COMPRESSION OF IMAGE PATCHES FOR LOCAL FEATURE EXTRACTION

Mina Makar, Chuo-Ling Chang, David Chen, Sam S. Tsai and Bernd Girod

Information Systems Laboratory, Department of Electrical Engineering, Stanford University
{mamakar, chuoling, dmchen, sstsai, bgirod}@stanford.edu

ABSTRACT

Local features are widely used for content-based image retrieval and object recognition. We present an efficient method for encoding digital images suitable for local feature extraction. First, we find the patches in the image corresponding to the detected features. Then, we extract these patches at their characteristic scale and orientation and encode them for efficient transmission. A Discrete Cosine Transform (DCT) with adaptive block size is used for patch compression. We compare this method to directly compressing feature descriptors using transform coding. Experimental results show the superior performance of our technique. Image patches can be compressed to rates around 55 bits/patch (18x compression relative to uncompressed SIFT feature descriptors) and still achieve good image matching performance.

Index Terms— Image compression, image matching, feature descriptors, transform coding

1. INTRODUCTION

Many applications in computer vision, pattern recognition, and image processing require the use of local image features. Examples of robust local features include Scale-Invariant Feature Transform (SIFT) [1] and Speeded-Up Robust Features (SURF) [2].

In an image matching framework, keypoints are detected from database and query images. Then, feature descriptors are calculated for every keypoint. Matching between two descriptors is usually evaluated using the L_2 distance. SIFT is generally regarded as one of the best feature extraction algorithms for being robust against many image deformations [3]. In SIFT, keypoint detection is based on maxima detection in Difference of Gaussians (DoG) pyramid and the descriptor consists of histograms of gradients in a patch located around the detected keypoint.

For applications where data are transmitted over a network for feature detection and image matching, it is desirable that the amount of data sent is as low as possible. Some recent work proposed solutions to compress the feature descriptors using Karhunen-Loeve (KLT) transform coding [4]. The KLT matrix is trained from the statistics of sample descriptors. Others investigated dimensionality reduction via principle component analysis [5] or linear discriminant analysis [6].

In this paper, instead of compressing the feature descriptors directly, we explore an approach that compresses image patches centered on the keypoints used for feature extraction. We can exploit the fact that most parts of the image are not used for feature extraction. By sending only the patches containing the keypoints after compressing them efficiently to low bit rates, we can obtain better image matching results than sending the compressed features. Building upon existing image coding standards, we found that using JPEG [7] to send an image for the purpose of feature extraction yields results worse than compressing the feature descriptors directly. However, we achieve superior results with a simplified version of the recently proposed Direction-Adaptive Partitioned Block Transform (DA-PBT) [8].

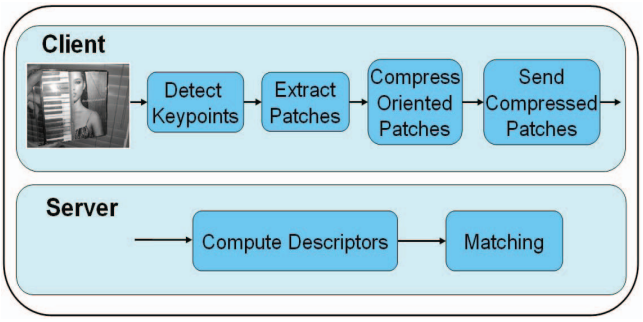

Fig. 1. System design for patch compression

The DA-PBT first divides the image into macro-blocks (16 x 16). Within each macro-block, it chooses the best block size (16 x 16, 8 x 8 or 4 x 4) and the best among 9 directional modes. The best block size and the best direction are selected via minimizing a Lagrangian cost function. In this paper, we only use the adaptive block size capability of DA-PBT with no directional modes. We refer to this transform as the Adaptive Block-size Discrete Cosine Transform (AB-DCT).

The rest of the paper is organized as follows. Sec. 2 discusses the proposed system design. In Sec. 3, we analyze the performance of patch compression using different image coding techniques and compare it to compression of the feature descriptors. In Sec. 4, we present experimental results obtained with a database of CD covers illustrating that our compression technique works well for a practical image retrieval problem.

2. SYSTEM DESCRIPTION

Fig. 1 represents the proposed design of the system. On the client side, SIFT keypoint detection is performed [1]. For every keypoint, we define a grid of a fixed size, e.g., 16 x 16 points, centered at the keypoint, rotated according to the keypoint orientation, and scaled so that the area it covers is representative of the image feature surrounding the keypoint. The image is sampled at the grid and rearranged into a square patch. This is done for all

the keypoints and then all patches are stacked into a single image to be compressed by the encoder. We use the AB-DCT followed by a context-adaptive arithmetic coder for entropy coding, which is more efficient than the entropy coding method adopted in [8].

