remote sensing

MDPI

Article

Towards Real-Time Service from Remote Sensing: Compression of Earth Observatory Video Data via Long-Term Background Referencing

Jing Xiao [1,2,3], Rong Zhu [1,3], Ruimin Hu [1,4], Mi Wang [3], Ying Zhu [3], Dan Chen [2,*] and Deren Li [3]

1 National Engineering Research Center for Multimedia Software, School of Computer Science, Wuhan University, Wuhan 430072, China; jing@whu.edu.cn (J.X.); zhurong@whu.edu.cn (R.Z.); hrm@whu.edu.cn (R.H.)
2 Research Institute of Wuhan University in Shenzhen, Shenzhen 518000, China
3 Collaborative Innovation Center of Geospatial Technology, Wuhan University, Wuhan 430072, China; wangmi@whu.edu.cn (M.W.); yzhu1003@whu.edu.cn (Y.Z.); drli@whu.edu.cn (D.L.)
4 Hubei Key Laboratory of Multimedia and Network Communication Engineering, Wuhan 430079, China
* Correspondence: dan.chen@whu.edu.cn; Tel.: +86-027-6875-2037

Received: 23 April 2018; Accepted: 4 June 2018; Published: 5 June 2018

Abstract: City surveillance enables many innovative applications of smart cities. However, the real-time utilization of remotely sensed surveillance data via unmanned aerial vehicles (UAVs) or video satellites is hindered by the considerable gap between the high data collection rate and the limited transmission bandwidth. High efficiency compression of the data is in high demand. Long-term background redundancy (LBR) (in contrast to local spatial/temporal redundancies in a single video clip) is a new form of redundancy common in Earth observatory video data (EOVD). LBR is induced by the repetition of static landscapes across multiple video clips and becomes significant as the number of video clips shot of the same area increases. Eliminating LBR improves EOVD coding efficiency considerably. First, this study proposes eliminating LBR by creating a long-term background referencing library (LBRL) containing high-definition geographically registered images of an entire area. Then, it analyzes the factors affecting the variations in the image representations of the background. Next, it proposes a method of generating references for encoding current video and develops the encoding and decoding framework for EOVD compression. Experimental results show that encoding UAV video clips with the proposed method saved an average of more than 54% bits using references generated under the same conditions. Bitrate savings reached 25–35% when applied to satellite video data with arbitrarily collected reference images. Applying the proposed coding method to EOVD will facilitate remote surveillance, which can foster the development of online smart city applications.

Keywords: big surveillance video data; high efficiency compression; redundancy across videos; background; moving objects

1. Introduction

Dynamic Earth observatory video data (EOVD) has enabled many innovative smart city applications (e.g., smart transportation, sewage disposal monitoring, and disaster management). Remote surveillance via unmanned aerial vehicles (UAVs) and video satellites has become a new trend in smart city development. However, receiving the EOVD immediately after its capture in order to meet the real-time demands of dynamic remote sensing data analysis and service in smart cities is a key problem.

The substantial gap between the EOVD data collection rate and the transmission bandwidth has greatly restricted remote surveillance applications in smart cities. Taking satellite Jilin-1 as an example, a single frame of satellite video data is about 12,000 × 5000 pixels with frame rate of 15 fps, resulting in 20 Gbps of video data. However, the transmission channel for real-time transmission between satellites and the Earth is only 10–20 Mbps. Even with the latest coding standard—high-efficiency video coding (HEVC) with a compression ratio of 300:1—the gap to be bridged is still 3- to 6-fold; thus, efficient data compression techniques are in high demand. Although the situation is alleviated to some extent for data transmission from short distance UAVs, the situation deteriorates rapidly as the data receiving distance increases. To solve this problem, much work has been done to reduce the data size through dictionary learning-based data representation. One excellent work is the incremental K-SVD method for spatial big data representation [1,2]. Another representative work is the low-rank dictionary [3,4]. However, as we focus on the representation of continuous data sequences in the pixel domain, these methods cannot be directly applied to compress the remote sensing video data.

EOVD are video clips taken from high space (e.g., 500–600 km for video satellites and hundreds of meters for UAVs) in which the majority of the picture's content is landscape with small foreground objects, in contrast to the common videos with foreground objects as the major content. Moreover, remote surveillance for smart city applications produces large overlaps in the surveillance video data collected over an extended time period. Since the landscape changes slowly, the overlapping areas will have similar backgrounds across video clips, giving rise to a new form of redundancy, called long-term background redundancy (LBR) in this paper. Taking all the videos on a large temporal scale, LBR becomes significant as the background repetition dramatically increases. Thus, eliminating LBR in EOVD will significantly improve coding efficiency and support real-time smart city video applications.

Most widespread video coding strategies commonly adopt intra/inter-frame prediction to explore similarities in local spatial/temporal domains [5,6], effectively eliminating most local redundancies within a single video clip. Moreover, to further reduce the redundancy in ground surveillance video data induced by static backgrounds, Reference [7] proposed generating short-term, high-quality reference frames of backgrounds to improve the prediction accuracy for those areas. While this study achieved efficient coding for a single video source, the similarity measurement is subject to changes in visual appearances due to projection and illumination variations of the background on large spatial and temporal scales.

Several multisource data coding schemes have been proposed in recent years, which mainly focus on coding image sets from arbitrary views. Some researchers [8–10] have utilized scale-invariant feature transform (SIFT) features to measure the similarity between blocks from different images. Due to their invariance to rotation and robustness to illumination changes, SIFT features can build correlations among different images, thus achieving inter-image prediction to explore redundancies among multisource images. The same idea has been extended to duplicated video clips [11], where the redundancies between video clips are eliminated by referencing the basic video clip after adjusting for projection and illumination. While these methods have provided excellent ideas for exploiting redundancies across data sources within a dataset, the matched image blocks in pixel domains from different sources usually do not relate in reality and thus are not suitable for matching large areas like backgrounds in EOVD.

A reference library that records information common to all video data (e.g., libraries of two-dimensional (2D) vehicle images [12] or three-dimensional (3D) vehicle models [13,14] to eliminate redundancies caused by the repetition of similar vehicles) could efficiently eliminate redundancies across video clips. Unlike references from a dataset, a library-based method normally presents the basic knowledge of the encoded content and transformation. It is more efficient than only selecting references in the pixel domain, because this method reveals how the images relate in reality. Our method was developed based on this idea but focuses on using a library of backgrounds rather than foregrounds to eliminate LBRs in EOVD.

In this study, we developed a long-term background referencing library (LBRL)-based EOVD coding scheme according to the characteristics of the EOVD. First, we discussed the LBR induced by similarities among video clips taken of the same area throughout a temporal scale. Then, we analyzed the factors causing image representation variations of the background in different video clips. Based on that analysis, we proposed how to develop an LBRL for remote surveillance applications in smart cities. Next, we proposed a method to generate references based on the LBRL and the adjusted impact factor. Finally, we developed an encoding and decoding framework for EOVD compression.

Video clips from UAV and from video satellites were used to conduct experiments to evaluate the performance of the proposed method. A reference library built using the same conditions as those of the encoding video clips was used in the UAV case to represent how a good background reference can help to reduce the bitrate, and the results revealed that the proposed method can achieve 54% bitrate savings on average over the main profile of HEVC. In the satellite case, LBRL was developed from a Google Earth image [15], which attempted to simulate the usage of real historical remote sensing data. In this case, the bitrate savings were around 25%. In addition, we also tested to what extent different impact factors contribute to the bitrate savings.

There are three main contributions of this work:

(1) We analyzed the characteristics of Earth observatory video data, and discovered the long-term background redundancy among the videos collected of the same location at different times, which provides a chance to further compress the EOVD.
(2) We introduced the concept of a referencing library (the LBRL) as the fundamental infrastructure to facilitate the real-time collection of EOVD, which will further enhance online smart city applications.
(3) We proposed an LBRL-based reference generation method and the coding framework for EOVD, which can significantly reduce the bitrate compared to the coding standard for a single video source, helping to alleviate the difference between data collection bitrate and the space to Earth transmission bandwidth.

The remainder of this paper is organized as follows: Section 2 provides a literature review regarding related work. A detailed analysis of the LBR of EOVD and the development of an LBRL to eliminate LBR is illustrated in Section 3. The LBRL-based reference generating and encoding framework is developed in Section 4. Section 5 reports our experimental results, and Section 6 concludes the paper.

2. Related Work

Our work is related to the current single video coding method, the coding method for ground surveillance data considering the scene redundancy, and the coding method for multisource video clips. Therefore, we review the coding method from these three aspects.

2.1. Video Compression of Satellite Videos

In the initial stages of satellite development, satellite data were stored as remote sensing images. Satellite image compression methods can be divided into two methods: prediction-based and transformation-based. Prediction-based methods [16–18] use encoded pixels to estimate the current pixel value based on the correlation between pixels or bands of satellite images. Transformation-based methods [19,20] regard satellite data as a generalized, stationary random field [21]; its three-dimensional orthogonal transformation [22] maximizes the information concentrated in a small number of transform coefficients, thereby removing the maximum amount of spatial redundancy and inter-spectrum redundancy. The above methods were designed for a single image frame. Although they effectively removed spatial redundancy in the image, removing the redundancy caused by the correlation between the images was difficult. In recent years, with the development of video satellites, general video compression standards have been integrated into satellites. For example, Skysat [23], a video satellite launched by Skybox, was outfitted with video

compression standard H.264 [5]. General video compression standards use local spatial-temporal prediction models in small-scale space-time ranges to process local, short-term data; they cannot, however, remove geographical background redundancy from satellite video.

