PRESERVING SIFT FEATURES IN JPEG-ENCODED IMAGES

Jianshu Chao and Eckehard Steinbach

Institute for Media Technology, Technische Universität München, Munich, Germany

ABSTRACT

For image compression applications where the information sink is not a person but a computer algorithm, the image encoder should control the encoding process in such a way that the important and relevant features of the image are preserved after compression. In this paper, our goal is to preserve the strongest SIFT features for JPEG-encoded images. We analyze the relevant characteristics of SIFT features and categorize the image Macroblocks into several groups. Then we propose a novel rate-distortion model which is based on the SIFT feature matching score. The dependency between the quantization table in the JPEG file and the common Lagrange multiplier is obtained from a training image database. Then for a given image quality we exploit this relationship to perform R-D optimization for each group. Our results show that the proposed algorithm achieves better feature preservation when compared to standard JPEG encoding. The proposed approach is fully standard compatible.

Index Terms— SIFT features, JPEG, variable quantization, R-D optimization

1. INTRODUCTION

Image compression has become a key technology for multimedia applications. Several successful standards (e.g. JPEG and JPEG2000) have been defined and are widely deployed in consumer electronic products. Typically, the image encoder assumes that a human being is the information sink that enjoys the content and hence operates in such a way that the perceivable distortion for a given bit budget is minimized. Or in other words, the lossy compression schemes behind these standards are optimized for and adapted to the human visual system. However, more and more application scenarios emerge where the information sink is not a person but a computer algorithm (e.g. face detection, object retrieval, event detection, object tracking, driver assistance, location recognition, surveillance, etc.). In these scenarios the adaptation to the human visual system is no longer necessary and the encoder should control the encoding process in such a way that the important and relevant features of the image are preserved after compression. In fact, typical encoding artifacts (e.g. blockiness) quickly confuse feature-based image analysis approaches.

The survey in [1] presents an overview of related work in the area of feature preserving image compression. Previous work has mainly focused on features such as edges, contours and textures, and the proposed coding frameworks are not standard-compatible. In our work, we are mainly interested in the preservation of important image features for object detection, matching and tracking. The Scale Invariant Feature Transform (SIFT) [2] is one of the most popular state-of-the-art feature extraction algorithms. Our goal is to preserve the most important SIFT features when compressing images at relatively low bit rate in a standard compatible manner. The basic idea is illustrated in Fig. 1. When these features are preserved satisfactorily, the subsequently detection, matching and tracking tasks will achieve high performance. We present and validate our proposed feature-preserving image compression approach based on JPEG images for SIFT features.

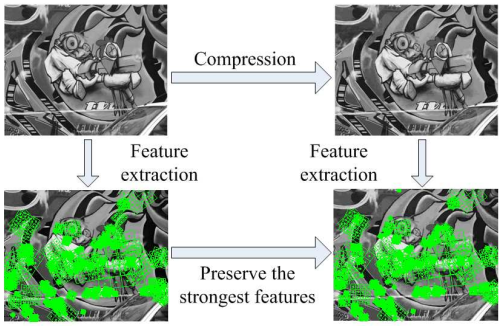

Fig. 1. The extraction of SIFT [2] features on an original and a JPEG compressed image. Ideally, the feature detector should return the same set of features.

The remainder of this paper is organized as follows. In Section 2, the evaluation criterion for feature preservation is presented. In Section 3, first we propose a simple approach for the encoding of those blocks without relevant features. Then we propose an R-D optimization method for the blocks which contain features that should be preserved. Section 4 presents the results which demonstrate that the proposed approach leads to improved feature preservation when compared to traditional JPEG encoding. Section 5 concludes the paper.

2. EVALUATION CRITERION

In the evaluation of feature detectors, the repeatability of interest points, which compares the position and scale of features, is widely used. We follow this approach and, similar to [2], a feature is defined to be repeatable, if: (1) the scale of the new feature in the SIFT scale space is within a factor of $\sqrt{2}$ of the original scale and (2) the location is within σ pixels of the original feature where σ is the scale of the feature in the original image. Following the approach in [3], we use the matching score as the evaluation criterion for feature preservation. A correct match is found if the feature is repeatable and its descriptor is additionally the nearest neighbor in the descriptor space. The matching score is then defined as the ratio between the number of correct matches and the number of original features in the uncompressed image.

$$matching\ score = \frac{\#correct\ matches}{\#original\ features} \tag{1}$$

This work has been supported by a PhD grant from the China Scholarship Council for Jianshu Chao

The criterion is simple and easy to compute considering the circular regions of SIFT features.

