Exact global motion compensation for holographic video compression

RAEES KIZHAKKUMKARA MUHAMAD1,2,*, DAVID BLINDER1,2, ATHANASIA SYMEONIDOU1,2, TOBIAS BIRNBAUM1,2, OSAMU WATANABE1,3, COLAS SCHRETTER1,2 AND PETER SCELKENS1,2

1 Vrije Universiteit Brussel (VUB), Dept. of Electronics and Informatics (ETRO), Belgium
2 imec, Kapeldreef 75, B-3001 Leuven, Belgium
3 Dept. of Electronics and Computer Systems, Takushoku University, Tokyo, Japan

* kraees@etrovub.be

Abstract: Holographic video requires impractical bitrates for storage and transmission without data compression. We introduce an end-to-end compression pipeline for compressing holographic sequences with known ground truth motion. The compression strategy employs a motion compensation algorithm based on the rotational transformation of angular spectrum. Residuals arising from the compensation step are represented using short-time Fourier transforms and quantized with uniform mid-rise quantizers whose bit depth is determined by a Lagrangian rate-distortion optimization criterion where the distortion metric is the mean squared error. Experiments use computer generated holographic videos and we report Bjøntegaard delta peak signal to noise ratio gains of around 20 dB when compared to traditional image/video codecs.

© 2019 Optical Society of America

1. Introduction

Holography introduced in the late 1940s by Dennis Gabor [1] has attracted a considerable amount of attention from researchers due to its ability to store and recreate the wavefront emanating from a 3-dimensional object on a 2-dimensional surface [2,3]. The wavefront is encoded as an interference pattern formed by the interaction of an object wavefield and a reference wavefield and is typically represented as a scalar complex-valued wavefield which contains both amplitude and phase information. Since holograms store the actual wavefront, they record/recreate physical reality and can account for visual cues like full parallax, stereopsis etc. while naturally resolving the accommodation-vergence conflict [4]. However, they present significant computational challenges due to the extremely high resolutions required. This requirement is imposed due to a fundamental property of holography which relates the angular field of view ($\theta_{\text{FOV}}$) of the hologram to its pixel pitch $p$ and wavelength of light $\lambda$ used to record the hologram, which is given by the grating equation [5]

$$\sin \left( \frac{\theta_{\text{FOV}}}{2} \right) = \frac{\lambda}{2p}. \quad (1)$$

For example, a monochromatic hologram for red light (633 nm) having a resolution 64K $\times$ 64K and recorded with a pixel pitch of 1 $\mu$m offers a viewing angle of 36° and a viewing aperture of size 6.4 cm $\times$ 6.4 cm. Increasing the pixel pitch would increase the aperture size but will reduce the field of view. A dynamic video sequence involving this hologram quantized at 8 bits per pixel (bpp) for the real and imaginary components while being recorded at a frame rate of 30 frames per second would require a data rate of around 2 Terabit/sec without compression. Thus for practical storage/transmission requirements, data compression is essential for holographic video.

Traditional image and video codecs are not suited for holographic data due to fundamental differences between the signals found in holographic data and natural imagery [6]. Image codecs
like JPEG utilize the fact that most of the energy in natural imagery is present in lower frequencies. The compression strategy deployed by these codecs essentially discard high frequency data and store only the information in the lower frequencies. Holographic data, on the other hand, can present with information across the entire frequency spectrum which can be understood as a consequence of the angular spectrum interpretation of lightwave propagation [2]. Each component in the Fourier decomposition of the hologram will correspond to the complex amplitude of light received at some particular angle on the hologram plane.

Typical video codecs utilize the temporal redundancy that exists between different frames of a video by performing an operation called motion compensation. At a high level, motion compensation tracks how objects are transitioning from one frame to the other such that only relevant information is stored. Since objects are completely localized in the spatial domain for conventional videos, the approach taken in these codecs is to divide the frame into blocks in the spatial domain and see how blocks in one frame can be mapped to other blocks in another frame at a different point in time. However, in holography (almost) every pixel contains information about an object in the recorded scene and as a consequence, the motion compensation algorithm can only account for in-plane motion of an object [7].

Therefore, to obtain a significant reduction in data rate, solutions designed specifically for holograms are required. Several techniques for compressing holograms have been proposed for static scenes. In [8], they utilize the regularity in fringe patterns that are characteristic to holography for compression purposes. Initially, the hologram is decomposed using a wavelet transform into different frequency subbands. The decomposed subbands are further analyzed using a bandelet transform that can determine the geometric flow inherent in the fringes. Another transform-based method proposed in [9] is to use Fresnelets which are obtained by applying the Fresnel transform on B-spline wavelets, in conjunction with a set portioning in hierarchical trees (SPIHT) coding, to obtain significant improvements. However, the usefulness of Fresnelets depends on scene depth [10] and do not guarantee good frequency localization [11]. In [12], a comparative study of scalar quantization and vector quantization methods for compressing static holograms was undertaken where they show modest compression gains can be achieved by using these techniques.

For dynamic scenes, a motion compensation algorithm was introduced in [7] designed for compressing holographic video sequences of a single object where the 3D rigid body motion of the object is known - like in the case of computer-generated holography (CGH). The motion compensation algorithm is based on scalar diffraction theory, wherein small rotations are represented using a paraxial approximation which enables modelling of the motion using a superset of Linear Canonical Transforms called the Affine Canonical Transform. Moving Picture Experts Group (MPEG)-4 part II was used in [13] to compress holographic sequences, where they observed a marginal performance improvement when using inter mode that uses the temporal redundancy between the frames compared to intra mode that utilizes only spatial redundancy. Similarly, [14] compared the compression performance between High Efficiency Video Coding (HEVC/H.265) and Advanced Video Coding (AVC/H.264) video codecs for compressing phase shifted holographic sequences and observed HEVC obtained better compressibility when compared to AVC. However, they observed the inter mode offered no improvement over intra mode which suggests the motion compensation performed by these codecs is not able to utilize temporal redundancy present in holographic videos. There also exist several methods for fast generation of computer generated holographic video sequences based on incremental changes between the different frames. In [15], objects in the scene are divided into sub-objects, and the individual contributions of these sub-objects are calculated with the assumption that only a fraction of these sub-objects undergoes an update between frames. Hence, recalculating the hologram from scratch for each frame can be avoided when generating holographic videos. While high compression ratios can be obtained, real time generation of holographic videos would still
require significant computational power at the decoder.

In this work, we will introduce an end-to-end compression pipeline designed specifically for compressing holographic video sequences where the ground truth motion is known, with the following contributions:

- A motion compensation algorithm is introduced which is derived analytically using the rotational transformation of the angular spectrum method [16] that rigorously satisfies the Helmholtz equation.
- The residuals, remaining after applying the motion compensation algorithm, are compressed with an STFT (Short-Time Fourier Transform) based approach that takes into account the relationship of these residuals in the space-frequency domain.

