INTERFRAME COMPRESSION WITH SELECTIVE UPDATE FRAMEWORK OF LOCAL FEATURES FOR MOBILE AUGMENTED REALITY

Jumpei Koyama[1], *Mina Makar*[2a], *Andre F. Araujo*[3], *and Bernd Girod*[3]

[1]Media Processing Systems Laboratories, Fujitsu Laboratories Ltd., JAPAN,
[2]Qualcomm Inc., USA,
[3]Department of Electrical Engineering, Stanford University, USA
koyama.junpei@jp.fujitsu.com, mmakar@qti.qualcomm.com, {afaraujo, bgirod}@stanford.edu

ABSTRACT

Mobile augmented reality (MAR) systems utilize local features to represent structures such as corners or blobs in query videos and database images. Local features are compressed and sent to servers to perform query-database matching. The compression efficiency is important for the server-based MAR systems in terms of network latency. Conventional compression methods handle video frames independently. In recent years, an interframe compression technology, which exploits temporal correlation over frames, has been proposed. It propagates local features temporally and realizes significant bitrate reduction without degradation in matching performance. However, this system requires a large amount of information when inserting independently compressed frames to update the status of local features. In this paper, we propose the selective update framework for local features and give simple algorithms to make use of the framework. Experimental results show that our proposal achieves gradual refreshes of local features, avoiding transmission of large amounts of data.

Index Terms— mobile augmented reality, image matching, local features, canonical patches, interframe compression

1. INTRODUCTION

Mobile augmented reality (MAR) systems may link the virtual world with the physical world by image-based retrieval technology. In these days, they have become widespread by the evolution of smartphones and the emerging of wearable devices such as Google Glass [1].

Most MAR systems employ server-based operations. First, query images or videos are captured by mobile devices such as smartphones. Then, keypoints are detected and local features are extracted from the image patches at those keypoints. Local feature extraction might use the Scale-Invariant Feature Transform (SIFT) [2], the Speeded Up Robust Features (SURF) [3] or Compressed Histogram of Gradients (CHoG) [4], for example. Local features are compressed and sent to remote servers to perform matching with database images. The compression process reduces the bitrate of local features and alleviates the burden to uplink of wireless networks. The compression technology is mainly categorized into three parts. The first one is binary hashing such as Locality Sensitive Hashing (LSH) [5] and Spectral Hashing (SH) [6]. The second one is transform coding as typified by Karhunen-Loeve Transform (KLT) and Independent Component Analysis (ICA) [7]. The third one is vector quantization. The representative methods are flat and hierarchical *k*-means (FKM and HKM) [8] and Product Quantization (PQ) [9]. In addition, Moving Picture Experts Group (MPEG) is developing a standard of compression technologies named the Compact Descriptors for Visual Search (CDVS) [10].

Another interesting approach for MAR systems is a patch encoding technology proposed by Makar et al [11]. In the architecture, image patches are compresses and sends to servers. This architecture enables to move the complex computational process of calculating descriptors from mobile devices to servers.

Conventional keypoint detection and local feature compression methods handle frames of query videos independently. Recently, a novel compression method, which utilizes interframe temporal correlation, is proposed. It achieves substantial bitrate reduction, compared to conventional methods [12]. The interframe compression inserts independent compression frames to refresh local features when the propagated local features can't keep representing local features of processing frames. The insertion needs a large amount of bits and it causes a large delay on the uplink of wireless network.

In this paper, we propose the interframe compression with a selective update framework for local features. We also give simple algorithms which utilize the framework. The proposed method enables gradual refresh of local features by adding a small number of independently compressed local features in every frame. The gradual refresh realizes stable low network delay because it doesn't insert large sets of independently encoded features at any given point in time.

[a]M. Makar performed the work while at Stanford University.

