Disparities selection controlled by the compensated image quality for a given bitrate

Imen Kadri, Gabriel Dauphin, Anissa Mokraoui & Zied Lachiri

Signal, Image and Video Processing

ISSN 1863-1703

SIViP
DOI 10.1007/s11760-020-01643-1

Springer

Signal, Image and Video Processing
https://doi.org/10.1007/s11760-020-01643-1

ORIGINAL PAPER

Disparities selection controlled by the compensated image quality for a given bitrate

Imen Kadri[1,2] · Gabriel Dauphin[1] · Anissa Mokraoui[1] · Zied Lachiri[2]

Received: 31 October 2019 / Revised: 20 December 2019 / Accepted: 16 January 2020

Abstract
A stereoscopic image consists of two views rendering a depth sense. Indeed each eye is constrained to look at one view, and the small objects displacements across the two views are interpreted as an indication of depth. These displacements are exploited as specific inter-view redundancies from a compression viewpoint. The classical still compression scheme, called disparity-compensated compression scheme, compresses one view independently of the second view, and a block-based disparity map modeling the displacements is losslessly compressed. The difference between the original view and its disparity predicted view is then compressed and used by the decoder to compute the compensated view to improve the disparity predicted view. However, a proof of concept work has already shown that selecting disparities according to the compensated view, instead of the predicted view, yields increased rate-distortion performance. This paper derives from the JPEG-coder, a disparity-dependent analytic expression of the distortion induced by the compensated view. This expression is embedded into an algorithm with a reasonable numerical complexity approaching the performance obtained with the proof of concept work. The proposed algorithm, called fast disparity-compensated block matching algorithm, provides at the same bitrate an average performance increase as compared to the classical stereoscopic image coding schemes.

Keywords Stereoscopic image · Compression · Disparity compensation · Block matching algorithm · JPEG-distortion

1 Introduction

A stereoscopic image is composed of two views which are perceived as two viewpoints of a single 3D-scene, thanks to a technical device. Applications concern the entertainment industry, video games, medical field and cartography [1]. From an information technology viewpoint, all these displayed contents require a very large amount of data which causes issues with storage, transmission and sometimes real-time displaying. Such data is used in many 3D-research activities [2] to estimate the depth map, generally assuming that objects look the same when seen from different views, which happens to be not so common [3]. Research in compression aims at reducing that amount of data by exploiting redundancies. This paper focuses on the stereoscopic images compression [4–6] where the depth map is *not* by itself an issue and it is needed only to explain the differences between the two views. The horizontal distance between the two similar points is called the *disparity* and is inversely related to the depth. The depth map is sometimes encoded as a disparity map as for lifting schemes where the view synthesis is achieved using a set of predict and update filters in a multi-resolution context. Correlations between depth map texture and motion are exploited in [3]. In [7], the authors used also view synthesis optimization, meaning that the choice of the depth map takes also into account the reconstruction of the other view, while using a different framework, this idea is at the core of our present work. Besides, it should be said that high performance is achieved when different techniques are combined as in Multiview Video Coding (MVC) extension of H264/AVC video

✉ Imen Kadri
imenkadri@gmail.com

Gabriel Dauphin
gabriel.dauphin@univ-paris13.fr

Anissa Mokraoui
anissa.mokraoui@univ-paris13.fr

Zied Lachiri
zied.lachiri@enit.tn.rnu

[1] L2TI, Institut Galilée, Université Paris 13, Villetaneuse, France

[2] LSITI, Ecole Nationale d'Ingénieur de Tunis, Université de Tunis El Manar, Tunis, Tunisia

Published online: 14 February 2020

coding standard [8] which has been subjectively evaluated in [9].

As in [10,11], this paper proposes to work with the original framework, called the disparity-compensated compression scheme (DCC), exploiting the stereoscopic image redundancy. It consists in coding separately a reference view, losslessly encoding an estimated disparity map and then encoding a residual image. The transmitted information enables the decoder to reconstruct the reference view, and using the disparity map to compute a predicted view to which is added the decoded residual image. Note that the DCC scheme shares some similarities with the depth and view synthesis representation in that depth information is here modeled as a block-based disparity map and the texture information is featured by the lossy-encoded residual image. The DCC scheme is very similar to motion/disparity compensation implemented in the HEVC/MVC (extension of the H264/AVC) video coding standards.

Research within this framework has achieved increased performance when estimating the disparity map, by taking into account its own bit-cost in [12,13] and its limited predicting capacity [14], by using blocks of arbitrary shapes in [15], and by addressing also the illumination compensation in [16]. Investigating the statistical properties of the residual, reference [17] uses a DCT-based coder for non-occluded 8×8-blocks and a 3-level Haar-based coder for occluded 8×8-blocks to encode the residual instead of the JPEG-coder [18]. Reducing the numerical complexity is also a significant research issue. Examples include selecting optimal hyper parameter values thanks to allocation modeling in [19] as opposed to an exhaustive search in [20] reducing the search area in [21] and using embedded coding scheme that can be truncated at any point to obtain the best reconstruction for a given bitrate [17].

At the core of our work is the idea that the estimation of the disparity should take into account the ability of the residual coder to refine the predicted view, instead of assuming that the best predicted view yields the best compensated view. In the context of the JPEG-residual encoder, a proof of concept using a very greedy algorithm has already shown increased performance in [22]. Our contribution is the design of an algorithm with a reasonable numerical complexity, able to select the disparity according to the compensated predicted view in order to improve the rate-distortion performance of the compressed stereoscopic image.

This paper is organized as follows. Section 2 summarizes the basic concepts of the classical DCC scheme. Section 3 shows how finding the best performing disparity map. Section 4 reviews the greedy disparity-compensated block matching (DCBM) algorithm to solve the optimization problem. Section 5 proposes a fast extension of the DCBM algorithm. Section 6 discusses the simulation results. Section 7 concludes the paper.

2 Basic concepts and notations

This paper deals with rectified stereoscopic images using the classical DCC scheme. Notations, used in the following sections, are summarized in Fig. 1 presenting the DCC scheme where the dashed line separates the encoder (above) from the decoder (below).