The compressed patches are sent to the server side where a SIFT descriptor is calculated from every patch. The query descriptors are then classified through a scalable vocabulary tree (SVT) [9], searching greedily for the nearest neighboring database descriptors. The SVT enables fast search through a large database and identifies a small set of most probably matching database images. The positions of the keypoints also should be transmitted to the server for performing geometric consistency check. The descriptors of the database images are calculated using the same procedure used for query images.

Note that SIFT descriptors are calculated from the gradients of the pixel values within the patch. Thus, we need not send the mean value of the patch since it will not affect the gradients. Accordingly, the mean value of each patch is subtracted before encoding to reduce the coding rate.

Because we are only concerned with matching the descriptors but not the visual quality of the image, the patches can be compressed coarsely. Moreover, the sender extracts the keypoint locations, as well as scale and orientation of the patches, based on the uncompressed image data. Therefore, this aspect of image matching is not affected by lossy patch compression.

3. PERFORMANCE OF PATCH COMPRESSION

In order to study the effect of patch compression on feature quality, we extract patches from the Winder-Brown dataset [10]. We use 10,000 patch pairs that match and 10,000 patch pairs that do not match. These patches are extracted at their characteristic scale and then oriented so that the maximum gradient is along the vertical direction. As the authors in [10] indicate that the jitter errors in their datasets are less than those present in a real situation, controlled amounts of jitter based on the recommended values in [10] are added to the patches. The patches used in our experiments can be found in [11].

We compress the first patch in each patch pair using the AB-DCT. Our initial experiments were done using the DA-PBT. However, since patches are already oriented, the use of the directional modes did not improve compression significantly. This led us to use the AB-DCT, i.e., switching off the DA-PBT directional modes. The compressed patch serves as the query patch. The other patch in the pair is not compressed and serves as the database patch. We then compute SIFT descriptors [1] for the query and the database patches and measure the L_2 distance between the resulting descriptors for each pair.

These distances are used to build two histograms representing the PDFs of distances for the matching and non-matching patches. Smaller overlap between the two PDFs is better since it implies a lower probability of matching error. Using the two PDFs, we obtain the receiver operating characteristic (ROC) curve [10] which plots correctly detected matches as a fraction of all true matches against incorrectly detected matches as a fraction of all true non-matches.

Targeting a bit rate lower than direct compression of feature descriptors [4], we are most interested in rates below 100 bits/patch. Fig. 2 shows the ROC curves for the AB-DCT compression at rates around 78 and 39 bits/patch. Dotted lines

