Robust Multi-Bit Watermarking for Free-View Television Using Light Field Rendering

Huawei TIAN†*, Yao ZHAO†*, Zheng WANG†, Nonmembers, Rongrong NI†, Member, and Lunming QIN†, Nonmember

SUMMARY With the rapid development of multi-view video coding (MVC) and light field rendering (LFR), Free-View Television (FTV) has emerged as a new entertainment equipment, which can bring more immersive and realistic feelings for TV viewers. In FTV broadcasting system, the TV-viewer can freely watch a realistic arbitrary view of a scene generated from a number of original views. In such a scenario, the ownership of the multi-view video should be verified not only on the original views, but also on any virtual view. However, capacities of existing watermarking schemes as copyright protection methods for LFR-based FTV are only one bit, i.e., presence or absence of the watermark, which seriously impacts its usage in practical scenarios. In this paper, we propose a robust multi-bit watermarking scheme for LFR-based free-view video. The direct-sequence code division multiple access (DS-CDMA) watermark is constructed according to the multi-bit message and embedded into DCT domain of each view frame. The message can be extracted bit-by-bit from a virtual frame generated at an arbitrary viewpoint with a correlation detector. Furthermore, we mathematically prove that the watermark can be detected from any virtual view. Experimental results also show that the watermark in FTV can be successfully detected from a virtual view. Moreover, the proposed watermark method is robust against common signal processing attacks, such as Gaussian filtering, salt & peppers noising, JPEG compression, and center cropping.

key words: free-view television, light field rendering, 3-D watermarking

1. Introduction

In recent years, significant consumer mass markets of 3-D videos have developed, including professional applications (e.g., scientific visualization and medical devices) and entertainment (e.g., 3-D cinemas, 3-D TV broadcasting and 3-D gaming) [1]. Entertainment companies in particular, such as Twentieth Century Fox Film Corporation, DreamWorks SKG, Walt Disney Pictures, Pixar Animation Studios and other film corporations have produced many excellent 3-D videos. 3-D watermarking for 3-D content protection is proposed as a significant solution for the developing of 3-D industry. According to the embedding and extraction spaces, the 3-D watermarking can be classified into three categories: 3-D/3-D, 3-D/2-D, and 2-D/2-D watermarking [2]–[4], [6]. In 3-D/3-D watermarking [7], [8], the watermark is embedded into and extracted from a 3-D geometric model, to protect the copyright of the 3-D mesh model. The 3-D/2-D watermarking [9] aims to embed watermark in 3-D objects and extract it from its 2-D projection images, intending to protect 2-D representations of a textured 3-D object. The 2-D/2-D watermarking [2]–[5] deals with the content protection of image-based 3D data representations (e.g., FTV) in which both the embedding and extraction are applied in 2-D space directly. Since 3-D broadcasting data are usually captured and transmitted in image-based representations, the 2-D/2-D watermarking becomes one of the most popular approaches for 3-D content protection.

All 3-D video systems should provide the ability to interactively control the viewpoint, a feature that has been termed free-viewpoint video by the MPEG Ad-Hoc Group on 3-D Audio and Video [10]. In addition, standardization on multi-view coding has been delivered by ISO and ITU bodies, and FTV is expected to be the next goal for standardization [1], [11], [12]. Recently, generating a realistic arbitrary view of a scene from a number of original views has become faster and cheaper with the advances in image based rendering (IBR) [13]. One of the main applications is FTV, where viewers can freely select the viewing position and angle via IBR on the transmitted multi-view videos.

Similar to the copyright problems of images [14], mono-view video [15]–[17], 3-D geometric structure [7], [8], and depth-image-based rendering 3D image/video [4], the copyright problem for free-view video can also be treated by an appropriate watermarking method. However, there are more challenging requirements, compared to the well-studied mono-view video watermarking. The owner of the multi-view video should verify his/her ownership, not only on the original views of the multi-view video, but also on any virtual view, which is generated using IBR from the original views. A 3-D watermarking scheme has recently emerged after the progresses in IBR techniques [2], [3]. Koz et al. proposed a watermarking scheme in which watermark is inserted into each view frame of multiview video in [2], [3]. The watermark is modulated with each view frame filtered by a high-pass filter, and spatially added onto the view frame. The watermark is a pseudo-random sequence generated from a Gaussian distribution with zero mean and unit variance. The normalized correlation between the virtual view frame and the equally rendered watermark is calculated to verify the existence of watermark. In [5], Apostolidis et al. tested several Mathematical Distributions as watermark generators and the parameters of the watermarking...
scheme, in order to evaluate their efficiency and functionality in terms of watermark’s robustness for cases of rendered images of FTV. In [18], Ramachandra et al. proposed a SIFT (scale invariant feature descriptor) based fingerprinting mechanism of FTV which can identify the view interpolation attack and the change of focus/display plane attack.