In Fig. 1, $\mathbf{I}_l$ (upper left corner) denotes the left view chosen here as the reference view. It feeds a lossy encoder denoted C_{q_l} (upper left corner) where $q_l \in \mathbf{Q}_l$ is its quality factor and $\mathbf{Q}_l$ is a set containing all allowed values. The bit stream output is transmitted to the decoder (left downward arrow connecting the dashed line). This bit stream is decoded by D_l yielding a reconstructed left view denoted $\widehat{\mathbf{I}}_l$ (lower left corner) as follows:

$$\widehat{\mathbf{I}}_l = D_l(C_{q_l}(\mathbf{I}_l)). \tag{1}$$

Note that the framework chosen uses a close loop as this bit stream yields also $\widehat{\mathbf{I}}_l$ in the encoder through D_l (center upper part). $\widehat{\mathbf{I}}_l$ feeds the remaining compressing part. Such a choice reduces the distortion as $\mathbf{I}_l$ is not available to the decompressing part, but it also increases the numerical complexity as the remaining compressing part depends on the choice of q_l. $\mathbf{I}_r$ (center of the upper part) represents the original right view. With $\widehat{\mathbf{I}}_l$, it is used by the disparity estimator (DE) to yield a disparity map denoted $\mathbf{d}$ using the well-known BM algorithm. $\mathbf{d}$ is then used by the Image Predictor (IP) to transform $\widehat{\mathbf{I}}_l$ into the predicted view, denoted $\mathbf{I}_p$.

More specifically, $\widehat{\mathbf{I}}_l$ and $\mathbf{I}_r$ are decomposed into K non-overlapping blocks of same size. The upper left corner of the k-block is indicated by coordinates (i_k, j_k). The pixels contained in the k-block are referred to by $(i_k + \Delta\mathrm{i}, j_k + \Delta\mathrm{j})$ where $(\Delta\mathrm{i}, \Delta\mathrm{j})$ spans $\mathscr{B}$, a set listing all internal-block displacements (including $(0, 0)$).

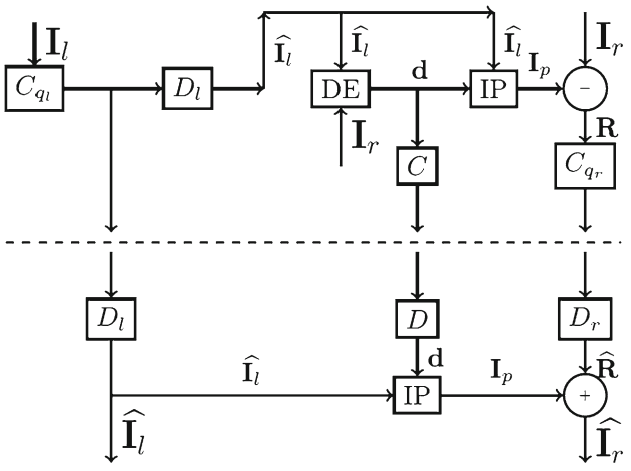

Fig. 1 DCC scheme where the encoder (above) is separated from the decoder (below) by a dashed line

$\mathbf{d}$ is an array of K disparity values denoted as $d_1, \ldots, d_K$. It describes the K right horizontal-shifts by which, in the IP-block, each $\widehat{\mathbf{I}}_l$-block is transformed into an $\mathbf{I}_p$-block:

$$\mathbf{I}_p\left(\begin{bmatrix} i_k + \Delta\mathrm{i}, \\ j_k + \Delta\mathrm{j} \end{bmatrix}\right) = \widehat{\mathbf{I}}_l\left(\begin{bmatrix} i_k + \Delta\mathrm{i}, \\ j_k + \Delta\mathrm{j} + d_k \end{bmatrix}\right), \tag{2}$$

where k ranges from 1 to K and $(\Delta\mathrm{i}, \Delta\mathrm{j})$ spans $\mathscr{B}$. This IP-block is shown on the upper right part in Fig. 1. To simplify notations, we do not indicate here the $\mathbf{d}$-dependency of $\mathbf{I}_p$.

The BM algorithm, in the DE-block, consists in selecting for each k-block, the disparity value d_k for which the k-block $\mathbf{I}_p$-values resemble most the k-block $\mathbf{I}_r$-values in the sense that the mean squared error is minimized as follows:

$$d_k(q_l) = \underset{d\in\mathbf{S}}{\arg\min} \sum_{(\Delta\mathrm{i},\Delta\mathrm{j})\in\mathscr{B}} \left(\widehat{\mathbf{I}}_l\begin{bmatrix} i_k + \Delta\mathrm{i} \\ j_k + \Delta\mathrm{j} + d \end{bmatrix} - \mathbf{I}_r\begin{bmatrix} i_k + \Delta\mathrm{i} \\ j_k + \Delta\mathrm{j} \end{bmatrix}\right)^2, \tag{3}$$

where $\mathbf{S}$ contains all allowed disparity values.

As $\widehat{\mathbf{I}}_l$ is q_l-dependent, the disparity value found, d_k is also q_l-dependent. C (center upper part) is a lossless encoding operation of the disparity map $\mathbf{d}$. The resulting bit stream is transmitted to the decoder (center downward arrow connecting the dashed line) which recovers the exact disparity map $\mathbf{d}$, through D, being the inverse operation of C as follows:

$$\mathbf{d} = D(C(\mathbf{d})). \tag{4}$$

The recovered disparity map is used with $\widehat{\mathbf{I}}_l$ by the second IP-block to yield according to Eq. (2), $\mathbf{I}_p$, this time in the decoder. This second IP-block is at the bottom in Fig. 1. $\mathbf{R}$ (upper right corner) represents the residual image, that is the difference between the original right view and its prediction:

$$\mathbf{R} = \mathbf{I}_r - \mathbf{I}_p. \tag{5}$$

C_{q_r} (upper right corner) is a lossy encoding operation where $q_r \in \mathbf{Q}_r$ is its quality factor and $\mathbf{Q}_r$ is the set of all allowed values. C_{q_r} compresses $\mathbf{R}$ into a bit stream transmitted to the decoder (right downward arrow connecting the dashed line). D_r, being the inverse operation of C_{q_r}, is used in the decoder to get an approximation of $\mathbf{R}$ denoted $\widehat{\mathbf{R}}$. By reversing Eq. (5), the decoder gets an approximation of $\mathbf{I}_r$ denoted as $\widehat{\mathbf{I}}_r$ and given by:

$$\widehat{\mathbf{I}}_r = \mathbf{I}_p + D_r(C_{q_r}(\mathbf{R})). \tag{6}$$

In general, $\widehat{\mathbf{I}}_r$ is closer to $\mathbf{I}_r$ than $\mathbf{I}_p$ and this improvement of $\mathbf{I}_p$ is being referred to as *compensation*.