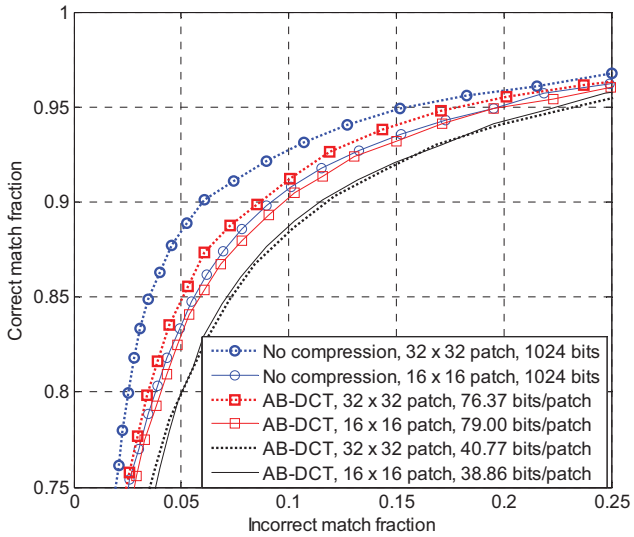

Fig. 2. ROC curves for AB-DCT compressed patches

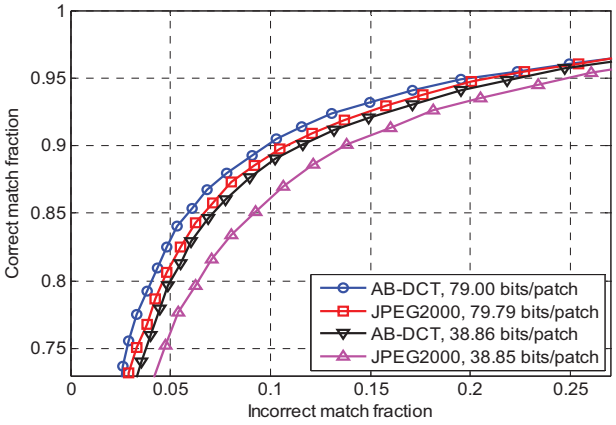

Fig. 3. ROC curves for AB-DCT and JPEG2000 patches

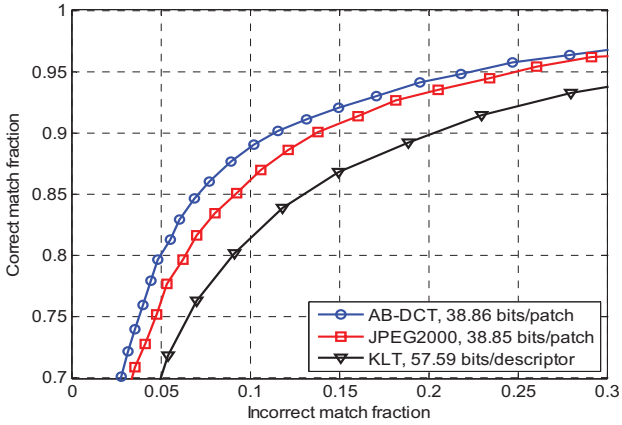

Fig. 4. ROC curves for KLT feature descriptor compression

indicate patch size of 32 x 32 while solid lines indicate 16 x 16. The original patch size is 64 x 64 and the center footprint is cropped after patch orientation. The ROC curve without compression is plotted for reference. At high rates, a larger patch size works better because more information is preserved around the keypoint. At lower rates, a smaller patch size is better since it

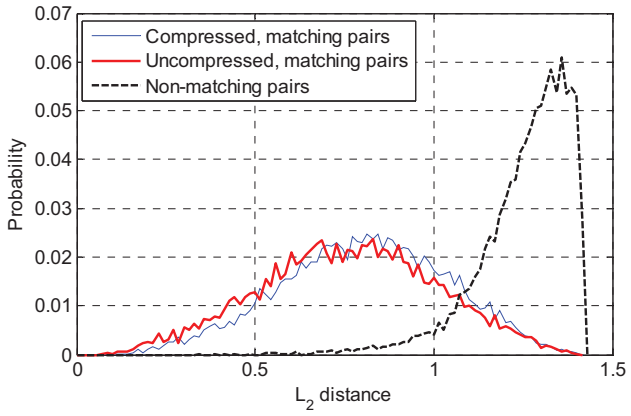

Fig. 5. PDF of L_2 norm distances for matching patch pairs

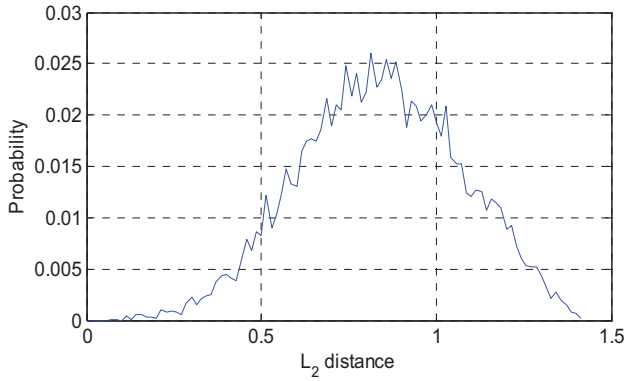

Fig. 6. Distances between original and compressed descriptors

achieves comparable performance with fewer bits (Fig. 2). Based on the previous curves we select to use 16 x 16 patches since our goal is to work at the lowest possible rate. This block size will be used throughout the rest of the paper.

Using the JPEG standard we cannot achieve these very low rates while obtaining acceptable matching performance. For comparison, we compressed the patches using JPEG2000 [12] at comparable rates. Fig. 3 shows a comparison between the AB-DCT and the JPEG2000 ROC curves at rates around 79 and 39 bits/patch. Similarly, the low rate of 39 bits/patch cannot be achieved using feature descriptor compression via transform coding as reported in [4]. Fig. 4 shows the ROC curve we obtain with descriptor compression at a rate of 57.6 bits/descriptor. Patch compression at a rate of 38.9 bits/patch is plotted for comparison.

From this experiment, we see that the AB-DCT outperforms the JPEG2000 standard for patch compression. Also, the efficiency of patch compression compared to feature descriptor compression [4] is obvious. In general block transforms are more suited for compressing patch images than subband coding transforms such as the DWT in JPEG2000 [12] because of consistency with patch edges. Thus, due to its design as a block transform and the use of adaptive block sizes with efficient entropy coding, the AB-DCT is a very good choice for application in patch compression.

