SIFT FEATURE-PRESERVING BIT ALLOCATION FOR H.264/AVC VIDEO COMPRESSION

Jianshu Chao[*] *and Eckehard Steinbach*

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

ABSTRACT

Compression artifacts in low-quality videos strongly influence the performance of feature matching algorithms. In order to achieve reasonable feature matching performance even for low bit rate video, we propose to allocate the bit budget during compression such that the important features are preserved. Specifically, we present two bit allocation approaches to preserve the strongest SIFT features for H.264 encoded videos. For both approaches, we first categorize the Macroblocks in a Group of Pictures into several groups according to the scale specific characteristics of SIFT features. In our first approach a novel R-D model based on the *matching score* is applied to allocate the bit budget to these groups. In our second approach, in order to reduce the computational complexity, we analyze the detector characteristics of correctly matched pairs and propose a R-D optimization method based on the *repeatability* metric. Our experiments show that both approaches achieve better feature preservation when compared to standard video encoding which is optimized for maximum picture quality. The proposed approaches are fully standard compatible and the encoded videos can be decoded by any H.264 decoder.

Index Terms— SIFT features, H.264, R-D optimization

1. INTRODUCTION

Video encoding algorithms normally take a human centric approach and maximize the subjective quality of the compressed content by exploiting known limitations of the human visual perception system. In some emerging scenarios (e.g., object retrieval and tracking, driver assistance services, location recognition, surveillance, etc.), the video content is, however, processed by computer algorithms rather than consumed by a person. In these scenarios there is no need that the lossy video compression schemes maximize the visual quality. As a matter of fact, the performance of typical video analysis algorithms is strongly influenced by encoding artifacts. The authors in [1] coin in this context the term *critical video quality* which is used to determine the smallest video bit rate without degrading the accuracy of face detection and face tracking algorithms.

The Scale Invariant Feature Transform (SIFT) [2], which is a widely used robust local feature detection algorithm, has been shown to easily get confused by compression artifacts [3] [4]. In this paper, we are interested in the preservation of SIFT features during low bit-rate video encoding. There is little previous work on SIFT feature-preserving image/video compression. [3] proposes to compress the relevant local patches after feature extraction in a non-standard compatible manner. In our previous work [5] we propose a JPEG compatible approach for SIFT feature-preserving still image compression. The core idea of [5] and the two bit-allocation approaches proposed in this paper is to control the encoding process such that the most important and relevant features are preserved satisfactorily for low bit rate encoding. The subsequently applied video analysis approaches (e.g. detection, matching and tracking) will then achieve better performance.

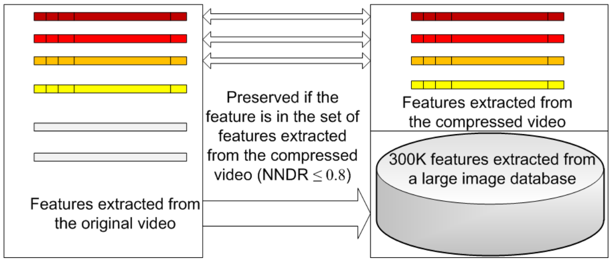
Preserved if the feature is in the set of features extracted from the compressed video (NNDR ≤ 0.8)
Features extracted from the compressed video
300K features extracted from a large image database
Features extracted from the original video

Fig. 1. Criterion of feature preservation.

The H.264/AVC video compression standard [6] is widely deployed in consumer electronic products and multimedia communications applications. Hence, the goal in this work is to preserve SIFT features in H.264-encoded videos. In the remainder of this paper we present and validate our proposed feature-preserving video compression schemes. In Section 2, the evaluation criterion for feature preservation is presented. In Section 3, we describe a descriptor-based R-D optimization approach for the Macroblocks (MBs) which contain features that should be preserved in the compressed video. In Section 4 we propose an alternative detector-based R-D optimization approach which is based only on the detection of likely locations, scales and orientations of interest points. The proposed detector-based approach is computationally faster compared to the descriptor-based approach. Section 5 presents experimental results which show that the proposed approaches lead to improved SIFT feature preservation when compared to normally H.264-encoded video. Section 6 concludes the paper.