The bitrate, denoted by b, is deduced from the bit streams $C_{q_l}(\mathbf{I}_l)$, $C(\mathbf{d})$ and $C_{q_r}(\mathbf{R})$:

$$b\left(\mathbf{I}_l, \mathbf{d}, \mathbf{I}_r, q_l, q_r\right) = \frac{|C_{q_l}(\mathbf{I}_l)| + |C(\mathbf{d})| + |C_{q_r}(\mathbf{R})|}{|\mathbf{I}_l| + |\mathbf{I}_r|}, \tag{7}$$

where $|\cdot|$ is the set cardinal number, here it helps counting, above, the number of bits and, below, the number of pixels.

3 Optimization problem statement

The aim of a coding/decoding scheme is a trade-off between getting the highest quality (i.e., visual rendering) while using the least amount of bits accounted for by Eq. (7). In this paper, this trade-off is rephrased into finding the best quality within a constrained bit budget. The mean squared error between $(\widehat{\mathbf{I}}_l, \widehat{\mathbf{I}}_r)$ and $(\mathbf{I}_l, \mathbf{I}_r)$ is used as the cost function to be minimized with respect to a bit budget, b_a. More specifically, the mean squared error of the k-block of an image $\mathbf{I}'$ as compared to that of an image $\mathbf{I}$ is:

$$J_k\left(\mathbf{I}', \mathbf{I}\right) = \frac{1}{|\mathscr{B}|} \sum_{(\Delta\mathrm{i},\Delta\mathrm{j})\in\mathscr{B}} \left(\mathbf{I}'\begin{bmatrix} i_k + \Delta\mathrm{i} \\ j_k + \Delta\mathrm{j} \end{bmatrix} - \mathbf{I}\begin{bmatrix} i_k + \Delta\mathrm{i} \\ j_k + \Delta\mathrm{j} \end{bmatrix}\right)^2. \tag{8}$$

Averaging J_k over all blocks yields J:

$$J\left(\mathbf{I}', \mathbf{I}\right) = \frac{1}{K}\sum_{k=1}^{K} J_k\left(\mathbf{I}', \mathbf{I}\right). \tag{9}$$

The cost function is then defined as:

$$\mathscr{J}\left(\widehat{\mathbf{I}}_l, \mathbf{I}_l, \widehat{\mathbf{I}}_r, \mathbf{I}_r\right) = \frac{1}{2}J\left(\widehat{\mathbf{I}}_l, \mathbf{I}_l\right) + \frac{1}{2}J\left(\widehat{\mathbf{I}}_r, \mathbf{I}_r\right). \tag{10}$$

This choice of cost function gives way to an optimization problem. $\widehat{\mathbf{I}}_r$ is actually $(q_l, q_r, \mathbf{d})$ dependent as stated by Eqs. (1), (2), (5) and (6). $\widehat{\mathbf{I}}_l$ is q_l dependent (see Eq. (1)). Such dependencies are indicated here:

$$\begin{cases} \mathbf{d}(q_l, q_r) = \underset{\mathbf{d}\in\mathbf{S}^K}{\arg\min}\ J(\widehat{\mathbf{I}}_r(q_l, q_r, \mathbf{d}), \mathbf{I}_r) \\ (q_l, q_r) = \underset{q_l\in\mathbf{Q}_l,\ q_r\in\mathbf{Q}_r,\ b\le b_a}{\arg\min}\ \mathscr{J}\left(\widehat{\mathbf{I}}_l(q_l), \mathbf{I}_l, \widehat{\mathbf{I}}_r(q_l, q_r, \mathbf{d}(q_l, q_r)), \mathbf{I}_r\right) \end{cases} \tag{11}$$

where b, defined in Eq. (7), depends on $\mathbf{I}_l, \mathbf{d}, \mathbf{I}_r, q_l, q_r$. $\mathbf{S}^K$ is the set of all arrays of size K whose components are in $\mathbf{S}$, and b_a is the expected bitrate.

Investigating the link between the BM algorithm and this optimization problem, Eq. (3) is recasted into:

$$d_k(q_l) = \underset{s\in\mathbf{S}}{\arg\min}\ J_k(\mathbf{I}_p, \mathbf{I}_r). \tag{12}$$

When considering the whole array of disparities, Eq. (12) becomes:

$$\mathbf{d}(q_l) = \underset{\mathbf{d}\in\mathbf{S}^K}{\arg\min}\ J(\mathbf{I}_p, \mathbf{I}_r). \tag{13}$$

Equation (13) is different from Eq. (11) only in that $\mathbf{I}_p$ is considered instead of $\widehat{\mathbf{I}}_r$. This difference is actually the decoded–encoded residual as stated by Eqs. (5) and (6):

$$\widehat{\mathbf{I}}_r - \mathbf{I}_p = D_r\left(C_{q_r}\left(\mathbf{I}_r - \mathbf{I}_p\right)\right). \tag{14}$$

Hence, the BM algorithm can be regarded as a suboptimal solution of Eq. (11), where the effect of the choice of the disparity on the residual, and the residual impact on the distortion, are neglected. Note that from then on, this DCC algorithm is referred to as BM algorithm.

4 Review of DCBM algorithm

This section presents the strategy of the disparity-compensated block matching (DCBM) algorithm already developed in [22]. The DCBM algorithm is different from the BM algorithm in that Eq. (11) is no longer simplified into Eq. (13). The DCBM algorithm is derived from a different suboptimal solution involving much greater numerical complexity. Indeed the algorithm is computed in $K+1$ steps. In the first step, the disparity map is computed using the BM algorithm. This initial disparity map has the K following components:

$$d_k(0, q_l) = \underset{d\in S}{\arg\min}\, J_k\left(\mathbf{I}_p, \mathbf{I}_r\right), \tag{15}$$

where k ranges from 1 to K. Note that at this point $\mathbf{d}(0, q_l)$ does not depend on q_r.