In fact, the approach in [2], [3], [5] is intended to embed only one bit of information, i.e., presence or absence of the watermark. In real application scenarios, many more bits are usually needed to represent a watermark, such as a meaningful mark or signature, etc., so these approaches are quite limited.

In this work, we propose a multi-bit watermarking scheme for free-view video. A direct-sequence code division multiple access (DS-CDMA) [19] watermark is constructed according to the multi-bit message and embedded in DCT domain of each view frame. The message is extracted bit-by-bit with a correlation detector from a watermarked view frame or a virtual frame generated for an arbitrary view. We also mathematically prove that the watermark information can be detected from the virtual view generated for an arbitrary view. Extensive experimental results demonstrate the robustness against various attacks.

The rest of the paper is organized as follows. In Sect. 2, the light field rendering (LFR) [20] approach is introduced firstly, which is one of the competing IBR technology for FTV systems [21]. Section 3 describes the details of watermark embedding and detection procedure. In addition, we mathematically prove that the watermark can be detected from the virtual view generated for an arbitrary view. The experimental results are illustrated in Sect. 4, and finally some conclusions are drawn in Sect. 5.

2. Light Field Rendering

In literatures, light field approach is the most well-known and preferred IBR technique. The first reason is that it does not require geometry information but only relies on scene images which can be easily captured by common digital cameras [22]. Second, it avoids building complex models, such as depth values or image correspondences, to extract the image values. Third, the new view can be constructed in real time and is independent of the scene complexity (only related with the size of the rendered image).

In practice, a light ray is parameterized as a line by its intersections with two parallel planes, namely the camera plane and the focal plane (see Fig. 1 (a)). In Fig. 1 (a), a light ray is shown and indexed as an integer 4-tuple \((u_0, v_0, s_0, t_0)\), where \((u_0, v_0)\) and \((s_0, t_0)\) are the intersections of the light ray with camera and focal planes, respectively. The two planes are discrete so that a finite number of light rays can be recorded. If the light rays from all the points on the focal plane arrive at one point on the camera plane, then an image is generated (2-D array of light rays). Therefore, the two planes can be interpreted as a 2-D array of images, as shown in Fig. 1 (b). To generate a virtual view of the object for a randomly selected view-point, the light ray for each pixel of the rendered image is calculated by quad linear interpolation existing nearby light rays in the image array. In the LFR, bilinear interpolation (BI) and nearest neighborhood interpolation (NI) are two main interpolation methods while generating the virtual view from the original camera locations. Figure 2 shows the two interpolation methods.
In BI-based LFR, the intensity value of each rendered pixel is obtained from a weighted sum of pixels belonging to four neighboring cameras (shown in Fig. 2). The BI function is defined as:

\[
I_i(x_0, y_0) = W_{\text{nearest}}(I_1(s_0, t_0), I_2(s_0, t_0), I_3(s_0, t_0), I_4(s_0, t_0))
\]

\[
= (1-a)(1-b)I_1(s_0, t_0) + a(1-b)I_2(s_0, t_0)
+ (1-a)bI_3(s_0, t_0) + abI_4(s_0, t_0)
\]

where \(I_1, I_2, I_3\) and \(I_4\) are four light field images corresponding to the cameras at the four neighboring positions on the camera plane. \(I_i\) is the rendered image corresponding to the virtual camera. \(a\) is the v-distance between \((u_0, v_0)\) and the position of Cam. 1, \(b\) is the u-distance between \((u_0, v_0)\) and the position of Cam. 1 as shown in Fig. 2. \((s_0, t_0)\) is the intersection of the light ray with the focal plane. If the light ray does not intersect a grid point of focus plane, the value of \((s_0, t_0)\) is set as the value of the nearest grid point of focus plane.