2.2. Video Compression of Surveillance Videos

Surveillance videos characteristically have fixed scenes and slight changes in the background. For these characteristics, surveillance video compression methods can be divided into LRSD (low-rank sparse decomposition)-based and background modeling methods. LRSD-based methods [24–26] employ LRSD to decompose the input video into low-rank components representing the background and sparse components representing the moving objects, which are encoded by different methods. Background modeling methods [7,27–30] use background modeling technology to build background frames for reference that improve the compression efficiency by improving the prediction accuracy. These surveillance video compression methods only apply to local spatial-temporal redundancy in single-source video; they do not consider the similarity of the background when the same region is captured by multisource satellite videos and cannot cope with apparent differences in the area due to shooting time, posture, height, and other factors.

2.3. Video Compression of Multisource Image/Video Data

Multisource image/video data refers to the collection of images/videos obtained by multiple shooting devices at various times from different positions. They contain a large number of similar images with common pixel distributions, features, and backgrounds. With the development of cloud technology, cloud-based image compression has attracted substantial interest [8,31,32]. These methods use cloud historical data to compress images by searching for similar images in the cloud data as a reference to improve prediction accuracy. Concurrently, compression methods [9–11,33,34] for image sets were developed using cloud historical data as a reference. The basic idea is to cluster images via image content, organize those images into a pseudo sequence, and code them like a video. The compression methods for multisource image/video data are designed from the perspective of image features, which usually mine similarities between image blocks by matching feature points. Moreover, multiscale features for image representation are proposed to extend representation from single payload to multiple payloads, as being proposed in References [35–38], which is also a way to build relations between multiple data sources. However, computational complexity is high, and the actual correspondence between the selected image block and the coding object is often lacking, which is not conducive to large-area matching.

3. Long-Term Background Referencing Library

First, this section will exploit LBR in EOVD and discuss what factors are important to eliminate LBR. Then, we develop an LBRL to represent the long-term background.

3.1. A New Redundancy Induced by Background Repetition

LBR is a new type of redundancy found in remotely captured video clips shot of the same location. It is caused by the similarity among the repeated background in different video clips. In the long term, LBR shows the following characteristics: structural consistency and appearance variance. To facilitate its expression, with A representing the area shot by a certain video clip and $\mathcal{A}$ the entire area of a smart city, $A \subset \mathcal{A}$. The background represented in a video clip of area A is denoted as B.

Structural consistency: Since landscapes change slowly, the structure of a certain area can be assumed to be consistent within a time period. Therefore, different video clips shot of this area will reveal the same structure of the area. As shown in Figure 1a, there are two frames taken from two video clips shot of the same location at different time by two satellites. Even though there are some differences in the image representations, we can easily judge that this is the same place according to the same structure.

Appearance variance: Due to the different conditions under which the video clips are captured, such as natural conditions (e.g., atmosphere, illumination) and device conditions (e.g., sensors), the images representations of the same area will have some variations. As shown in Figure 1a and the magnified part in Figure 1b, we can find variances in viewing angle, color, and quality. Thus, we discuss the appearance variance in these aspects.

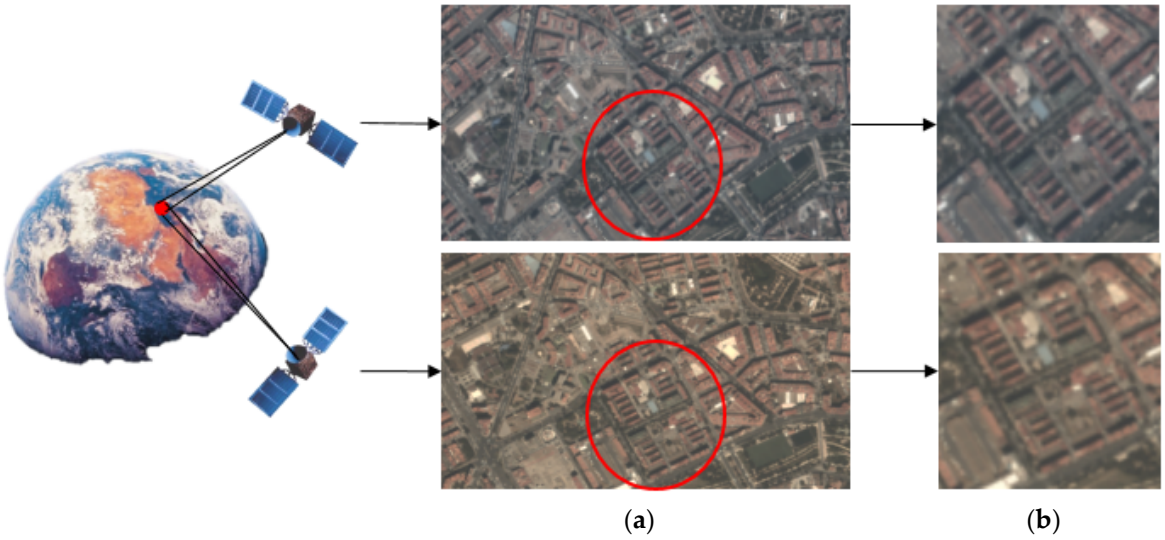


Figure 1. Appearances of two video clips shot of the same location at different time by two different satellites. (**a**) A sample frame taken from each of the video clips; the same structure indicates the same location. (**b**) Magnified area showing the variances in projection, color, and quality.

(1) **Projection difference:** a location in a specific video clip can be represented by the projection of that area into the image plane, which is:

$$B = P_v(A) \tag{1}$$

where B is the background in a picture and $P_v(A)$ is the projection of area A into a video clip. Since the projection is decided by the position and angle of the camera, it changes for every frame.

(2) **Radiometric difference:** the color of an image is affected by changes in the area's environmental radiation. Radiation changes can be modeled because the factors causing them, such as illumination, are limited in the long term. Therefore, the image representation of an area can be expressed as follows:

$$I_R(B) = M_I \cdot B = M_I \cdot P_v(A) \tag{2}$$

where M_I is a radiation model that converts from the reference background to the current image, and $I_R(B)$ is the image representation of the background after radiation change.

(3) **Quality difference:** EOVD image quality is affected by many factors. Some are related to the sensor itself, such as the optical imaging system, electrical signal conversion, and motion of the platform. These factors remain stable for a certain video clip, leading to consistent quality degradation for that video clip. Therefore, the image representation of an area can be expressed as follows:

$$I(B) = M_q \cdot I_R(B) = M_q \cdot M_I \cdot P_v(A) \tag{3}$$

where M_q is the quality degradation of a certain satellite, and $I_D(B)$ is the final image representation of the area.

3.2. Development of an LBRL

Current video coding standards using intra- and inter-frame prediction are very efficient at eliminating short-term redundancies. However, using such a prediction method across multiple video clips is uncommon, mostly because of how the image representation changes due to variations in projection, radiation, and quality. As a result, the same area is recorded every time it is captured, leading to a waste of transmission bandwidth.

Creating an LBRL to eliminate LBR addresses this redundant transmission issue. Ideally, an LBRL should do the following:

(1) Be able to cover the entire area of smart city applications.
(2) Be robust enough to handle changes in image representation due to various viewing angles.
(3) Be compatible with changes in the visual appearance of the background caused by radiation changes and quality degradation.

Therefore, we proposed an LBRL composed of basic, high-resolution reference images of a smart city's area, which can support three essential transformations: projection transformation related to each frame, and radiometric adjustment plus quality adjustment related to a video clip. The formation of the LBRL is shown in Figure 2.

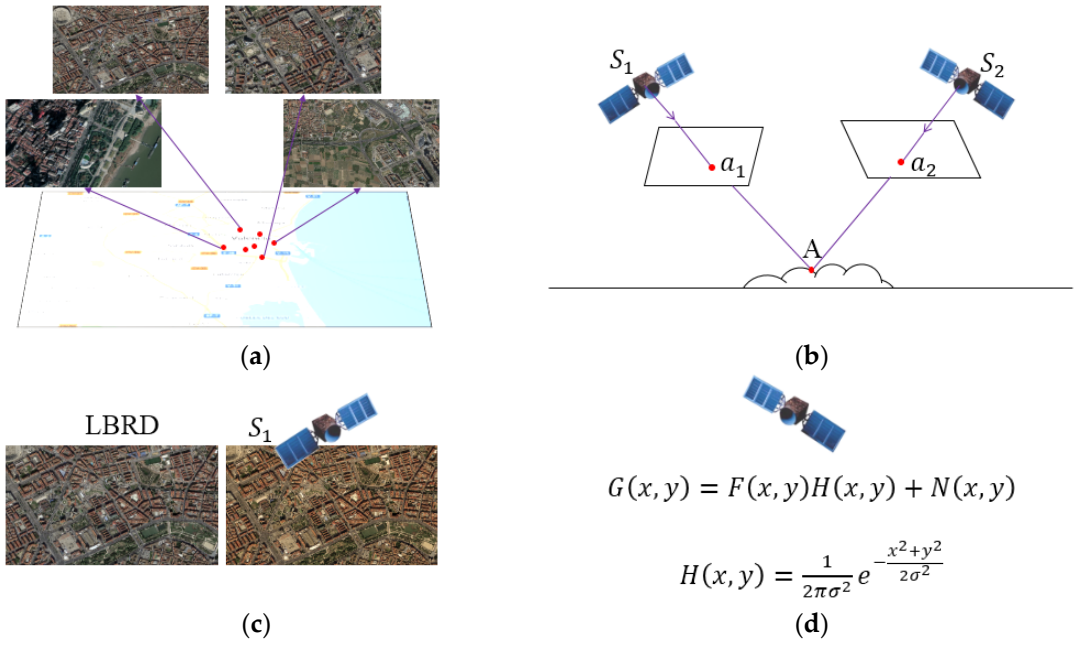


Figure 2. Sketch of the long-term background referencing library (LBRL) for long-term background redundancy (LBR) elimination. (**a**) Corrected historical images registered at corresponding locations in the scale of a smart city, forming the foundation library of the LBRL; (**b**) projection transformation; (**c**) radiometric adjustment; (**d**) quality adjustment.

