Thousand to One: An Image Compression System via Cloud Search

Chen Zhao
Peking University
Beijing, China
zhaochen@pku.edu.cn

Siwei Ma
Peking University
Beijing, China
swma@pku.edu.cn

Wen Gao
Peking University
Beijing, China
wgao@pku.edu.cn

ABSTRACT

With the advent of the 'big data' era, a huge number of images are produced every day. Traditional image compression methods no longer satisfy the demand to store and transmit them. In this paper, we face this challenge and take advantage of the correlations existing between images to achieve a higher compression rate. We propose an image compression system that encodes each image by referencing its correlated images in the cloud. We first extract features from an image and retrieve its similar image from the massive images in the cloud by comparing these features. Then we preprocess the retrieved picture by applying projective transformation and illumination compensation to obtain multiple reference images of higher prediction accuracy. By taking advantage of the redundancy between the reference images and the current image, we encode the current image through prediction coding techniques. The experimental results demonstrate that the proposed method outperforms JPEG and HEVC intra coding by 61.5% and 21.3% on average, respectively. It has an average compression ratio of over a thousand to one.

Keywords

Image compression, big data, cloud-scale coding, and image retrieval

1. INTRODUCTION

In recent years, since smart phones and digital cameras are more and more popular, a huge number of images are acquired and stored. Consequently, people have higher and higher demand for the capacity of disks and other storage devices. The number of images is exploding so rapidly that we have to seek for other solutions to store them. To deal with the insufficiency of personal storage devices, cloud storage provided by some Internet companies is emerging these years, such as google drive, dropbox, rapidshare, Instagram, Baidu cloud disk, etc. But this just shifts the storage pressure from an individual to a company, and it cannot address the issue fundamentally. The explosion of data does not only put higher requirement for hardware, but is also a severe challenge for energy and labor cost.

In order to efficiently store massive images, we need to do image compression beforehand. Traditional image coding methods (such as JPEG, JPEG 2000), can remove the redundancy inside a picture. Since they only exploit the spatial correlations, however, the compression ratio cannot satisfy the storage requirement in the 'big data' era.

The big image data are not only a challenge for us, but provide opportunities or possibilities to further compress the images. Since the cost of taking a picture is reducing, people tend to take a couple of pictures at the same location or of the same objects. Although these pictures may have different illumination, shooting angles, etc., they may have high similarity after some processing. Besides the pictures in the same album, different people may take pictures at the same places at various time. If these pictures are uploaded to the cloud disk, the correlations between them can be exploited for compression.

Compressing images by utilizing the inter-image correlations is an effective way to further remove redundancy. Scholars did explorations on image set coding/album compression as earlier as in the 1990's and proposed many good methods [1]-[12]. These methods seek to find an efficient way to organize the images such that the image sequence can be encoded through inter-image prediction. There are still researchers nowadays working to compress image albums more efficiently. Most of these methods, however, do calculations to obtain the coding order for all the images in one entire album, which cannot directly apply to compressing cloud images. First, the number of images in the cloud is so huge that it is not realistic to compute an optimal coding order for the whole. Second, new pictures are produced and uploaded to the cloud every second, and consequently, we cannot acquire all the pictures for calculation at one time. Therefore, a new image compression system is to be designed for the current 'big data' scenario, in order to achieve compression with higher efficiency by utilizing cloud data.

Huanjing Yue etc. first proposed a cloud-based solution for image coding in [13]. In their solution, the encoder extracts image features and compresses the descriptors; the decoder reconstructs the images by searching for similar image patches in the cloud using the decoded descriptors. Since only the compressed features and downsampled original images are stored or transmitted, the bitstream is greatly reduced compared to traditional image coding methods. But this solution largely depends on the existence of similar image patches in the cloud --- once no similar images are found, the reconstruction quality would be quite low. Moreover, even if there exist similar images, the objective fidelity of the reconstructed picture to the original picture is even lower than JPEG.