In NI-based LFR, the intensity value of each pixel for VI-based LFR, the intensity value of each rendered image is generated from the pixel intensity belonging only to the nearest camera. The NI function can be defined as:

\[
I_i(x_0, y_0) = W_{\text{nearest}}(I_1(s_0, t_0), I_2(s_0, t_0), I_3(s_0, t_0), I_4(s_0, t_0))
\]

\[
= W_{\text{nearest}}(I_i(s_0, t_0), I_j(s_0, t_0), f, a, b)
\]

\[
= \begin{cases} 
1 & \text{if } a \leq 1 - a \& b \leq 1 - b \\
2 & \text{if } 1 - a < a \& b \leq 1 - b \\
3 & \text{if } a \leq 1 - a \& 1 - b < b \\
4 & \text{if } 1 - a < a \& 1 - b < b.
\end{cases}
\]

3. Proposed Watermarking Method

3.1 Watermarking Embedding

In the proposed watermarking scheme, the spread spectrum-based [23] embedding method is used to embed watermark into each image of the light field image array. The watermarking embedding procedure is demonstrated in Fig. 3 and summarized as follows:

1) Generate \(M\) uncorrelated 1-D binary pseudo random sequences \(p_i, i = 1, \ldots, M\), as signature patterns using the private key as seed. Each of these sequences has zero mean and takes values from binary alphabet \([-1, 1]\). \(M\) is the number of bits in the watermark message. The length of \(p_i\) is \(N, N > M\).

2) Create a 1-D DS-CDMA watermark signature \(W_1\) by modulating the watermark message with the patterns generated in Step 1), i.e., \(W_1 = \sum_{i=1}^{M} w_i p_i\), where \(w_i\) is the \(i\)th bit (i.e., -1 or 1) in the watermark message \(w = [w_1, w_2, \ldots, w_i, \ldots, w_M]\).

3) Convert the 1-D signature \(W_1\) into a 2-D signature \(W_2\) in a pre-selected zigzag scan (e.g., mid-range DCT coefficients); other coefficients are set to zero.

4) Apply the inverse discrete cosine transform (IDCT) to the 2-D signature \(W_2\) to produce \(W\).

5) According to the camera calibration information of an original light field image, we select a part of final watermark signature \(W\) corresponding to the original light field image. As shown in Fig. 3, the selected part is embedded into corresponding original light field image \(I\) using the formula:

\[
I' = I + \lambda W,
\]

where \(\lambda\) is the watermarking strength. It produces the watermarked light field image \(I'\). The embedding procedure can make sure that the same watermark is added to the pixels of different camera views, whose corresponding light rays are intersected at the same pixel in the focal plane. Furthermore, it can avoid the superposition of the different watermarks from different camera views in the interpolation during the rendering.

The whole procedure is equivalent to embedding the watermark signature \(W\) into the DCT domain of the light field image.

3.2 Watermarking Extraction

Rather than dealing with the general attacks for image and video watermarking, the major challenge of FTV is extracting the watermark message from an arbitrary view generated by LFR. The strategy of estimating the position and rotation for the imagery view has been investigated by Koz et al. in [3], therefore we can only focus on the state that the position and rotation of the virtual camera is known. The following
steps are taken to decode the watermark message in a rendered image $I_r$:

1) Regenerate $M$ uncorrelated 1-D binary pseudo random sequence $p_i$, $i = 1, \ldots, M$, using the same key as in Step 1) of watermarking embedding. $M$ is the number of bits in the watermark message. Each of these sequences has zero mean and takes values from binary alphabet $\{-1, 1\}$;  

2) Convert the 1-D pseudo random sequence $p_i$ into a 2-D $p_i^T$ in a pre-selected zigzag scan (e.g., mid-range DCT coefficients), other coefficients are set to zero;  

3) Apply the IDCT to the 2-D pseudo random sequence $p_i^T$ to produce $P_i$;  

4) Apply the same rendering operation as generating the arbitrary view $I_a$ to watermark signature $P_i$, in order to generate a rendered watermark $P_i^r$ (assuming the position and rotation of the virtual camera is known);  

5) Decode the watermark message bit-by-bit using a correlation detector. That is, the $i$th bit of the watermark message is decoded as  

$$ \hat{w}_i = \begin{cases} 1, & \text{corr}(I_i^r, P_i^r) > T, \\ -1, & \text{corr}(I_i^r, P_i^r) \leq T, \end{cases} $$

where $\text{corr}(\cdot)$ is used to compute the correlation coefficient of two vectors and $T$ is a threshold. Selecting the threshold is a classical decision estimation problem in which we wish to minimize both the rate of false negatives and false positives. Reader can refer to [23] for more detail of selecting threshold. The extracted watermark message is $\hat{w} = [\hat{w}_1, \hat{w}_2, \ldots, \hat{w}_i, \ldots, \hat{w}_M]$, $\hat{w}_i \in \{-1, 1\}$.