2. EVALUATION CRITERION AND VALIDATION METHOD

The features we want to preserve are those which exhibit a detector response above a certain threshold during detection. Ideally, the descriptors of these features before and after video compression would be identical if they are preserved perfectly. This is, however, impossible for lossy video compression. In our work, a feature is defined to be preserved if the descriptor in the original image has its corresponding nearest neighbor descriptor in the compressed image. To this end, we extract the features in the original image and compare them with about 300K features extracted from a reference image database and the features in the compressed image, examining whether the matched descriptors are in the compressed image. According to [2], it is better to reject false matches by comparing the nearest neighbor to the second nearest neighbor in the descrip-

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

tor space, which is called nearest neighbor distance ratio (NNDR) matching strategy. In our experiments, the NNDR is set to 0.8 as proposed in [2] and we use FLANN [7] to get the nearest neighbors in the feature database. If the NNDR is greater than 0.8, the feature preservation is assumed to have failed. This process is illustrated in Figure 1. The "correct matches" are the corresponding matched pairs in the uncompressed and compressed images which meet the above NNDR criterion . Following the approach in [4], we use the matching score as the evaluation criterion for feature preservation. The matching score is defined as the ratio between the number of correct matches and the number of original features extracted in the uncompressed video.

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

In the process of encoding the H.264 video, the matching scores between the original frames and the compressed frames are calculated. In our experiments we use VLFeat 0.9.13 [8] for SIFT feature extraction. The video compression is performed using the H.264/AVC reference software JM 17.2 [9].

3. DESCRIPTOR BASED BIT ALLOCATION

Our main goal in this paper is to design and validate R-D optimization strategies for SIFT feature-preserving video compression. After normal MSE-based R-D optimized mode decision is performed, we modify the encoder control using a novel distortion measure to determine which quantization parameter (QP) to choose for each MB in order to preserve the most important features. The available bit budget is used in such a way that the uncompressed and compressed videos produce as many matching features as possible.

3.1. Bit allocation for blocks without relevant features

A SIFT feature covers only a local image area and the descriptor is calculated within a 4×4 array of histograms with 8 orientations. The video quality of MBs which do not contribute to the important features has no effect on the matching score. Similar to [5], these MBs are compressed using the worst QP=51 in conformity with the H.264 standard.

3.2. Categorization of blocks containing features

Next, a R-D optimized QP decision is applied to those MBs containing features. The SIFT features in one frame typically extend across several MBs and overlap with each other. The traditional R-D optimization block by block is hence not appropriate. The different MBs have varying importance for feature preservation. From our previous work [5], we know the sensitivity of features detected at different scales to compression artifacts. In general there are fewer features at higher scales while they have higher matching scores. This is because features in lower scales occupy smaller regions, so they are more easily influenced by compression artifacts. Large scale features cover large image areas and contain more information, thus have a higher discriminative power which makes it easier to match them [4]. Hence, we allocate more bit budget to the error-prone small scale features.

Following the approach proposed in [5], we analyze the relevant characteristics of SIFT features and categorize the image Macroblocks into several groups according to the scale values of the features. The following depicts our Macroblock categorization method:

Step 1. Calculate SIFT features in each octave separately for all frames.

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

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

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

Step 5. Perform R-D optimization from group 1 to the last group.

For reduced computational complexity, we select the MBs for entire octaves (3 scales by default) instead of for individual scales. The number of groups is hence reduced by a factor of 3. At low bit rate many MBs of P-frames are coded using the skip mode in H.264. In this mode, the MBs are copied from the previous frame and no residual signal is transmitted. So the QP of the current MB can not be tuned in the current frame. Due to this, when calculating the matching score in our proposed feature-preserving R-D optimization the whole GOP is treated as a unit, i.e, one group includes MBs across all frames in a GOP in Step 2.

3.3. Rate-Distortion Optimization

We detect the important features from the original video and the corresponding matched features in the H.264 encoded video. The matching score for a GOP is the ratio between the total number of important features and the number of matches in the compressed frames. Here we define the R-D model for optimization as follows.