In order to address the image compression issue in the 'big data' scenario, in this paper, we propose a novel image compression system via cloud search. For each newly uploaded picture, the system does compression by utilizing the massive images already existing in the cloud disk. Different from the above-mentioned image set coding that calculates the coding structure of all the images, this system adopts an 'encoding-one-image-immediately-after-it-is-uploaded' processing mechanism. Thus it ensures the feasibility in terms of processing time in the 'big data' scenario. Additionally, in contrast to the coding solution of [13], our system does image retrieval in the encoder rather than the decoder. This not only tremendously reduces decoding time, but also makes it possible to do intra-image coding in the absence of similar images, such that the objective fidelity is guaranteed.

978-1-4673-7478-1/15/$31.00 ©2015 IEEE

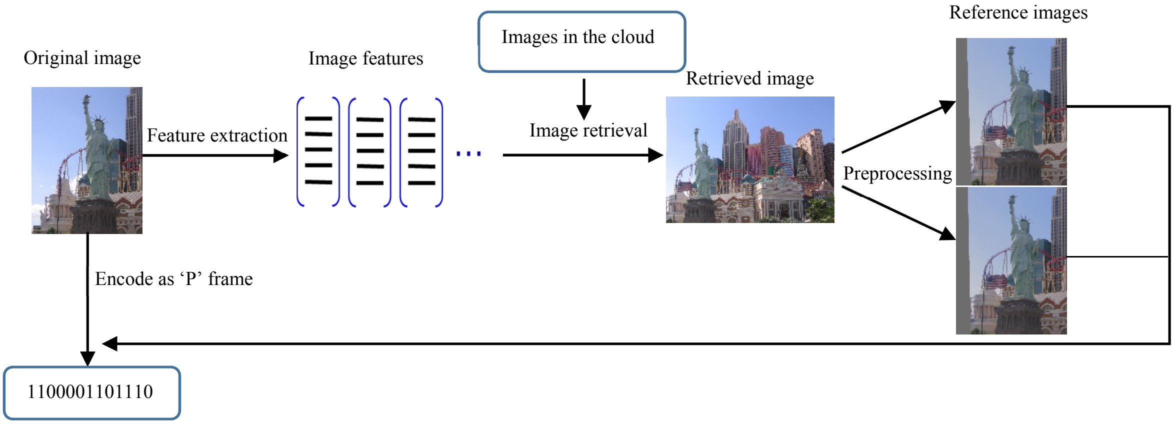


Figure 1. Framework of the proposed cloud-based image compression system

The remainder of this paper is organized as follows. Section 2 describes the proposed image compression system in detail; Section 3 provides the experimental results of the proposed system; Section 4 concludes the paper.

2. CLOUD BASED IMAGE COMPRESSION SYSTEM

We propose a cloud-based image compression system, which takes advantage of the massive images in the database and compresses images by exploiting inter-image correlations. An overview of the system framework is depicted in Fig. 1. The system considers compressing a newly uploaded image which is in the raw data format (if the image is already compressed using some other coding method such as JPEG, our system first decodes the image into raw data and then does the following compression steps).

We assume that there are already a number of images in the cloud--- the huge image database in which it is very probable to find a similar image for the newly uploaded image. Thus we can take advantage of the correlations between this similar image and the current image and compress it as a video 'P' frame. There are four main steps in this image compression system. First, we extract features from the original image and then use these image features to retrieve a similar image in the cloud. Third, we preprocess the retrieved image with different parameters to obtain a few (Fig. 1 depicts two) transformed images as reference for the current image. Finally, the current image is treated as a 'P' frame in a video sequence by referencing the two images and encoded using some video coding technologies, such as the H.265 standard.

In the following subsections, we discuss the key technologies in the four steps.

2.1 Feature extraction and image retrieval

In order to find the most correlated image from the cloud-scale image database, an efficient image distance metric should be considered. A lot of methods have been proposed in the literature [1], [11], [12] to evaluate image similarities for encoding an image set, but when the number of images exceeds a certain scale, these methods would be quite time-consuming.

We are inspired by the emerging mobile visual search methods and consider using the MPEG standardized compact descriptor for visual search (CDVS) [14], [15] as our image features and using image retrieval methods to find the correlated image. Fig. 2 illustrates the processing steps to produce a CDVS of an image. We can see from this pipeline that part of the SIFT features are selected to form local descriptors and global descriptors, which are compressed and aggregated respectively. Each CDVS is very small in size after compression, thus it saves a lot of time when comparing with other images features. At the same time, thanks to its global descriptor, it has as high accuracy as SIFT features.