The article is structured as follows: Section 2 provides a brief overview of our compression pipeline. Section 3 describes the motion compensation algorithm in detail while Section 4 discusses the origin as well as the methodology used to compress the residuals that arise from the algorithm. Section 5 has a detailed study on the results obtained by our solution under different conditions and we conclude in Section 6 along with possible suggestions on the future work.

2. Overview of the compression pipeline

Our solution is designed for lossy compression of holographic sequences where the ground truth motion in the scene is known. Additionally, we consider the case of arbitrary 3D rigid body motion of a single object. Section 6 will elaborate on the direction of research required for extending this method to a truly arbitrary scene with multiple objects, each performing independent rigid body motion.

![Block diagram of the proposed holographic video coder.](image)

(a) The encoding procedure initially uses the previous decoded frame to predict the current frame using motion compensation. The residual between the predicted frame and the ground truth frame is compressed.

(b) The decoding procedure performs the motion compensation step and decodes the compressed residual to obtain the decoded current frame.

The flow of data through the encoder is shown in Fig. 1(a) and the encoder will perform the following steps for compressing a frame. The first frame is assumed to be available at the decoder and the compression will begin from the second frame.

1. The previous lossy encoded frame available at the decoder known as the reference frame is used along with the motion information in the scene by the motion compensation algorithm to predict the current frame.
2. The residual between the ground truth current frame and the motion compensated current frame is obtained in the spatial domain.

3. A representation of the residual in space-frequency domain termed space-frequency blocks (SFB) is obtained by using the STFT.

4. The SFB’s are quantized using a uniform mid-tread quantizer that can dynamically allocate the number of bits assigned to a space-frequency block based on a Lagrangian rate-distortion optimization criterion.

5. The output of the quantizer along with required side information is stored as our compressed bitstream.

The decoder is shown in Fig. 1(b) and will perform the following steps to decode the frame given the encoded frame, the previous decoded frame and the motion information of the scene.

1. The stored bitstream (encoded frame) is parsed and dequantized such that the lossy SFB representation of the residual is obtained.

2. The space-frequency representation is reversed such that the lossy residual is now obtained in the spatial domain.

3. The motion compensation algorithm is applied to the reference frame and is equivalent to the step performed by the encoder.

4. The decoded frame is obtained as the sum of the residual and the motion compensated hologram.

3. Motion compensation

This section describes the theoretical formulation for the motion compensation used by the holographic video codec.

3.1. Preliminaries

The origin of the cartesian coordinate system described by the axes $X, Y, Z$ is defined to be the center of the hologram, where $Z$-axis lies perpendicular to the hologram plane. The monochromatic field at position $(x, y, 0)$ lying on the hologram plane formed by light waves with wavelength $\lambda$ is denoted by $g(x, y; 0)$, while its two-dimensional Fourier Transform $\mathcal{F}\{g(x, y; 0)\} = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} g(x, y; 0)e^{-j2\pi(ux+vy)} \, dx \, dy$ is denoted by $G(u, v; 0)$. Free space propagation between the hologram plane and another parallel plane displaced by a distance $z$ along the direction of propagation can be modelled by the angular spectrum method [2] as

$$g(x, y; z) = \mathcal{F}^{-1}\{G(u, v; 0)H(u, v; z)\}.$$  \hspace{1cm} (2)

The transfer function for free space propagation $H(u, v; z)$ is given by

$$H(u, v; z) = \begin{cases} e^{i2\pi z\sqrt{u^2 + v^2 - \lambda^2}} & \text{if } u^2 + v^2 \leq \lambda^2 \\ 0 & \text{otherwise} \end{cases}.$$  \hspace{1cm} (3)

A rotation can be defined as a circular movement around an arbitrary axis. Any rotation around the origin can be decomposed into a series of counter-clockwise rotations of some $\theta_x, \theta_y$ and $\theta_z$ radians about the $X, Y$ and $Z$ axis performed in the specified order. The coordinate transformation of a point located at a displacement vector of $p = [x, y, z]^T$ from the origin undergoing these
rotations will have its transformed displacement vector $p'$ described by the rotation matrix $R(\theta_x, \theta_y, \theta_z)$ as

$$p' = R(\theta_x, \theta_y, \theta_z)p. \quad (4)$$

where the rotation matrix is given by

$$R(\theta_x, \theta_y, \theta_z) = \begin{bmatrix}
\cos(\theta_x) \cos(\theta_y) & \cos(\theta_x) \sin(\theta_y) \sin(\theta_z) - \sin(\theta_x) \cos(\theta_z) & \cos(\theta_x) \sin(\theta_y) \cos(\theta_z) + \sin(\theta_x) \sin(\theta_z) \\
\sin(\theta_x) \cos(\theta_y) & \sin(\theta_x) \sin(\theta_y) \sin(\theta_z) + \cos(\theta_x) \cos(\theta_z) & \sin(\theta_x) \sin(\theta_y) \cos(\theta_z) - \cos(\theta_x) \sin(\theta_z) \\
-\sin(\theta_y) & \cos(\theta_y) & \cos(\theta_z)
\end{bmatrix}. \quad (5)$$

### 3.2. Motion Compensation using the rotational transformation of angular spectrum

The core idea behind motion compensation is to apply the motion on the hologram plane instead of the object such that the same relative motion between the hologram and object occur. Rigid body motion comprises of rotations, translations or both. Arbitrary rigid body motion of a body can be described as a series of the following individual motions:

1. Rotations of $\theta_x, \theta_y$ and $\theta_z$ radians performed in the specified order about three orthogonal axes parallel to the $X, Y$ and $Z$ axes respectively by an object as shown by the cube in Fig. 2. The parallel axes intersect at the pivot point $(x_o, y_o, z_o)$, where the origin of the system is defined to be the center of the hologram shown by the plane in Fig. 2.

2. Translations of $x_t, y_t$ and $z_t$ along $X, Y$ and $Z$ axis.

![Fig. 2. Rotation of an object (cube) around a pivot point $P(x_o, y_o, z_o)$ for $\theta_x, \theta_y$ and $\theta_z$ radians about three orthogonal axes. These orthogonal axes are parallel to the $X, Y$ and $Z$ axis where the origin $O(0, 0, 0)$ is the center of the hologram, which is indicated by the plane.](image)

It is important to note that rotations are not commutative in general, i.e. the order of operations is important. On the other hand, translations are commutative. To predict the effect of arbitrary rigid body motion on the hologram, we will compensate for rotational motion first and then for translational motion.

To compensate for the rotational motion, the reference hologram $g_{\text{ref}}(*, *, 0)$ is translated to a virtual hologram parallel to the reference hologram and having the pivot point $(x_o, y_o, z_o)$ as its origin. The virtual hologram $g_1(*, *, 0)$ shown in Fig. 3(a) described in the new coordinate system will be

$$g_1(x, y; 0) = g_{\text{ref}}(x + x_o, y + y_o; z_o). \quad (6)$$
Applying the space-shift theorem in the frequency domain on Eq. (6) we get
\[ G_1(u, v; 0) = G_{\text{ref}}(u, v; z_o)e^{j2\pi(ux_o+vy_o)}. \] (7)

The angular spectrum method for free-space propagation from Eq. (3) gives
\[ G_{\text{ref}}(u, v; z_o) = G_{\text{ref}}(u, v; 0)H(u, v; z_o). \] (8)

From Eq. (7) and Eq. (8) we have
\[ G_1(u, v; 0) = G_{\text{ref}}(u, v; 0)H(u, v; z_o)e^{j2\pi(ux_o+vy_o)}. \] (9)

Fig. 3. Equivalent relative rotation between object and virtual hologram.

