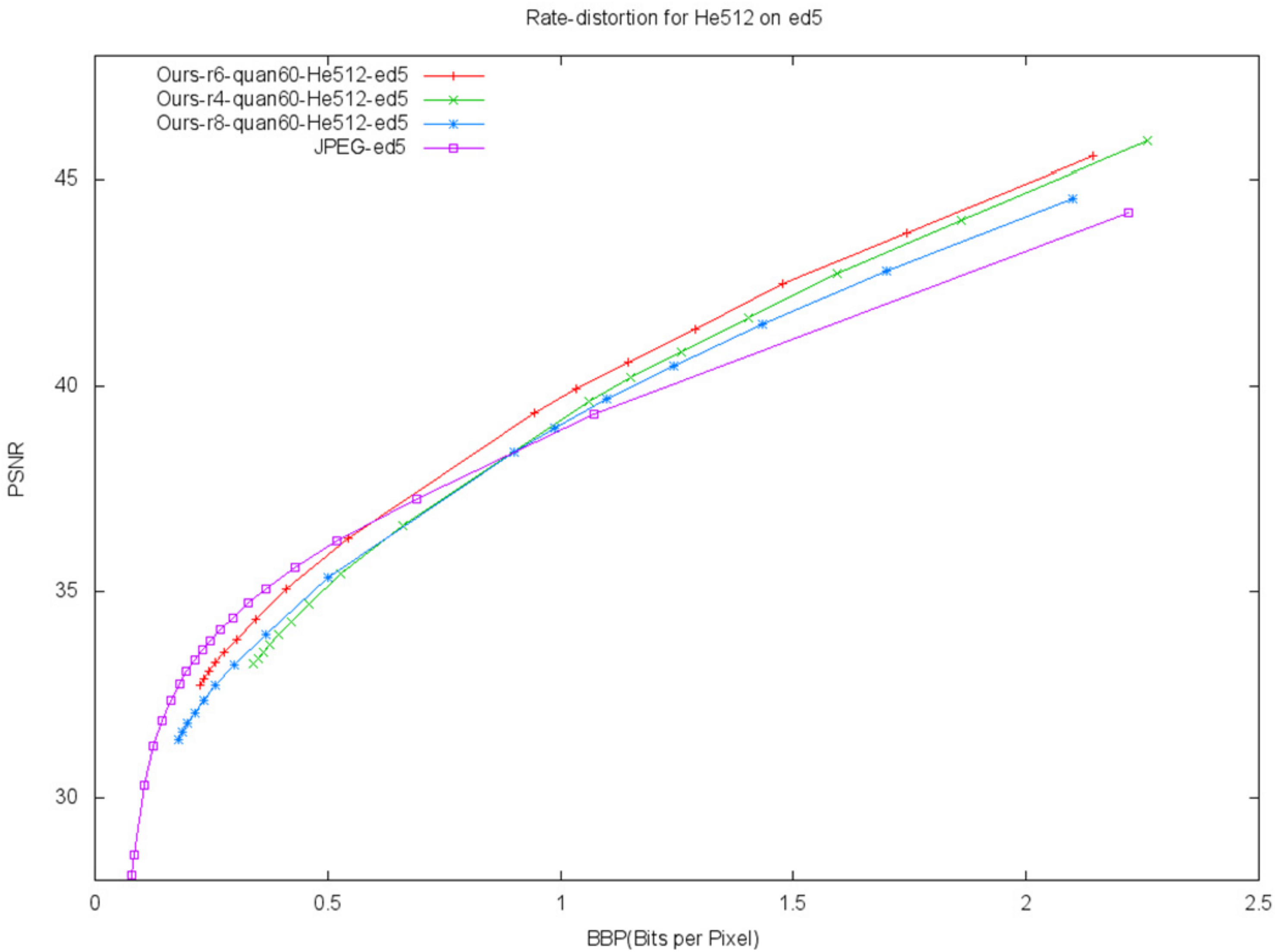
Residual after 50/100 quantization | Histogram after 50/100 quantization | reconstruction

(b)

