# DashCam Video Compression using Historical Data

Biao Ma and Amy R. Reibman
School of Electrical and Computer Engineering
Purdue University
West Lafayette, Indiana, USA

***Abstract*—While dashcam videos (DCVs) are used to document unanticipated situations such as accidents, they are often used to create a training set for vehicle detection and vehicle behavior modeling. This requires a substantial volume of stored DCVs. In this paper, we propose a system that effectively compresses these DCVs by taking advantage of existing historical data. First, our video retrieval and alignment preprocessors construct a reference video for a new DCV based on GPS information and ORB (Oriented FAST and Rotated BRIEF) features. The 3D-HEVC encoder jointly compresses these two videos. Our illumination matching algorithm makes the system robust over different illumination conditions. Our system reduces the bit-rate around 30% when tested over 80 sequences and 3 different scenarios: highway, boulevard and downtown.**

***Index Terms*—Dashcam video, video compression, 3D-HEVC**

## I. Introduction

Dashcams, the in-vehicle cameras mounted on dashboards, are designed to record videos that can provide evidence about traffic accidents. As a part of intelligent vehicle research, data-driven approaches [1], [2] use dashcam videos (DCVs) as training samples to detect vehicles or model vehicle behavior. However, to model complex traffic and vehicle behavior, a substantial amount of DCVs is necessary; these can be collected by vehicle-video-sharing [3] or vehicle-to-broadband cloud (V2B) platforms [4]. For these platforms, an effective DCV compression approach is desired. Our work addresses this problem while other vehicle video compression techniques [5], [6] mainly focus on on-road surveillance videos rather than DCVs.

When a vehicle travels the same route on two occasions, the two DCVs contain both static objects and new or moving objects. Because vehicles may have the same location in different DCVs, the static objects may be similar, which makes the corresponding frames highly correlated. As a result, instead of compressing these videos separately, it is reasonable to consider compressing correlated videos jointly.

Recently, multi-view video coding tools (MV-HEVC and 3D-HEVC) [7] have been designed for efficiently encoding multi-view videos. Multi-view videos are videos that are recorded simultaneously by different cameras. DCVs are not multi-view videos, so it is difficult to use MV-HEVC or 3D-HEVC directly. First, unlike in multi-view videos, static objects in DCVs may be recorded under different illumination and the relative viewpoints of different DCVs may not be fixed. Second, users may drive at a different speed, so the corresponding frames will not match temporally. Our goal is to take advantage of the existing framework of MV-HEVC/3D-HEVC as much as possible, by applying several preprocessors to create a specific sequence that matches illumination condition and contents across the timeline for the video to be encoded. Specifically, there are two tasks: finding related video pairs and finding their temporal relationships.

There is little research that considers these two tasks applied to dashcam videos. However, recent research in near duplicated videos (NDVs) can provide some useful ideas for both two tasks. NDVs [8] are generated by replicating or editing a video in different ways, such as changing frame order, using different compression parameters or modifying image scales. Most work on NDVs [9]–[12] concentrate on key-frame-based video retrievals. Given the results, not only is a matched video found but also a general timeline alignment can be created. This is because the changes between key-frames are based on temporal resampling, such as equally-spaced sampling (e.g. halving or doubling the frame rate). These research apply global features [9], [10] or quantized local features [11], [12] to achieve fast retrieval. In addition, video compression of NDVs [12] shows that a light regulator and homographic transformations performs well for NDVs.

However, the contents of DCVs vary due to the changing vehicle speed, weather conditions and driving routes. As a result, unlike the relationship between NDVs, the correspondence between two DCVs cannot be modeled using regular temporal resamplings. Thus, key-frame-based algorithms cannot provide an accurate timeline alignment. Also, DCVs can be difficult to distinguish temporally since many similar on-road scenes do not have distinguishing features. So global features or quantized local features cannot provide a confident video retrieval. For these two reasons, the approach in [12] does not perform well when compressing DCVs.