We used historical, geographical registered images that had been corrected to develop an LBRL of an area for EOVD referencing. These images were stitched together to cover the entire area of a smart city. The geographical attribute was used to facilitate image matching during the referencing process in the EOVD encoding. The approximate area was determined according to the initial video's data positioning. Image data in an LBRL can be updated when a static ground change is detected from new video data.

Since the three transformations were highly related to each video clip, we did not include them in the LBRL but made the transformations available within the LBRL-based reference generator, which will be described in the next section.

4. LBRL-Based Reference Generation and Coding Framework

This section first details how to generate references from an LBRL for a newly collected video clip through geometrical matching for projection transformation, radiometric adjustment, and quality adjustment of the background image. Then, it describes the encoding and decoding scheme based on a generated background reference. An overview of the proposed LBRL-based EOVD coding framework is illustrated in Figure 3.

4.1. Generating a Background Reference

Prediction from references is known to be the most important process in the current video coding process to remove redundancies. As in block-based prediction, coding efficiency is directly related to the similarity between the reference and the current encoding frame. We generated background references for the current frame in the following sequence: geometrical matching, radiometric adjustment, and quality adjustment.

4.1.1. Geometric Matching

The initial position of a captured video clip is decided through the satellite global positioning system (GPS), which is used to decide an approximate captured area in the geographical registered library. A buffer is added outside the approximate captured area to cope with the positioning error of GPS. The reference image containing the captured area with the buffer is then cropped for the geometrical matching between the reference image and the captured video frame.

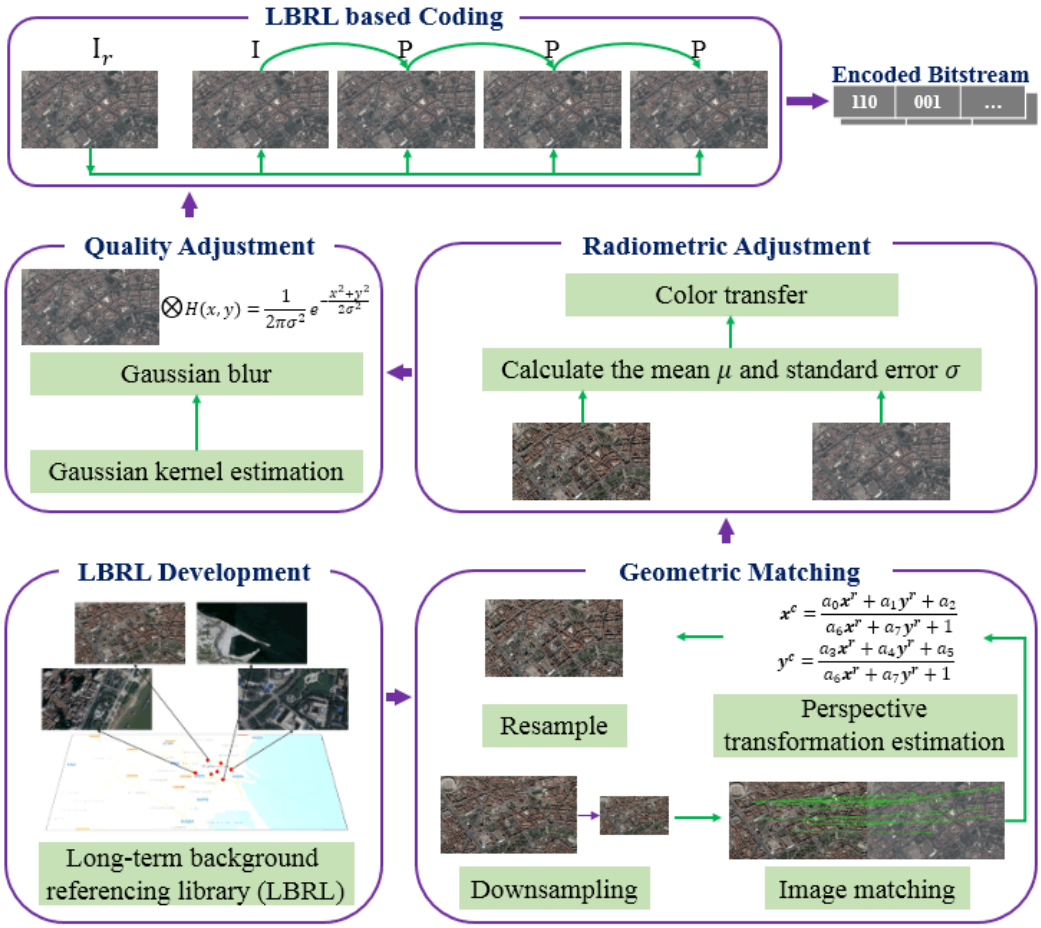


Figure 3. Overview of the proposed background reference image generation and the Earth observatory video data (EOVD) encoding using the generated reference image.

Geometrical matching to the LBRL locates the correct shooting area of the current frame and transforms the area reference image from the LBRL to the target frame through projection transformation. This process consists of downsampling the reference image, matching correspondence points, estimating the perspective transformation, and resampling the image.

SIFT feature matching [39] is normally used to find correspondence points. Due to differences in resolution, quality, and radiation between the basic area image from the LBRL and the current frame, however, sufficient correspondence point pairs often cannot be obtained, resulting in incorrect matches. Therefore, we first downsampled the high-resolution image from the LBRL to convert it to an image with ground resolution similar to the video data. The ground resolution of the current video was obtained from satellite documentation. The approximate shooting area was estimated from the online geopositioning of the satellite imagery. Then, to match the downsampled reference image with the current video frame, we adopted the improved SIFT matching method [40] that was developed for multisensor remote sensing image matching, which developed a distinctive order based on a self-similarity descriptor that was robust against illumination differences.

Based on correspondence point pairs, the perspective transformation was estimated by solving the following mapping functions:

$$x_i^c = \frac{a_0 x_i^r + a_1 y_i^r + a_2}{a_6 x_i^r + a_7 y_i^r + 1} \text{ and } y_i^c = \frac{a_3 x_i^r + a_4 y_i^r + a_5}{a_6 x_i^r + a_7 y_i^r + 1} \tag{4}$$

where $(x_i^c,\ y_i^c)$ are the coordinates of points in the current video and $(x_i^r,\ y_i^r)$ are the corresponding points in the downsampled reference images from the LBRL. After estimating the perspective transformation, we generated the geometrically transformed reference image I_r^g using Equation (4).

4.1.2. Radiometric Adjustment

We employed the color transfer model proposed in Reference [41] to adjust the radiation of the geometrically transformed reference image I_r^g to correspond with the current video frame. Since video data is recoded in the YUV color space, the radiometric adjustment was also conducted in this color space. Since a YUV color space is similar to a $l\alpha\beta$ color space, in which the first channel is lightness and the other two channels are color components, we adopted a color transform model similar to the one proposed for the $l\alpha\beta$ color space in our work:

$$\begin{bmatrix} Y_t \\ U_t \\ V_t \end{bmatrix} = \begin{bmatrix} \frac{\sigma_c^Y}{\sigma_s^Y} & 0 & 0 \\ 0 & \frac{\sigma_c^U}{\sigma_s^U} & 0 \\ 0 & 0 & \frac{\sigma_c^V}{\sigma_s^V} \end{bmatrix} \left(\begin{bmatrix} Y_s \\ U_s \\ V_s \end{bmatrix} - \begin{bmatrix} \overline{Y_s} \\ \overline{U_s} \\ \overline{V_s} \end{bmatrix} \right) + \begin{bmatrix} \overline{Y_c} \\ \overline{U_c} \\ \overline{V_c} \end{bmatrix} \tag{5}$$

where $\begin{bmatrix} Y_t & U_t & V_t \end{bmatrix}^T$ and $\begin{bmatrix} Y_s & U_s & V_s \end{bmatrix}^T$ are the color values in the radiometrically adjusted reference images I_r^c and I_r^g, respectively. $\begin{bmatrix} \overline{Y_s} & \overline{U_s} & \overline{V_s} \end{bmatrix}^T$ and $\begin{bmatrix} \sigma_s^Y & \sigma_s^U & \sigma_s^V \end{bmatrix}^T$ are the mean and standard deviations of YUV from I_r^g, while $\begin{bmatrix} \overline{Y_c} & \overline{U_c} & \overline{V_c} \end{bmatrix}^T$ and $\begin{bmatrix} \sigma_c^Y & \sigma_c^U & \sigma_c^V \end{bmatrix}^T$ are the current frame's values. By using this model, the color of the reference I_r^g was adjusted according to the color statistics of the current frame.

4.1.3. Quality Adjustment

Images in the current satellite video data usually appeared blurrier than the reference image; thus, the quality of the reference image was adjusted to correspond to the quality of the current video frame. In this paper, we adjusted the quality based on the previously obtained reference image I_r^c after geometrical matching and radiometric adjustment, generating the final reference image I_r.