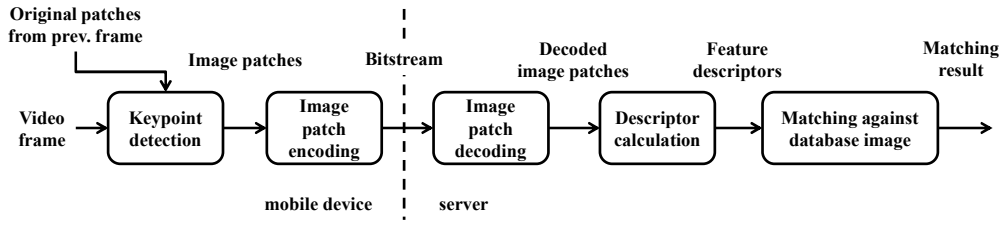


Fig. 1. Diagram of interframe compression for image patches

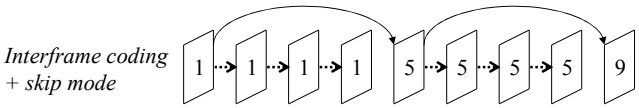


Fig. 2. Example of reference structure of interframe compression with skip mode. Same image patches are used for frames with same numbers.

The remainder of this paper is organized as follows. Section 2 gives a brief explanation on the conventional interframe compression method. Based on that, we show the selective update framework for local features and explain the details of keypoint detection algorithms for the compression method in Section 3. Section 4 shows experimental results of keypoints detection and matching performance. Finally, we conclude this paper in Section 5.

2. INTERFRAME COMPRESSION OF LOCAL IMAGE PATCHES

The interframe compression method makes use of temporal correlation between consecutive frames for keypoint detection and image patch or descriptor compression. In this paper, we focus on the patch compression architecture. Figure 1 shows the diagram of the interframe compression. First, a mobile device performs matching between original image patches in a previous frame and a current video frame to detect keypoint locations. It also extracts image patches at the locations in the current frame and compresses them using the patches from the previous frame as reference. The server decodes the compressed patches by using the side information sent from mobile devices. Finally, the server extracts descriptors from the decoded patches and performs matching with database descriptors extracted previously.

2.1. Temporally coherent keypoint detection

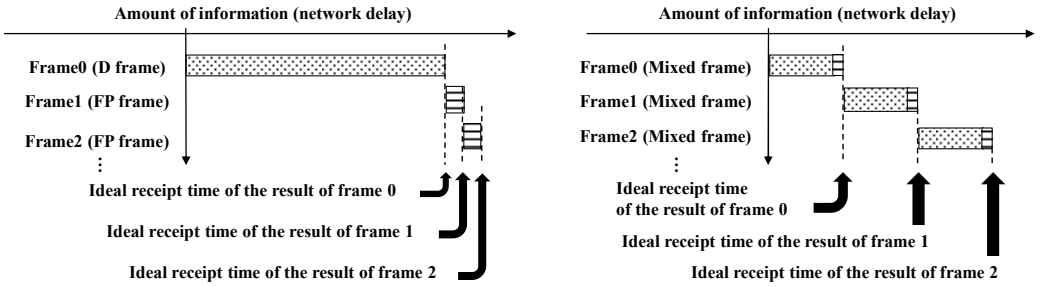
Typically, consecutive frames in videos have strong correlation of pixel values. The interframe compression method performs motion estimation between image patches in the previous frame and the current video frame to locate keypoints. This detection process is called Temporally Coherent keypoint Detection (TCD) [12].

In TCD, keypoints in the first video frame are detected independently, for example using the methods developed by the CDVS standard. The detection process is called Independent keypoint Detection (ID) and the frames where the process is applied are called Detection frames (D-frames). For consecutive frames, TCD is performed. The frames where TCD is applied are called Forward Propagation frames (FP-frames). The process finds locations, orientations and scales of keypoints in the current frame. Since this motion estimation is based on a similarity transformation, it is robust to those features, which are commonly observed when videos are recorded by mobile devices. If the sum of absolute differences (SAD) between image patches from the previous and current video frames is larger than a certain threshold, the motion estimation for the patch is considered to fail and the tracking for this patch is terminated, in order to avoid propagation of significantly different patches in the same track to following frames.