Instead of using local features, some algorithms for video alignment or video synchronization provide different ideas for timeline alignment. [13], [14] are based on image intensity and use heavily downsampled images as the descriptors to save computations. This enables the application of global optimization algorithms. In [13], a dynamic Bayesian network is used to find an optimal correspondence relationship between frames. [14] tries to find the relationship by minimizing the cost function based on image difference.

Note that [13], [14] are based on a strong condition that the relative viewing angle between different videos cannot be too large and the vehicle must maintain the same lane. When this is not satisfied, experimental results demonstrate our system has significantly improved compression efficiency

978-1-5090-5966-9/16/$31.00 ©2016 IEEE

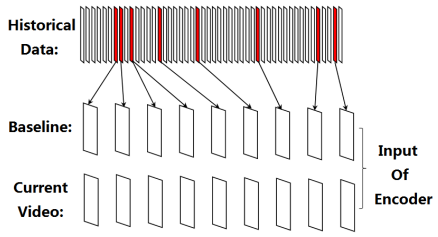



Fig. 1: Generation of the baseline video

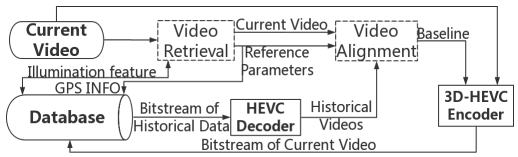



Fig. 2: Our proposed DCV compression system

relative to the matching algorithm in [14]. Further, we have a different goal. [13] uses an interval containing 3 to 6 frames to measure the goodness of each matching result. If the matching result falls into the interval, it is considered a true correspondence. However, we are interested in finding an exact frame correspondence that gives the highest compression efficiency.

To compress DCVs, we propose a video retrieval and alignment system based on ORB (Oriented FAST and Rotated BRIEF) [15]. In the next section, the global system is illustrated. A GPS-based video retrieval model and an ORB-based video alignment preprocessor are proposed in section III. Also, the illumination model based on 3D-HEVC and a novel illumination matching algorithm are described. Experimental results are shown and analyzed in section IV. Finally, we will conclude our work in section V.

## II. System Description

The general idea of our system is to assemble a set of frames (called the "baseline") from a collection of previously encoded videos (the "historical data") so the baseline can be used to effectively predict the current video (See Fig. 1). The content of baseline frames should match the current frame as much as possible. Instead of being aligned at a pixel level, the corresponding frames only need to have objects of nearly similar scales.

The whole structure of our system is shown in Fig. 2. The two preprocessors, video retrieval and video alignment, are applied to identify correlations between DCVs. The video retrieval preprocessor is based on GPS information provided by the dashcam. Searching in the database, it identifies reference video sequences that have identical location and similar illumination condition as the current video. Given the parameters (the video index and frame interval) of the reference video sequence, the video alignment preprocessor produces the baseline video which corresponds on a frame-by-frame basis with the initial incoming video. Then, the system uses a 3D-HEVC encoder to jointly encode the baseline and current video. Finally, side information generated by video alignment

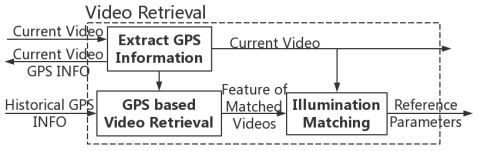



Fig. 3: The Video Retrieval Preprocessor

preprocessor, the GPS information and the compressed bitstream of the current video are stored. The decoder accesses the same database.

The 3D-HEVC encoder assumes the videos in both views share a common timeline, and enables predictions between frames that are at the same time instances. In our case, the baseline is view 0 and the current video is encoded as view 1 using inter or intra predictions and interview predictions. We apply 3D-HEVC in our system, so that we can take advantage of its illumination compensation algorithm.

## III. Proposed DCV Compression Method

In this section, we describe in detail our two preprocessors: video retrieval and video alignment.

### A. Video Retrieval

Our video retrieval pre-processor selects the best reference video. First, the GPS-based retrieval finds all videos for the same route. Then our illumination algorithm searches for the best reference for the 3D-HEVC illumination compensation model.