We applied a 2D Gaussian blur filter to simulate the quality degradation of the satellite video. Since we assumed the quality degradation was homogeneous over the whole image, we adopted an isotropic Gaussian model and set the mean of the Gaussian distribution to 0, leaving the standard deviation σ to be defined according to the difference between I_r^c and the current video frame.

In practice, a Gaussian blur filter can be converted to a 5×5 kernel, the values of which can be represented by a polynomial of σ according to their distance to the kernel center. In this way, the image value after Gaussian blur can be represented by a function of σ; thus, σ can be obtained by minimizing the pixel value differences between I_r^c after Gaussian blur and the current frame. To reduce computational complexity, we stochastically selected N 5×5 blocks from I_r^c and their corresponding pixels from the current frame and minimized the objective function to obtain the optimized parameter σ as follows:

$$\underset{\sigma}{\operatorname{argmin}} \sum_{k=0}^{N} \left(G(\sigma) * B_r^k - p_c^k \right) \tag{6}$$

where B_r^k is the kth block from I_r^c, and p_c^k is the kth corresponding pixel value from the current frame. $G(\sigma)$ is the Gaussian kernel, whose values are defined by the parameter σ as follows:

$$a_{ij} = \frac{1}{\sum_{ij} a'_{ij}} \cdot a'_{ij} \ (i, j = -2, -1, 0, 1, 2) \tag{7}$$

$$a'_{ij} = e^{-\frac{(i^2+j^2)}{2\sigma^2}} \ (i, j = -2, -1, 0, 1, 2) \tag{8}$$

The final reference image I_r was obtained by $G(\sigma) * I_r^c$. The next section will introduce how to use an LBRL reference image to encode and decode EOVD.

4.2. Encoding and Decoding Scheme

The LBRL-based encoding and decoding of satellite videos requires the LBRL reference image at both the encoding and decoding ends. Figure 4 illustrates the overall LBRL-based encoding framework. The encoding process can be described as follows:

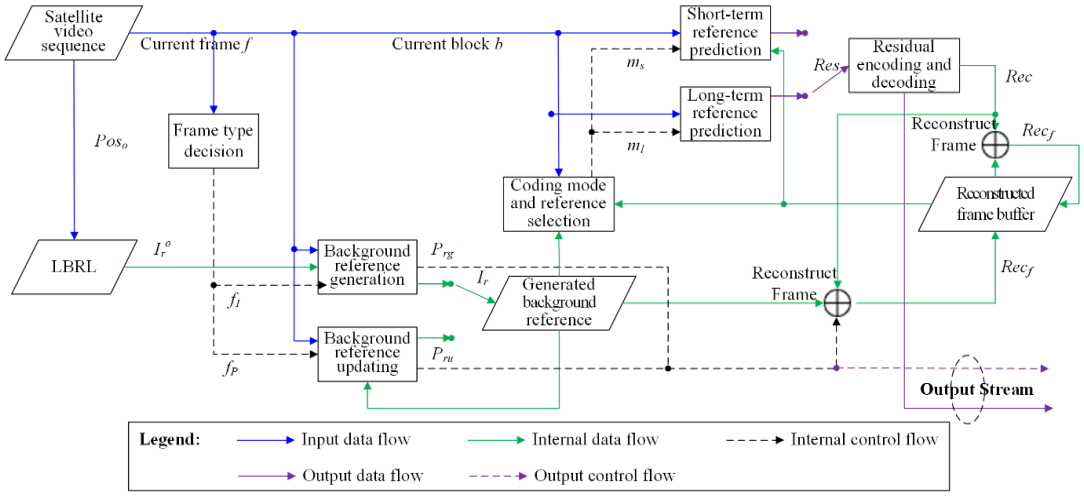


Figure 4. The overall framework of the LBRL-based encoding of EOVD.

Step 1. **Generating the background reference.** Initially, or for an I frame, f_I, we used the proposed LBRL-based reference generation method described in Section 3.1 to initialize a background reference (denoted by I_r in Figure 4) for the encoding of I frames. Since the generated background reference was not sent, we needed to send the control data together with the encoded frame to reconstruct the reference at the decoder. The control data for generating the background reference

was denoted as P_{rg}, including the perspective transformation matrix $PT = \begin{bmatrix} a_0 & a_1 & a_2 \\ a_3 & a_4 & a_5 \\ a_6 & a_7 & 1 \end{bmatrix}$ in geometric matching, $\begin{bmatrix} \overline{Y_c} & \overline{U_c} & \overline{V_c} \end{bmatrix}^T$ and $\begin{bmatrix} \sigma_c^Y & \sigma_c^U & \sigma_c^V \end{bmatrix}^T$ in radiometric adjustment, and σ in quality adjustment. The generated background reference was stored in a temporal buffer in the encoder to update the reference for subsequent frames. At the same time that a newly generated I_r was put into the reference buffer, previous data were removed from the buffer.

Step 2. **Updating the background reference.** For P frames, f_P, immediately after an I frame, the radiometric conditions and quality degradation did not vary markedly, only the projections changed slightly. Therefore, we only updated the perspective transformation from the background reference for the last frame. The output control data was denoted as P_{ru}, including a new PT matrix. The updated reference image was then added to the reference buffer.

Step 3. **Calculating candidate modes and performing predictions.** A generated or updated background reference was added to the coding reference list. For any I frame, besides the traditional intra-picture prediction, a long-term prediction (denoted by m_l) taking I_r as the additional reference could also be performed. Since inter-picture prediction is normally more efficient than intra-picture prediction, it is more efficient at reducing the bitrate. Then, for P frames, both short-term (denoted by m_s) and long-term predictions could be selected by referring to the adjacent frames or background reference, respectively. As proven by Reference [3], a high-quality background reference can help reduce the bitrate of blocks in P frames.

Step 4. **Encoding and reconstructing the current block.** Rate-distortion was applied to select the best encoding mode. By performing the predictions, residuals (denoted as *Res*) were computed and encoded by transform, scaling, quantization, and entropy coding. Frames were reconstructed (denoted as Rec_f) to provide short-term frame references by reconstructing each block by adding the block reference to the decoded block residuals. The reconstructed frames were stored in the reconstructed frame buffer to provide the reference list.

After encoding a video clip, the parameters for the reference generation and prediction, as well as encoded residuals, are output. After being transmitted from the remote sensing platforms to the server on Earth, video clips are decoded by reconstructing the background references from the LBRL using the reference generating or updating parameters.

5. Experiments and Results

5.1. Experimental Setup

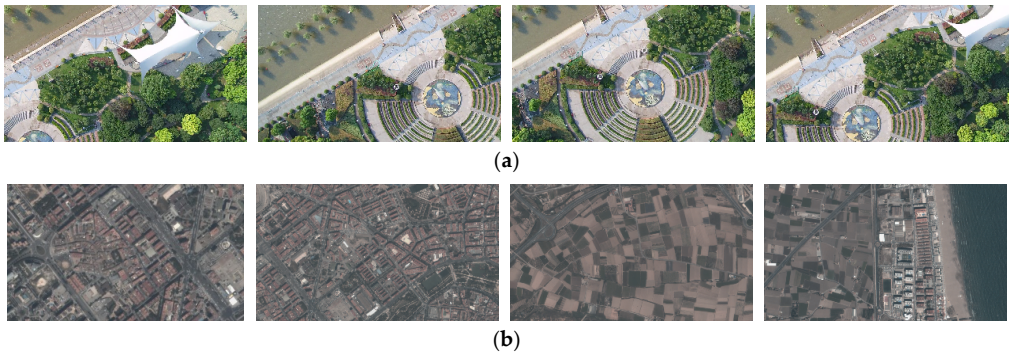
We evaluated the effectiveness of the proposed LBRL-based EOVD compression method by carrying out extensive experiments. Two types of EOVD datasets were used in this paper as follows:

Video clips from UAVs: Four video clips from UAVs captured over Yangtze river park, Wuhan, China, were employed to evaluate performance, as shown in Figure 5a. These video clips were captured once a day for four consecutive days by a Yuneec Typhoon H UAV. The flight height was fixed at 100 m. Each original video clip contained 300 frames of 1080p (1920 × 1280) resolution, 15 fps. The videos contain slow flight and fast flight. Two other videos in the same area were captured to extract frames as data to develop a reference image library for this test. In total, the area contained nine stitched key frame pictures.

Video clips from satellites: Four video clips from satellite Jilin-1 over Valencia, Spain were used in the experiment (Figure 5b), containing two video clips of building, one of farmland, and one of seaside areas. To facilitate the coding process, we did not directly use the original 12,000 × 5000 resolution, but cut out video clips with a of size 1080p with 300 frames, 15 fps. As access to the historical satellite data was limited in our experiments, images from Google Earth [15] were employed as the LBRL images. The Google Earth images were already geographically registered images with high resolutions.

Because the reference images for the UAV video clips were captured by the same device under similar conditions, this group of tests mainly focused on evaluating how effective the background library was at video coding. The reference images for the satellite video clips came from a different satellite under significantly different sensing conditions. This reality scene was used to test whether the algorithm for radiometric and quality adjustments in reference generation was effective.

In the experiment, we conducted two implementations of the proposed method based on two standard codecs for different testing purposes (details shown in Table 1). It can also be implemented on other codecs since it mainly provides an extra encoding reference. The first implementation was based on the low-delay configuration of an *HEVC* test model HM16.8, named *LBRL-HEVC*. This implementation was compared to the unmodified *HEVC* codec to test the effectiveness of the proposed method on EOVD compression, since *HEVC* can achieve the highest compression ratio. This testing was implemented at a four-core Intel i5 CPU on a 2.6 GHz platform.