3. BIT ALLOCATION

Our main goal is to design and validate rate-distortion optimization strategies for image compression for which the human perception oriented distortion measure used traditionally is replaced by a distortion measure that helps us to preserve SIFT features. For this we have to modify the encoder control and in particular address the quantization and bit allocation problem. We use the available bit budget in such a way that feature detection algorithms applied to the uncompressed and compressed image produce as many identical SIFT features as possible.

The software we use for extracting SIFT features is VLFeat 0.9.9 from [4]. The images are compressed using the JPEG codec from [5] with variable quantization according to the JPEG standard extension syntax [6]. The test images are from [3] and for these images we aim to preserve the 200 strongest features. The strongest features are those which lead to the largest detector response during detection.

3.1. Bit allocation for blocks without relevant features

In the process of feature extraction, the SIFT descriptor is computed from a 4×4 array of histograms with 8 orientations. The size of the array in the image is mσ where σ is the scale of the feature and m = 3.0. So the SIFT descriptor is calculated from a square with an edge length of 12σ pixels. We start our evaluation with a simple test: The blocks within 6σ radius of a feature are compressed using normal compression quality while other parts which contain no features are compressed at much lower quality. One could even think of deleting these areas which are irrelevant for feature extraction. Conceptually, this corresponds to the approach in [7] which after feature extraction compresses the relevant local image patches in a non standard-compatible way.

To validate our idea we compress all relevant MBs at the same quality level (50) and the irrelevant MBs using worst quality. For the sake of compatibility with the JPEG extension (variable quantization [6]), blocks which do not contribute to the strongest features are compressed using non-linear Q_SCALE and SCALE_CODE = 31. Our results show that the bit rate of the normally compressed JPEG file (image *graf*, quality 50) is 0.9817 bpp and the matching score is 96.5%. In comparison, the bit rate of the image compressed using the simple feature-preserving approach is 0.7413 bpp and the matching score is also 96.5%. Of course this is expected because SIFT features can be extracted and matched individually. Due to this fact, we can allocate more bits to SIFT areas and other blocks use the worst quality. This represents our first approach of bit allocation. The resulting images are shown in Fig. 2.

Fig. 2. Image quality 50, blocks containing no features use the SCALE_CODE 31 (Left image: *graf*; Right image: *bark*)

Fig. 3 shows the matching score for traditional JPEG and the simple bit allocation method described above (from now on referred to as "approach 1") as a function of bit rate. The matching scores are not influenced while the bit rates decrease significantly.

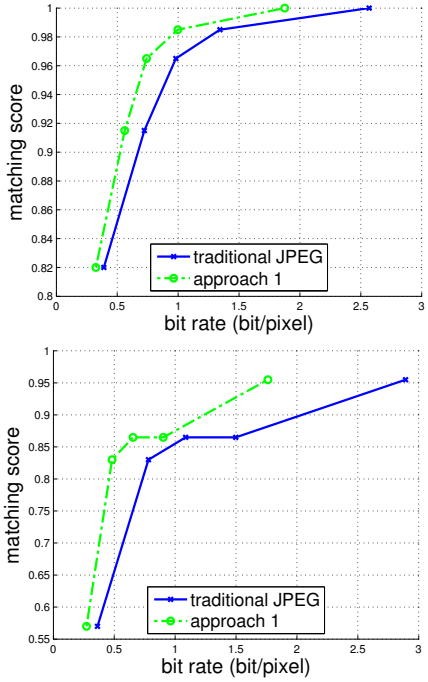

Fig. 3. Matching scores as a function of bit rate. Blue: traditional JPEG images; Green: approach 1 for bit allocation (Upper: test image *graf*; Lower: test image *bark*)

3.2. Feature analysis

In order to distribute the bit budget even better for the blocks that contain SIFT features or parts thereof, an improved rate-distortion optimization strategy can be applied. Each SIFT feature has a scale and the feature containing blocks of different scales can overlap, i.e., one image block can be relevant for several features. Hence a simple bit allocation block by block will not be optimal. The main challenge that we address in this paper is how to allocate the bit budget (adapt the quantization) such that as many important features as possible are preserved after compression.