We further studied the probability distribution of the L_2 distances between the descriptors of the matching pairs in the case where we compress the first patch using the AB-DCT and compare it to the case of without compression. The resulting PDFs are shown in Fig. 5 and the PDF for the compressed non-matching

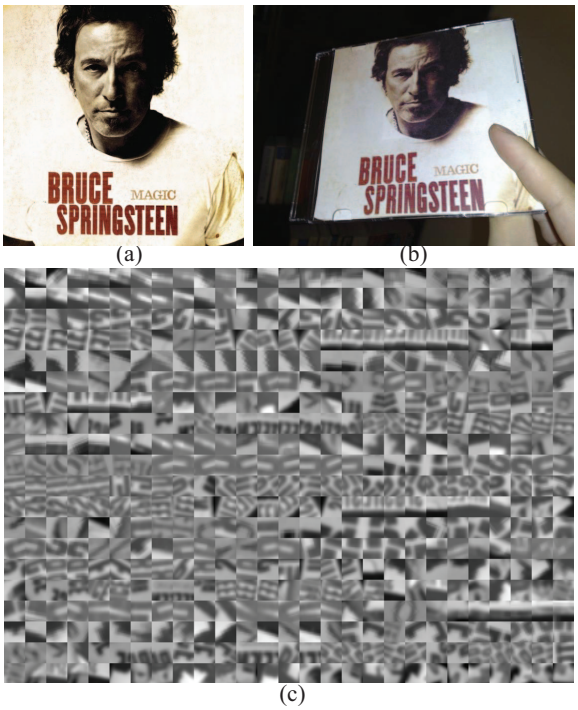

Fig. 7. (a) Example database image, (b) query image, (c) patches extracted from query image

pairs is plotted for reference. Although the PDF of the L_2 norm distances is shifted to the right in the case of compressing the patches, the difference between the PDFs is small for large distances (above 1.1 in the figure). These are the distances where we generally put the threshold for the match decision.

Another interesting observation is shown in Fig. 6. This figure represents the PDF of the L_2 distances between the descriptor extracted from the uncompressed patches and those extracted from the same patches but after compression at 38.9 bits/patch using the AB-DCT. We see that the variation in the L_2 distances between descriptors caused by compression is comparable to that caused by the change in appearance between different patches containing the same scene (Uncompressed in Fig. 5). This means the variation can still be handled by the robust image matching mechanism.

4. IMAGE MATCHING RESULTS

In this section, the technique of patch compression using the AB-DCT is applied to the CD cover recognition problem where we have a database of 1800 clean CD covers and we photograph 50 CDs using a camera phone to represent the query images. All query images have a single matching CD cover in the database. Fig.7 shows an example database-query image pair and the patches extracted from the query image. Mid-gray represents zero level. The complete set of query images is found in [13].

First, a pairwise matching test was performed on query images. Descriptors calculated from the compressed query image patches are matched with those calculated from the uncompressed patches of the matching database image. The ratio test [1] is performed to determine matching descriptors and then RANSAC [14] is used for geometric consistency check. In this experiment, we target at minimizing the average rate spent per patch while

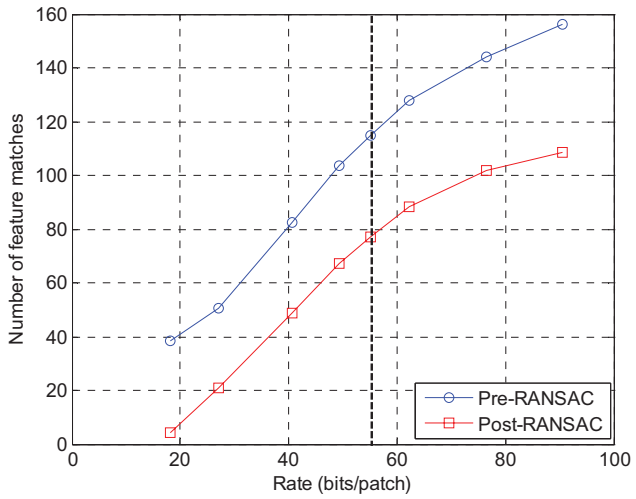

Fig. 8. Relation between average bit rate per patch and average number of feature matches

getting a sufficient number of Post-ratio-test and Post-RANSAC feature matches to indicate a matching image.

Fig. 8 represents the relation between the average rate per patch and the number of matched features Pre-RANSAC (after the ratio test) and Post-RANSAC. These results are averaged over the 50 image pairs and are based on using the SIFT algorithm for feature detection and description. The number of feature matches increases as more rate is spent to encode the image patches.