The object and the virtual hologram before and after applying rotation is shown for an example in Fig. 3(a) and 3(b), where the object is rotated around the vertical axis (parallel to \( Y \)-axis and passing through the pivot point). To an observer in the object rotating with an angular velocity \( \Omega_1 \) around the pivot point, an arbitrary point located on the virtual hologram plane at a displacement vector \( r \) from the pivot point will move with an instantaneous velocity \( v_1 \) due to the rotation of the observer’s coordinate system [17] and is given by
\[ v_1 = -\Omega_1 \times r, \] (10)

where \( \times \) is the cross product between vectors. Now, the object is kept stationary while the virtual hologram plane is rotated by an angular velocity \( \Omega_2 \) around the pivot point. The observer will now record a velocity of \( v_2 \) given by
\[ v_2 = \Omega_2 \times r. \] (11)

If \( \Omega_1 = -\Omega_2 \) then \( v_1 = v_2 \) which implies the same relative motion has occurred. For the example of rotation of the object around the vertical axis shown in Fig. 3(b), the same relative rotation can be achieved by keeping the object stationary and rotating the virtual hologram around the vertical axis in the opposite direction and is shown in Fig. 3(c). Therefore, to predict the change in the virtual hologram due to the rotation of the object, we keep the object stationary and apply rotations of \(-\theta_x, -\theta_y, -\theta_z\) radians around orthogonal axes that intersect at the pivot point.
where the motion compensation transfer function will be

\[
R(-\theta_x, -\theta_y, -\theta_z) = \begin{bmatrix}
a_1 & a_4 & a_7 \\
a_2 & a_5 & a_8 \\
a_3 & a_6 & a_9
\end{bmatrix}
\]  

(12)

This rotation can be achieved by using the rotational transformation of the angular spectrum method [16], a method for formulating rotation of the hologram plane described with respect to its origin. The elements of the rotation matrix in Eq. (12) serve as parameters for the auxiliary functions used for computing the rotated hologram. The rotated virtual hologram is given by the product of the resampled input hologram and a Jacobian as

\[
G_2(u, v; 0) = G_1(\alpha(u, v), \beta(u, v); 0)[J(u, v)],
\]

(13)

where the nonlinear resampling functions are

\[
\begin{align*}
\alpha(u, v) &= a_1u + a_2v + a_3w(u, v), \\
\beta(u, v) &= a_4u + a_5v + a_6w(u, v), \\
w(u, v) &= \sqrt{u^2 - v^2},
\end{align*}
\]

(14a)

(14b)

(14c)

and the Jacobian is

\[
J(u, v) = (a_2a_6 - a_3a_5)\frac{u}{w(u, v)} + (a_3a_4 - a_1a_6)\frac{v}{w(u, v)} + (a_1a_5 - a_2a_4).
\]

(15)

The hologram \(G_3(*, *; 0)\) obtained in Eq. (13) is propagated back to the origin of our reference hologram in a similar manner as in Eq. (6) and (9) to obtain

\[
G_3(u, v; 0) = G_2(u, v; 0)H(u, v; -z_o)e^{-j2\pi(\alpha u \cdot + v)\cdot z_o}.
\]

(16)

g_{3}(*, *, 0) represents an intermediate motion compensated hologram accounting for the rotations undertaken by the object and only compensation for translational motion remains. The relationship between the hologram \(g_{3}(*, *, 0)\) and the motion compensated hologram \(g_{mc}(*, *, 0)\) defined with respect to the latter’s local origin is given by

\[
g_{mc}(x, y; 0) = g_3(x - x_o, y - y_o; -z_o).
\]

(17)

This is again similar to the transformation described in Eq. (6) and (9) and the motion compensated hologram in the frequency domain will be

\[
G_{mc}(u, v; 0) = G_3(u, v; 0)H(u, v; -z_o)e^{-j2\pi(\alpha u \cdot + v)\cdot z_o}).
\]

(18)

Eq. (9), (13), (16) and (18) can be combined to obtain an equation of the form

\[
G_{mc}(u, v; 0) = G_{ref}(\alpha(u, v), \beta(u, v)) \cdot H_{mc}(u, v) ,
\]

(19)

where the motion compensation transfer function will be \(H_{mc}(u, v) =

\[
H(u, v; -z_o)H(\alpha(u, v), \beta(u, v); z_o)](u, v)|e^{-j2\pi(\alpha u \cdot + v)\cdot z_o}(|, \beta(u, v) - v).)
\]

(20)
The motion compensated hologram in the spatial domain is obtained by the inverse fourier transform as

\[ g_{mc}(x, y; 0) = \mathcal{F}^{-1}\{G_{mc}(u, v; 0)\}. \]  

(21)

To implement Eq. (19) we use the cubic spline interpolation on the two-dimensional Fast Fourier Transform (2D-FFT) of the reference hologram to obtain the resampled hologram in the frequency domain, followed by a transfer function multiplication. The final motion compensated hologram is obtained using a two-dimensional Inverse Fast Fourier Transform (2D-IFFT).

4. Compression of Residuals

4.1. Residuals - localization in space and frequency

The motion compensation solution is based on the rotational transformation of the angular spectrum method which rigorously satisfies the Helmholtz equation [16]. However, in practice, due to the finite size and discrete nature of digital holograms, there will exist a residual signal between the ground truth and the motion compensated hologram.

(a) Rotation about axis parallel to X-axis.
(b) Rotation about axis parallel to Y-axis.
(c) Rotation about axis parallel to Z-axis.
(d) Translation along X-axis.
(e) Translation along Y-axis.
(f) Translation along Z-axis (towards hologram plane).

Fig. 4. The amplitude (black indicates absence of signal) of a zero-padded motion compensated hologram in the spatial domain is shown here for different types of rotations and translations. The rectangular border in the center indicates the spatial boundary of the ground truth hologram.

Formally speaking, the mutual information that can be utilised between successive frames is described by the intersection of the space-frequency regions of the motion compensated frame and the current frame to be encoded [7]. More intuitively, this effect can be viewed by spatial projections and nonlinear mappings in the frequency domain.

The information in the reference hologram frame is projected onto virtual hologram planes that do not share the same spatial boundary due to the motion of the object. The effect of the motion compensation for rotations and translations in the spatial domains are shown in Fig. 4. The information obtained from the reference frame that gets projected outside the spatial boundary of the ground truth hologram is discarded while the missing information needs to be filled in.
Besides the missing information, for the rotations around axes parallel to the $X$ or $Y$ axes and translations along the $Z$ axis, we observe a spatially varying error that is minimal at the center of the hologram and increases as we approach the spatial boundary of the motion compensated hologram.