As video frames are being processed, the propagated keypoints become unable to represent the current video frame because they do not describe newly emerging local features. Therefore, interframe compression refreshes local features by inserting D-frames. First, the system performs brute force matching between image patches from ID and TCD keypoints in the current frame. Then it calculates the average SAD of the matching patch pairs. If the average is larger than a threshold, the system inserts a D-frame.

2.2. Canonical patch encoding

The image patches in the first D-frame in a sequence are compressed independently. Other D-frames performs image patch reordering in a current frame based on image patch differences to detect most similar image patches in the previous frame. Then it compresses the differences and sends them with reordering information to a server. On the other hand, the patches in an FP-frame have strong correlation with image patches from the previous frame. For FP-frames, the patch compression process performs predictive encoding between current and previous patches, similarly to video encoding architectures. The actual patch encoding process is similar to JPEG. It performs an orthogonal transform of pixel differences and uses entropy coding to compress the transformed differences. A patch size of 8x8 is used throughout this process. Note that while the TCD process uses original pixel values of previous and current image patches, the patch encoding must use decoded image patches of previous encoded and decoded patches.
Finally, drastic bitrate reductions can be achieved by



(a) Conventional interframe compression (b) Proposed method

Fig. 3. Ideal reception times of retrieval results in the interframe compression and the proposed method. The dotted bars represent required bits for independently compressed local features and the horizontal line bars represent those for temporally compressed local features.

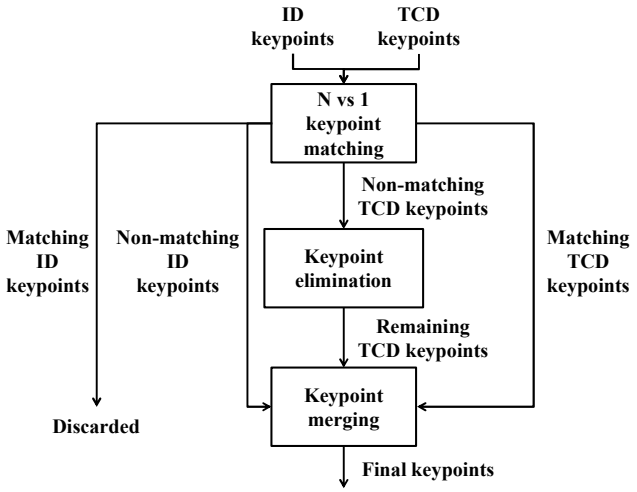


Fig. 4. Detailed flowchart of keypoint detection with selective update framework.

sending compressed image patches for only the D-frames. This mode is called *skip mode*, and is illustrated in Figure 2: the image patches from frame 1 are propagated temporally and used as local image patches from frames 2, 3 and 4, for example. In [12], it was shown that this process reduces dramatically the required bitrate while providing competitive recognition results.

3. SELECTIVE UPDATE FRAMEWORK

As explained Section 2.1, the conventional interframe compression scheme inserts the frames where keypoints and image patches are processed independently in terms of time to refresh local features when propagated local features can no longer represent accurately local features in the current frame. Independently compressed image patches need much larger bitrate when compared with temporally compressed ones. The scheme using independently compressed image patches may incur large delays before obtaining retrieval results (see Figure 3 (a)), due to lower uplink network bitrate. The insertion of independently compressed image patches mainly occurs at the moment of scene changes and appearance of new objects. This means that users have to wait longer than usual when they query video frames containing a new object.

To overcome this drawback, we introduce a selective update framework of local features which selects independent compression and interframe compression at the level of keypoints. We also show simple algorithms to make use of this framework. The selective compression enables gradual refresh by distributing independently compressed patches over several video frames. It can alleviate the maximum amount of information in the frame by gradual transmission of independently compressed patches and realize equal interval reception of the results as shown by Figure 3 (b).