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

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

where D_{ms} is the distortion metric used in R-D optimization for relevant MBs with SIFT features. The distortion is 0 when the features in one GOP are all preserved while 1 means no features are preserved after compression. 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.

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

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

Each video has its own properties so it is difficult to estimate the Lagrange multiplier analytically. Hence, currently we use experimentally determined λs. For this we encode the video with all QPs to determine the R-D characteristics. For a fixed λ, the optimal QP can be chosen on the convex hull of the R-D points. Hence, the minimization of the individual J_i can be written as:

$$QP_i^* = \arg\min_{QP} J_i,\ QP \in \{0, 1, 2, ..., 51\} \tag{5}$$

As Step 5 above shows, first we perform R-D optimized QP selection in group 1 because of the high vulnerability of these smaller scale features. The QP for the MBs in the first group is fixed after finding the minimum J. Then the R-D optimization process is repeated for group 2. This continues until the last group.

4. DETECTOR BASED BIT ALLOCATION

The approach described in Section 3 is computationally expensive because the descriptors need to be calculated and compared with the

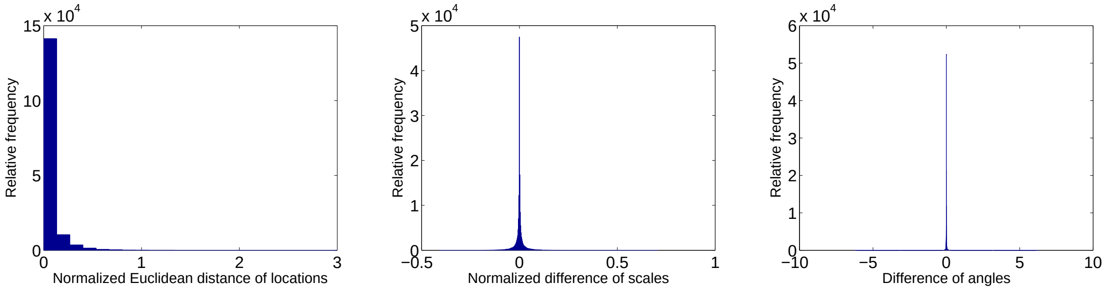

Fig. 2. Relative frequencies of locations, scales and angles.

300K feature descriptor database. [10] defines the ϵ-*repeatability* of point pairs which is widely used as the detector evaluation criterion in their experiments. The authors examine the repeatability for location errors ϵ ranging from 0.5 to 5 pixels. [4] introduces the *overlap error* which is defined as the error of region-to-region ellipses from a pair of affine region detectors. The repeatability score is calculated as the ratio between the number of feature pairs which have less than 40% overlap error and the smaller number of features in the pair of images. In [2], a feature is defined to be repeatable if the scales, locations and orientations of a pair of features are similar within certain tolerances.

Ideally, we need to set the parameters such that the repeatable features are identical to those obtained via correctly matched descriptors according to our previous descriptor matching strategy. From [2] we can see that the repeatability curves are all very close to the percentage of descriptors correctly matched to a large database, indicating that the repeatability criterion has a close relationship with the matching score. In this paper, we use a similar method as [2]. According 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; (2) the location is within σ pixels of the original feature where σ is the scale of the feature in the original image; (3) the orientations of two features are required to be within 15 degrees (optional). However, we have to examine the criterion carefully instead of using these parameters directly in our work. If the parameters of repeatability are too loose, then more false matches would be included. In contrast, the true matches would be expelled if the parameters are too strict.

4.1. Statistical observations

In order to determine the relationship between the detector characteristics and the correctly matched descriptors, we gather statistics about them. In our experiment, the first images of the "graf, bikes, cars, bark, trees, ubc, wall" sets [4] are first converted to YUV files, and then compressed as I frames. The SIFT parameter *peakthresh* is tuned to generate 500 to 700 features in the original images. Each descriptor is compared to the 300K database and also the descriptors in the compressed image using a QP from 0 to 51. In total, there are nearly 160K correctly matched pairs and the characteristics of all original detectors and the matched ones in the compressed images are the statistical data we need.