3.3 Mathematical Analysis of the Watermarking Scheme

We mathematically prove that the watermark can be detected from the virtual view image generated for an arbitrary view. Firstly, we assume $I_a$ is the rendered image corresponding to the virtual camera, $I_1$, $I_2$, $I_3$, and $I_4$ are four light field images corresponding to the cameras at the four neighboring positions on the camera plane. Let $I_i^r$ denote the image of virtual view rendered from watermark light field images, and $I_1^r$, $I_2^r$, $I_3^r$, and $I_4^r$ denote the four watermarked light field images corresponding to $I_1$, $I_2$, $I_3$, and $I_4$, $I_1^r = I_1 + \lambda W$, $I_2^r = I_2 + \lambda W$, $I_3^r = I_3 + \lambda W$, and $I_4^r = I_4 + \lambda W$, where $W$ is the final watermark signature in Step 5) of watermarking embedding procedure and $\lambda$ is the watermarking strength. As shown in Fig. 2, BI and NI are two main interpolation methods while generating the virtual view from the original camera locations. Therefore, we discuss the problem under two states.

3.3.1 In Bilinear Interpolation-Based LFR

According to the BI function (1), the pixel $I_i^r(x_0, y_0)$ in the image of virtual view rendered from watermarked light field images is computed as the following equation:

$$ I_i^r(x_0, y_0) = F_{\text{bilinear}}(I_i^r(s_0, t_0), I_i^r(s_0 + t_0, t_0), I_i^r(s_0 + t_0, s_0 + t_0), I_i^r(s_0, s_0 + t_0)) = (1-a)(1-b)I_i^r(s_0, s_0) + a(1-b)I_i^r(s_0, t_0) + (1-a)bI_i^r(s_0, t_0) + abI_i^r(t_0, t_0), $$

where $a$ is the $x$-distance between $(u_0, v_0)$ and the position of Cam. 1, $b$ is the $y$-distance between $(u_0, v_0)$ and the position of Cam. 1 as shown in Fig. 2. The final watermark signature $W$ which is embedded into each original light field image is same, therefore $I_1^r = I_1 + \lambda W$, $I_2^r = I_2 + \lambda W$, $I_3^r = I_3 + \lambda W$, and $I_4^r = I_4 + \lambda W$. And then

$$ I_i^r(x_0, y_0) = F_{\text{bilinear}}(I_i^r(s_0, t_0), I_i^r(s_0, t_0), I_i^r(s_0, t_0), I_i^r(s_0, t_0)) = (1-a)(1-b)I_i^r(s_0, t_0) + a(1-b)I_i^r(s_0, t_0) + (1-a)bI_i^r(s_0, t_0) + abI_i^r(t_0, t_0) = [(1-a)(1-b)I_i^r(s_0, t_0) + a(1-b)I_i^r(s_0, t_0) + (1-a)bI_i^r(s_0, t_0) + abI_i^r(t_0, t_0)] = W(s_0, t_0), $$

Because $I_i^a(x_0, y_0) = I_i^r(x_0, y_0) + \lambda W$, we can get the following equation:

$$ I_i^r(x_0, y_0) = I_i^r(x_0, y_0) + \lambda W(s_0, t_0). $$

Because the watermark embedded into each original light field image is same, we can get the following equation:

$$ W(x_0, y_0) = F_{\text{bilinear}}(W(s_0, t_0), W(s_0, t_0), W(s_0, t_0), W(s_0, t_0)) = W(s_0, t_0), $$

therefore,

$$ I_i^r(x_0, y_0) = I_i^r(x_0, y_0) + \lambda W^r(x_0, y_0), $$

where $W^r = \sum_{i=1}^{M} w_i P_i$, $w_i$ is the $i$th bit (i.e., -1 or 1) in the watermark message $w = [w_1, w_2, \ldots, w_i, \ldots, w_M]$.

In watermarking extraction, in order to detect the $i$th bit of watermark message we apply the same rendering operations to generate a rendered watermark $P_i^r$ and $\text{corr}(I_i^r, P_i^r)$ is used to decode the $i$th bit $\hat{w}_i$ of the watermarking. According to (9), $\text{corr}(I_i^r, P_i^r)$ can be computed as the following equation:

$$ \text{corr}(I_i^r, P_i^r) = \text{corr}(I_i^r + \lambda W^r, P_i^r) = \text{corr}(I_i + \lambda W^r + \lambda W, P_i^r) = \text{corr}(I_i, P_i^r) + \lambda \text{corr}(W, P_i^r) = \text{corr}(I_i, P_i^r) + \lambda \text{corr}(W, P_i^r) + \lambda \text{corr}(W^r, P_i^r) = \lambda w_i. $$
According to (4) and (10), we can obtain the conclusion that the watermark message can be successfully detected from the virtual view generated for an arbitrary view with the proposed watermarking scheme in the NI-based LFR.