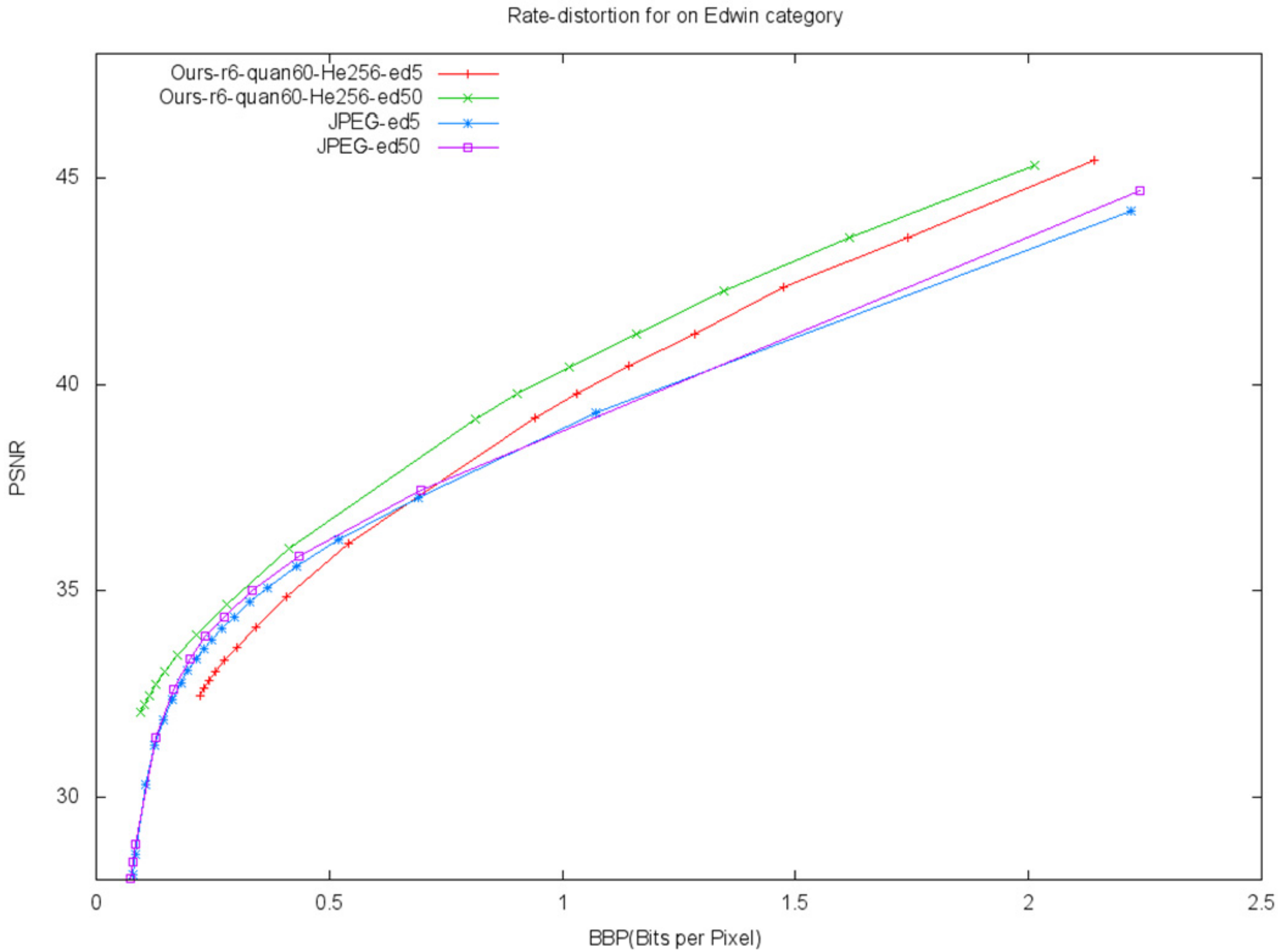
Experiments -- Dataset

pg ed flo hc esb

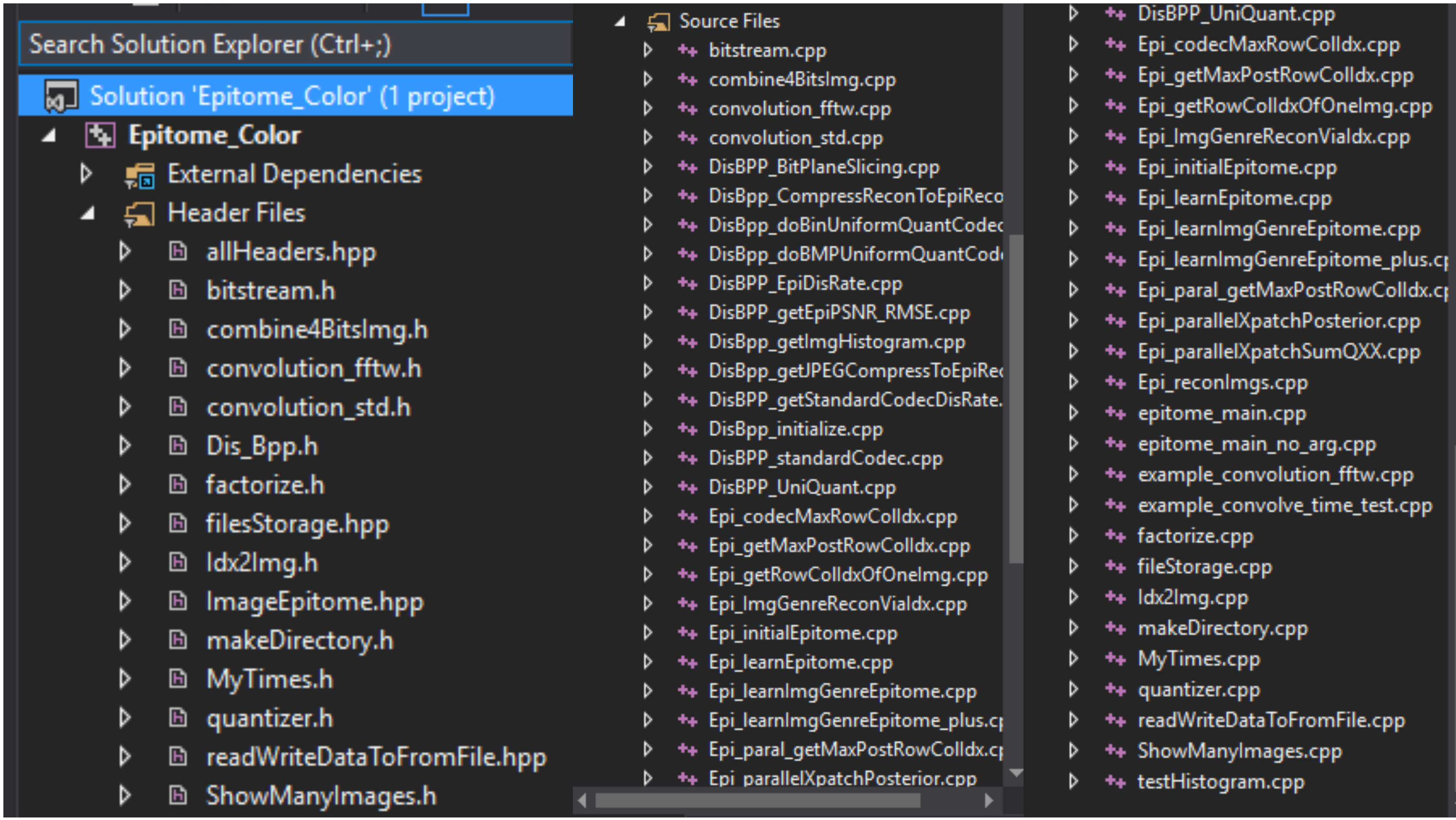
Experiments -- Result (1)