The goal at step $t \in \{1, \ldots, K\}$ is to select the k-block disparity, denoted, for now, as s. We assume that a disparity map $\mathbf{d}(t-1, q_l, q_r)$ has already been computed at step $t-1$. For each $s \in \mathbf{S}$, a predicted image $\mathbf{I}_p(t, q_l, q_r, s)$ is computed taking into account s on the tth block and $d_k(t-1, q_l, q_r)$ for all other blocks:

$$\mathbf{I}_p(t, q_l, q_r, s)\begin{bmatrix} i_k + \Delta\mathrm{i} \\ j_k + \Delta\mathrm{j} \end{bmatrix} = \begin{cases} \widehat{\mathbf{I}}_l\begin{bmatrix} i_k + \Delta\mathrm{i} \\ j_k + \Delta\mathrm{j} + d_k(t-1, q_l, q_r) \end{bmatrix} & \text{if} \quad k \neq t \\ \widehat{\mathbf{I}}_l\begin{bmatrix} i_k + \Delta\mathrm{i} \\ j_k + \Delta\mathrm{j} + s \end{bmatrix} & \text{if} \quad k = t \end{cases} \tag{16}$$

with $(\Delta\mathrm{i}, \Delta\mathrm{j})$ spanning $\mathscr{B}$ and k ranging from 1 to K.

Compensation transforms $\mathbf{I}_p(t, q_l, q_r, s)$ into $\widehat{\mathbf{I}}_r(t, q_l, q_r, s)$ as follows:

$$\widehat{\mathbf{I}}_r(t, q_l, q_r, s) = \mathbf{I}_p(t, q_l, q_r, s) + D_r C_{q_r}\left(\mathbf{I}_r - \mathbf{I}_p(t, q_l, q_r, s)\right). \tag{17}$$

Finally, $J(\widehat{\mathbf{I}}_r, \mathbf{I}_r)$ is computed and the best disparity is selected as follows:

$$d_k(t, q_l, q_r) = \begin{cases} d_k(t-1, q_l, q_r) & \text{if} \quad k \neq t \\ \underset{s\in\mathbf{S}}{\arg\min}\ J\left(\widehat{\mathbf{I}}_r(t, q_l, q_r, s), \mathbf{I}_r\right) & \text{if} \quad k = t \end{cases} \tag{18}$$

Note that the increased numerical complexity when using DCBM, stems from the necessity, to code and decode a new image, at each block and then each time a new disparity value is considered. The DCBM algorithm is summarized in Algorithm 1.

Algorithm 1 DCBM algorithm

Input: $\mathbf{I}_l, \mathbf{I}_r, q_l, q_r$
Output: $C_{q_l}(\mathbf{I}_l), C(\mathbf{d}), C_{q_r}(\mathbf{R}), b, \mathscr{J}$
Compute $C_{q_l}(\mathbf{I}_l)$, $\widehat{\mathbf{I}}_l$ with Eq. (1) and $J(\widehat{\mathbf{I}}_l, \mathbf{I}_l)$ with Eqs. (8) and (9)
Compute $\mathbf{d}(0, q_l)$ with Eq. (15) using $\mathbf{I}_p$ defined by Eq. (2)
for all $t \in \{1 \ldots K\}$ **do**
  for all $s \in \mathbf{S}$ **do**
    Compute $\mathbf{I}_p(t, q_l, q_r, s)$ with Eq. (16) using $\mathbf{d}(t-1, q_l, q_r)$
    Compute $\widehat{\mathbf{I}}_r(t, q_l, q_r, s)$ with Eq. (17)
    Compute $J\left(\widehat{\mathbf{I}}_r(t, q_l, q_r, s), \mathbf{I}_r\right)$ with Eq. (9)
  end for
  Select $\mathbf{d}(t, q_l, q_r)$ with Eq. (18) using all s-values of $J(\widehat{\mathbf{I}}_r, \mathbf{I}_r)$
end for
Get $\mathbf{d} = \mathbf{d}(K, q_l, q_r)$ and compute $C(\mathbf{d})$
Compute $\mathbf{I}_p$ with Eq. (2) using $\mathbf{d}$
Compute $\mathbf{R} = \mathbf{I}_r - \mathbf{I}_p$ and $C_{q_r}(\mathbf{R})$ with Eq. (5)
Compute $\widehat{\mathbf{I}}_r$ with Eq. (6) and $J(\widehat{\mathbf{I}}_r, \mathbf{I}_r)$ with Eq. (9)
Compute $\mathscr{J}$ with Eq. (10) using $J(\widehat{\mathbf{I}}_l, \mathbf{I}_l)$ and $J(\widehat{\mathbf{I}}_r, \mathbf{I}_r)$
Compute $b(\mathbf{I}_l, \mathbf{d}, \mathbf{I}_r, q_l, q_r)$ with Eq. (7) using $C_{q_l}(\mathbf{I}_l), C(\mathbf{d}), C_{q_r}(\mathbf{R})$

5 Proposed FDCBM algorithm

Due to the interesting performance of the DCBM algorithm (see [22]), this section proposes a Fast version of this algorithm called FDCBM algorithm. The novelty is that disparity selection is no longer based on the computation of $\widehat{\mathbf{I}}_r$ with all its pixel values. The underlying idea of the developed algorithm is first discussed, and then an explicit formula of the JPEG-codec distortion is derived. Blocks of size 8×8 pixels are considered knowing that an extension to a larger block size is possible.

5.1 FDCBM algorithm underlying idea

This section considers that the size of $\mathscr{B}$ is 8×8 and more specifically that the disparity-related blocks are exactly the JPEG-related blocks.

Introduce first some new notations. Define $\widehat{\mathbf{R}} = D_r C_{q_r}(\mathbf{R})$ the reconstructed residual at the decoder, and $\mathbf{I}_k$ any matrix of size 8×8:

$$\begin{cases} \mathbf{R}_k(\Delta\mathrm{i}, \Delta\mathrm{j}) = \mathbf{R}(i_k + \Delta\mathrm{i}, j_k + \Delta\mathrm{j}) \\ \widehat{\mathbf{R}}_k(\Delta\mathrm{i}, \Delta\mathrm{j}) = \widehat{\mathbf{R}}(i_k + \Delta\mathrm{i}, j_k + \Delta\mathrm{j}) \\ \|\mathbf{I}_k\|^2 = \frac{1}{|\mathscr{B}|}\sum_{(\Delta\mathrm{i},\Delta\mathrm{j})\in\mathscr{B}} (\mathbf{I}_k(\Delta\mathrm{i}, \Delta\mathrm{j}))^2 \end{cases} \tag{19}$$