To match a database image, the number of Post-RANSAC matches should exceed a certain threshold. Using a rate of 55.18 bits/patch, we obtained correct matches for 48 image pairs where the PSNR of the image patches was around 29 dB. Note also that operating on the actual patches without mean removal requires a higher average rate of 60.96 bits/patch.

Second, we test retrieval performance for the entire database of 1800 CD covers. The query image patches are compressed to a rate of 55.18 bits/patch as it gives good results for pairwise matching (This bit rate is indicated with a dashed line in Fig. 8). The CD image database contains 1.5 million features. For fast search through the database, the query image features are classified using an SVT [9]. We assume that the right match is among the 25 top matching results from the SVT. The SVT top matches are then pairwise compared by ratio test and RANSAC, and the database image with the most Post-RANSAC feature matches is presented as the correct match. However, if all the 25 top matches yield Post-RANSAC results below a certain threshold, this indicates that no match was found. Using this framework, we again were able to achieve 48 correct matches for our 50 query CD images. Using the same query images without compressing the patches achieves 49 correct matches while requiring much higher rate to send the uncompressed patches or the uncompressed SIFT features. Typically, an uncompressed 128-dimensional SIFT feature vector is represented by 1024 bits. This means we obtain around 18x rate saving with a negligible effect on image matching performance.

For comparison with feature descriptor compression, the descriptors extracted from the uncompressed patches were compressed using the method in [4] to a rate of 63.25 bits/descriptor and the retrieval performance was tested using the same SVT. Only 46 correct matches were achieved despite using a higher bit rate. This shows the efficiency of the proposed technique in terms that it outperforms feature descriptor compression in a practical image retrieval problem.

5. CONCLUSION

Compression of feature patches enables significant rate reduction when the goal is to use these patches for image matching. We present an efficient framework for patch extraction and compression. We found the AB-DCT to be a good choice for patch compression outperforming current image coding standards. Transmitting compressed patches yields rate saving of around 18x relative to uncompressed SIFT descriptors and gives better results than compression of feature descriptors.

6. REFERENCES

[1] D. Lowe, "Distinctive image features from scale-invariant keypoints", *International Journal of Computer Vision,* vol. 60, pp. 91–110, November 2004.

[2] H. Bay, T. Tuytelaars, and L. V. Gool, "SURF: speeded up robust features", *European Conference on Computer Vision*, pp. 404–417, Graz, Austria, May 2006.

[3] K. Mikolajczyk and C. Schmid, "Performance evaluation of local descriptors", *IEEE Trans. on Pattern Analysis and Machine Intelligence,* vol. 27, pp. 1615–1630, October 2005.

[4] V. Chandrasekhar, G. Takacs, D. Chen, S. Tsai, J. Singh, and B. Girod, "Transform coding of image feature descriptors", *SPIE Visual Communications and Image Processing (VCIP)*, San Jose, California, January 2009. (Submitted)

[5] Y. Ke and R. Sukthankar, "PCA-SIFT: a more distinctive representation for local image descriptors", *Conference on Computer Vision and Pattern Recognition*, pp. 511–517, Washington DC, USA, June 2004.

[6] G. Hua, M. Brown, and S. Winder, "Discriminant embedding for local image descriptors", *International Conference on Computer Vision*, pp. 1–8, Rio de Janeiro, Brazil, October 2007.

[7] ITU-T and ISO/IEC JTC1, "Digital compression and coding of continuous-tone still images", in *ISO/IEC 10918-1 – ITU-T Recommendation T.81*, September 1992.

[8] C.-L. Chang and B. Girod, "Direction-Adaptive Partitioned Block Transform for Image Coding," *IEEE International Conference on Image Processing,* 2008 (to appear)

[9] D. Nister and H. Stewenius, "Scalable recognition with a vocabulary tree", *Conference on Computer Vision and Pattern Recognition*, pp. 2161–2168, New York, NY, USA, June 2006.

[10] S. Winder and M. Brown, "Learning Local Image Descriptors", *Conference on Computer Vision and Pattern Recognition*, pp. 1–8, Minneapolis MN, USA, June 2007.

[11] "Noisy patches database", www.stanford.edu/~vijayc/patches

[12] C. Christopoulos, A. Skodas and T. Ebrahimi, "The JPEG-2000 Still Image Coding System: An Overview," *IEEE Trans. Consumer Electronics*, vol. 46, no. 4, pp. 1103-1127, Nov. 2000.

[13] "CD query images", http://msw3.stanford.edu/~dchen/CDD/

[14] M. Fischler and R. C. Bolles, "Random sample consensus: a paradigm for model fitting", *Communications of ACM*, vol. 24, no. 6, pp. 381–384, June 1981.