After this step, we can retrieve multiple images from the cloud database. Then we select the top one as the candidate similar image

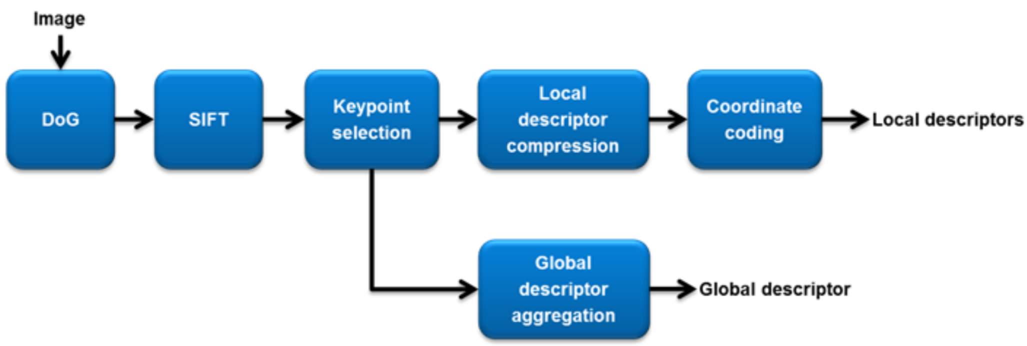


Figure 2. Compact descriptor extraction pipeline in CDVS TM [14]

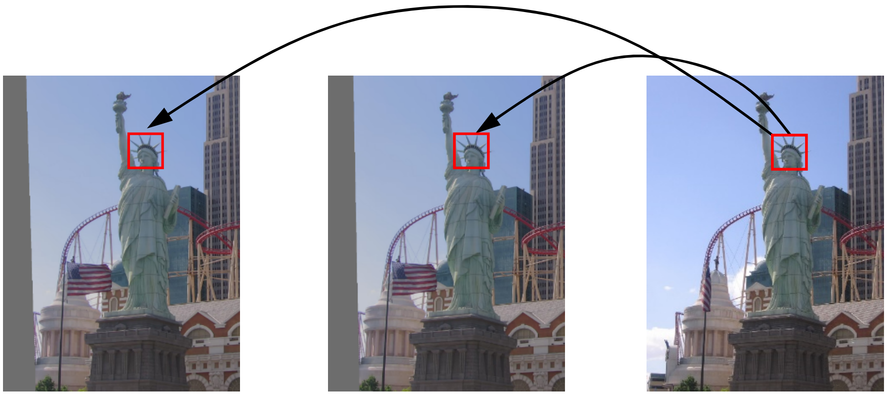


Figure 3. Predictive coding for the current image by block matching with the reference pictures

for the image to be encoded (it is also possible that there exist no similar image for the current image).

2.2 Preprocessing for the retrieved image

In a cloud-scale database, there is a high probability to find an image that has the same or similar content as the current image. But these two images may differ significantly in the block level, because they may be taken with different angles, focal length, exposure time, etc. The retrieved image might not be suitable to be directly utilized as a reference to predict the current image. Thus, before the coding step, we preprocess the retrieved image to obtain one or more images that are more similar to the current image in a lower level.

The preprocessing step includes geometric deformation and illumination compensation [1]. The geometric deformation is calculated using the matching feature points between the current image and the similar image. According to the principle of projective transformation [16], a transformation matrix, which contains information of image rotation, translation, scaling, etc., can be computed with every four pairs of matching features. Considering that there are usually many more matching points between two similar images, multiple transformation matrix can be obtained. We choose the optimal transformation matrix by solving an optimization problem, which can be formulated as the following energy function

$$E = d + \eta \times s, \tag{1}$$

in which, d is the data term, which represents the distance between the current image and the retrieved image after transformation using a certain matrix; s is the smoothness term, which represents the connectivity between the feature points of the current image ---- connectivity is formulated using Delaunay triangulation of the feature points; the weighting parameter η trades off the contributions of the two terms. We solve Eq. (1) through graph cut to obtain the best N solutions, and accordingly, we get N transformation matrices. N determines the number of reference pictures for the current image and it is usually a small number.

Using the calculated matrices, we transform the retrieved similar image to obtain N pictures, which are further processed through illumination compensation to produce N reference pictures.