So as to be consistent with notations defined in Sect. 2, indexes of these 8×8 matrices start from 0: $\Delta\mathrm{i}, \Delta\mathrm{j} \in \{0, \ldots 7\}$. Note that because of the above block-related assumption, $\widehat{\mathbf{R}}_k$ can also be considered as the decoded–encoded 8×8 matrix $\mathbf{R}_k$:

$$\widehat{\mathbf{R}}_k = D_r C_{q_r}(\mathbf{R}_k). \tag{20}$$

Our main claim is that the relevant pixel values are those of $\mathbf{R}_k$ and that J_k measures the mean squared distortions yielded by the compression and decompression of $\mathbf{R}_k$:

$$\begin{aligned} J_k\left(\widehat{\mathbf{I}}_r, \mathbf{I}_r\right) &= J_k\left(\mathbf{I}_p + \widehat{\mathbf{R}}, \mathbf{I}_p + \mathbf{R}\right) = J_k\left(\widehat{\mathbf{R}}, \mathbf{R}\right) \\ &= \left\| D_r C_{q_r}(\mathbf{R}_k) - \mathbf{R}_k \right\|^2. \end{aligned} \tag{21}$$

The first equality is obtained with Eqs. (5) and (6). The second equality uses an additive-invariance property derived from Eq. (8). The third equality is computed using Eqs. (8), (19) and (20).

5.2 JPEG encoding modeling

This section is interested in what JPEG encoding causes distortions, namely the quantization of the DCT-components:

$$D_r C_{q_r}(\mathbf{R}_k) = \mathbb{IDCT}\left[Q_{q_r}(\mathbb{DCT}[\mathbf{R}_k])\right], \tag{22}$$

where Q_{q_r} is the 8×8-JPEG-quantizer.

As DCT is an orthogonal transformation, it preserves the L2 norm:

$$\begin{aligned} &\left\| D_r C_{q_r}(\mathbf{R}_k) - \mathbf{R}_k \right\|^2 \\ &= \left\| \mathbb{DCT}\left[D_r C_{q_r}(\mathbf{R}_k)\right] - \mathbb{DCT}[\mathbf{R}_k] \right\|^2. \end{aligned} \tag{23}$$

Combining Eqs. (22) and (23), a minimized formula of the mean squared distortions is obtained:

$$\begin{aligned} &\left\| D_r C_{q_r}(\mathbf{R}_k) - \mathbf{R}_k \right\|^2 \\ &= \left\| Q_{q_r}(\mathbb{DCT}[\mathbf{R}_k]) - \mathbb{DCT}[\mathbf{R}_k] \right\|^2. \end{aligned} \tag{24}$$

The explicit formula uses the following information extracted from the JPEG-codec (see [23]). The DCT of an 8×8 matrix is:

$$\mathbb{DCT}[\mathbf{I}_k] = T^T \mathbf{I}_k T, \tag{25}$$

where T is an 8×8 orthogonal matrix defined as follows:

$$\begin{aligned} T_{\Delta\mathrm{i},\Delta\mathrm{j}} = &\frac{1}{\sqrt{8}} \cos\left(\pi \frac{(2\Delta\mathrm{j}+1)\Delta\mathrm{i}}{16}\right) \\ &\times \begin{cases} 1 & \text{if } \Delta\mathrm{i} = 0 \\ \sqrt{2} & \text{if } 1 \le \Delta\mathrm{i} \le 7 \end{cases} \end{aligned} \tag{26}$$

The JPEG-quantizer transforms an 8×8-matrix into an 8×8-matrix:

$$Q_{q_r}(\mathbf{I}) = \left[\mathrm{Round}\left(\frac{\mathbf{I}(\Delta\mathrm{i}, \Delta\mathrm{j})}{\mathscr{Q}(\Delta\mathrm{i}, \Delta\mathrm{j})\alpha(q_r)}\right)\mathscr{Q}(\Delta\mathrm{i}, \Delta\mathrm{j})\alpha(q_r)\right]_{\Delta\mathrm{i},\Delta\mathrm{j}} \tag{27}$$

using a nonlinear mapping transforms q_r into a scaling factor (see [24]):

$$\alpha(Q) = \begin{cases} \frac{50}{Q} & \text{if } Q \le 50 \\ 2 - \frac{Q}{50} & \text{if } Q > 50 \end{cases} \tag{28}$$

Experimentations have shown that $J_k\left(\widehat{\mathbf{I}}_r, \mathbf{I}_r\right)$ is not exactly equal to $\left\| Q_{q_r}(\mathbb{DCT}[\mathbf{R}_k]) - \mathbb{DCT}[\mathbf{R}_k] \right\|^2$, and the latter depends on q_l, q_r and on the k-block disparity, s. So the following notation is used:

$$\tilde{J}_k(q_l, q_r, s) = \left\| Q_{q_r}(\mathbb{DCT}[\mathbf{R}_k]) - \mathbb{DCT}[\mathbf{R}_k] \right\|^2. \tag{29}$$

Finally, the k-block disparity is selected as:

$$d_k(q_l, q_r) = \arg\min_{s\in\mathbf{S}} \tilde{J}_k(q_l, q_r, s). \tag{30}$$

5.3 Derived FDCBM algorithm

Instead of computing large-scale images with DCBM algorithm, only 8×8-matrices are computed yielding to an approximation of $J_k(\widehat{\mathbf{I}}_r, \mathbf{I}_r)$ (i.e., $\tilde{J}_k(q_l, q_r, s)$) using Eq. (29). Moreover, instead of selecting the k-block disparity based on $J(\widehat{\mathbf{I}}_r, \mathbf{I}_r)$, it is based on the minimization of $\tilde{J}_k(q_l, q_r, s)$. The numerical complexity of FDCBM algorithm is then definitely much lower than that of DCBM algorithm. It remains higher than that of the BM algorithm, not only because of the complexity of Eq. (29) but also because it takes into account q_l and q_r, whereas BM takes into account only q_l. The FDCBM algorithm is summarized in Algorithm 2.