*1) GPS-based Retrieval:* The video retrieval preprocessor shown in Fig. 3 uses the GPS information that was recorded along with the current video to query within the database for reference video sequences that have similar physical locations. Our database is based on a local map divided into several geographic regions, which allows multi-thread computations. The size of every region is 4 $km^2$ as suggested by [16]. When new videos are captured, their GPS information is extracted on a per-second basis and segmented based on the geographic regions. For all the GPS information in each region, we search the matching sequences amid the database.

*2) Illumination Matching:* Methods based on GPS or local features cannot provide any information about the illumination conditions which can dramatically influence the color appearance of the objects. Consequently, block matching may not find an effective result among the outputs of the GPS-based retrieval. To solve this problem, we adopt two methods.

First, we adopt an illumination compensation method. 3D-HEVC includes a block-based illumination compensation model [17] which uses a scale parameter and an offset parameter to compensate the reference block. These parameters are calculated using the rows and the columns just before the reference and current blocks. However, the block matching algorithm is prior to the compensation. When the illumination difference is large, there is a high probability that objects are mismatched during this compensation. These mismatchings have the minimum displaced frame difference (DFD). However, in these cases, illumination compensation is unlikely to further reduce the displaced frame difference. As a result, the

978-1-5090-5966-9/16/$31.00 ©2016 IEEE

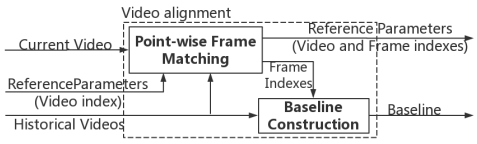



Fig. 4: The Video Alignment Preprocessor

compensation model in 3D-HEVC only works well when the illumination difference is small.

Because our application creates videos that may have significantly different illumination conditions, applying only the 3D-HEVC illumination compensation algorithm is not very effective. Therefore, secondly, we design an illumination matching algorithm, applied prior to 3D-HEVC compression, to leverage its illumination compensation algorithm.

For the trip in each region, there are likely to be multiple outputs for the GPS-based retrieval if there are several historical videos. Based on these outputs, our illumination matching algorithm narrows the results down to a single matched video. It uses global features to identify a reference video with the most similar illumination conditions to the current video. For convenience, we demonstrate our algorithm within one geographic region.

We assume the current video has matching locations with several videos stored in the database since our compression system is based on a continually growing database. Since these videos are captured at the same location, the reflectivity of each immobile object's surface is approximately unchanged. Similar illumination conditions produce similar color distributions, so a feature based on color distributions is useful.

We adopt the statistical mean and the contrast feature proposed in [18] as the descriptor. It is calculated as:

$$C = \sigma/(\alpha_4)^{\frac{1}{4}}$$

where $\sigma$ is the standard deviation and $\alpha_4$ is the fourth moment of the pixel values.

To incorporate the geometric information of the light intensity, each frame is equally divided into four quadrants. Frames are divided into upper and lower quadrants to eliminate the influence of the clouds and sun, since the clouds and sun only appear in the sky. Frames are divided into left and right quadrants to separately describe the position of shadows and sun. The average statistic histogram of every quadrant is calculated over 300 frames. Then the feature of this sequence is constructed using the mean and the contrast of every quadrant.

### B. Video Alignment

As shown in Fig. 1, once a single matching video has been selected, we need to complete a timeline alignment which ensures every frame in the current video is matched with a historical frame (chosen from the reference sequence) that is the best reference for it. Fig. 4 shows the video alignment preprocessor. It uses a point-wise frame matching algorithm based on ORB [15] to complete a precise timeline alignment. Using the frame indexes of the matching procedure and the decoded reference video, the baseline video is constructed. Some earlier works also applied local-feature-based homographic transformations on the matching frames, which we show here is not effective for our application.

*1) Timeline Alignment:* Since the GPS only can provide 1 Hz location information for DCVs, the local visual feature is used to construct a matching sequence that precisely aligns with the current video temporally.