The nonlinear sampling of the motion compensated hologram obtained in Eq. (19) is a frequency domain operation where the frequency coordinates of the motion compensated hologram are obtained from the reference hologram via the mapping $(u,v) \leftrightarrow (a_1u + a2v + a3w(u,v), a4u + a5v + a6w(u,v))$ in Eq. (14). For fast implementation in a digital system, interpolation of the 2D-FFT is used to obtain the non-linearly sampled hologram. The FFT defined for an equidistant sampled grid introduces errors due to the non-equidistant sampled grid used here [16], where the warping of the frequency grid is dependent on the frequency coordinates. Furthermore, the accuracy of the FFT decreases when the spectrum is sampled far from the zero frequency [16]. As a consequence, we can expect some frequency neighborhoods to be more affected than others.

The residuals originate from two sources - one which is localized in space and another in frequency. Any representation of the residuals in either only the spatial domain or only the frequency domain will not be sparse, as a signal localized in space will spread out in frequency and vice versa for the signal localized in frequency. This forms the motivation of our design - to decompose the residual signal in both space and frequency.

### 4.2. Space-frequency blocks

After performing motion compensation for a hologram of size $A \times B$, the encoder will obtain the residual signal $g[\ast, \ast]$ that exists between the ground truth hologram $g_{gt}[\ast, \ast]$ and the motion compensated hologram $g_{mc}[\ast, \ast]$.

$$g[x,y] = g_{gt}[x,y] - g_{mc}[x,y],$$

where $x \in \{1,2,\ldots,A\}$ and $y \in \{1,2,\ldots,B\}$. (22)

The residual hologram $g[\ast, \ast]$ is divided in the spatial domain into uniformly sized sub-holograms $g_{s}^{k,l}[\ast, \ast]$ of size $N_x \times N_y$.

$$g_{s}^{k,l}[x,y] = g[(k-1)N_x + x, (l-1)N_y + y],$$

where $x \in \{1,2,\ldots,N_x\}$, $y \in \{1,2,\ldots,N_y\}$, $k \in \{1,2,\ldots, A/N_x\}$ and $l \in \{1,2,\ldots, B/N_y\}$. (23)

The orthonormal 2D-DFT of these spatial blocks is obtained as

$$G_{s}^{k,l}[u,v] = \frac{1}{\sqrt{N_xN_y}} \sum_{x=1}^{N_x} \sum_{y=1}^{N_y} g_{s}^{k,l}[x,y]e^{-2\pi j((u-1)x/N_x + (v-1)y/N_y)},$$

where $u \in \{1,2,\ldots,N_x\}$ and $v \in \{1,2,\ldots,N_y\}$. (24)

Fig. 5. Illustration of the procedure described by Eqs. (22-25) to obtain space-frequency blocks by using the STFT.
The obtained DFTs are further sub-divided into blocks of size $N_u \times N_v$ - termed space-frequency blocks (SFB).

$$G_{sf}^{k,l,m,n}[u,v] = G_s^k([m-1]N_u + u, (n-1)N_v + v),$$

where $u \in \{1,2,...,N_u\}$, $v \in \{1,2,...,N_v\}$, $m \in \{1,2,...,\frac{N_u}{N_s}\}$ and $n \in \{1,2,...,\frac{N_v}{N_c}\}$. \hfill (25)

The chain of operations done here to obtain the SFB’s are shown in Fig. 5. The SFB represents the fundamental unit of information in our compression system. Each SFB $G_{sf}^{k,l,m,n}[*,*]$ is quantized separately using an uniform mid-tread quantizer to produce a quantized SFB represented by $\hat{G}_{sf}^{k,l,m,n}[*,*]$. The decoder will reverse the operations mentioned above on $\hat{G}_{sf}^{k,l,m,n}[*,*]$ to obtain $\hat{g}[*,*]$. The decoder will perform the motion compensation (equivalent to the step at the encoder) to obtain the compressed video frame $\hat{g}_{ct}[*,*]$ as

$$\hat{g}_{ct}[x,y] = \hat{g}[x,y] + g_{mc}[x,y],$$

where $x \in \{1,2,...,A\}$ and $y \in \{1,2,...,B\}$. \hfill (26)

Since the DFT is an orthonormal transform, the distortion measured as the squared error between the ground truth hologram and the compressed hologram will be

$$\sum_{x=1}^{A} \sum_{y=1}^{B} |\hat{g}_{ct}[x,y] - \hat{g}_{ct}[x,y]|^2 = \sum_{k=1}^{A} \sum_{l=1}^{B} \sum_{m=1}^{N_u} \sum_{n=1}^{N_v} D^{k,l,m,n},$$

where $D^{k,l,m,n} = \sum_{u=1}^{N_u} \sum_{v=1}^{N_v} |G_{sf}^{k,l,m,n}[u,v] - \hat{G}_{sf}^{k,l,m,n}[u,v]|^2$. \hfill (27)

Eq. (27) implies the total squared error between the ground truth hologram and the compressed hologram will be the sum of the squared error between their SFB’s. Since the residuals are localized in space and frequency we can expect the residual energy will also be also localized in some space-frequency neighborhood, which can be utilized for compression. Here, we have used an STFT with rectangular windows to achieve this localization. The number of divisions in the spatial domain $\left(\frac{A}{N_u}, \frac{B}{N_v}\right)$ and the number of divisions in the frequency domain $\left(\frac{N_u}{N_s}, \frac{N_v}{N_c}\right)$ will represent the degree of localization in space and frequency, respectively. For the same hologram of size $A \times B$, increasing the number of spatial divisions/spatial localization will reduce the number of frequency divisions/frequency localization. Intuitively, it can be seen that for a given target rate, we must allocate more bits to encode those residual SFB’s whose energy is high. The next sections will describe the type of quantization and the bit allocation method used by our holographic video coder.

4.3. Quantization and bit allocation

The real and imaginary coefficients of residual SFB’s obtained in Eq. (25) for the purpose of compression are quantized using a uniform scalar mid-rise quantizer, with the same quantizer used for all coefficients within a residual SFB. The uniform scalar mid-rise quantizer can be defined by its quantization range $[X_{\min}, X_{\max}]$ and the number of quantization outputs/levels $L$, which determines its step size $\Delta = \frac{X_{\max} - X_{\min}}{L}$. In our case, since the residual SFB’s are zero mean...
we use $X_{\text{max}} = -X_{\text{min}}$ and the output of the quantizer will be given by Eq. (28).

$$Q(x, L, X_{\text{max}}) = \begin{cases} 0 & \text{if } L = 1 \\ \left(\frac{x}{L} + 0.5\right) \frac{2X_{\text{max}}}{L} & \text{else if } x < -X_{\text{max}} \\ \left(\left\lfloor \frac{xL}{2X_{\text{max}}} \right\rfloor + 0.5\right) \frac{2X_{\text{max}}}{L} & \text{else if } -X_{\text{max}} \leq x \leq X_{\text{max}} \\ \left(\frac{L}{2} - 0.5\right) \frac{2X_{\text{max}}}{L} & \text{otherwise} \end{cases}.$$  (28)

The input-output relationship of a uniform scalar mid-rise quantizer with $L = 8$ levels and $X_{\text{max}} = -X_{\text{min}} = 4$ is shown in Fig. 6(a). The quantizer used for some SFB block with space-frequency indices $k, l, m, n$ is defined by

1. The number of levels used by the quantizer will be given by $2^{b_{k, l, m, n}}$, where $b_{k, l, m, n}$ is the number of bits assigned to the quantizer. In our implementation $b_{k, l, m, n}$ is an integer and can vary from 0 (which means no quantization and storage) to 8.