3.1. Keypoint detection

We perform keypoint selection in the keypoint detection part to realize the selective update framework. First, TCD is performed using a current video frame and image patches from the previous frame. In parallel, independent keypoint detection is also processed. Then, we perform keypoint matching and elimination to determine which keypoints are to be kept. Finally, the keypoint detection part outputs the keypoint data and selection information. Image patches at the keypoint locations are compressed with the method selected in the detection part. Figure 4 shows the detail flowchart of the keypoint detection with the selective update framework. It is divided into three parts: 1) N vs 1 keypoint matching, 2) Keypoint elimination and 3) Keypoint merging.

3.1.1. N vs 1 keypoint matching

We perform the keypoint matching between ID and TCD keypoints. As results of the matching, we obtain four sorts of keypoints, *matching TCD keypoints*, *matching ID keypoints*, *non-matching TCD keypoints* and *non-matching ID keypoints*. The matching TCD keypoints avoid keypoint elimination and are merged because they represent local features in the current frame well. The matching ID

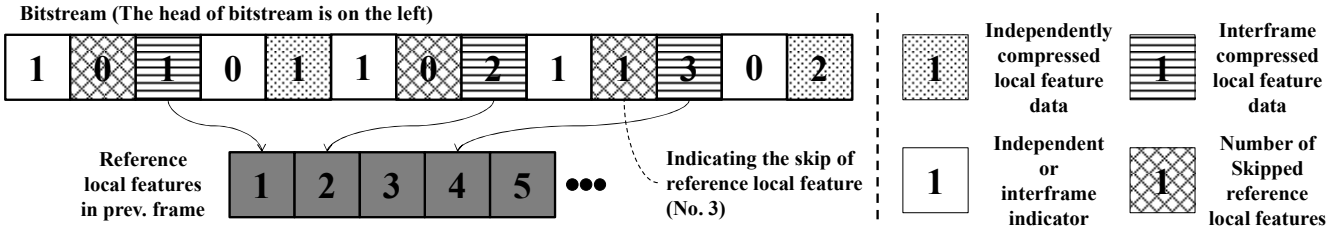


Fig. 5. Coding structure of interframe compression with selective update framework.

keypoints are discarded since they are represented well by corresponding matching TCD keypoints. The non-matching ID keypoints also avoid the elimination and are merged because they may represent local features from new objects appearing in the video. The non-matching TCD keypoints are given lower priority than other keypoints, and some of them will be eliminated in the later process.

We adopt N vs 1 matching for the matching process. The expression "N vs 1" means that the process allows multiple ID keypoints to match one TCD keypoint. The ID process sometimes detects similar keypoints in terms of vertical and horizontal positions, scale and orientation. The 1 vs 1 matching could discard only one ID keypoint and other ID keypoints would be propagated temporally. These keypoints would be accumulated at various positions in video frames and occupy a large portion of the keypoints to be sent over the network. Since the number of the keypoints to be sent is generally fixed, this would prevent the video frame from being represented with diverse keypoints. The N vs 1 matching can discard similar ID keypoints to avoid their propagation.

We perform the matching using thresholds obtained from data, determined as follows. First, we process ID and TCD in the same frame. We also perform matching between ID and TCD keypoints using Kullback-Leibler divergence and RANdom SAmple Consensus (RANSAC) [13], with an affine model, for geometry verification. This processing is done to each frames in the 6 sequences from the Stanford MAR dataset [14] and we define the obtained matching keypoint pairs as ground truth matching pairs. Using the ground truth data, we look for the best thresholds for differences of positions and orientation and ratio of scales between ID and TCD keypoints so as to maximize a receiver operating characteristic curve (ROC).

However, as for the difference of positions between ID and TCD keypoints, large fluctuations are often caused by scaling of original images. The fluctuations interfere with the keypoint matching. To deal with the problem, we replace the fixed thresholds with scale context thresholds. The scale values of ID keypoints are used for the context. The thresholds for all keypoint information are $th\_dx = th\_dy = \alpha * scale$, $th\_ds = 1.5$ and $th\_do = 0.5$ where th_dx and th_dy represent the threshold for difference of horizontal and vertical positions, respectively. The threshold th_ds and th_do are used for scale ratio and difference of orientation, respectively. We set the variable α to 1.5, based on preliminary experiments.