Our goal is to find a reference frame that gives the highest compression rate for the current frame. To achieve this, the objects' scale in the reference frame should be similar to the ones in the current frame. We use the number of matching points between two frames to measure their scale similarity. In the GPS and illumination matched video, the frame that has the most matching points is selected to be the reference frame for a current frame. This is because when similar scale is achieved, the resolution is similar. And key-points are matched only when the resolution is similar. The more area has similar scale, the more matching points the image pair has.

However, not all local features are equally effective. Large-patch-based local features such as ORB have higher accuracy. Scale-invariant features have lower accuracy, such as SIFT [19]. Although scale-invariance increases features' robustness, it also increases the number of matching points that have dissimilar scale. Thus, we select the ORB feature since it has a larger patch-size and is not completely scale-invariant [15]. We use RANSAC [20] to remove unstable matches.

The timeline alignment need not be real-time in our application. Our system can first compress the incoming video with low QP using standard HEVC. All preprocessings and the joint encoding process can be finished offline, which also enables our algorithm to be applied when many options with similar illumination are available.

*2) Image Alignment:* To reduce the bit-rate of compressing one NDV using another, [12], [21] align matching frames at the pixel level using one or more homographic transformations, with the goal of reducing both the motion vectors and the displaced frame difference (DFD). However, our performance results shown below demonstrate that this approach usually decreases the performance of our system by about 4%. The estimation of the transformation is strongly influenced by local feature matching and tends to be inaccurate for DCVs. Instead, to reduce the DFD, it is more important that as many objects as possible have the same scale in each frame; this is ensured by our timeline alignment and does not require a homographic transformations.

## IV. Experiments and Results

### A. Experimental setup

To test our system, we built a DCV database that contains 80 videos recorded in 3 scenarios: boulevard (24 videos), highway (35 videos) and downtown (21 videos). Videos of the same scenario were captured under 4 illumination conditions: cloudy days, and clear days' morning, noon and afternoon. All of the DCVs are recorded by KDLINKS X1 Digital Recorder with resolution $1920 \times 1080$.

We process each video with the video retrieval and alignment preprocessors. To show the compression results, we

978-1-5090-5966-9/16/$31.00 ©2016 IEEE

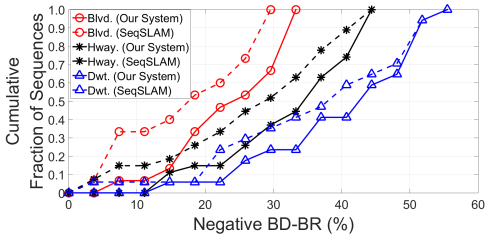



Fig. 5: The bit-saving across sequences

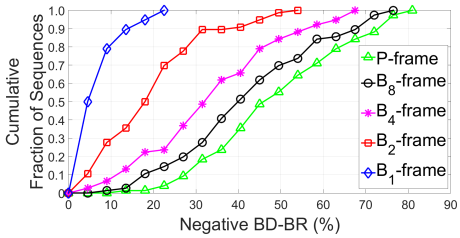



Fig. 6: The bit-saving based on distance to reference

use 300 frames of each output video of the video retrieval preprocessor. Each video is predicted using one historical video which is chosen by our illumination matching algorithm. To encode the DCVs, we use the version of 16.7 of test model of HEVC [22] and its 3D extension [23]. We encode each video using quantization parameters, 30, 35, 40, 45, and use BD-rate [24] to quantify the results.

The 80 videos are divided into 2 test sets. Test set I includes videos recorded on different days (21 videos for boulevard, 31 videos for highway and 17 videos for downtown). Test set II includes the remaining video pairs recorded at times that differ by an only few minutes. This ensures that the road-side objects and illumination condition are nearly identical.

### *B. Performance of our system*

Compared to standard HEVC, the performance of our system over test set II are 62.7% (downtown), 65.0% (highway) and 53.4% (boulevard). Since the difference between the reference video and current video are minimum, this is the upper-bound performance.