3.3.2 In Nearest Neighborhood Interpolation-Based LFR

According to the NI function (2), the pixel $I'_i(x_0, y_0)$ in the watermarked image of the virtual view is computed as the following equation:

$$I'_i(x_0, y_0) = F_{\text{nearest}}(I'_1(s_0, t_0), I'_2(s_0, t_0), I'_3(s_0, t_0), I'_4(s_0, t_0)) = F_{\text{nearest}}(I_1(s_0, t_0) + \lambda W(s_0, t_0), I_2(s_0, t_0) + \lambda W(s_0, t_0), I_3(s_0, t_0) + \lambda W(s_0, t_0), I_4(s_0, t_0) + \lambda W(s_0, t_0)) = I_f(s_0, t_0) + \lambda W(s_0, t_0), j \in \{1, 2, 3, 4\}.$$  

Because the watermark embedded into each original light field image is same, we can get the following equation:

$$W'(x_0, y_0) = F_{\text{nearest}}(W(s_0, t_0), W(s_0, t_0), W(s_0, t_0), W(s_0, t_0)) = W(s_0, t_0).$$

According to (2), (11), and (12), we can get the following equation:

$$I'_i(x_0, y_0) = I_i(x_0, y_0) + \lambda W'(x_0, y_0).$$  

In watermarking extraction, in order to detect the $i$th bit of watermark message we apply the same rendering operations to generate a rendered watermark $P'_i$ and $\text{corr}(I'_i, P'_i)$ is used to decode the $i$th bit $\hat{w}_i$ of the watermarking. According to (13), $\text{corr}(I'_i, P'_i)$ can be computed as the following equation:

$$\text{corr}(I'_i, P'_i) = \text{corr}(I_i, W'(P'_i)) = \text{corr}(I_i, \lambda(\sum_{j=1}^{M} w_j P'_i), P'_i) = \lambda w_i.$$  

According to (4) and (14), we can obtain the conclusion that the watermark message can be successfully detected from the virtual view generated for an arbitrary view with the proposed watermarking scheme in the NI-based LFR.

4. Experimental Results

To evaluate the imperceptibility and robustness of the proposed watermarking scheme, we conduct experiments on two common light fields, Buddha and Dragon [24]. The parameterization of the focal and camera plane for Buddha and Dragon light field is shown in Fig. 5. There are $32 \times 32$ original views in the Buddha light field array and Dragon light field array. The size of all view images is $256 \times 256$ pixels. The length of the pseudo random sequence $p_i$ is set as $N = 3 \times 10^4$. In Step 3) of watermark embedding procedure, the selection of mid-range DCT coefficients starts from the 1800th coefficient in the zigzag scan order. There are three kinds of watermark message in our experiments ($M = 50$ bits, $M = 100$ bit, and $M = 150$ bits). The watermark message is only embedded into the luminance of the color image in the simulations. The capacity of the watermarking scheme will triple, if watermark message is embedded into three components (i.e. RGB channels). The decoding bit-error rate (BER), defined as the ratio between the number of incorrectly decoded bits and the total number of embedded bits, is used to evaluate the robustness of the watermarking scheme. 100 different randomly generated watermark sequences are tried and the BER values are taken as the average of the 100 cases.

4.1 Imperceptibility Test

The peak signal to noise ratio (PSNR) value between the original and the watermarked image is a criterion for the watermark imperceptibility. The embedding distortion can be controlled by $\lambda$ in (3). The relationship between PSNR and $\lambda$ is shown in Fig. 6 when the length of watermark message $M = 50$ bits. In our experiments, the PSNR value is about 35dB. The PSNR value is the average of $32 \times 32$ views which are watermarked with 100 different randomly generated watermark sequences. An original view and watermarked view of Buddha light field array and Dragon light field array are presented in Fig. 7. As shown in Fig. 7, there is no visible difference between the original and watermarked views based on subjective assessment.

Typical rendered views for the original and water-
marked Buddha and Dragon are presented in Fig. 8, where the length of watermark message is \( M = 50 \) bits. In Buddha and Dragon, the camera position of \([0, 0, 2]\) and normal direction of (Position-A in Fig. 5) are taken as references. In order to describe the effect of translation and rotation, six cases are considered during the simulations as shown in Table 1.

In the third step of the imperceptibility tests, the visual qualities of the original and watermarked rendered videos are also tested. The first group of videos is obtained by sequencing the light field images according to the circular trajectory (Path I) on the camera plane in Fig. 5.