Algorithm 2 FDCBM algorithm

Input: $\mathbf{I}_l, \mathbf{I}_r, q_l, q_r$
Output: $C_{q_l}(\mathbf{I}_l), C(\mathbf{d}), C_{q_r}(\mathbf{R}), b, \mathscr{J}$
Compute $C_{q_l}(\mathbf{I}_l)$, $\widehat{\mathbf{I}}_l$ with Eq. (1) and $J(\widehat{\mathbf{I}}_l, \mathbf{I}_l)$ with Eqs. (8) and (9)
for all $k \in \{1 \dots K\}$ **do**
  for all $s \in \mathbf{S}$ **do**
    Compute $\mathbf{R}_k$ using $\widehat{\mathbf{I}}_l$ and $\mathbf{I}_r$ with Eqs. (19), (2) and (5)
    Compute $\tilde{J}_k(q_l, q_r, s)$ with Eq. (29)
  end for
  Select d_k with Eq. (30) using all s-values of $\tilde{J}_k(s)$
end for
Collect $\mathbf{d} = (d_1, \dots, d_K)$ and compute $C(\mathbf{d})$
Compute $\mathbf{I}_p$ with Eq. (2) using $\mathbf{d}$
Compute $\mathbf{R} = \mathbf{I}_r - \mathbf{I}_p$ and $C_{q_r}(\mathbf{R})$ with Eq. (5)
Compute $\widehat{\mathbf{I}}_r$ with Eq. (6) and $J(\widehat{\mathbf{I}}_r, \mathbf{I}_r)$ with Eq. (9)
Compute $\mathscr{J}$ with Eq. (10) using $J(\widehat{\mathbf{I}}_l, \mathbf{I}_l)$ and $J(\widehat{\mathbf{I}}_r, \mathbf{I}_r)$
Compute $b(\mathbf{I}_l, \mathbf{d}, \mathbf{I}_r, q_l, q_r)$ with Eq. (7) using $C_{q_l}(\mathbf{I}_l)$, $C(\mathbf{d})$, $C_{q_r}(\mathbf{R})$

6 Performance of the proposed algorithm

This section starts with a discussion on the validity of Eq. (29) on which the proposed FDCBM algorithm is based. To do so, simulations are conducted on synthetic data to measure the ability of this equation to reduce distortions more than the BM algorithm.

For each $q_r \in \{1, \dots, 99\}$, 200 stereoscopic images of size 256×256 are randomly drawn from independent uniform distributions (left views are not encoded) using ω ranging from 1 to 200 and $\mathbf{S} = \{-14, \dots, 15\}$. On each image, a block is randomly selected and for this block, the BM, DCBM and FDCBM algorithms yield three disparities denoted as $d_{\mathrm{BM}}(q_r, \omega), d_{\mathrm{DCBM}}(q_r, \omega), d_{\mathrm{FDCBM}}(q_r, \omega)$.

For each image and each algorithm, its mean squared distortion is computed and denoted as $J_k(q_r, d_{\mathrm{BM}}(q_r, \omega), \omega)$, $J_k(q_r, d_{\mathrm{DCBM}}(q_r, \omega), \omega)$ and $J_k(q_r, d_{\mathrm{FDCBM}}(q_r, \omega), \omega)$. These simulations clearly confirm that:

$$\begin{cases} J_k(q_r, d_{\mathrm{DCBM}}(q_r, \omega), \omega) \leq J_k(q_r, d_{\mathrm{BM}}(q_r, \omega), \omega) \\ J_k(q_r, d_{\mathrm{DCBM}}(q_r, \omega), \omega) \leq J_k(q_r, d_{\mathrm{FDCBM}}(q_r, \omega), \omega) \end{cases}$$

Moreover, most often, simulations show that:

$$J_k(q_r, d_{\mathrm{FDCBM}}(q_r, \omega), \omega) \leq J_k(q_r, d_{\mathrm{BM}}(q_r, \omega), \omega).$$

To see how $J_k(q_r, d_{\mathrm{FDCBM}}(q_r, \omega), \omega)$ is close to $J_k(q_r, d_{\mathrm{DCBM}}(q_r, \omega), \omega)$ as compared to $J_k(q_r, d_{\mathrm{BM}}(q_r, \omega), \omega)$, an average distortion reduction ratio is measured as follows:

$$\rho(q_r) = \frac{1}{200} \times \sum_{\omega=1}^{200} \frac{J_k(q_r, d_{\mathrm{BM}}(q_r, \omega), \omega) - J_k(q_r, d_{\mathrm{FDCBM}}(q_r, \omega), \omega)}{J_k(q_r, d_{\mathrm{BM}}(q_r, \omega), \omega) - J_k(q_r, d_{\mathrm{DCBM}}(q_r, \omega), \omega)}. \quad (31)$$

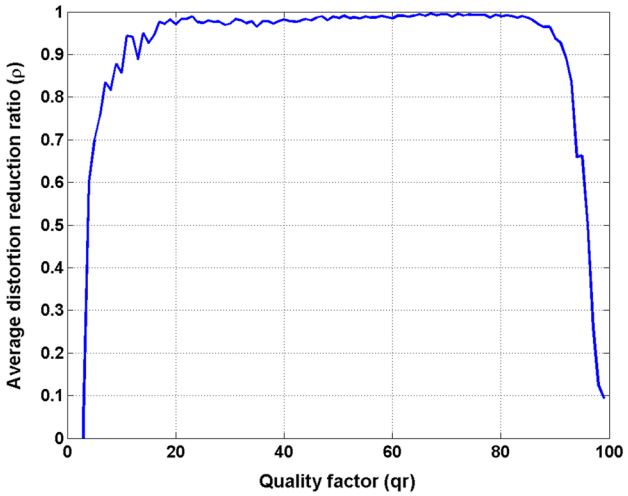

Fig. 2 Average distortion reduction ratio of BM–FDCBM compared to BM–DCBM on synthetic data (function of q_r)

Figure 2 illustrates the behavior of the ratio $\rho(q_r)$ when q_r ranges from 1 to 100. When q_r is between 15 and 90, on average and compared to the distortions left when using BM algorithm, FDCBM algorithm is able to reduce at least 90% of the distortions that DCBM algorithm is able to reduce.

The second part of this section concerns the simulation results performed on Middleburry dataset stereoscopic images [2]. To simplify the experiment, the left view is not compressed. Assume that the pixel values, on both views, are ranging from 0 to 255. The distortion of the predicted right view is measured using the peak signal-to-noise ratio ($PSNR$) given by $PSNR = 10 \log_{10}\left(\frac{255^2}{J(\widehat{\mathbf{I}}_r, \mathbf{I}_r)}\right)$. The rate, in bits per pixel (bpp), is measured only on the right view according to $b = \frac{|C(\mathbf{d})| + |C_{q_r}(\mathbf{R})|}{|\mathbf{I}_r|}$. The lossless coder, C, is here an arithmetic coder (see [25]). To reduce the numerical complexity, the set of quality factor values is reduced to $\mathbf{Q}_r = \{5, 10, 15, \dots, 90\}$. The set of all available disparities is $\mathbf{S} = \{0, \dots, 120\}$.