2. The quantization range $X_{\text{max}}^{k, l, m, n}$.

The task that remains is to determine for a given target rate $R_{\text{req}}$, the optimal values of $b_{k, l, m, n}$ and $X_{\text{max}}^{k, l, m, n}$ such that the overall mean squared distortion between the ground truth hologram and the compressed hologram is minimized.

The value of $b_{k, l, m, n}$ is determined using a Lagrangian rate-distortion optimization criterion which determines how many bits to assign to the SFB. For this procedure, we require the distortion produced by each SFB at different bit depths. We determine for each SFB the following tables, the optimal quantization range $X_{\text{max}}^{k, l, m, n}(b)$ and its corresponding distortion $D_{\text{lab}}^{k, l, m, n}(b)$ at all possible bit depths $b \in \{0, 1, ..., 8\}$.

The choice of the quantization range $X_{\text{max}}^{k, l, m, n}(b)$ for some bit depth $b$ depends on two conflicting errors, namely the overload error and the granular error. Having a high quantization range would ensure the overload error due to the input being outside the quantization range is minimized but will increase the granular error that arises from the increase in step-size [18]. The effect of quantization range on the total squared error measured in Signal to Noise Ratio (SNR) produced by the mid-rise quantizer for a typical SFB is shown in Fig. 6(b). As expected, the SNR as a function of quantization range is unimodal. Furthermore, it can be observed that the value of the optimal quantization range increases as the bit depth increases. This phenomenon is explained by
the decrease in step size due to the higher bit depth which reduces the impact of the granular error, while the impact of overload error remains the same.

The determination of the quantization range is done by a numerical optimization search algorithm called the golden-section search which is a technique for determining the extremum of a unimodal function. The method initializes with an initial search range containing the extremum and narrows the search range iteratively. Here, we perform the search by considering the squared error produced by the quantizer as a function of the quantization range. The initial search range for the quantization range will be between 0 and the largest coefficient (real/imaginary) in the SFB and the algorithm implemented is shown in Algorithm 1.

Algorithm 1 Golden-section search for determining $X_{\text{maxtab}}^{k,l,m,n}(b)$ and $D_{\text{tab}}^{k,l,m,n}(b)$

1: procedure DETERMINEQUANTIZATIONRANGE($G_{\text{sf}}^{k,l,m,n}[u,v], b$)
2: \[ \varphi = \frac{1 + \sqrt{5}}{2} \]
3: \[ c = 0 \]
4: for all $u \in \{1, 2, \ldots, N_u\}, v \in \{1, 2, \ldots, N_v\}$ do
5: \[ d_{\text{real}} = \max_{u,v} \left| \text{Re}(G_{\text{sf}}^{k,l,m,n}[u,v]) \right| \]
6: \[ d_{\text{imag}} = \max_{u,v} \left| \text{Im}(G_{\text{sf}}^{k,l,m,n}[u,v]) \right| \]
7: end for
8: \[ d = \max(d_{\text{real}}, d_{\text{imag}}) \quad \triangleright \text{Largest absolute valued coefficient in SFB} \]
9: \[ e = d - \frac{c - d}{\varphi} \]
10: \[ f = c + \frac{c - d}{\varphi} \]
11: for Number of iterations of golden section search do
12: for all $u \in \{1, 2, \ldots, N_u\}, v \in \{1, 2, \ldots, N_v\}$ do
13: \[ \hat{G}_{\text{e}}[u,v] = Q(\text{Re}(G_{\text{sf}}^{k,l,m,n}[u,v]), 2^b, e) + jQ(\text{Im}(G_{\text{sf}}^{k,l,m,n}[u,v]), 2^b, e) \]
14: \[ \hat{G}_{\text{f}}[u,v] = Q(\text{Re}(G_{\text{sf}}^{k,l,m,n}[u,v]), 2^b, f) + jQ(\text{Im}(G_{\text{sf}}^{k,l,m,n}[u,v]), 2^b, f) \]
15: end for
16: \[ D_{\text{e}} = \sum_{u=1}^{N_u} \sum_{v=1}^{N_v} |G_{\text{sf}}^{k,l,m,n}[u,v] - \hat{G}_{\text{e}}[u,v]|^2 \]
17: \[ D_{\text{f}} = \sum_{u=1}^{N_u} \sum_{v=1}^{N_v} |G_{\text{sf}}^{k,l,m,n}[u,v] - \hat{G}_{\text{f}}[u,v]|^2 \]
18: if $D_{\text{e}} < D_{\text{f}}$ then
19: \[ d = f \]
20: else
21: \[ c = e \]
22: end if
23: \[ e = d - \frac{c - d}{\varphi} \]
24: \[ f = c + \frac{c - d}{\varphi} \]
25: end for
26: \[ X_{\text{maxtab}}^{k,l,m,n}(b) = \frac{c + d}{2} \quad \triangleright \text{Minimum estimated as mid-point of last search range} \]
27: for all $u \in \{1, 2, \ldots, N_u\}, v \in \{1, 2, \ldots, N_v\}$ do
28: \[ \hat{G}_{\text{exx}}[u,v] = Q(\text{Re}(G_{\text{sf}}^{k,l,m,n}[u,v]), 2^b, c + d) + jQ(\text{Im}(G_{\text{sf}}^{k,l,m,n}[u,v]), 2^b, c + d) \]
29: end for
30: \[ D_{\text{tab}}^{k,l,m,n}(b) = \sum_{u=1}^{N_u} \sum_{v=1}^{N_v} |G_{\text{sf}}^{k,l,m,n}[u,v] - \hat{G}_{\text{exx}}[u,v]|^2 \]
31: return $X_{\text{maxtab}}^{k,l,m,n}(b), D_{\text{tab}}^{k,l,m,n}(b)$
32: end procedure

The final value of $X_{\text{maxtab}}^{k,l,m,n}$ can be obtained in two refinements. The first refinement calculates $X_{\text{maxtab}}^{k,l,m,n}(b)$ for all blocks and possible quantization levels by using a small number of iterations of
the golden section search algorithm for the bit allocation. Once the blocks and levels to be stored (typically a small fraction of the total SFB’s) are identified, this value can be refined for only these blocks at its respective quantization level. However, we observe that a near-optimal value of $X_{\text{max}}^{k,l,m,n}$ is obtained in less than 15 iterations of the golden section search algorithm, which implies the second refinement pass can be avoided to save computation with a very minimal hit to the distortion. In this case $X_{\text{max}}^{k,l,m,n} = X_{\text{max}}^{k,l,m,n}(b_{\text{sol}}^{k,l,m,n})$.