(**a**)

(**b**)

Figure 5. Typical frames from experimental data. (**a**) Unmanned aerial vehicle (UAV) video clips; (**b**) satellite video clips.

Although *HEVC* has achieved a very high compression ratio, it is not commonly applied in practice due to its computational complexity. In order to test the applicability of our method for practical use, the second implementation was based on the *x264* codec, named *LBRL-x264*, compared to the unmodified model of *x264*. *x264* is known as the fastest CPU implementation of video compression [42]. It is the most commonly used codec in the practice, including applications on UAVs and video satellite platforms. This testing was implemented at Nvidia Jetson TX2, which was selected as one part of an embedded system developed for a small satellite set to launch in the year 2020. Nvidia Jetson TX2 contains four ARM Cortex A57 cores and one GPU with 256 CUDA cores.

In the experiment, Bjøntegaard delta PSNR (BD-PSNR) and Bjøntegaard delta rate (BD-Rate) [43] were utilized as the metrics for the objective evaluation of coding performance. We also included subjective evaluation metrics for the satellite video data.

Table 1. Experimental configuration of two implementations.

	Testing Platform	Codec	Detailed Settings
First Implementation *LBRL-HEVC* vs. *HEVC*	Four-core Intel i5-4210m CPU @ 2.60 GHz	HM 16.8 [44]	Frame structure: Low Delay IPPP; GOP size = 8; QP = 22, 27, 32, 37; Max partition depth = 4; Fast search = Enable; Search range = 64; Intra period = 8; Rate control = −1; SA0 = 1;
Second Implementation *LBRL-x264* vs. *x264*	Nvidia Jetson TX2 contains four ARM Cortex A57 cores and one GPU with 256 CUDA cores	X264 [45]: version r2901 of 20 January 20 2018	Profile = baseline; GOP size = 8; Slice mode = 0; QP = 22, 27, 32, 37; Preset = ultrafast; Keyint = 8; Search range = 32; Rate control = −1;

5.2. Experiments with UAV Video Clips

In this experiment, the effectiveness of how a long-term background reference can improve the encoding efficiency was tested. We used the first implementation of the proposed method *LBRL-HEVC* against *HEVC* for this purpose.

The image data used to build LBRL was collected under the same conditions with the to-be encoded video clips. They were geographically corrected by using ground control points, and then stitched together. The developed LBRL for UAV data is shown in Figure 6a, with the size of 12.69 MB. Considering that the images in LBRL of UAV shared quite similar conditions with the UAV video data, we only conduct geometric matching to generate the reference image, without radiometric adjustment and quality adjustment. Taking one video clip in Figure 6d, the approximate area was firstly located in the LBRL (red rectangle) and then cropped out, as shown in Figure 6b. The geometric matching and transformation was conducted to convert the cropped image from LBRL to be of the same shape as the frames of the video clip.

The total encoding performance gains of the proposed *LBRL-HEVC* compared with *HEVC* are listed in Table 2 and the Rate-Distortion (RD) curve is shown in Figure 7. At the same PSNR, the proposed method averagely decreases 54.18% bitrate over *HEVC*. This result also corresponds to 4.32 dB PSNR gains over *HEVC* at the same bitrate. The bitrate reduction occurs mainly because of the bit savings from the I frame. Most of the prediction modes of the I frame changed from intra-frame prediction to inter-frame prediction, referencing the generated background reference images. The P frame can also reference both the generated background reference images and its previous frames.

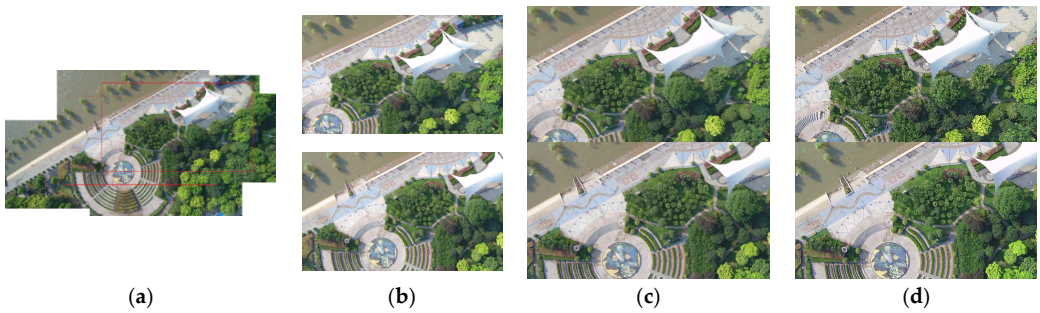


Figure 6. LBRL and reference images generation for UAV video clips. (**a**) The LBRL developed for the UAV video clip test; (**b**) cropped area from the LBRL (red rectangle in (**a**)) according to the to-be-encoded video clip; (**c**) geometrical transformed reference image; (**d**) to-be-encoded video clip. First row—UAV video clip a; second row—UAV video clip **d**.

Table 2. The overall BD-PSNR (dB) and BD-Rate (%) of *LBRL-HEVC* vs. *HEVC* with UAV data.

UAV	BD-PSNR	BD-Rate	UAV	BD-PSNR	BD-Rate
a	5.34	−62.77	c	3.51	−49.65
b	4.21	−53.32	d	4.21	−50.97
			Average	**4.32**	**−54.18**

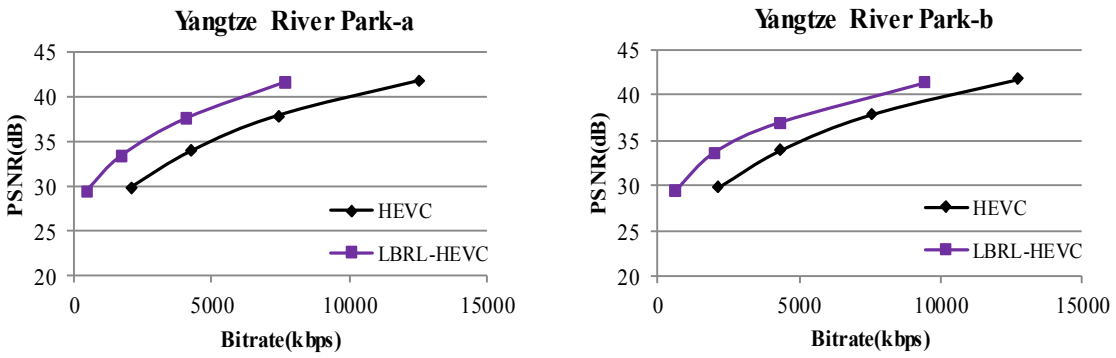


Figure 7. *Cont.*

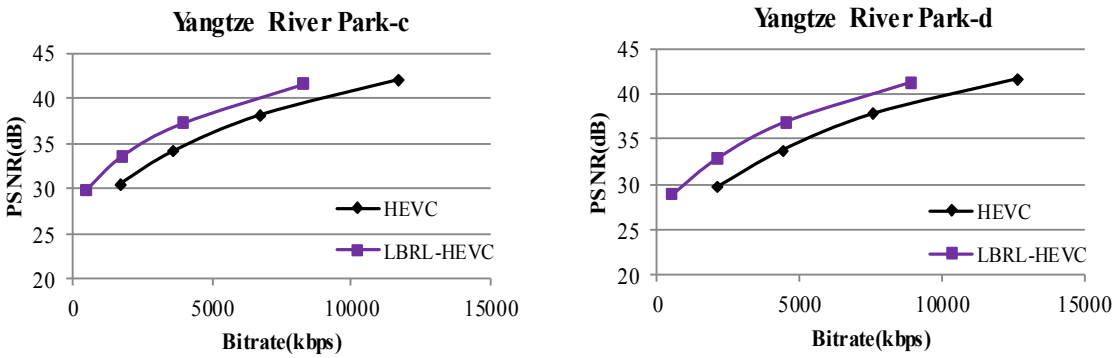


Figure 7. RD curves of *LBRL-HEVC* and *HEVC* for four video clips from UAV data.

5.3. Experiments with Satellite Video Clips

The LBRL in this case consisted of satellite images downloaded from Google Earth, due to the limited access to historical satellite data. Since the satellite video clips were in the city of Valencia, Spain, we built the LBRL for this city. The total land area of Valencia was 134.7 km^2, and the total size of Google Earth images covering this city was 5.93 GB. The size of the library was proportional to the land area, namely 45 MB/km^2 on average. Even considering one of the biggest cities, New York City, USA, with a land area of about 784 km^2, the size of the LBRL is less than 35 GB.

5.3.1. Intermediate Results from Background Reference Generation

The intermediate results for background reference generation from a Google Earth image (Figure 8b) for a certain video clip (sample frame in Figure 8a) are presented in Figure 8c,d. It is easy to notice that the Google Earth image is much shaper and the color is more brilliant than the captured satellite videos, so besides geometric matching, we also conducted radiometric adjustment and quality adjustment to generate the final reference image.