The rate-distortion curves, provided in Fig. 3, confirm the results stated above using "Art" stereoscopic image of Middlebury-dataset (2005) and blocks of size 8×8. Indeed, the performance (in terms of rate distortion) of the proposed FDCBM algorithm is similar to that of DBCM algorithm, which is, however, better than that of the classical BM algorithm and the reference-based block matching algorithm called (R algorithm) proposed in [26]. Figure 5 presents the decompressed right image "Aloe" extracted from Middleburry dataset (2006) using BM algorithm on the left side, R algorithm on the mid side and FDCBM on the right side. For each algorithm, blocks are of sizes 8×8 and $q_r \in \mathbf{Q}_r$ is set so that $b = 0.3$ bpp. When comparing both reconstructed views with the original view, it appears that the background cloth on right neighborhoods of each vertical leaf is wrongly

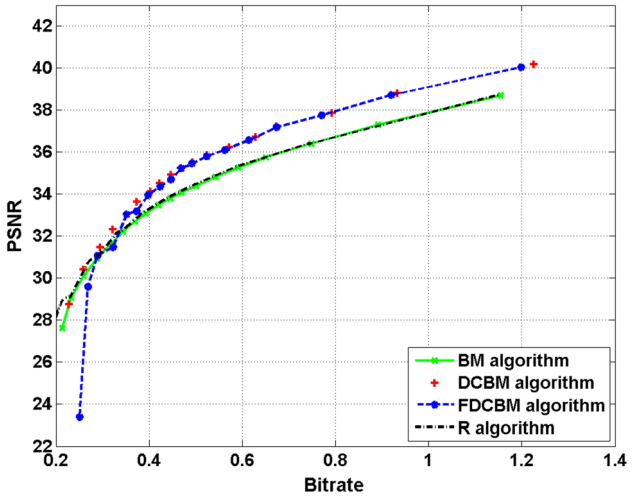

Fig. 3 Performance comparison of BM, DCBM, FDCBM and R algorithms using "Art" stereoscopic image of Middlebury-dataset (2005)

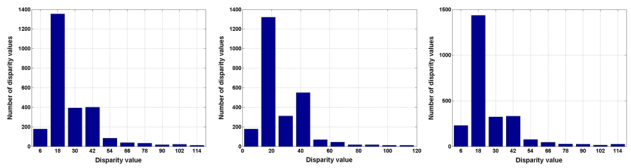

Fig. 4 Histogram of the disparity map yielded at $b = 0.3$ bpp using "Aloe" stereoscopic image: BM algorithm (left side), R algorithm (on the middle) and FDCBM algorithm (right side)

drawn. The reason may be that these neighborhoods are occluded in the left view. The BM and R algorithms yield a dotted structure, whereas the FDCBM algorithm yields a slightly blurred square texture. From a PSNR-viewpoint, the FDCBM-reconstructed view is closer to the original view (30.14 dB) than the BM-reconstructed view (29.5 dB) and the R-reconstructed image (29.6 dB).

Figure 4 shows the histograms of, on the left side, the BM-disparity map, on the mid side the R-disparity map, and on the right side, the FDCBM-disparity map for the same experiment. More specifically, selected disparity values are sorted into 10 bins, each bin is referred to by its average disparity value on the horizontal axis. The vertical axis indicates the number of blocks for which the disparity value falls into a given bin. (The total number of blocks for that image is 2726.) Both histograms are right skewed, showing that for most blocks it did not prove useful to consider disparity values greater than 50. A closer look shows that, on the right-hand side, the two first columns are slightly bigger and the two following columns are slightly smaller. This means that for this specific image, on average FDCBM algorithm tends to select smaller disparity values than BM and R algorithms (Fig. 5).

Figure 6 provides the reconstructions of the "Dwarves" right view from Middleburry dataset (2005) using BM (in the

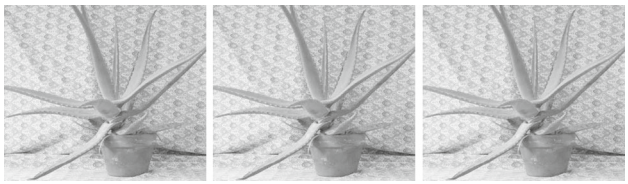

Fig. 5 Reconstructed "Aloe" right view: BM algorithm (left side); R algorithm (mid side) and FDCBM algorithm (right side)

Fig. 6 Reconstructed "Dwarves" right view: BM algorithm (left side); R algorithm (mid side) and FDCBM algorithm (right side)

left), R (on the middle) and FDCBM (on the right) algorithms. A zoom on the vase (bottom right) shows that FDCBM gives much closer reconstruction to the original view.

As for numerical complexity, FDCBM algorithm (consuming 17 s) is 3388 times quicker than DCBM algorithm (consuming 4 h), 6.8 times slower than BM algorithm (consuming 2.5 s) and 1.5 times slower than R algorithm (consuming 12 s). This has been measured on the "Aloe" stereoscopic image with block of 8×8 size using Matlab in a Windows environment on a computer using one processor with four cores at a frequency of 3.7 GHz.

The Bjøntegaard metric [27] is used here to quantify the increase in performance of FDCBM algorithm as compared to BM and R algorithms. Based on four rate-distortion points for each algorithm (roughly [0.3, 0.4, 0.5, 0.6] bpp), it computes an average PSNR increase or an average bitrate decrease. As for the "Art" stereoscopic image, FDCBM algorithm yields on average a PSNR increase of, respectively, 0.78 dB and 0.52 dB compared to the BM and R algorithms. To simplify its reading, the stereoscopic images have been sorted by their increase in PSNR performance to compare FDCBM with BM algorithms.

Table 1 shows on columns 2 and 3 that, on average, for all stereoscopic images, FDCBM is better performing than BM, and the difference ranges from 0.42 up to 1.69 dB. It seems difficult to understand why this difference is higher for some images and lower on other images. For instance, "Cloth3" and "Cloth4" appear at both ends of the table and yet have similar appearance. The same comment applies to "Baby1" and "Baby3." And both "Midd1," "Midd2" and "Lampshade1," "Lampshade2" have similar appearance and yet each pair has quite different performance increases. It is interesting to note that the stereoscopic image having the least PSNR-performance increase (+0.17 dB), namely "Plastic," is having a rather important bitrate decrease (−15.73%).