The Lagrangian multiplier based rate-distortion optimization method used to determine the bit depth $b_{\text{sol}}^{k,l,m,n}$ from [19] can be applied when the quantizer bit allocation is chosen from a finite set. The allocation is done based on the actual distortion produced by the quantizer for different bitdepths and does not depend on the output of the quantizer to satisfy certain statistical properties, as is the case for other methods [20].

Let $B_{\text{sol}}$ be a bit allocation vector that contains the bit allocation $b_{\text{sol}}^{k,l,m,n}$ for all $k, l, m, n$ blocks corresponding to some solution satisfying $0 \leq b_{\text{sol}}^{k,l,m,n} \leq 8$ and let $\mathcal{B}$ be the set that contains all possible $B_{\text{sol}}$. The squared error produced by the bit allocation vector is denoted as $D(B_{\text{sol}})$ and the rate by $R(B_{\text{sol}})$.

From Eq. (27), the squared error of the hologram will be

$$D(B_{\text{sol}}) = \sum_{k=1}^{A} \sum_{l=1}^{B} \sum_{m=1}^{N_x} \sum_{n=1}^{N_y} D_{\text{tab}}^{k,l,m,n}(b_{\text{sol}}^{k,l,m,n}).$$

The rate of a SFB as a function of the bit depth assigned to it will be

$$R(b_{\text{sol}}^{k,l,m,n}) = \begin{cases} 0 & \text{if } b_{\text{sol}}^{k,l,m,n} = 0 \\ \left\lceil \log_2\left(\frac{AB}{N_uN_v}\right) \right\rceil + 32 + 2b_{\text{sol}}^{k,l,m,n}N_uN_v & \text{if } 1 \leq b_{\text{sol}}^{k,l,m,n} \leq 8 \end{cases}.$$ 

In the case of zero bit assignment, we will not store any information. For other cases $2b_{\text{sol}}^{k,l,m,n}N_uN_v$ bits are required to store the levels chosen by the quantizer for the real and imaginary coefficients of the SFB. The side information refers to additional information required at the decoder and comes from three sources.

1. $\left\lceil \log_2\left(\frac{AB}{N_uN_v}\right) \right\rceil$ bits are required to indicate the position of the SFB as the total number of SFB’s are $\frac{AB}{N_uN_v}$.
2. The quantization range used stored as a 32 bit single precision floating point number.
3. 3 bits required to indicate the bit depth used so that the decoder is able to parse the bitstream.

The final bitstream is obtained by concatenating the individual bitstreams of those SFB’s that are
chosen for storage as shown in Fig. 7. The total rate will be given by

$$R(B_{sol}) = \sum_{k=1}^{\Lambda} \sum_{l=1}^{N_y} \sum_{m=1}^{N_y} \sum_{n=1}^{N_n} R(b_{sol}^{k,l,m,n}).$$  

(31)

The rate-distortion optimization procedure for the Lagrangian multiplier method is expressed by

$$\min_{B_{sol} \in S} (D(B_{sol}) + \Lambda R(B_{sol})) \text{ for } 0 \leq \Lambda < \infty.$$  

(32)

The bit allocation vector $B_{sol}(\Lambda)$ obtained as the solution for some $\Lambda$ to the above minimization problem is termed the unconstrained solution. $B_{sol}(\Lambda)$ is obtained by individually solving [19]

$$\min_{b_{sol} \in \{0,1,\ldots,8\}} (D_{tab}(b_{sol}^{k,l,m,n}) + \Lambda R(b_{sol}^{k,l,m,n})).$$  

(33)

In case there are multiple solutions to $B_{sol}(\Lambda)$, then the value of $\Lambda$ is called singular. If the value of $\Lambda$ is such that the rate of the unconstrained solution satisfies $R(B_{sol}(\Lambda)) = R_{req}$, then the required bit-assignment is $b_{sol}^{k,l,m,n}(\Lambda)$. It can be seen that increasing the value of $\Lambda$ will reduce the rate of the unconstrained solution $R(B_{sol}(\Lambda))$. However, it is not possible to directly determine which value of $\Lambda$ corresponds to the required rate constraint. On the other hand, it is not necessary to sweep all values of $\Lambda$ from 0 to $\infty$ to obtain the required value of $\Lambda$. In [19], it is shown rigorously that all solutions of $\Lambda$ located in between two singular values are the same. Therefore, it is only necessary to check $\Lambda$ at the singular values (finite in number) to determine if the required rate solution has been obtained.

Algorithm 2 from [19] determines an initial guess for $\Lambda$ and solves for the unconstrained solution. The search direction for $\Lambda$ is determined by the unconstrained solution obtained. If the rate for the unconstrained rate solution is greater than the required rate, the singular value of $\Lambda$, closest from above to the estimated value, is used and the process is repeated until the required rate is obtained. On the other hand, if the rate for the unconstrained rate solution/solutions is smaller than the required rate, the singular value of $\Lambda$, closest to the estimated value from below, is used.

5. Results

5.1. CGH method, hologram and codec parameters

<table>
<thead>
<tr>
<th>Table 1. Hologram and Codec Parameters</th>
</tr>
</thead>
<tbody>
<tr>
<td>Number of frames</td>
</tr>
<tr>
<td>Resolution</td>
</tr>
<tr>
<td>Pixel Pitch</td>
</tr>
<tr>
<td>Aperture Size</td>
</tr>
<tr>
<td>Field of view</td>
</tr>
<tr>
<td>Size of spatial block $(N_x \times N_y)$</td>
</tr>
<tr>
<td>Size of frequency block $(N_u \times N_v)$</td>
</tr>
</tbody>
</table>

The proposed holographic video compression technique is evaluated by using CGH generated from point clouds employing the rigorous spherical wave propagation equation as given below.
Algorithm 2 Lagrangian rate-distortion optimization for determining bit depth allocation