We calculate the Euclidean distance of the locations of matched pairs and then normalize the value by the original feature's scale, i.e, $\sqrt{(x-xc)^2+(y-yc)^2}/s$, where (x, y) is the original location, (xc, yc) is the new location in the compressed image, and s is the feature scale. Figure 2 (left) shows the relationship between the normalized Euclidean distance and its distribution from these 160K pairs. In our experiment, if $\sqrt{(x-xc)^2+(y-yc)^2}/s < 1$ the new feature has a high probability of being a correct match (the location requirement). Figure 2 (middle) presents the distribution of the normalized difference of scales, i.e, $|s - sc|/s$, where s is the original scale, sc is the new scale in the compressed image. The new scale should be very close to the original scale and we require $|s - sc|/s < 0.25$ (the scale requirement) in our experiments. Figure 2 (right) shows the difference of angles of correctly matched pairs. From the distribution we can see that the variance of the angle difference is smaller than the distribution of locations or scales, since the descriptors are easily affected by the angle errors. There are some possible correct matches around 2π, e.g., if an original angle θ is 0.001 and the corresponding angle θc is 6.281 respectively, then the $\theta - \theta c = -6.28$. So we require $|\theta - \theta c| < 0.2094$ *or* $||\theta - \theta c| - 6.2832| < 0.2094$ (the angle requirement, 0.2094 is 12 degrees). In summary, if a detected feature in the original image has a corresponding detected feature in the compressed image which meets the above three requirements, the original feature is repeatable. So the repeatability is defined as follows:

$$repeatability = \frac{\#repeatable\ features}{\#original\ features} \quad (6)$$

4.2. Rate-Distortion Optimization

The process of Rate-Distortion optimization is similar to Section 3.3 except that we now use a R-D model based on repeatability as follows.

$$D_{re} = 1 - repeatability(GOP) \quad (7)$$

$$J = D_{re} + \lambda R \quad (8)$$

where D_{re} is the distortion metric used in the detector-based R-D optimization for relevant MBs with SIFT features. The distortion is 0 when the detectors in one GOP are all repeatable while 1 means no detectors are repeatable after compression. In this approach, there is no need to calculate the descriptors and compare them with the 300K descriptor database. The flow diagram of the two approaches is illustrated in Figure 3.

5. RESULTS

In our experiments, we encode the videos using the Baseline Profile with RD-Optimization enabled using high complexity mode. The GOP size is 12 and its structure is IPPP$\cdot\cdot\cdot$. Only the first frame is encoded as an I frame, and one previous frame is used as a reference frame for P frames. The peak threshold in the VLFeat is set to 12 for image intensities within [0, 255] in order to extract the strongest features and all other parameters are used by default. The test video we use is the CIF format video *Tempete* [11] (100 frames are encoded and the frame rate is 30 fps) since it contains a specific object and the camera zooms out producing different scales of this scene. We

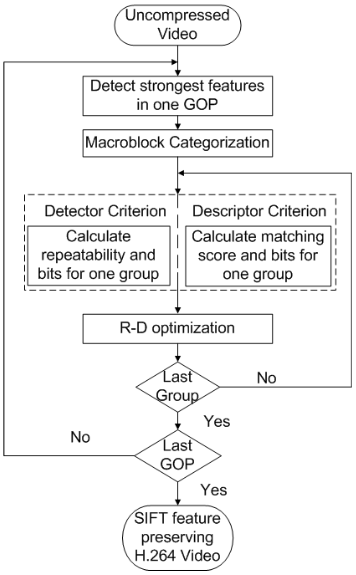
Uncompressed Video
Detect strongest features in one GOP
Macroblock Categorization
Detector Criterion
Descriptor Criterion
Calculate repeatability and bits for one group
Calculate matching score and bits for one group
R-D optimization
Last Group
No
Yes
No
Last GOP
Yes
SIFT feature preserving H.264 Video

Fig. 3. Flow diagram of the two proposed R-D optimization approaches.

optimize the video encoding according to the matching score based scheme described in Section 3 and the repeatability based scheme in Section 4, respectively. The QPs are selected as 30, 35, 40, 45, 50 for the normally encoded H.264 video. The experimental λ is varied over 2, 1, 0.5, 0.3 and 0.1 to generate suitable bit rates. Figure 4 shows the final matching scores and the repeatability values of normally H.264 encoded videos and our approaches as a function of bit rate. It can be seen that both proposed approaches lead to improved SIFT feature preservation.