We do illumination compensation in order to further make the reference pictures as close as the current image. It is formulated using the following equation

$$I'' = \alpha I' + \beta, \tag{2}$$

where I' represents the deformed image; I'' is the image after illumination compensation; α and β are illumination parameters, the estimation of which is obtained by solving the following optimization problem

$$\min \ I - (\alpha I' + \beta), \tag{3}$$

in which I represents the current image. We use the pixel values of the matching feature points of the images I and I' for calculation of this equation. Through solving a partial differential equation, the values of the parameters α and β are obtained. Then according to Eq. (2), a picture after illumination compensation is obtained for each deformed picture.

After the preprocessing, N pictures are obtained as the reference pictures for the current image (in Fig. 1, two reference pictures are shown as an example).

2.3 Encode the current image as 'P' frame

We encode the current image by block matching with the N reference pictures obtained from the previous step, as shown in Fig. 3. Concretely, we utilize the inter-frame predictive coding in video coding standards (such as HEVC), and treat the current image as a 'P' frame with the N reference pictures.

Meanwhile, we also consider the case where there exists no similar image in the cloud disk. If the retrieved image differs from the current image in content, they are not possible to be close in the pixel level even after preprocessing. In this case, if we still use inter-image prediction coding, the overhead may be larger than the saved bits. Thus we utilize the mode decision mechanism based on rate-distortion optimization (RDO) so that the encoder adaptively determines whether to apply inter-image coding or intra-image coding according to the rate-distortion cost (RD Cost, as shown in Eq. (4)). When the RD cost of inter-image coding is higher, the intra-image coding is applied.

$$\min J = D + \lambda R. \tag{4}$$

In Eq. (4), J is RD Cost; λ is the Lagrange parameter; D and R represent the distortion and consumed bits under a certain coding mode. The encoder selects the mode that has the smallest RD Cost for the current image block.

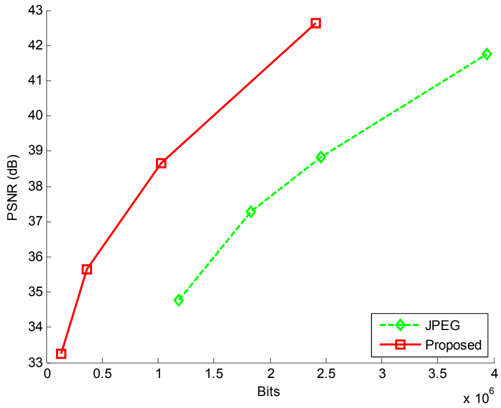


Figure 4. RD performance comparison between the proposed algorithm and JPEG

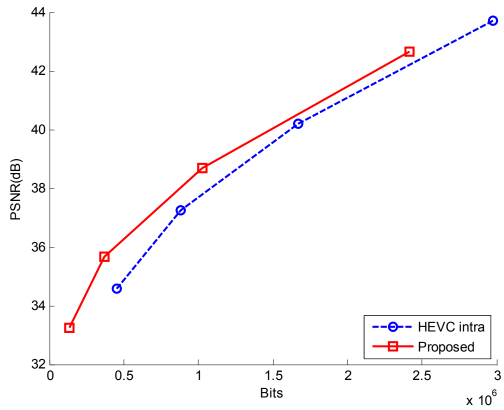


Figure 5. RD performance comparison between the proposed algorithm and HEVC intra

3. EXPERIMENTAL RESULTS

In order to verify the effectiveness of the proposed image compression system, we do extensive experiments using the ‘INRIA Holidays’ image set [17]. In the experiment, we select 126 test images from the ‘query’ images and use the other images as the cloud database images.

For all the test images, we encode them using the proposed system, JPEG and HEVC intra (encode the current frame using HEVC intra predictive coding techniques). Fig. 4 demonstrates the average rate-distortion performance comparison of the proposed system and JPEG; Fig. 5 demonstrates the average rate-distortion performance comparison of the proposed system and HEVC intra. We can see that the proposed system outperforms JEPG by a large scale, and that it is obviously better even than the highly-efficient intra coding of the state-of-the-art video coding standard HEVC. It is demonstrated that compared to JPEG, the proposed algorithm saves up to 96.2% bits and 61.5% bits on average; that compared to HEVC intra, it saves up to 84.7% bits and 21.3% bits on average.