First we make an observation concerning SIFT features and the matching score. The curves in Fig. 4 show the matching scores separately for each octave of the SIFT scale space for images compressed with JPEG quality level 50. The seven different curves are from the first images of the "graf, bikes, cars, bark, trees, ubc, wall" sets respectively [3]. Only a few octaves contain SIFT features. From the curves, we can conclude that the lower the octave, the more vulnerable the features are. That's because features in lower octaves have smaller scales, so they are easily influenced by JPEG compression. In comparison, the features which have large scales are more robust against compression artifacts. Large scale features contain more information, thus have a higher discriminative power which makes them easier to match [3]. Hence, if we want to preserve the strongest features across all scales, we have to allocate more bit budget to the error-prone small scale features.

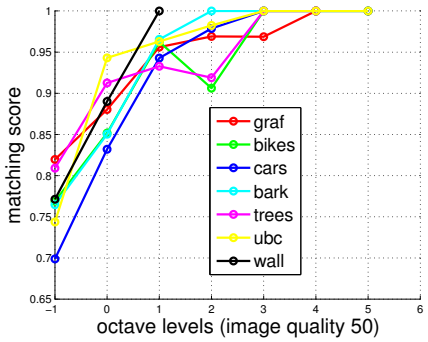

Fig. 4. Matching scores in each octave

3.3. Categorization of blocks containing features

Based on the observation above, the categorization of blocks can be done as follows:

Step 1. Calculate SIFT features in each octave separately.

Step 2. Find the blocks containing features in the first octave, then tag these blocks as group 1.

Step 3. Find the relevant blocks in the next octave. If the blocks haven't been tagged before, tag them as a new group.

Step 4. Repeat step 3 until the last octave level is reached.

In our experiment, for the *graf* and *bark* test images, 4 and 5 octaves contain features, respectively. Consequently 4 respective 5 groups are obtained. The reasons for this categorization are the following: First, the matching score cannot be increased by changing the quantization parameter of one single block. We have to adjust the quantization parameter of all blocks which are relevant for one feature together. Second, small scale features are more vulnerable to compression and they should be treated preferentially. Third, small scale features use higher quality after performing quantization optimization and the relevant blocks' information will be kept well. The possible overlapping parts with larger scale features will not influence the successive allocation process for large scale features.

3.4. Rate-Distortion Optimization

We assume that the matching score has a direct relationship with the DCT coefficient quantizer. Next we propose a novel distortion measure and design an encoding mode and quantization control strategy. If the compressed image has the exact same set of features (i.e. no distortion of features), the matching score should be 1, otherwise it is smaller. We treat (1 - matching score) as our distortion metric. Blocks with SIFT features are compressed using the following rate-distortion model:

$$D_{ms} = 1 - matching\ score \tag{2}$$

$$J = D_{ms} + \lambda R \tag{3}$$

Since the total distortion is the sum of the individual distortions, the optimization of (3) is simplified by minimizing the cost function separately for each group. The rate distortion cost of each group is:

$$J = \sum_{i=1}^{N} J_i = \sum_{i=1}^{N} D_i + \lambda(\sum_{i=1}^{N} R_i) \tag{4}$$

where N is the number of groups, D_i is the distortion of the *i-th* group; R_i is the rate of the *i-th* group. We minimize the J_i separately, using a common Lagrange multiplier λ.