1: procedure DetermineBitAllocation(D_{tab}^r(\ast), R_{req})
2: \[ \Lambda = \frac{1}{A^B} \sum_{k=1}^{A} \sum_{l=1}^{B} \sum_{m=1}^{C} \sum_{n=1}^{D} D_{tab}^{k,l,m,n}(4) - D_{tab}^{k,l,m,n}(5) \] \quad \triangleright \text{Initial guess for } \Lambda
3: if \( B_{sol}(\Lambda) \) is non-singular then
4: \( \Lambda_{\text{cur}} = B_{sol}(\Lambda) \)
5: else
6: \( \Lambda_{\text{cur}} = \min_{|R(B_{\text{cur}}(\Lambda)) - R_{req}|} B_{sol}(\Lambda) \)
7: end if
8: if \( R(\Lambda_{\text{cur}}) < R_{req} \) then
9: while \( R(\Lambda_{\text{cur}}) < R_{req} \) do \quad \triangleright \text{Find next singular value of } \Lambda \text{ from below}
10: \quad \text{for all } k \in \{1, 2, ..., N_k\}, l \in \{1, 2, ..., N_l\}, m \in \{1, 2, ..., N_m\}, n \in \{1, 2, ..., N_n\} \text{ do}
11: \quad \quad \text{for all } s^{k,l,m,n} \in \{b_{cur}^{k,l,m,n} + 1, r_{cur}^{k,l,m,n} + 2, ..., 8\} \text{ do}
12: \quad \quad \quad \text{where } s^{k,l,m,n} \neq 0 \text{ and } b \in s^{k,l,m,n}
13: \quad \quad \quad \Lambda = \min_{k,l,m,n,b} \frac{D_{tab}^{k,l,m,n}(b^{k,l,m,n}) - D_{tab}^{k,l,m,n}(b)}{R(b) - R(b^{k,l,m,n})}
14: \quad \quad \end{for all s^{k,l,m,n}}
15: \quad \end{for all k \in \{1, 2, ..., N_k\}}
16: \end{while R(\Lambda_{\text{cur}}) < R_{req}}
17: else if \( R(\Lambda_{\text{cur}}) > R_{req} \) then
18: while \( R(\Lambda_{\text{cur}}) > R_{req} \) do \quad \triangleright \text{Find next singular value of } \Lambda \text{ from above}
19: \quad \text{for all } k \in \{1, 2, ..., N_k\}, l \in \{1, 2, ..., N_l\}, m \in \{1, 2, ..., N_m\}, n \in \{1, 2, ..., N_n\} \text{ do}
20: \quad \quad \text{for all } s^{k,l,m,n} \notin \{0, 1, ..., b_{\text{cur}}^{k,l,m,n} - 1\} \text{ do}
21: \quad \quad \quad \text{where } s^{k,l,m,n} \neq 0 \text{ and } b \in s^{k,l,m,n}
22: \quad \quad \quad \Lambda = \min_{k,l,m,n,b} \frac{D_{tab}^{k,l,m,n}(\Lambda_{\text{cur}}) - D_{tab}^{k,l,m,n}(b)}{R(b) - R(b_{\text{cur}}^{k,l,m,n})}
23: \quad \quad \end{for all s^{k,l,m,n}}
24: \quad \end{for all k \in \{1, 2, ..., N_k\}}
25: \end{while R(\Lambda_{\text{cur}}) > R_{req}}
26: end if
27: end if
28: \( b_{\text{cur}}^{k,l,m,n} = b' \)
29: return \( B = \Lambda_{\text{cur}} \)
30: end procedure
for a hologram of size $A \times B$.

$$g[x, y] = \sum_{j \in P} \frac{a_j}{r_j} e^{i2\pi r_j / \lambda},$$

where $x \in \{1, 2...A\}$ and $y \in \{1, 2...B\}$,

and $r_j = \sqrt{(x - x_j)^2 + (y - y_j)^2 + (z_h - z_j)^2}$ is the distance between the hologram pixel $(x, y, z_h)$ and the point $j$ with amplitude $a_j$ present in the point cloud set $P$, calculated for the wavelength $\lambda$ for the coherent light source and the hologram at depth $z_h$. The distortion measure used for objective evaluation is the Signal to Noise Ratio (SNR) calculated on the numerically reconstructed object plane hologram located at the centroid of the object. $g_{\text{com}}[x, y]$ and $g_{\text{gt}}[x, y]$ represents the reconstructed compressed hologram and the ground truth hologram in the spatial domain, respectively. With this the SNR is given as

$$\text{SNR} = 10 \log_{10} \left( \frac{\sum_{x=1}^{A} \sum_{y=1}^{B} |g_{\text{gt}}[x, y]|^2}{\sum_{x=1}^{A} \sum_{y=1}^{B} \left( |g_{\text{gt}}[x, y]| - |g_{\text{com}}[x, y]| \right)^2} \right).$$

The SNR is a mean-squared error based metric, where a high value of SNR implies large deviations from the original signal are negligible. More specifically, the mean squared error and consequently the SNR penalizes large errors heavily compared to small errors. While this property may be undesirable in some applications, mean-squared error based metrics have been used extensively in traditional image analysis. A suitable metric for analyzing holograms is an ongoing research problem [5, 21].

The Bjøntegaard Delta Peak Signal to Noise Ratio (BD-PSNR) [22] is a metric used to compare two codecs across some rate region. The BD-PSNR gain of codec $A$ over codec $B$ for a rate region will be given by the surface area that lies between the rate-SNR curves of the two codecs, where the rate axis is logarithmically scaled. A positive BD-PSNR gain for codec $A$ corresponds to better rate-distortion performance by codec $A$ over codec $B$ for the rate region, while a negative gain corresponds to the opposite assessment.

The hologram parameters for the tests conducted and the sizes of the spatial and frequency block for the proposed holographic video codec are given in Table 1.

5.2. Impact of depth and complexity of object

<table>
<thead>
<tr>
<th>Object</th>
<th>Length × Width</th>
<th>Depth</th>
<th>Number of points</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cat</td>
<td>1.3 mm × 3.8 mm</td>
<td>7.5 mm</td>
<td>27,894</td>
</tr>
<tr>
<td>Venus</td>
<td>2.9 mm × 4.1 mm</td>
<td>2.0 mm</td>
<td>104,537</td>
</tr>
<tr>
<td>Dragon</td>
<td>1.8 mm × 4.0 mm</td>
<td>2.6 mm</td>
<td>1,804,728</td>
</tr>
</tbody>
</table>

The effect of depth and the complexity of the object used is evaluated by testing point clouds of different objects, namely ‘Cat’, ‘Venus’, and ‘Dragon’. The physical dimensions and cardinalities of the point clouds tested are shown in Table 2. ‘Cat’ features the largest depth, while ‘Dragon’ comprises of the largest number of points. The centroids of these objects are kept at a depth of 20 mm from the hologram plane and the objects are rotated around the axis parallel to the Y-axis at 1° per frame and passing through the centroid.

The BD-PSNR gain obtained by the proposed codec over HEVC Inter mode and JPEG 2000 for the rate region from 0.1 to 1.0 bpp is shown in Table 3. We observe that the proposed codec
offers a gain of around 27 dB with respect to HEVC Inter mode and around 29 dB with respect to JPEG 2000. For similar motion, the effect of depth and the complexity of the object is not significant.

![Images of reconstructions](image.png)

Fig. 8. Reconstructed central view in object plane

The reconstruction of the holograms from the different codecs in the object plane is shown in Fig. 8 for a bitrate of 0.1 bpp. The proposed codec is visually identical to the ground truth, while there is a strong presence of background noise in both JPEG 2000 and HEVC Inter mode. JPEG 2000 also features aliasing which further degrades the reconstruction. A close-up reconstruction for the object ‘Cat’ is shown in Fig. 9. The points lying close in depth to the reconstruction plane should appear as fine spots, which is the case for the proposed codec. For HEVC Inter mode and JPEG 2000, we see the point cloud is not resolved to the same detail which implies the depth perception offered by these codecs is poorer than the proposed codec.