Fig. 6 illustrates the bdrate performance [18] of the proposed algorithm compared to HEVC intra. Negative values of bdrate represent performance gain and positive values represent performance loss. It is shown that for all the test images, the

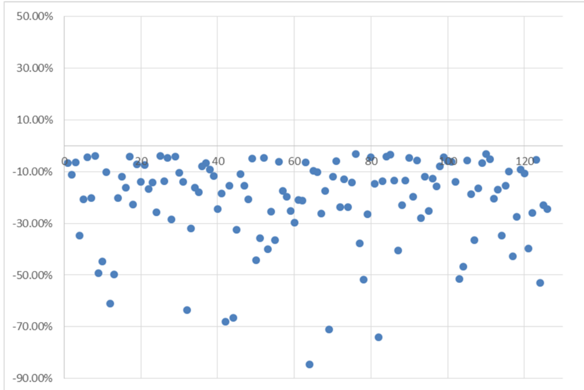


Figure 6. The bdrate performance of the proposed algorithm versus HEVC intra. Each point represents an image. The x-axis is the image order and the y-axis is the bdrate for each test image.

proposed algorithm has higher performance than HEVC intra and for many images, it has very large gain.

Fig. 7 illustrates the subjective quality of the reconstructed image by the proposed algorithm and the two comparative methods. For a fair comparison, we make the bitrates of all methods as close as possible. The bits per pixel (bpp) of the proposed algorithm and HEVC intra is of the level of 0.01; the bpp of JPEG is set 0.1 since the reconstruction quality is quite poor even at bpp = 0.1 and it is not necessary to use even lower bitrates for comparison. We can see from the figure that the proposed algorithm not only achieves the highest PSNR score, but also has the best subjective quality of reconstructed image. By contrast, the reconstructed image of JPEG has obvious blocking artifact and HEVC intra loses a lot of texture details. Our algorithm has high fidelity to the original image even at a compression ratio of over a thousand to one (the bpp of the original YUV420 image without compression is 12 and the bpp of our algorithm is 0.01, thus the compression ratio is 12/0.01).

4. CONCLUSIONS

We propose an image compression system via cloud search in this paper. The system takes advantage of the images in the cloud database, and encodes an image by exploiting inter-image correlations. This system has a much higher compression performance compared to traditional image coding methods. It provides a novel framework for image coding in the ‘big data’ era and has promising application in compressing images in Internet cloud disks and social network images.

Possible future work includes organization of the cloud database, more accurate image retrieval, improving the preprocessing step, etc.

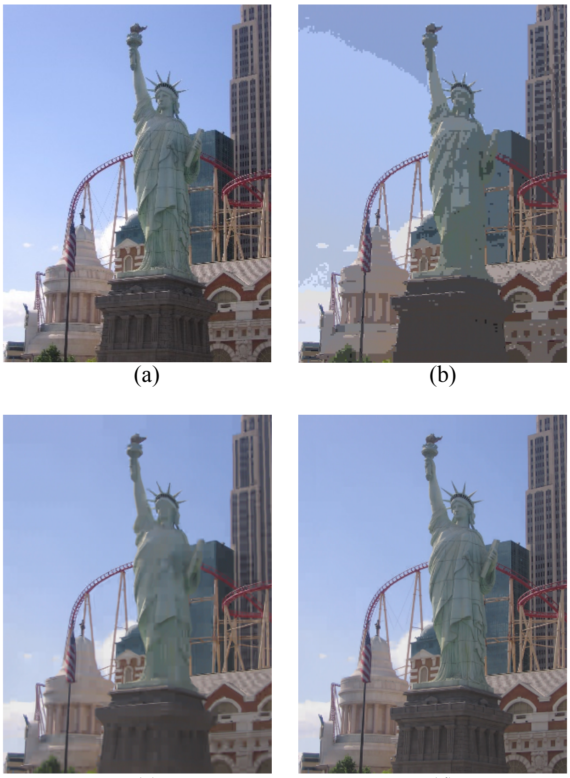
5. ACKNOWLEDGMENTS

This work is supported in part by the National High-tech R&D Program of China (863 Program, 2015AA015903), National Nature Science Foundation of China (61322106), which are gratefully acknowledged.