### Table 1 Six cases for the creation of rendered views in the Dragon light field and the Buddha light field.

<table>
<thead>
<tr>
<th>Transl. on ( u )?</th>
<th>Transl. on ( v )?</th>
<th>Rotat. ?</th>
<th>Position</th>
<th>Direction</th>
<th>Label</th>
</tr>
</thead>
<tbody>
<tr>
<td>I</td>
<td>No</td>
<td>No</td>
<td>( [0,0,2] )</td>
<td>( [0,0,-1] )</td>
<td>A</td>
</tr>
<tr>
<td>II</td>
<td>Yes</td>
<td>No</td>
<td>( [0,0,3] )</td>
<td>( [0,0,-1] )</td>
<td>B</td>
</tr>
<tr>
<td>III</td>
<td>Yes</td>
<td>Yes</td>
<td>( [1,5,0,2] )</td>
<td>( [1,0,-1] )</td>
<td>C</td>
</tr>
<tr>
<td>IV</td>
<td>Yes</td>
<td>No</td>
<td>( [1,5,0,2] )</td>
<td>( [1,0,-1] )</td>
<td>D</td>
</tr>
<tr>
<td>V</td>
<td>Yes</td>
<td>Yes</td>
<td>( [1,5,0,2] )</td>
<td>( [1,0,-1] )</td>
<td>E</td>
</tr>
<tr>
<td>VI</td>
<td>Yes</td>
<td>Yes</td>
<td>( [1,5,0,2.5] )</td>
<td>( [-1,0,-1] )</td>
<td>F</td>
</tr>
</tbody>
</table>

The second group of videos is obtained by sequencing the light field images according to the straight trajectory (Path II) on the camera plane in Fig. 5. The rendered videos from the original and watermarked Buddha and Dragon for the illustrated Path I and Path II in Fig. 5 are given in https://sites.google.com/site/hwtianwatermark. Based on subjective assessment, there is no visible difference between the watermarked and original rendered videos.

### 4.2 Robustness Test for Rendering

1) Rendering from Six Typical Positions: In the robustness tests, the extraction is applied for different imagery views based upon the virtual camera position and orientation. Six cases in Buddha and Dragon light field are considered in the simulations as shown in Table 1. These cases cover the translation, rotation and scaling for the rendered views. The robustness tests of the six cases are evaluated with the average BER of 100 random watermark sequences are shown in Fig. 9. From Fig. 9 we can see that the proposed watermarking scheme performs very well on different imagery views. For Buddha and Dragon, the BER values are all 0 in the six typical cases when the length of watermark message is 50 bits. In general, the BER of watermarking scheme increases along with the increasing of the length of watermark message. However, the robustness of the proposed watermarking scheme performs also very well when the length of watermark message increases. As shown in Fig. 9(a) and (b), the largest BER is only 0.005 for Buddha light field when the interpolation method is BI. The largest BER is 0.03 for Dragon light field with BI as shown in Fig. 9(c) and (d). From Fig. 9 we can see that all BER values are very low when the length of watermark message \( M = 50, 100, \) and 150 bits, respectively. Therefore the proposed watermarking scheme is robust against the typical rendering cases covering translation, rotation and scaling.

2) Rendering from the Circular Trajectory: In the second group of robustness tests, the extraction scheme is applied for eight different imagery view-points on the circular trajectory (Path I in Fig. 5). Eight virtual cameras are located at Path I in Fig. 5 with \( \theta = -40^\circ, -30^\circ, -20^\circ, -10^\circ, 10^\circ, 20^\circ, 30^\circ, \) and \( 40^\circ \), respectively. The robustness tests of the eight cases evaluated with the average BER of 100 random watermark sequences are shown in Fig. 10. All the BER values of the watermarking scheme are all very low when \( M = 50, 100, \) and 150 bits, respectively. The largest BER is only 0.002. Therefore the proposed watermarking method based upon the virtual camera position and orientation is robust against the typical rendering cases covering translation, rotation and scaling.

---

![Fig. 7](image-url) The performance of the proposed watermarking scheme on original views of light field array. (a) One original view in Buddha light field array. (b) Watermarked view of (a). (c) One original view in Dragon light field array. (d) Watermarked view of (c).

![Fig. 8](image-url) The performance of the proposed watermarking scheme on six typical rendered views. The 1st row: rendered views for original Buddha. The 2nd row: rendered views for watermarked Buddha. The 3rd row: rendered views for original Dragon. The 4th row: rendered views for watermarked Dragon. The 1st column: Position-A in Fig. 5. The 2nd column: Position-B in Fig. 5. The 3rd column: Position-C in Fig. 5. The 4th column: Position-D in Fig. 5. The 5th column: Position-E in Fig. 5. The 6th column: Position-F in Fig. 5.
scheme is robust enough against rendering when a virtual view is generated on the circular trajectory.