3.1.2. Keypoint elimination

The TCD keypoints that don't match ID keypoints in the N vs 1 matching are considered as candidates for the keypoint elimination. The elimination is executed to make space for addition of non-matching ID keypoints. In this paper, the process randomly eliminates candidate keypoints.

3.1.3. Keypoint merging

The final process is keypoint merging. It merges non-matching independently detected keypoints, matching TCD keypoints and non-eliminated TCD keypoints in descending order of Difference of Gaussian peaks. We select the top N keypoints and compress them to send to servers over the network.

3.2. Coding structure

Figure 5 shows the coding structure employed in the selective update framework. It consists of two parts:

- Coding mode indicator (1 bit)
- Number of skipped reference local features (9bit)

The part *coding mode indicator* represents which coding method is used for the processing local feature data. If the indicator indicates interframe compression, the part *number of skipped reference local features* follows. This number informs decoders of how many reference local features in the previous frame should be skipped. Though the number is represented by fixed bit length, other encoding methods can obviously achieve better compression performance. In Figure 5, since the first *coding mode indicator* represents interframe compression, the *number of skipped reference local features* part follows. The second *coding mode indicator* represents the independent compression method. In this case, local feature data follows. The case of the third local feature is the same as the first one. The fourth indicator represents the interframe compression and the following number shows that one feature is to be skipped. In this case, one reference local feature (the third feature in reference local feature) is skipped and the fourth reference local feature is used for compression.

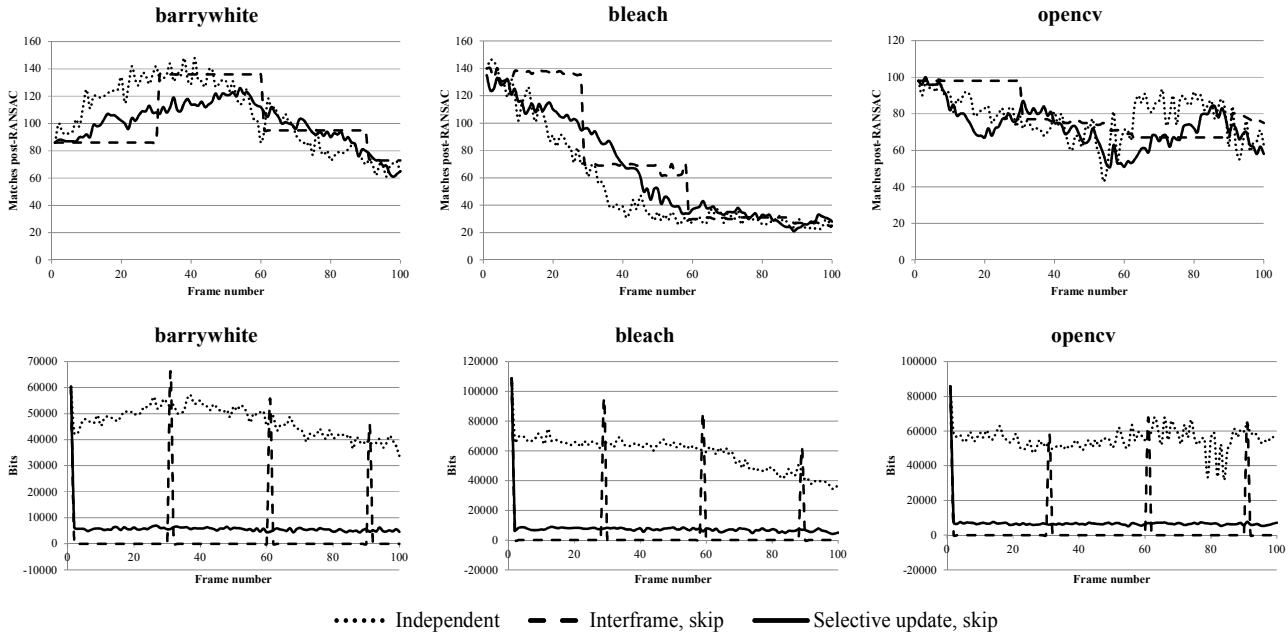


Fig. 6. Numbers of post-RANSAC matches (top row) and bitrates (bottom row) of all methods in 3 sequences.