6. REFERENCES

[1] Z. Shi, X. Sun and F. Wu “Feature based Image Set Compression,” IEEE Conference on Multimedia and Expo (ICME), 2013.

[2] Shi, Zhongbo, Xiaoyan Sun, and Feng Wu. "Multi-model prediction for image set compression." In Visual



(a) (b) (c) (d)

Figure 7. Subjective quality of the reconstructed image by the proposed algorithm, JPEG and HEVC intra. (a)Original image; (b)Reconstructed image by JPEG, bpp=0.107, PSNR=26.11dB;(c)Reconstructed image by HEVC intra, bpp=0.014, PSNR=30.22dB; (d)Reconstructed image by the proposed algorithm, bpp=0.013，PSNR= 34.88dB。

Communications and Image Processing (VCIP), 2013, pp. 1-6. IEEE, 2013.

[3] Kosmas Karadimitriou and John M. Tyler, “The centroid method for compressing sets of similar images,” Pattern Recognition Letters, vol. 19, no. 7, pp. 585 – 593, 1998.

[4] Yurij S. Musatenko and Vitalij N. Kurashov, “Correlated image set compression system based on new fast efficient algorithm of karhunen-loeve transform,” pp.518–529, 1998.

[5] Samy Ait-Aoudia and Abdelhalim Gabis, “A comparison of set redundancy compression techniques,” EURASIP J. Appl. Signal Process., vol. 2006, pp. 216–216, Jan. 2006.

[6] Cheng H. Li X. Schmieder, A., “A study of clustering algorithms and validity for lossy image set compression,” in Proceedings of the 2009 International Conference on Image Processing, Computer Vision and Pattern Recognition (IPCV’09), 2009, pp. 501 –506.

[7] A. George, E. Yasser, H.[redacted]Meer, “Quadtree-Based Centroid Technique for Compressing Sets of Similar Medical Images,” Proc. 4th International Conference on Intelligent Systems Design and Applications, ISDA 2004.

[8] Chi-Ho Yeung, Oscar C. Au, Ketan Tang, Zhiding Yu, Enming Luo, Yannan Wu, and Shing Fat Tu, “Compressing similar image sets using low frequency template,” in Multimedia and Expo (ICME), 2011 IEEE international Conference on, july 2011, pp. 1 –6.

[9] C. Nielsen and Xiobo Li, “MST for lossy compression of image sets,” in Data Compression Conference, 2006. DCC 2006. Proceedings, march 2006, pp. 1 pp. –463.

[10] B. Gergel, H. Cheng “A unified framework for image set compression,” ww1.ucmss.com

[11] B. Gergel, A. Schmieder, X. Li, and H. Cheng. A study of prediction measures for lossy image set compression. In Proceedings of the 2008 International Conference on Image Processing, Computer Vision, & Pattern Recognition (IPCV’08), pages 69–74, 2008.

[12] J. Ranger and H. Cheng, “Structural similarity as a prediction metric in lossy image set compression,” in ICVP 2012. Proceedings, 2012.

[13] Huanjing Yue, Xiaoyan Sun, Jingyu Yang, and Feng Wu. "Cloud-Based Image Coding for Mobile Devices—Toward Thousands to One Compression." IEEE TRANSACTIONS ON MULTIMEDIA 15, no. 4 (2013): 845.

[14] CDVS1. (2011). Call for Proposals for Compact Descriptors for Visual Search, N12201. Turin, Italy: ISO/IEC JTC1/SC29/WG11.

[15] Ling-Yu Duan, Jie Lin, Jie Chen, Tiejun Huang, Wen Gao, "Compact Descriptors for Visual Search," IEEE MultiMedia, vol.21, no.3, pp.30,40, July-Sept. 2014.

[16] R. Szeliski. Computer vision: algorithms and applications.Springer, 2010.

[17] http://lear.inrialpes.fr/~jegou/data.php

[18] G. Bjontegaard, “Calculation of average PSNR difference between RD-curves,” in Proc. ITU-T Q.6/SG16 VCEG 13th Meeting, Austin, TX, Apr. 2001, Doc. VCEG-M33.