### 5.3. Impact of different types of rotations and translations

The centroid of the point cloud object ‘Venus’ mentioned in Table 2 is kept at a depth of 20 mm from the hologram plane and the following motions are tested for both the Y-axis and the Z-axis

- Rotation of the object around the axis parallel to the coordinate axis (Y/Z) and passing through the centroid with 0.25°, 1° and 5° per frame,
<table>
<thead>
<tr>
<th>Object</th>
<th>Codec</th>
<th>HEVC Inter</th>
<th>JPEG 2000</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cat</td>
<td>27.06 dB</td>
<td>28.98 dB</td>
<td></td>
</tr>
<tr>
<td>Venus</td>
<td>27.20 dB</td>
<td>28.81 dB</td>
<td></td>
</tr>
<tr>
<td>Dragon</td>
<td>26.93 dB</td>
<td>28.75 dB</td>
<td></td>
</tr>
</tbody>
</table>

- Translation of the object along the coordinate axis (Y/Z) with 8.53 µm, 17.67 µm and 34.13 µm per frame.

We do not test rotation and translation with respect to the X-axis as it is analytically similar to the Y-axis. Since JPEG 2000 is an intra only codec, we observe that the rate-distortion performance of JPEG 2000 is invariant to the type of motion performed. Therefore, to analyze the effect of motion we only include the results for HEVC Inter mode and the proposed video codec.

The rate-distortion performance obtained by the proposed codec and HEVC Inter mode is shown in Fig. 10. For both the proposed codec and HEVC Inter mode the lowest SNR was observed for the rotation of the object around the axis parallel to Y-axis (also applicable to X-axis). We observe that HEVC Inter mode is able to compensate for the in-plane motions which is to be expected [7], and to a lesser degree the translations along Z-axis. The BD-PSNR gain of the proposed codec over HEVC Inter evaluated over the bit-range from 0.1 to 1.0 bpp for the different types of rotations and translations is shown in Table 4.
We observe that the BD-PSNR gain offered by the proposed codec is dependent on the type and magnitude of the applied motion. For rotations about the axis parallel to $Y$-axis, the gain decreases as the magnitude of motion increases. On the other hand, for rotations about the axis parallel to $Z$-axis we observe the opposite phenomenon. For translations along $Y$-axis a constant gain of around 19 dB is obtained, while for translations along the $Z$-axis we obtain slightly higher gains for the lowest and the highest magnitude of motion.

![Fig. 10. Effect of different types of rotations and translations on rate-distortion performance](image)

**Table 4. BD-PSNR gain of the proposed codec over HEVC Inter for different motions (0.1 to 1.0 bpp)**

<table>
<thead>
<tr>
<th>Motion</th>
<th>Gain (dB)</th>
</tr>
</thead>
<tbody>
<tr>
<td>$0.25^\circ/\text{f - Y}$</td>
<td>29.04</td>
</tr>
<tr>
<td>1$^\circ/\text{f - Y}$</td>
<td>27.31</td>
</tr>
<tr>
<td>$5^\circ/\text{f - Y}$</td>
<td>13.83</td>
</tr>
<tr>
<td>$0.25^\circ/\text{f - Z}$</td>
<td>6.15</td>
</tr>
<tr>
<td>1$^\circ/\text{f - Z}$</td>
<td>12.32</td>
</tr>
<tr>
<td>$5^\circ/\text{f - Z}$</td>
<td>24.02</td>
</tr>
<tr>
<td>8.53 $\mu$m/\text{f - Y}$</td>
<td>19.02</td>
</tr>
<tr>
<td>17.67 $\mu$m/\text{f - Y}$</td>
<td>19.92</td>
</tr>
<tr>
<td>34.13 $\mu$m/\text{f - Y}$</td>
<td>18.37</td>
</tr>
<tr>
<td>8.53 $\mu$m/\text{f - Z}$</td>
<td>23.18</td>
</tr>
<tr>
<td>17.67 $\mu$m/\text{f - Z}$</td>
<td>19.59</td>
</tr>
<tr>
<td>34.13 $\mu$m/\text{f - Z}$</td>
<td>22.09</td>
</tr>
</tbody>
</table>

5.4. Gain of motion compensation and residual compression pipeline

To quantify the gain of the motion compensation and the residual compression pipeline, 'Venus' detailed in Table 2 is rotated at an angular speed of 1$^\circ$ per frame around the vertical axis parallel to $Y$-axis and passing through the centroid kept at 20 mm from the hologram plane. The video sequence is compressed using:
1. HEVC Inter mode and JPEG 2000.

2. The motion compensation algorithm is applied and the residuals obtained in Eq. (22) is compressed using HEVC Intra mode and JPEG 2000.

3. The entire pipeline of the proposed video codec is tested.

The rate-distortion curves obtained by these methods are shown in Fig. 11. For the rate region from 0.1 to 1.0 bpp, the motion compensation algorithm improves the BD-PSNR for JPEG 2000 and HEVC by around 16.84 dB and 20.04 dB, respectively. The entire pipeline provides a BD-PSNR gain of 28.81 dB and 27.20 dB over JPEG 2000 and HEVC respectively for the same bitrate range.

![Fig. 11. Rate-Distortion performance for different codecs](image)

6. Conclusion

In this work, we have introduced a motion compensation algorithm and a space-frequency representation for compressing holographic video sequences involving an object with known motion. The motion compensation in traditional video codecs like HEVC can only account for in-plane motion. Instead, our motion compensation algorithm is based on the rotational transformation of the angular spectrum method and can compensate for arbitrary 3D rigid body motion. For experiments, we used computer-generated holography for generating video content. The end-to-end compression pipeline demonstrated improvements of about 20 dB in terms of BD-PSNR for most of the tested scenarios. With compression, it is now possible to transmit a 30 frames per second 4K holographic video at less than 48 Megabits per seconds, making possible the transmission over regular high-speed Internet, without noticeable objective/subjective loss. In contrast, a ten fold larger bandwidth is required when using the HEVC encoder.

We can expect further gains by using an entropy coding scheme using the levels chosen by the quantizer. Ongoing work extends the technique for scenes with multiple objects by employing local motion compensation after a per-object segmentation is applied. Furthermore, for compensating videos with unknown motion, a motion estimation algorithm could be developed for determining the various rotation and translation parameters using solely the input video frames.
Funding
European Union Seventh Framework Programme (FP7/2007-2013)/ERC Grant Agreement n.617779 (INTERFERE).

Acknowledgements
Preliminary results of this work were presented at the Digital Holography and Three-Dimensional Imaging Conference in 2019 held in Bordeaux, under the title 'Exact Compensation of Rotational Motion for Holographic Video Compression'. The point cloud objects 'Cat', 'Venus' and 'Dragon' are courtesy of Tosca hi-resolution 3D database v1.0, Direct Dimensions Inc. (www.dirdim.com) and Stanford Computer Graphics Laboratory respectively. The authors would like to thank Gillian Assi and Margo Audiens for their help with some of the illustrations used in this work. We would also like to express our sincere gratitude to the anonymous reviewers for their valuable feedback.

References