4.3 Robustness Test against Other Attacks

We also evaluate the robustness of the watermarking method against common signal processing attacks, because they could occur in the transmission chain of FTV. For these tests, we consider six typical positions and eight different imagery view-points on the circular trajectory (Path I shown in Fig. 5). $M = 50$ bits. The performance is shown in Tables 2 and 3. We choose a set of attacks as listed in Table 4 to
Table 3 Robustness against various attacks for eight cases of circular trajectory.

<table>
<thead>
<tr>
<th>LF Image</th>
<th>Attack</th>
<th>Nearest Neighborhood Interpolation</th>
<th>Bilinear Interpolation</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>−40°</td>
<td>−30° −20° −10° 10° 20° 30° 40°</td>
<td>−40° −30° −20° −10° 10° 20° 30° 40°</td>
</tr>
<tr>
<td>Buddha</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Dragon</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Table 4 A set of common signal processing attacks and cropping attacks for robustness evaluating.

<table>
<thead>
<tr>
<th>Attack Types</th>
<th>Parameters</th>
<th>Marks</th>
</tr>
</thead>
<tbody>
<tr>
<td>Medina filter</td>
<td>( size = 2 \times 2 )</td>
<td>( A_1 )</td>
</tr>
<tr>
<td>Mean filter</td>
<td>( size = 2 \times 2 )</td>
<td>( A_2 )</td>
</tr>
<tr>
<td>Gaussian filter</td>
<td>( size = 3 \times 3 )</td>
<td>( A_3 )</td>
</tr>
<tr>
<td>Uniform noise</td>
<td>( \beta = 0.01 )</td>
<td>( A_4 )</td>
</tr>
<tr>
<td>Salt &amp; peppers noise</td>
<td>( scale = 0.04 )</td>
<td>( A_5 )</td>
</tr>
<tr>
<td>Gaussian noise</td>
<td>( variance = 0.01 )</td>
<td>( A_6 )</td>
</tr>
<tr>
<td>JPEG Compression</td>
<td>( ratio q = 70% )</td>
<td>( A_7 )</td>
</tr>
<tr>
<td>Cropping</td>
<td>( percent = 30% )</td>
<td>( A_8 )</td>
</tr>
</tbody>
</table>

The attacked image with adding uniform noise is

\[
I'(x,y) = I(x,y) \cdot (1 + \beta \cdot n(x,y)),
\]

where \( I(x,y) \) is the pixel grayscale value of an input image at \((x,y)\), \(\beta\) is a parameter that controls the strength of the additive noise, \(n(x,y)\) is noise with uniform distribution, zero mean and unit variance, and \(I'(x,y)\) is the pixel grayscale value of the attacked image.

Tables 2 and 3 show the effect of the 17 different attacks listed in Table 4 on the watermark. Corresponding to the robustness test results, we have the following observations. 1) Common Filtering Operations (A_1~A_6): Tables 2 and 3 show that the watermark is robust enough to median, mean and Gaussian filter operations. Especially, BER values for Buddha under filter operations are all 0 in Table 3. 2) Additive Noises (A_7~A_12): The watermarking scheme is robust to additive noise including uniform noise, salt & peppers noise, and Gaussian noise operations. As shown in Tables 2 and 3, most of BER values under additive noise operations are 0. 3) Image Compression (A_13~A_15): The robustness of watermarking scheme performs very well against JPEG compression as shown in Tables 2 and 3. Even in the worst case, the mean is a 4×4 matrix

\[
\begin{bmatrix}
0.0001 & 0.0044 & 0.0044 & 0.0001 \\
0.0044 & 0.2411 & 0.2411 & 0.0044 \\
0.0044 & 0.2411 & 0.2411 & 0.0044 \\
0.0001 & 0.0044 & 0.0044 & 0.0001
\end{bmatrix}
\]

examine the robustness of the watermark. These attacks include median filtering of size 2×2 and 3×3, mean filtering of size 2×2 and 3×3, Gaussian filtering of size 3×3 and 4×4, adding uniform noise, Gaussian noise, adding salt & peppers noise with \( scale = 0.04 \) and \( scale = 0.05 \), JPEG compression and center cropping. The Gaussian filter matrix of size 3×3 is

\[
\begin{bmatrix}
0.0113 & 0.0838 & 0.0113 \\
0.0838 & 0.6193 & 0.0838 \\
0.0113 & 0.0838 & 0.0113
\end{bmatrix}
\]