Fig. 5 shows the performance of our system under different scenarios using test set I. The x-axis represents the percentage of bit-saving in BD-rate. Every point in this figure represents the fraction of sequences on which our system does not exceed the corresponding percentage of bit-saving. Performance improves to the right. It shows that our algorithm performs well in all three scenarios, although it performs better in the downtown and highway scenario than in the boulevard scenario. The average amount of bit-saving for the downtown, highway and boulevard scenarios are 39.7%, 31.8% and 23.5% respectively. The downtown scenario has the highest performance because it is insensitive to illumination conditions. It has small sky area and a regular shadow shape, which is easy for the illumination compensation module. The boulevard scenario has the lowest performance. It is more sensitive to illumination conditions and has a large sky area like the highway scenario and a more complex shadow. It requires more videos to construct a dense database so that the best reference video can have a similar enough illumination condition to the current video.

Fig. 6 shows the average performance of our system for each type of frame using all videos. Here the subscript denotes the distance away from the reference frame in the current view. The average efficiency for P, $B_8$, $B_4$, $B_2$ and $B_1$ frames are 48.1%, 41.5%, 32.9%, 18.6% and 6.0%. The further the frame is away from the reference frame in the current video, the higher the compression efficiency. This is because for those frames far away from the reference frame in the current video, the aligned reference frame in baseline can provide a better prediction. However, if the timeline alignment is not optimal, the compression efficiency will decrease correspondingly.

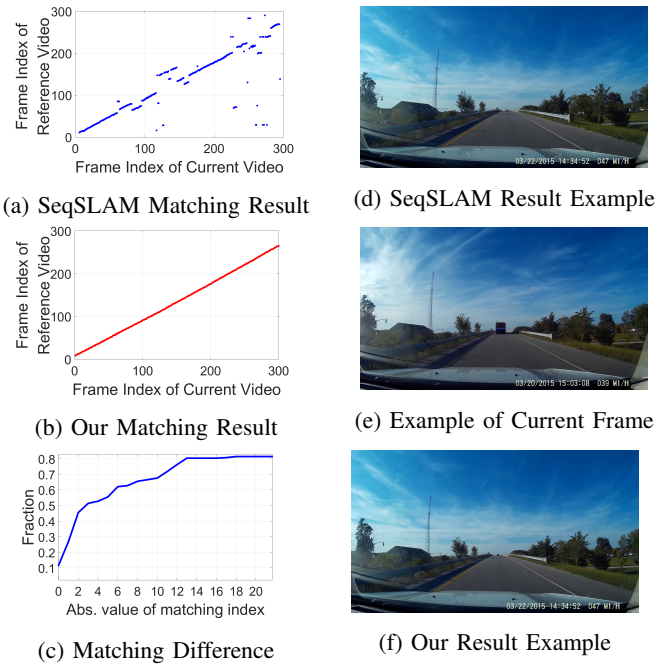
### *C. Comparison to other methods*

In Fig. 5, we also present the performance using SeqSLAM [14], which uses image intensity for timeline alignment. Its average performance for the downtown, highway and boulevard scenarios are 35.1%, 25.5% and 16.7%. The main reason that our algorithm outperforms SeqSLAM is that we have different definitions of "best matching". We define the best matching frame to be the one with the most similar scale as the current frame. SeqSLAM chooses the frame that has the smallest difference to the current frame as the optimal one. These two approaches are the same only if the difference between viewing angles is quite small. In Fig. 5, our algorithm is much better than SeqSLAM when the compression efficiency is lower. This is because the strong assumption in SeqSLAM is not satisfied in these cases because the viewing angles are not quite similar. Although our algorithm also cannot achieve high performance in these cases, it is more powerful than SeqSLAM when viewing angles are not quite similar. For cases which have similar viewing angles, our algorithm and SeqSLAM both have high and similar compression efficiency. SeqSLAM performs as well as our algorithm for only 3 sequences in test set I, although the upper-bound performance on test set II are nearly identical.