Experiments -- Result (2)



C++ Code


Search Solution Explorer (Ctrl+;)
Solution 'Epitome_Color' (1 project)
Epitome_Color
External Dependencies
Header Files
allHeaders.hpp
bitstream.h
combine4BitsImg.h
convolution_fftw.h
convolution_std.h
Dis_Bpp.h
factorize.h
filesStorage.hpp
Idx2Img.h
ImageEpitome.hpp
makeDirectory.h
MyTimes.h
quantizer.h
readWriteDataToFromFile.hpp
ShowManyImages.h
Source Files
bitstream.cpp
combine4BitsImg.cpp
convolution_fftw.cpp
convolution_std.cpp
DisBPP_BitPlaneSlicing.cpp
DisBpp_CompressReconToEpiReco
DisBpp_doBinUniformQuantCodec
DisBpp_doBMPUniformQuantCod
DisBPP_EpiDisRate.cpp
DisBPP_getEpiPSNR_RMSE.cpp
DisBpp_getImgHistogram.cpp
DisBpp_getJPEGCompressToEpiRe
DisBPP_getStandardCodecDisRate.
DisBpp_initialize.cpp
DisBPP_standardCodec.cpp
DisBPP_UniQuant.cpp
Epi_codecMaxRowColIdx.cpp
Epi_getMaxPostRowColIdx.cpp
Epi_getRowColIdxOfOneImg.cpp
Epi_ImgGenreReconViaIdx.cpp
Epi_initialEpitome.cpp
Epi_learnEpitome.cpp
Epi_learnImgGenreEpitome.cpp
Epi_learnImgGenreEpitome_plus.c
Epi_paral_getMaxPostRowColIdx.c
Epi_parallelXpatchPosterior.cpp
DisBPP_UniQuant.cpp
Epi_codecMaxRowColIdx.cpp
Epi_getMaxPostRowColIdx.cpp
Epi_getRowColIdxOfOneImg.cpp
Epi_ImgGenreReconViaIdx.cpp
Epi_initialEpitome.cpp
Epi_learnEpitome.cpp
Epi_learnImgGenreEpitome.cpp
Epi_learnImgGenreEpitome_plus.c
Epi_paral_getMaxPostRowColIdx.c
Epi_parallelXpatchPosterior.cpp
Epi_parallelXpatchSumQXX.cpp
Epi_reconImgs.cpp
epitome_main.cpp
epitome_main_no_arg.cpp
example_convolution_fftw.cpp
example_convolve_time_test.cpp
factorize.cpp
fileStorage.cpp
Idx2Img.cpp
makeDirectory.cpp
MyTimes.cpp
quantizer.cpp
readWriteDataToFromFile.cpp
ShowManyImages.cpp
testHistogram.cpp

Summary

- By exploiting the similarity and repetition of shape and texture within and among images, we have presented an effective multi-image compression approach, which works well at high compression rates for the images in our dataset, and even at low bit rate for some image categories in the dataset.

- Appropriate encoding and decoding of the epitome and transform maps, makes our approach outperform JPEG, when the compression bit rate is larger than 0.8-1.0 bits per pixel.

- When the number of images increase to some extend, the compression performance will be also improved.

Future Work

- Improve the epitomic reconstruction quality, by introducing more sophisticated transform which incorporates intensity variation and geometric deformation.
- Further efficiently encoding the transform mapping.
- Accelerate the epitome learning using GPU.
- Apply our approach to video compression.