Finally, in order to compare the three strategies clearly, the total number of matches for all frames in one H.264 video are counted to show how many features can be preserved. Figure 5 shows that our proposed two approaches both increase the number of matching features significantly compared to the normally encoded H.264 video. These numbers also indicate that the repeatability used in our optimization process is reasonable since it has a close connection with the matching performance. A video demo of our approaches can be viewed from YouTube[1]. In the videos with feature preservation, the human visual system is not considered and there are some visual inconsistencies due to large QP changes between MBs.

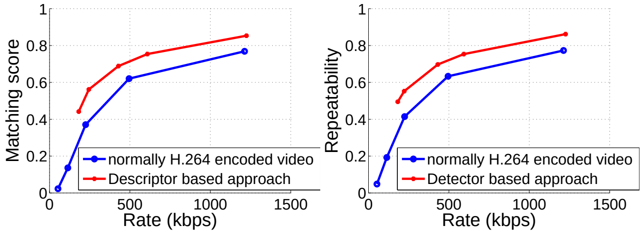

Fig. 4. Matching scores as a function of bit rate (left). Repeatability as a function of bit rate (right).

6. CONCLUSION

In this paper, we propose two bit-allocation strategies for the preservation of SIFT features in H.264 encoded video. In our first approach, a matching score based R-D optimization process is applied. In the second approach, we propose a repeatability based method for bit allocation, which is computationally faster. The results show that our proposed approaches can both achieve better matching performance when compared to the normally H.264 encoded video. The video encoded by our approaches is fully standard compatible and any H.264 decoder is able to decode the video. It should be noted that both approaches proposed in this work are computationally too complex for real-time encoding.

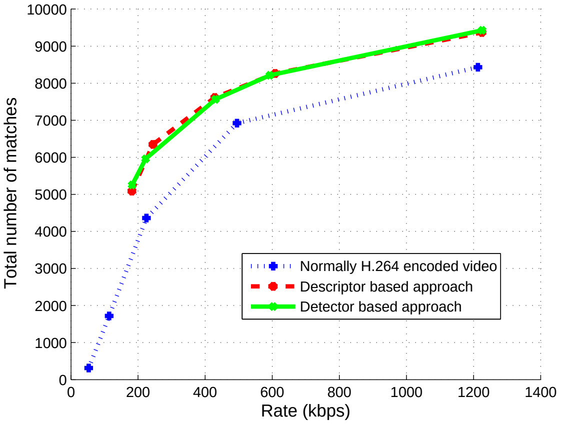

Fig. 5. The total number of matches as a function of bit rate.

7. REFERENCES

[1] P. Korshunov and W.T. Ooi, "Critical Video Quality for Distributed Automated Video Surveillance," *ACM Multimedia, Singapore*, November 2005.

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

[3] 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, Taipei, Taiwan*, April 2009.

[4] 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*, November 2005.

[5] J. Chao and E. Steinbach, "Preserving SIFT Features in JPEG-encoded Images," *IEEE International Conference on Image Processing, Brussels, Belgium*, September 2011.

[6] JVT, "Draft ITU-T recommendation and final draft international standard of joint video specification (ITU-T rec. H.264-ISO/IEC 14496-10 AVC)," March 2003.

[7] M. Muja and D. G. Lowe, "Fast Approximate Nearest Neighbors with Automatic Algorithm Configuration," *International Conference on Computer Vision Theory and Application, Lisboa, Portugal*, February 2009.

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

[9] http://iphome.hhi.de/suehring/tml/.

[10] C. Schmid, R. Mohr, and C. Bauckhage, "Evaluation of Interest Point Detectors," *International Journal of Computer Vision, vol. 37, no. 2*, June 2000.

[11] http://trace.eas.asu.edu/yuv/.

[1] http://www.youtube.com/watch?v=gVKL-qTkf-M