Fig. 7a and Fig. 7b show the timeline alignment results of our algorithm and SeqSLAM on a sample sequence from the highway scenario. In this sequence, the viewing angles are a little different, which makes the result of SeqSLAM unstable. This also causes the two algorithms to have quite different results. We take the difference of these two matching results and plot the distribution in Fig. 7c. Among 300 frames, around 50% of frames have more than a 4 frames difference. And around 20% of frames have more than a 20 frames difference. Note that 4 frames is already a large distance for video compression. Because the timeline alignment of SeqSLAM is not optimal, its compression efficiency is lower than ours.

Figs. 7d to 7f show an example of matching frames for a P-frame. The frame selected by our algorithm shares the same scale across most of the frame, while the frame selected by SeqSLAM matches well only on the left side of the frame. Large differences appear on the right side, which requires more bits for the DFD. For this sequence, our algorithm achieves 29.4% average compression improvement while SeqSLAM

978-1-5090-5966-9/16/$31.00 ©2016 IEEE




(a) SeqSLAM Matching Result

(b) Our Matching Result

(c) Matching Difference

(d) SeqSLAM Result Example

(e) Example of Current Frame

(f) Our Result Example

Fig. 7: Matching Results of SeqSLAM and Our algorithm

achieves 21.0%. For this specific frame, our algorithm achieves a 49.7% savings while SeqSLAM only achieves 37.1%.

If we use homographic transformations discussed in section III-B2, the performance of our system decreases 4.2% (downtown), 4.7% (highway) and 2.2% (boulevard) while SeqSLAM decreases 2.7%, 2.9% and 1.0%. The inaccurate matches of SeqSLAM are less affected by the homographic transform than our more accurate matches are.

## V. Conclusion

In this work, we construct a DCV compression system that takes advantages of the historical data recorded from previous trips on the same route. The video retrieval and alignment preprocessors we design to construct a reference video using the historical data enables us to leverage the power of the 3D-HEVC standard. Our experiments show that significant compression improvements are achieved when DCVs are jointly compressed with a carefully-formed reference video. On average, we achieve around 30% bit-rate savings.

While we described the system assuming it would be used by a single vehicle, driving repeated routes on different days, our compression framework is also relevant for multiple-vehicle systems. For example, it can be used by a bus company to archive DCVs on a regular route.

We will continue to refine the preprocessors to improve system efficiency. For example, we will add adaptive models to decide which part of the video is necessary to be processed in order to save more computations. We also consider improvements to the illumination compensation model to make it effective across a broad range of conditions.

## References


[1] S. Sivaraman, B. Morris, and M. Trivedi, "Learning multi-lane trajectories using vehicle-based vision," in *IEEE International Conference on Computer Vision Workshops (ICCV Workshops)*, 2011, pp. 2070–2076.

[2] H. Cho and S.-Y. Hwang, "High-performance on-road vehicle detection with non-biased cascade classifier by weight-balanced training," *EURASIP Journal on Image and Video Processing*, vol. 2015, no. 1, pp. 1–7, 2015.

[3] C.-Y. Chiang, S.-M. Yuan, S.-B. Yang, G.-H. Luo, and Y.-L. Chen, "Vehicle driving video sharing and search framework based on GPS data," in *Genetic and Evolutionary Computing*. Springer, 2014, pp. 389–397.

[4] M. Faezipour, M. Nourani, A. Saeed, and S. Addepalli, "Progress and challenges in intelligent vehicle area networks," *Communications of the ACM*, vol. 55, no. 2, pp. 90–100, 2012.

[5] J. Xiao, L. Liao, J. Hu, Y. Chen, and R. Hu, "Exploiting global redundancy in big surveillance video data for efficient coding," *Cluster Computing*, vol. 18, no. 2, pp. 531–540, 2015.

[6] X. Zhang, T. Huang, Y. Tian, and W. Gao, "Overview of the IEEE 1857 surveillance groups," in *IEEE International Conference on Image Processing (ICIP)*, 2013, pp. 1505–1509.

[7] G. Tech, Y. Chen, K. Müller, J.-R. Ohm, A. Vetro, and Y.-K. Wang, "Overview of the multiview and 3D extensions of High Efficiency Video Coding," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 26, no. 1, pp. 35–49, 2016.