3.5. Lagrange Multiplier

JPEG codecs provide a selection of qualities from 1 to 100 which correspond to 100 values used to scale the standard quantization table. The final quantization table in the JPEG file is generated according to the selected scaling factor. The approximate relationship between the recommended scales (S) and image qualities (Q) in [5] is:

$$S = \begin{cases} 50/Q & \text{if } Q < 50, \\ 2 - 0.02 * Q & \text{if } Q \geq 50. \end{cases} \tag{5}$$

Now we use a training database to determine the relationship between the common λ and the image qualities following the method described in [8]. The uncompressed image database is from [9] where the first 1200 color images are encoded at all quality levels. The matching score for each quality and the bit rate are recorded. Note that the matching score is discrete because it is calculated from the ratio of correct matches and the number of original features in one image. For a fixed value of λ, the minimization of the Lagrangian cost function in equation (3) is used to obtain the optimum quality Q for an image. Fig. 5 shows the relative frequency of the optimum qualities Q for the Lagrange multiplier λ varied over nine values: 2, 1.5, 1, 0.5, 0.3, 0.15, 0.1, 0.05 and 0.01. From the chart, we can see that for high and low qualities there is a close relationship with λ. The middle λs span a large area of qualities. In our experiment, in contrast we also find that for a given quality level, the matching score does not change significantly even if λ varies a lot within a certain range when we optimize the image. In other words, the image qualities seem not very sensitive to λ within a specific range. So it is sufficient to obtain the approximate relationship between the quality levels and λ.

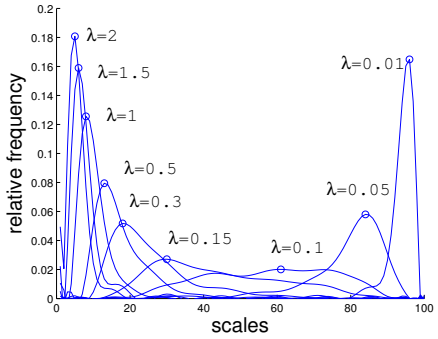

Fig. 5. Relative frequency of the chosen optimum qualities vs. quality levels

Next, we map the JPEG qualities to real scales S which are used to multiply the standard quantization table. Now each tested λ value corresponds to a unique quantization scale S having the highest relative frequency. We use these data points to fit a curve. The fitted curve, shown in Fig. 6, is the approximation of the relationship between the scale and the Lagrange multiplier:

$$\lambda \approx 0.012 \cdot S^2 + 0.08 \cdot S \tag{6}$$

When the scale is small, λ tends to be linear with S; when the scale is large, λ tends to be quadratic with S. A target quantization table is included in the JPEG file according to the input quality level. Before we perform R-D optimization using equation (3), we use equation (6) to calculate λ for the given quality level.

According to the JPEG standard extension [6], we use non-linear Q_SCALE and in this case the block has the same quantizers as the

target quantization table if its SCALE_CODE (SC) equals 12. The quantizers for each block can be changed according to the SC of each block. We select SC from 9 to 15 in order to confine the bit budget. Hence, the minimization of the individual J_i can be written as:

$$SC_i^* = \arg\min_{SC} J_i,\ SC \in \{12, 12 \pm 1, 12 \pm 2, 12 \pm 3\} \quad (7)$$

The variable quantization is performed by tuning the SCALE_CODE so that the optimum SC_i for each group is obtained which minimizes the Lagrangian cost function. R-D optimization is first performed in group 1 which contains the small scale features by adjusting SC_1 in equation (7). The quantizers for the first group are fixed after finding the optimum SC_1. Then the R-D optimization process is repeated for group 2. This continues until the last group.