The Gaussian filter matrix of size 4×4 is

\[
\begin{bmatrix}
0.0001 & 0.0044 & 0.0044 & 0.0001 \\
0.0044 & 0.2411 & 0.2411 & 0.0044 \\
0.0044 & 0.2411 & 0.2411 & 0.0044 \\
0.0001 & 0.0044 & 0.0044 & 0.0001
\end{bmatrix}
\]

which is the pixel grayscale value of an input image at \((x,y)\), \(\beta\) is a parameter that controls the strength of the additive noise, \(n(x,y)\) is noise with uniform distribution, zero mean and unit variance, and \(I'(x,y)\) is the pixel grayscale value of the attacked image.
case JPEG compression with data compression ratio 30% for the two interpolation schemes in six typical cases and eight cases of circular trajectory, most of BER are very low. 4) Cropping ($A_{16} \sim A_{17}$): Tables 2 and 3 show the watermark resistance to cropping ranging from 30% ~ 50%. Most of BER are 0. In the worst case (Dragon, NI, and typical Case IV), the BER is only 0.092.

Although the main characteristic attack for free-view video is rendering operations during virtual view generation, the proposed watermarking is robust against typical signal processing. From the above observations, we can see that the proposed watermarking scheme is not only resistant to common signal processing but also robust against combined signal processing attacks and light field rendering. Especially, the robustness against median filter, mean filter, Gaussian filter, adding uniform noise, JPEG compression and center cropping of the watermarking scheme performs very well.

5. Conclusion

There are very few watermarking schemes for LFR-based free-view video. Koz et al. proposed a watermarking approach for the free-view video. However, it is only a one-bit watermarking scheme, which seriously impacts its usage in practical scenarios. There are no multi-bit watermarking schemes for free-view video in the current literature. Therefore, a robust multi-bit watermarking scheme for free-view video is proposed in this paper. The watermark message is embedded into DCT domain of each view using DS-CDMA watermarking method. The watermark extraction is carried out in the DCT domain of virtual view generated for an arbitrary view with a correlation detector. Furthermore, we mathematically proved that the watermark can be detected from the virtual view generated for an arbitrary view. Experimental results show that the watermark for FTV can be detected from virtual views generated at an arbitrary viewpoint. Moreover, the proposed scheme is resistant to common signal processing including lowpass filtering, adding noise, JPEG compression and cropping. More importantly, the watermarking scheme is robust against combined signal processing attacks and light field rendering operation.

Acknowledgements

The authors appreciate Prof. Zhouchen Lin and Dr. Gang Cao for very helpful discussions and suggestions during the work on this paper. This work was supported in part by 973 Program (2011CB302204), the NSF of China for Distinguished Young Scholars (61025013), National NSF of China (61073159, 61272355), and the Program for Changjiang Scholars and Innovative Research Team in University (IRT201206).

References

Huawei Tian received the B.E. degree in computer science and technology from the College of Computer Science, Chongqing University, Chongqing, China, in 2006. He received the Ph.D. degree in the Institute of Information Science, Beijing Jiaotong University, Beijing, China, in 2013. Since July 2013, he has been with the faculty of the People’s Public Security University of China. His research interests include image processing, pattern recognition, digital watermarking and steganography, digital forensics, visual attention, and 3-D videos.

Yao Zhao received the B.S. degree from Fuzhou University, Fuzhou, China, in 1989 and the M.E. degree from the Southeast University, Nanjing, China, in 1992, both from the Radio Engineering Department, and the Ph.D. degree from the Institute of Information Science, BJTU, in 1996. He became an Associate Professor at BJTU in 1998 and became a Professor in 2001. From 2001 to 2002, he worked as a senior research fellow in the Information and Communication Theory Group, Faculty of Information Technology and Systems, Delft University of Technology, Delft, The Netherlands. He is now the Director of the Institute of Information Science, BJTU.

Zeng Wang received the B.E. degree in computer science and M.E. degree in signal and information processing from Beijing Jiaotong University, Beijing, China, in 2009 and 2011, respectively. His research interests mainly focus on the pattern recognition and 3D videos.

Rongrong Ni received the Ph.D. degree in signal and information processing from the Institute of Information Science, Beijing Jiaotong University (BJTU), Beijing, China, in 2005. Since Spring 2005, she has been with the faculty of the School of Computer and Information Technology and the Institute of Information Science, BJTU, where she has been an Associate Professor since 2008.

Lunming Qin received the B.E. degree in automation and the M.E. degree in signal and information processing from Beijing Jiaotong University, Beijing, China, in 2003 and 2006, respectively, where he is currently working toward the Ph.D. degree in signal and information processing in the Institute of Information Science.