4. EXPERIMENTAL RESULTS

We conduct matching between query videos and database images to evaluate the performance of the proposed method. We use 6 video sequences ("barrywhite", "bleach", "opencv", "polish", "titanic" and "wangbook") from Stanford MAR Dataset [14]. These videos are obtained by recording book and CD covers, a detergent container and a shoe polish sponge. Since these videos are recorded using a handheld smartphone, they include different amounts of translation, scaling and rotation. Every video sequence consists of 100 frames. The recording condition is 30 fps with 640x480 resolution. In addition, we calculate descriptors from database images for the matching process.

We compare 3 methods described below.

1. Independent compression
2. Interframe compression + skip
3. Selective update framework + skip (proposed method)

The experimental setup is common to all methods. First, we extract keypoints from query videos and compress the image patches at the obtained keypoint locations. On the server, we decode the compressed image patches and extract descriptors from the decoded patches. Finally, we perform matching between query descriptors and database descriptors. We evaluate the matching performance by bitrates and numbers of matches after geometry verification. We use RANSAC for geometry verification. In these experiments, we set the maximum number of compressed keypoints N to 400 and quantization parameter used in the image patch compression to 8. We set the number of independently compressed local features to be added in every frame to 25 in the proposed method. By eliminating temporally propagated local features to the number and refilling dependent compressed local features in every frame, we realize the gradual refresh of local features and can control the bitrates of compressed image patches.

4.1. Performance of selective update framework

Figure 6 shows the number of post-RANSAC matches (first row) and bitrates (second row) of all methods in "barrywhite", "bleach" and "opencv" sequences.

Table 1. Bitrates, maximum amount of information in one frame (max bits) and averages of numbers of post-RANSAC matches for all methods in all sequences.

	Independent			Interframe + skip			Selective update + skip		
Sequence	Bitrate (kbps)	Max bits (kb)	Avg. Matches	Bitrate (kbps)	Max bits (kb)	Avg. Matches	Bitrate (kbps)	Max bits (kb)	Avg. Matches
barrywhite	1414.0	57.2	107.0	68.5	66.3	102.4	184.1	6.9	99.8
bleach	1753.5	74.8	54.6	105.6	95.6	71.0	243.9	8.9	64.6
opencv	1658.8	68.1	77.9	83.0	67.8	79.8	219.4	7.7	72.7
polish	1355.1	54.0	37.4	70.4	65.9	34.3	190.4	6.9	37.3
titanic	1608.1	69.8	76.4	78.1	67.9	72.7	203.7	7.8	65.8
wangbook	1995.5	79.0	46.4	136.0	91.7	38.2	231.9	8.2	36.2

The horizontal axis shows the frame number. The results of the proposed, independent and interframe compression methods are shown by solid, dotted and dashed lines, respectively. As stated in Section 2.1, the interframe compression inserts independent compressed frames to refresh local features when they can't represent the state of local features in the current frame. This process causes the extremely large peak bits observed for the interframe compression. Moreover, when the insertion happens, the numbers of matches abruptly becomes equal to those of independent compression. On the other hand, with respect to the number of matches, the results of proposed method can smoothly track those of the interframe compression with slight bit consumption. Table 1 shows the bitrates, maximum amount of information in one frame (max bits) and the averages of the number of post-RANSAC matches of all methods in all sequences. The independent compression scheme produces the largest value among three methods in terms of the bitrate and the max bits. The interframe compression utilizes temporal correlation over frames and drastically reduces the bitrate compared with independent compression with minor drops of the numbers of post-RANSAC matches. However the values of the max bits are about the same as those of the independent compression because of the insertion of independently compressed features. Note that since the interframe coding employs predictive patch compression with reordering information for D-frames in the middle of a sequence, the values of max bits don't match those of the independent coding. The proposed method needs slightly higher bitrates compared with the interframe compression since it updates the small number of local features every frame to refresh gradually. However, the gradual refresh can free us from the insertion of independent compressed frames, and realize the 10 times reduction of max bits without large drops of the matching performance.