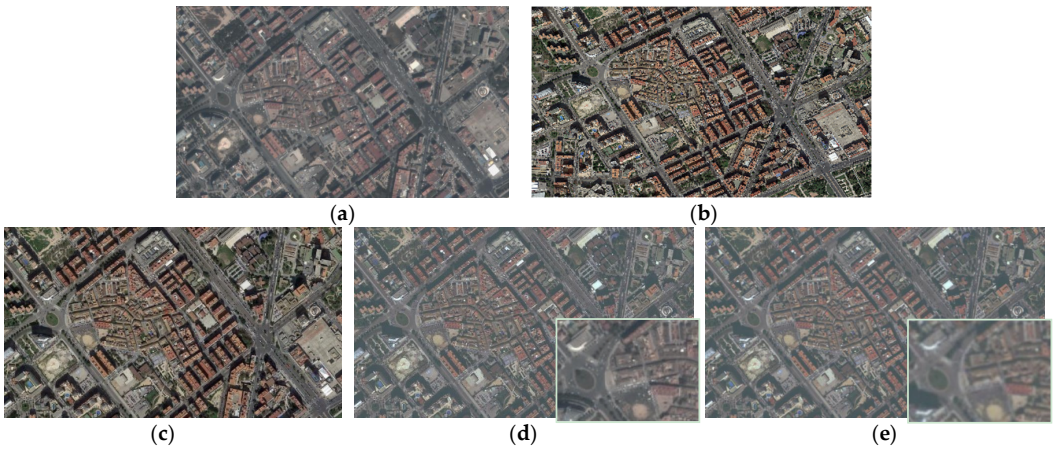


Figure 8. Reference images. (**a**) Sample frame from satellite video clips building-1; (**b**) cropped image from a large image downloaded from © Google Earth; (**c**) reference image I_r^g after geometric matching; (**d**) reference image I_r^c from I_r^g after radiometric adjustment; (**e**) final background reference image I_r from I_r^c after quality adjustment.

From the intermediate result, we can see that the proposed background referencing generation method can successfully handle the image representation variances caused by illumination and sharpness differences. However, current strategy will not work well with the problems which cause a change of representation of remote sensing images: (1) shadow movement; (2) projection

difference of tall buildings; (3) huge illumination change; (4) scene change due to seasons variation e.g., vegetation; (5) landscape change. The solution to these problems might require multimode reference images for one region in LBRL, together with image translation techniques and other methods for background reference generation. An updating strategy for LBRL is also required to handle the landscape change problem.

5.3.2. Results of *LBRL-HEVC*

The improvement of coding efficiency was tested first using the implementation *LBRL-HEVC*. In this test, the coding results from references generated with only radiometric adjustment (*Only-RA*) and with only quality adjustment (*Only-QA*) were also compared to analyze the effectiveness of the radiometric and quality adjustment in generating good background references.

The coding results of *LBRL-HEVC* compared with *HEVC* are presented in Table 3. In general, the average bitrate savings can reach up to 24.93%. Compared to the averaged bitrate savings with UAV data, it was proven that the similarity of the background reference had great effectiveness on the improvement of the EOV data compression ratio.

We can also notice that in different video clips with different video content, the highest bitrate reduction appeared with farmland, where there were few tall buildings. Since we did not consider the elevation change, we could not correct the projection difference in our geometric matching, leading to low efficiency prediction for places containing projection differences. The seaside video clip had the lowest bitrate reduction, which was probably due to the negative influence of waves in the water area.

Comparing the results from *Only-RA* and *Only-QA* with *HEVC*, the coding efficiency was not obviously improved. This might be because with only one process, there were still great differences between the background reference and the encoding video frames in the pixel domain, resulting in non-valid inter-frame predictions. From the experimental data, we can conclude that the quality adjustment was a bit more important than the radiometric adjustment for background reference generation.

The RD curves for the tested satellite video clips are shown in Figure 9, revealing results similar to those we obtained from Table 2. The RD curves for *Only-RA* and *Only-QA* almost overlapped with *HEVC*, showing no significant improvement. The curves for the proposed method were higher than the other curves for the four video clips, representing the general effectiveness of the proposed method in bitrate reduction for satellite videos.

Table 3. The overall BD-PSNR (dB) and BD-Rate (%) of *Only-RA*, *Only-QA*, and *LBLR-HEVC* vs. *HEVC* with satellite data.

Method	Satellite Jilin-1	BD-PSNR	BD-Rate
Only-RA (Only Radiometric Adjustment)	Building-1	0.20	−5.08
	Building-2	0.16	−4.75
	Farmland	0.53	−9.65
	Seaside	0.08	−3.83
	Average	**0.24**	**−5.83**
Only-QA (Only Quality Adjustment)	Building-1	0.26	−6.38
	Building-2	0.26	−6.76
	Farmland	0.61	−13.86
	Seaside	0.22	−6.68
	Average	**0.34**	**−8.42**
LBRL-HEVC (Both RA and QA)	Building-1	1.19	−26.21
	Building-2	0.95	−23.80
	Farmland	1.76	−33.04
	Seaside	0.65	−16.68
	Average	**1.14**	**−24.93**

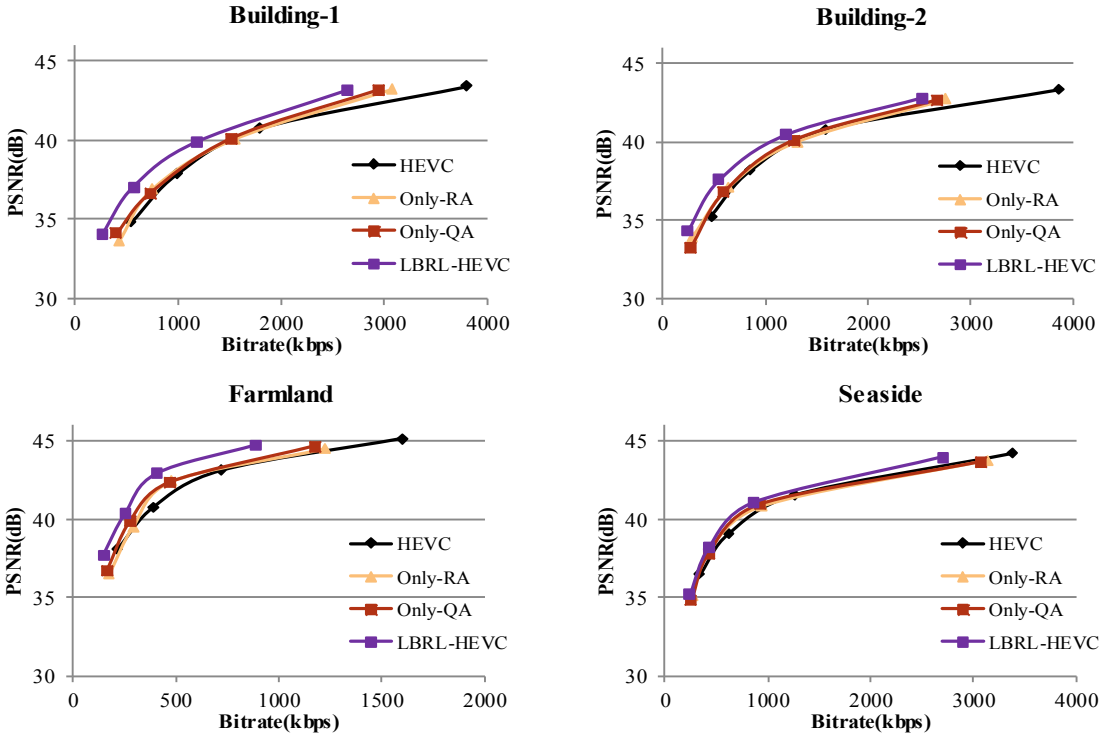


Figure 9. RD curves of *HEVC*, *Only-RA*, *Only-QA*, and *LBRL-HEVC* from four satellite video clips.

In general, the implementation of the proposed method on *HEVC* (*LBRL-HEVC*) proved that we can generate effective background references from the Google Earth images, and that the compression ratio can be successfully increased. The bitrate reduction of satellite data was less than that of UAV data, which was mainly due to the similarity between the reference image in LBRL and the current video data.

5.3.3. Results of *LBRL-x264*

In this section, the effectiveness of the proposed method in the embedded system of real applications is evaluated. The results of *LBRL-x264* compared to *x264* are presented in Table 4. Similar to the results of *LBRL-HEVC* compared to *HEVC*, *LBRL-x264* can reduce around 32.77% bitrate compared to *x264* at the same PSNR, and the quality improvement was on average 1.7 dB at the same bitrate.

Table 4. The overall BD-PSNR (dB) and BD-Rate (%) of *LBRL-x264* vs. *x264* with satellite video data.

Satellite Video	BD-PSNR	BD-Rate	Satellite Video	BD-PSNR	BD-Rate
a	2.13	−36.30	c	2.20	−40.47
b	1.42	−29.88	d	1.05	−24.43
			Average	**1.70**	**−32.77**

The detailed results are plotted in Figure 10, together with the curves of *LBRL-HEVC* and *HEVC*. As shown in the curves of *x264* and *LBRL-x264*, the differences between them were bigger at the lower part than the higher part. The lower part of the curves covered the range of the selected bitrate for transmission, where obvious bitrate reduction can be observed. More details are shown in the visual results in Figure 11. We can also notice that the bitrates from *HEVC* were much lower than the results from *LBRL-x264* or *x264*. However, the HEVC-based codecs cannot be implemented on UAV or satellite platforms, due to the computational complexity presented in Section 5.4.

The visual comparisons for video clips Builidng-2 and Seaside are shown in Figure 10. In the visual comparison, we selected target bitrates of around 500 kbps. The video clips in the test were 1080p of 1920 × 1280 resolution, and the original video data of 12,000 × 5000 resolution was 30 times that of the tested video clips. With the same quality, the encoded original video data stream would be 15 Mbps, which was within the required range of 10–20 Mbps transmission bandwidth between satellites and Earth.