[8] J. Liu, Z. Huang, H. Cai, H. T. Shen, C. W. Ngo, and W. Wang, "Near-duplicate video retrieval: Current research and future trends," *ACM Computing Surveys (CSUR)*, vol. 45, no. 4, p. 44, 2013.

[9] S.-C. S. Cheung and A. Zakhor, "Fast similarity search and clustering of video sequences on the world-wide-web," *IEEE Transactions on Multimedia*, vol. 7, no. 3, pp. 524–537, 2005.

[10] L. Shang, L. Yang, F. Wang, K.-P. Chan, and X.-S. Hua, "Real-time large scale near-duplicate web video retrieval," in *Proceedings of the international conference on Multimedia*, 2010, pp. 531–540.

[11] C.-L. Chou, H.-T. Chen, C.-C. Hsu, C.-P. Ho, and S.-Y. Lee, "Near-duplicate video retrieval by using pattern-based Prefix tree and temporal relation forest," in *IEEE International Conference on Multimedia and Expo (ICME)*, 2014, pp. 1–6.

[12] H. Wang, M. Ma, and T. Tian, "Effectively compressing Near-Duplicate Videos in a joint way," in *IEEE International Conference on Multimedia and Expo (ICME)*, 2015, pp. 1–6.

[13] F. Diego, D. Ponsa, J. Serrat, and A. M. López, "Video alignment for change detection," *IEEE Transactions on Image Processing*, vol. 20, no. 7, pp. 1858–1869, 2011.

[14] M. J. Milford and G. F. Wyeth, "SeqSLAM: Visual route-based navigation for sunny summer days and stormy winter nights," in *IEEE International Conference on Robotics and Automation (ICRA)*, 2012, pp. 1643–1649.

[15] E. Rublee, V. Rabaud, K. Konolige, and G. Bradski, "ORB: an efficient alternative to SIFT or SURF," in *IEEE International Conference on Computer Vision (ICCV)*, 2011, pp. 2564–2571.

[16] G. Schroth, R. Huitl, D. Chen, M. Abu-Alqumsan, A. Al-Nuaimi, and E. Steinbach, "Mobile visual location recognition," *IEEE Signal Processing Magazine*, vol. 28, no. 4, pp. 77–89, 2011.

[17] H. Liu, J. Jung, J. Sung, J. Jia, and S. Yea, "3D-CE2. h: Results of illumination compensation for inter-view prediction," *ITU-T SG16 WP3 and ISO/IEC JTC1/SC29/WG11 JCT3V-B0045*, 2012.

[18] H. Tamura, S. Mori, and T. Yamawaki, "Textural features corresponding to visual perception," *IEEE Transactions on Systems, Man and Cybernetics*, vol. 8, no. 6, pp. 460–473, 1978.

[19] D. G. Lowe, "Distinctive image features from scale-invariant keypoints," *International journal of computer vision*, vol. 60, no. 2, pp. 91–110, 2004.

[20] M. A. Fischler and R. C. Bolles, "Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography," *Communications of the ACM*, vol. 24, no. 6, pp. 381–395, 1981.

[21] Z. Shi, X. Sun, and F. Wu, "Photo album compression for cloud storage using local features," *IEEE Journal on Emerging and Selected Topics in Circuits and Systems*, vol. 4, no. 1, pp. 17–28, 2014.

[22] Joint Collaborative Team on Video Coding (JCT-VC), "HEVC Test Model," https://hevc.hhi.fraunhofer.de/svn/svn_HEVCSoftware/.

[23] Joint Collaborative Team on Video Coding (JCT-VC), "3D-HEVC Test Model," https://hevc.hhi.fraunhofer.de/svn/svn_3DVCSoftware/.

[24] G. Bjontegaard, "Calcuation of average PSNR differences between RD-curves," *Doc. VCEG-M33 ITU-T Q6/16, Austin, TX, USA, 2-4 April 2001*.


978-1-5090-5966-9/16/$31.00 ©2016 IEEE