5. CONCLUSION

We propose a selective update framework for the interframe compression of local features for mobile augmented reality systems and give simple algorithm to utilize the framework. The proposed method realizes gradual refreshment of local features by selecting temporally compressed local features to be eliminated and adding independently compressed local features in place of the eliminated ones every frame. The selection is performed by matching between independent and temporally detected local features and by evaluating the priority of matching and non-matching local features.

Experimental results show that the proposed method realizes a 10 times reduction of maximum amount of information in one frame with slight bitrate increases for gradual refreshment with comparable matching performance.

6. REFERENCES

[1] Google-Glass, 2013, http://www.google.com/glass/start.

[2] D. Lowe, "Distinctive Image Features From Scale-Invariant Keypoints," *Int'l Journal of Computer Vision*, vol. 60, pp. 91-110, Nov. 2004.

[3] H. Bay, T. Tuytelaars, and L. V. Gool, "SURF: Speeded Up Robust Features," *Proc. of the European Conf. on Computer Vision*, pp. 404-417, 2006.

[4] V. Chandrasekhar, G. Takacs, D. Chen, S. Tsai, R. Grzeszczuk and B. Girod, "CHoG: Compressed histogram of gradients," *Proc. of the IEEE Int'l Conf. on Computer Vision and Pattern Recognition*, pp.2504-2511, Jun. 2009.

[5] C. Yeo, P. Ahammad and K. Ramchandran, "Rate-efficient visual correspondences using random projections," *Proc. of the IEEE Int'l Conf. on Image Processing*, pp. 217-220, Oct. 2008.

[6] Y. Weiss, A. Torralba and R. Fergus, "Spectral Hashing," *Proc. of Neural Information Proccesing. Systems*, pp. 1753-1760, Dec. 2008

[7] V. Chandrasekhar, G. Takacs, D. Chen, S. Tsai, J. Singh and B. Girod, "Transform coding of image feature descriptors," *Proc. of Visual Communications and Image Processing Conf.*, pp. 725710-725710-9, Jan. 2009.

[8] D. Nister and H. Stewenius, "Scalable Recognition with a Vocabulary Tree," *Proc. of IEEE Conf. on Computer Vision and Pattern Recognition*, pp. 2161-2168, Jun. 2006.

[9] H. Jegou, M. Douze and C. Schmid, "Product Quantization for Nearest Neighbor Search," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, Vol. 33, no. 1, pp. 117-128, Jan. 2011.

[10] B. Girod, V. Chandrasekhar, R. Grzeszczuk and Y. Reznik, "Mobile Visual Search: Architectures, Technologies, and the Emerging MPEG Standard," *IEEE Multimedia*, vol. 18, no. 3, pp. 86-94, Mar. 2011.

[11] M. Makar, C.-L. Chang, D. Chen, S. Tsai and B. Girod, "Compression of Image Patches for Local Feature Extraction," *Proc. of the IEEE Int'l Conf. on Acoustics, Speech and Signal Processing*, pp. 821-824, 2009.

[12] M. Makar, S. Tsai, V. Chandrasekhar, D. Chen and B. Girod, "Interframe Coding of Canonical Patches for Mobile Augmented Reality," *Proc. of the IEEE Int'l Symposium on Multimedia*, pp. 50-57, Dec. 2012.

[13] M. Fischler and R. Bolles, "Random Sample Consensus: A Paradigm for Model Fitting with Applications to Image Analysis and Automated Cartography," *Communications of the ACM*, vol. 24, issue 6, pp. 381-395, Jun. 1981.

[14] M. Makar, A. Araujo and D. Chen, "Stanford Streaming Mobile Augmented Reality Dataset," Aug. 2013. http://purl.stanford.edu/ph459zk5920.