As shown in the pictures, if encoded at nearly the same bitrate, the LBRL-based methods can provide better visual quality than that achieved from the corresponding codec. Comparing different decoded pictures from the same frame, the visual qualities were consistent with the PSNR values; namely, lower PSNR corresponded to lower visual quality. *LBRL-HEVC* can provide almost the same result visually as the original frame, especially that for Seaside with 39.25 dB. When the quality degraded to 35–37 dB from *HEVC*, decoded pictures tended to be blurry. Since the compression ratio is lower in *LBRL-x246* and *x264*, the qualities of the decoded pictures were obviously lower than those from *HEVC*. We can clearly notice the blocking artifacts in the pictures from *x264*. Taking the cars on the road in the Seaside video clip as an example to clarify the visual comparison, we can count six cars from the original picture. After encoding by *LBRL-HEVC*, five were remained, whilst only four cars were left in the decoded picture from *HEVC*. The shape of the cars became blurry in *LBRL-x264*, but we can still count five cars. The cars had almost disappeared from in the picture from *x264*.

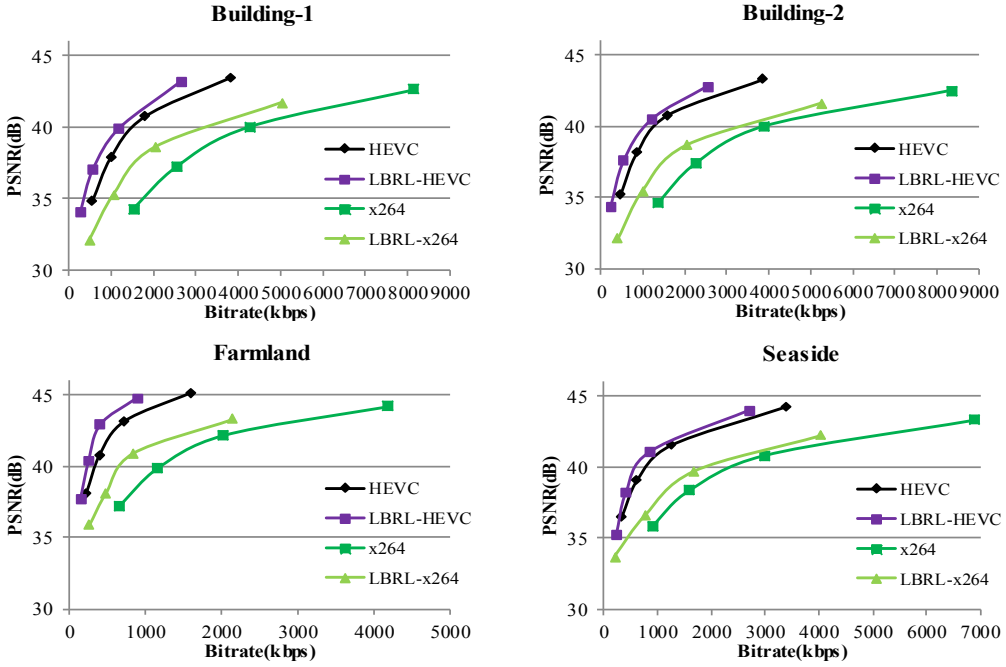


Figure 10. RD curves of *LBRL-x264*, *x264*, *LBRL-HEVC*, and *HEVC* from four satellite video clips.

5.4. Computational Complexity Analysis

In the proposed method, the additional computational cost comes from the generation of reference images, including geometric matching, radiometric adjustment, quality adjustment, and the resampling of the images, as well as the long-term prediction for the I frame. The computational complexity of other encoding processes is the same as that for HEVC. The LBRL is built offline, thus we do not take it into consideration in the computational complexity analysis.

The computational complexity was measured by frames per second (fps) and tested separately on two implementations. *LBRL-HEVC* and *HEVC* were tested using a laptop with i5 CPU. The total additional time for background reference generation was 5.52 s for the I frame and 3.50 s for the P frame, since the P frame did not need radiometric adjustment and quality adjustment. The tests for

LBRL-x264 and *x264* were carried on Nvidia Jetson TX2. The algorithms for SIFT matching, radiometric adjustment, and resampling were accelerated by the parallel processing on the GPU. Therefore, th total processing times for generating background reference images for the I frame and P frame were around 96.1 ms and 45.8 ms, respectively.

The comparison of the computational speed of the two implementations on different platforms are reported in Table 5. The QP for encoding was uniformly set to 32 for the comparison. Because the computational complexity was quite high for HEVC, the additional cost for reference generation did not have obvious effects on the processing speed. Since the processing time was around 1 min for one frame, far from real-time processing, it cannot be used on UAV or satellite platforms. The computational speed reached more than 100 fps for *x264*, which left time for the proposed method to generate the background reference. The average processing speed was around 16.77 fps for *LBRL-x264*, a bit higher than 15 fps set for remote sensing video data. Therefore, our method implemented on *x264* could achieve the real-time processing of 1080p video data on a remote sensing platform.

Figure 11. Visual comparison between *HEVC*, *LBRL-HEVC*, *x264*, and *LBRL-x264*.

Table 5. Computational speed of *LBRL-HEVC*, *HEVC*, *LBRL-x264*, and *x264*, with satellite video data (fps).

	Intel i5 CPU @ 2.60 GHz		**Nvidia Jetson TX2**	
	LBRL-HEVC	***HEVC***	***LBRL-x264***	***x264***
Building-1	0.0157	0.0167	15.91	93
Building-2	0.0157	0.0167	16.25	99
Farmland	0.0175	0.0187	17.88	147
Seaside	0.0172	0.0184	17.04	125
Average	**0.0165**	**0.0176**	**16.77**	**116**

6. Conclusions

This paper proposes a long-term background referencing-based Earth observatory data encoding method for real-time collection, analysis, and applications in smart cities. The key idea is to build an LBRL covering the entire area of a smart city to represent the common appearance of the landscape. For each new captured video clip, the corresponding image of the shooting location from the library is cropped and converted according to the image representation of the area in the video clip. The converted image is used as the additional long-term reference for the encoding of I frames and P frames. Extensive experiments with UAV video data and satellite video data show that, the proposed LBRL-based EOV encoding method can save 25% to 54% of the total bitrate and achieve a significant gain in background coding performance over *HEVC* and *x264* correspondingly. The GPU implementation of the proposed method based on *x264* codec on Nvidia TX2 can achieve a real-time processing of the 1080p video data with 15 fps. By applying the *x264* implementation, the gap between the bitrate of video data and the bandwidth of the transmission channel can be reduced from 3–6-fold to 2–4-fold.

Compared with the existing short-term prediction-based coding methods for single video clips, the proposed method follows the characteristics of a large portion of static landscape in EOV data, in addition to making use of the existing information of the landscape. Moreover, the information is reformed and geographically organized in the library, rather than the original data form used in multisource coding methods. The geographically organized form of the library helps to facilitate the reference searching. The uniform representation of the landscape and its transformations guarantee a highly similar reference, which further improves the compression efficiency.

The proposed method does not completely solve the real-time transmission problem between remote sensing platforms and Earth, but provides an idea to make use of known information on Earth to reduce the information needed to be sent from remote sensing platforms. To further improve the compression efficiency of the propose method, we will further investigate the development of background referencing libraries from multiple sources of historical data, exploiting the extraction and representation of common knowledge from images taken under different conditions. Then, exploring more accurate radiometric and quality adjustment models, this method can possibly be implemented for different land cover types. A complete solution to the transmission problem calls for development in different fields, including computational platforms, new data transmission solutions, and improved data processing techniques.

Author Contributions: Conceptualization, D.C.; Data curation, M.W.; Formal analysis, J.X.; Funding acquisition, R.Z.; Investigation, J.X.; Methodology, J.X.; Project administration, R.H.; Software, J.X.; Supervision, D.L.; Validation, Y.Z.; Writing—original draft, J.X.; Writing—review and editing, D.C.

Acknowledgments: We gratefully acknowledge anonymous reviewers who read drafts and made many helpful suggestions. This work is supported by the National Natural Science Foundation of China (91738302, 61502348, 61671336), science and technology program of Shenzhen (JCYJ20150422150029092), Open Research Fund of State Key Laboratory of Information Engineering in Surveying, Mapping and Remote Sensing, Wuhan University (17E03), the Fundamental Research Funds for the Central Universities (413000048), and the EU FP7 QUICK project under Grant Agreement No. PIRSES-GA-2013-612652.

Conflicts of Interest: The authors declare no conflict of interest. The founding sponsors had no role in the design of the study; in the collection, analyses, or interpretation of data; in the writing of the manuscript, and in the decision to publish the results.