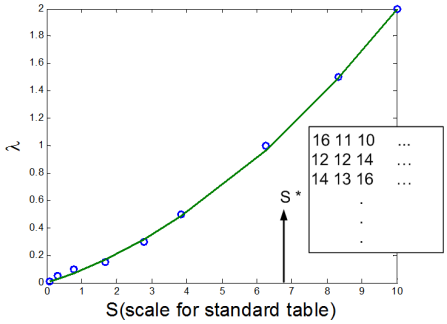

Fig. 6. Lagrange multiplier vs. scale for standard table

4. RESULTS

Fig. 7 shows experimental results for the two test images *graf* and *bark*. The blue curves are matching scores for traditional JPEG images; green curves for images after bit allocation with approach 1; red curves for images after R-D optimized bit allocation. Obviously, through the proposed R-D optimization, we can achieve better performance compared to traditional JPEG compression. We also see an additional improvement compared to approach 1 introduced in Section 3.1. We observe similar improvements for other images from [3]. From the results we also find that if the percentage of the blocks without relevant features is high, approach 1 already achieves most of the gains while approach 2 improves just a little on top of it. This is because the bit rate reduces a lot when many MBs use worst quality and just a few ones need to be tuned while performing R-D optimization. In contrast, approach 2 achieves larger gains than approach 1 when most of the blocks contain features. So the two approaches finally achieve improved SIFT feature preservation performance compared to traditional JPEG encoding.

5. CONCLUSION

In this paper we propose and validate a strategy for preserving SIFT features after JPEG image compression. We examine the properties of SIFT features and propose two methods to allocate the bit budget. Our results show that our proposed algorithm successfully preserves the strongest SIFT features even at low bit rates. The extension of the proposed approaches to other compression schemes (e.g JPEG2000 and H.264/AVC) or other features (e.g. SURF [10]) will be addressed in our future work.

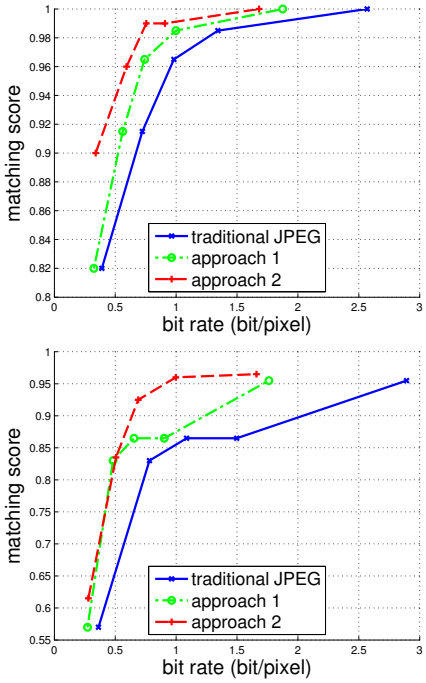

Fig. 7. Results of our feature-preserving bit allocation for two different test images. (Upper: test image *graf*; Lower: test image *bark*)

6. REFERENCES

[1] O.O.V. Villegas, R.P. Elias, and V.G.C. Sanchez, "Feature Preserving Image Compression: A Survey," *Proceedings of the Electronics, Robotics and Automotive Mechanics Conference (CERMA'06)*, 2006.

[2] D.G. Lowe, "Distinctive Image Feature from Scale-Invariant Keypoints," *International Journal of Computer Vision, vol. 2, no. 60, pp. 91-110*, 2004.

[3] K. Mikolajczyk, T. Tuytelaars, C. Schmid, A. Zisserman, J. Matas, F. Schaffalitzky, T. Kadir, and L.V. Gool, "A comparison of Affine Region Detectors," *International Journal of Computer Vision vol. 65, no. 1-2, pp. 43-72*, 2005.

[4] A. Vedaldi and B. Fulkerson http://www.vlfeat.org/.

[5] Independent JPEG Group http://www.ijg.org/.

[6] ITU-T Rec. T.84, *Information Technology - Digital Compression and Coding of Continuous-Tone Still Images: Extensions*, 1996.

[7] M. Makar, C.-L. Chang, D. Chen, S. Tsai, and B. Girod, "Compression of image patches for local feature extraction," *IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP), Taipei, Taiwan*, April 2009.

[8] G.J. Sullivan and T. Wiegand, "Rate-Distortion Optimization for Video Compression," *IEEE Signal Processing Magazine, vol. 15, no. 6, pp. 74-90*, Nov. 1998.

[9] G. Schaefer and M. Stich, "UCID - An Uncompressed Colour Image Database," *Storage and Retrieval Methods and Applications for Multimedia, pp. 472-480*, 2004.

[10] H. Bay, A. Ess, T. Tuytelaars, and L.V. Gool, "SURF: Speeded Up Robust Features," *Computer Vision and Image Understanding (CVIU), vol. 110, no. 3, pp. 346-359*, 2008.