References

1. Wang, L.; Lu, K.; Liu, P.; Ranjan, R.; Chen, L. IK-SVD: Dictionary Learning for Spatial Big Data via Incremental Atom Update. *Comput. Sci. Eng.* **2014**, *16*, 41–52. [CrossRef]
2. Song, W.; Deng, Z.; Wang, L.; Du, B.; Liu, P.; Lu, K. G-IK-SVD: Parallel IK-SVD on GPUs for sparse representation of spatial big data. *J. Supercomput.* **2017**, *73*, 3433–3450. [CrossRef]
3. Jing, X.Y.; Zhu, X.; Wu, F.; Hu, R.; You, X.; Wang, Y.; Feng, H.; Yang, J.Y. Super-resolution Person Re-identification with Semi-coupled Low-rank Discriminant Dictionary Learning. *IEEE Trans. Image Process.* **2017**, *26*, 1363–1378. [CrossRef] [PubMed]
4. Wu, F.; Jing, X.Y.; You, X.; Yue, D. Multi-view low-rank dictionary learning for image classification. *Pattern Recognit.* **2016**, *50*, 143–154. [CrossRef]
5. Wiegand, T.; Sullivan, G.J.; Bjontegaard, G.; Luthra, A. Overview of the H.264/AVC video coding standard. *IEEE Trans. Circuits Syst. Video Technol.* **2003**, *13*, 560–576. [CrossRef]
6. Sullivan, G.J.; Ohm, J.; Han, W.; Wiegand, T. Overview of the high efficiency video coding standard. *IEEE Trans. Circuits Syst. Video Technol.* **2012**, *22*, 1649–1668. [CrossRef]
7. Zhang, X.; Huang, T.; Tian, Y.; Gao, W. Background-modeling-based adaptive prediction for surveillance video coding. *IEEE Trans. Image Process.* **2014**, *23*, 769–784. [CrossRef] [PubMed]
8. Yue, H.; Sun, X.; Yang, J.; Wu, F. Cloud-based image coding for mobile devices—Toward thousands to one compression. *IEEE Trans. Multimedia* **2013**, *15*, 845–857.
9. Shi, Z.; Sun, X.; Wu, F. Feature-based image set compression. In Proceedings of the IEEE International Conference on Multimedia and Expo, San Jose, CA, USA, 15–19 July 2013; pp. 1–6.
10. Wu, H.; Sun, X.; Yang, J.; Zeng, W.; Wu, F. Lossless compression of JPEG coded photo collections. *IEEE Trans. Image Process.* **2016**, *25*, 2684–2696. [CrossRef] [PubMed]
11. Wang, H.; Tian, T.; Ma, M.; Wu, J. Joint compression of near-duplicate Videos. *IEEE Trans. Multimedia* **2017**, *19*, 908–920. [CrossRef]
12. Ma, C.; Liu, D.; Peng, X.; Wu, F. Surveillance video coding with vehicle library. In Proceedings of the IEEE International Conference on Image Processing, Beijing, China, 17–20 September 2017; pp. 270–274.
13. Xiao, J.; Liao, L.; Hu, J.; Chen, Y.; Hu, R. Exploiting global redundancy in big surveillance video data for efficient coding. *Clust. Comput.* **2015**, *18*, 531–540. [CrossRef]
14. Xiao, J.; Hu, R.; Liao, L.; Chen, Y.; Wang, Z.; Xiong, Z. Knowledge-based coding of objects for multi-source surveillance video data. *IEEE Trans. Multimedia* **2016**, *18*, 1691–1706. [CrossRef]
15. Google Earth V 7.1.5.1557. (7 July 2015). Valencia, Spain. Available online: https://www.google.com/earth/download/gep/agree.html (accessed on 12 March 2018).
16. Mielikainen, J.; Toivanen, P. Clustered DPCM for the lossless compression of hyperspectral images. *IEEE Trans. Geosci. Remote Sens.* **2003**, *41*, 2943–2946. [CrossRef]
17. Magli, E.; Olmo, G.; Quacchio, E. Optimized onboard lossless and near-lossless compression of hyperspectral data using CALIC. *IEEE Geosci. Remote Sens. Lett.* **2004**, *1*, 21–25. [CrossRef]
18. Toivanen, P.; Kubasova, O.; Mielikainen, J. Correlation-based band-ordering heuristic for lossless compression of hyperspectral sounder data. *IEEE Geosci. Remote Sens. Lett.* **2005**, *2*, 50–54. [CrossRef]
19. Penna, B.; Tillo, T.; Magii, E.; Olino, G. Transform coding techniques for lossy hyperspectral data compression. *IEEE Trans. Geosci. Remote Sens.* **2007**, *45*, 1408–1421. [CrossRef]
20. Liu, G.; Zhao, F. Efficient compression algorithm for hyperspectral images based on correlation coefficients adaptive 3D zerotree coding. *JET Image Process.* **2007**, *2*, 72–82. [CrossRef]

21. Cagnazzo, M.; Poggi, G.; Verdoliva, L. Region-based transform coding of multispectral images. *IEEE Trans. Image Process.* **2007**, *16*, 2916–2926. [CrossRef] [PubMed]
22. Ngadiran, R.; Boussakta, S.; Bouridane, A.; Syarif, B. Hyperspectral image compression with modified 3D SPECK. In Proceedings of the International Symposium on Communication Systems Networks and Digital Signal Processing, Newcastle upon Tyne, UK, 21–23 July 2010; pp. 806–810.
23. SkySat-C Generation Satellite Sensors. Available online: https://www.satimagingcorp.com/satellite-sensors/skysat-1/ (accessed on 12 March 2018).
24. Chen, C.; Cai, J.; Lin, W.; Shi, G. Surveillance video coding via low-rank and sparse decomposition. In Proceedings of the 20th ACM International Conference on Multimedia, Nara, Japan, 29 October–2 November 2012; pp. 713–716.
25. Chen, C.; Cai, J.; Lin, W.; Shi, G. Incremental low-rank and sparse decomposition for compressing videos captured by fixed cameras. *J. Vis. Commun. Image Represent.* **2015**, *26*, 338–348. [CrossRef]
26. Hou, J.; Chau, L.P.; Magnenat-Thalmann, N.; He, Y. Sparse low-rank matrix approximation for data compression. *IEEE Trans. Circuits Syst. Video Technol.* **2017**, *27*, 1043–1054. [CrossRef]
27. Guo, S.; Wang, Y.; Tian, Y.; Xing, P.; Gao, W. Quality-progressive coding for high bit-rate background frames on surveillance videos. In Proceedings of the IEEE International Symposium on Circuits and Systems, Lisbon, Portugal, 24–27 May 2015; pp. 2764–2767.
28. Yin, L.; Hu, R.; Chen, S.; Xiao, J.; Hu, J. A block-based background model for surveillance video coding. In Proceedings of the Data Compression Conference, Snowbird, UT, USA, 7–9 April 2015; p. 476.
29. Chen, F.; Li, H.; Li, L.; Liu, D.; Wu, F. Block-composed background reference for high efficiency video coding. *IEEE Trans. Circuits Syst. Video Technol.* **2017**, *27*, 2639–2651. [CrossRef]
30. Chakraborty, S.; Paul, M.; Murshed, M.; Ali, M. Adaptive weighted non-parametric background model for efficient video coding. *Neurocomputing* **2017**, *226*, 35–45. [CrossRef]
31. Song, X.; Peng, X.; Xu, J.; Wu, F. Cloud-based distributed image coding. *IEEE Trans. Circuits Syst. Video Technol.* **2015**, *25*, 1926–1940. [CrossRef]
32. Weinzaepfel, P.; Jégou, H.; Pérez, P. Reconstructing an image from its local descriptors. In Proceedings of the IEEE International Conference on Computer Vision and Pattern Recognition, Providence, RI, USA, 20–25 June 2011; pp. 337–344.
33. Au, O.; Li, S.; Zou, R.; Dai, W.; Sun, L. Digital photo album compression based on global motion compensation and intra/inter prediction. In Proceedings of the International Conference on Audio, Language and Image Processing, Shanghai, China, 16–18 July 2012; pp. 84–90.
34. Zou, R.; Au, O.C.; Zhou, G.; Dai, W.; Hu, W.; Wan, P. Personal photo album compression and management. In Proceedings of the IEEE International Symposium on Circuits and Systems, Beijing, China, 19–23 May 2013; pp. 1428–1431.
35. Lu, X.; Chen, Y.; Li, X. Hierarchical Recurrent Neural Hashing for Image Retrieval with Hierarchical Convolutional Features. *IEEE Trans. Image Process.* **2018**, *27*, 106–120. [CrossRef] [PubMed]
36. Lu, X.; Zhang, W.; Li, X. A Hybrid Sparsity and Distance-based Discrimination Detector for Hyperspectral Images. *IEEE Trans. Geosci. Remote Sens.* **2018**, *56*, 1704–1717. [CrossRef]
37. Lu, X.; Wang, B.; Li, X.; Zheng, X. Exploring Models and Data for Remote Sensing Image Caption Generation. *IEEE Trans. Geosci. Remote Sens.* **2017**, *56*, 2183–2195. [CrossRef]
38. Lu, X.; Zheng, X.; Yuan, Y. Remote Sensing Scene Classification by Unsupervised Representation Learning. *IEEE Trans. Geosci. Remote Sens.* **2017**, *55*, 5148–5157. [CrossRef]
39. Lowe, D.G. Distinctive Image Features from Scale-Invariant Keypoints. *Int. J. Comput. Vis.* **2004**, *60*, 91–110. [CrossRef]
40. Sedaghat, A.; Ebadi, H. Distinctive Order Based Self-Similarity descriptor for multi-sensor remote sensing image matching. *ISPRS J. Photogramm. Remote Sens.* **2015**, *108*, 62–71. [CrossRef]
41. Reinhard, E.; Ashikhmin, M.; Gooch, B.; Shirley, P. Color Transfer between Image. *IEEE Comput. Graph. Appl.* **2002**, *21*, 34–41. [CrossRef]
42. Ko, Y.; Yi, Y.; Ha, S. An efficient parallelization technique for *x264* encoder on heterogeneous platforms consisting of CPUs and GPUs. *J. Real-Time Image Process.* **2014**, *9*, 5–18. [CrossRef]
43. Bjontegaard, G. *Calculation of Average PSNR Difference between RD-Curves*; ITU-T SG16 Q.6 Doc.; Technical Report VCEG-M33; ITU-T: Austin, TX, USA, 2001.

44. HEVC Test Model, HM Reference Software. Available online: https://hevc.hhi.fraunhofer.de/ (accessed on 28 July 2017).
45. x264 Free Library/Codec, 64-bit, 8-bit Depth Version r2901. Available online: http://www.divx-digest.com/software/x264.html (accessed on 20 January 2018).