Table 1 Performance comparison between FDCBM, BM and R algorithms using Bjøntegaard metric on Middlebury-datasets (2005–2006)

Image	FDCBM versus BM algorithms		FDCBM versus R algorithms	
	ΔPSNR (dB)	bpp (%)	ΔPSNR (dB)	bpp (%)
Plastic	+0.17	−15.73	+0.41	−10.3
Cloth3	+0.31	−7.44	+0.74	−12.78
Midd1	+0.31	−4.79	+0.1	−6.5
Cloth1	+0.34	−9.08	+0.2	−8.11
Laundry	+0.37	−5.66	+0.48	−7.61
Computer	+0.39	−6.3	+0.35	−6.09
Baby1	+0.39	−8.17	+0.42	−10.03
Baby2	+0.42	−9.35	+0.4	−7.44
Wood1	+0.43	−8.9	+0.68	−17.64
Rocks2	+0.44	−9.89	+0.49	−13.83
Books	+0.46	−7.93	+0.37	−5.61
Aloe	+0.53	−11.7	+0.6	−14.2
Lampshade1	+0.54	−5.09	+0.66	−13.01
Rocks1	+0.56	−12.43	+0.5	−13.53
Bowling2	+0.57	−9.86	+0.77	−13.94
Midd2	+0.58	−10.51	+0.11	−3.82
Drumsticks	+0.58	−8.85	+0.41	−6.67
Dolls	+0.59	−10.65	+0.51	−10.51
Moebius	+0.65	−11.44	+1.24	−22.59
Cloth2	+0.66	−12.68	+0.73	−15.47
Baby3	+0.67	−12.74	+0.55	−10.55
Wood2	+0.72	−9.34	+0.65	−8.42
Monopoly	+0.75	−13.56	+0.44	−6.53
Cloth4	+0.76	−17.12	+0.8	−14.63
Art	+0.78	−11.36	+0.52	−8.81
Bowling1	+0.89	−14.47	+0.11	−20.98
Dwarves	+1.06	−17.48	+0.64	−12.64
Flowerpots	+1.12	−14.52	+1.12	−16.52
Lampshade2	+1.23	−22.76	+0.91	−18.23
Mean	+0.62	−11.38	+0.54	−11.62

Columns 3 and 4 of Table 1 summarize the performance of our proposal compared to the R algorithm. One can observe that the FDCBM algorithm achieves better performance compared to the R algorithm.

7 Conclusion

A new block-based disparity estimation technique called FDCBM algorithm has been developed. The purpose of this work is not to be competitive with stereoscopic image/video standards, but to first show the feasibility of the proposed approach as a proof of the concept. Where the classical technique selects each disparity so that the predicted image resembles most the right view, the proposed technique computes for each disparity the compensated image, and the selected disparity is the one yielding the highest similarity between the compensated image and the right view. The computation is done with an analytic expression derived here from the JPEG-codec. To reduce the numerical complexity, these computations are fed using only the considered block pixel values.

Tested on stereoscopic images, FDCBM algorithm is performing better than the classical disparity-compensated compression algorithm using a block matching and reference-based block matching disparity estimation technique. For example, as compared to the former, the increase in performance, at same bitrate, is ranging, depending on the stereoscopic image, from 0.17 up to 1.23 dB with an average of 0.62 dB. The underlying idea of this paper is not to replace the residual error encoding methods in the stereoscopic image/video standards by JPEG encoding but rather

to exploit the quantization parameters and tables, as specified in the standards, to better choose the disparities to improve the compensated view quality. Indeed, the residual error coding is traditionally based on an orthogonal transformation followed by a quantization process controlled by some parameters associated with quantization tables which need to be studied in future work. Moreover, only equal size blocks have been considered to show the interest of the proposed strategy. Blocks of variable size will be investigated in the near future.

References

1. Nam, K., Anh-Hoang, P., Munkh-Uchral, E., Ashraful, AMd, Ki-Chul, K., Mei-Lan, P., Jeong-Hyeon, L.: 3D display technology. Disp. Imaging **1**, 73–95 (2013)
2. Scharstein, D., Pal, C.: Learning conditional random fields for stereo. In: IEEE Conference on Computer Vision and Pattern Recognition, Minneapolis, MN, pp. 1–8 (2007)
3. Dufaux, F., Pesquet-Popescu, B., Cagnazzo, M.: Emerging Technologies for 3D Video: Creation, Coding, Transmission and Rendering, 1st edn. Wiley, Hoboken (2013)
4. Frossard, P., Schenkel, M.B., Luo, C., Wu, F.: Joint decoding of stereo jpeg image pairs. In: IEEE International Conference on Image Processing, pp. 2633–2636 (2010)
5. Ortis, A., Battiato, S.: A new fast matching method for adaptive compression of stereoscopic images. In Three-Dimensional Image Processing, Measurement (3DIPM), and Applications 2015. San Francisco, California, USA, February 10–12, p. 93930K (2015)
6. Ahlvers, Udo, Zölzer, Udo, Rechmeier, Stefan: FFT-based disparity estimation for stereo image coding. Proceedings 2003 International Conference on Image Processing (Cat. No.03CH37429), vol. 1, pp. I–761 (2003)
7. Schwarz, H., Bartnik, C., Bosse, S., Brust, H., Hinz, T., Lakshman, H., Marpe, D., Merkle, P., Müller, K., Rhee, H., Tech, G., Winken, M., Wiegand, T.: 3D video coding using advanced prediction, depth modeling, and encoder control methods. In: Picture Coding Symposium, pp. 1–4. (May 2012)
8. Li, S., Yu, M., Jiang, G., Choi, T.-Y., Kim, Y.-D.: Approaches to H. 264-based stereoscopic video coding. In: Third International Conference on Image and Graphics (ICIG'04), pp. 365–368 (Dec 2004)
9. Hanhart, P., Rerabek, M., Korshunov, P., Ebrahimi, T.: Subjective Evaluation of HEVC Intra Coding for Still Image Compression. Technical report, [JCT-VC contribution] AhG4, (Jan 2013)
10. Woo, W., Ortega, A.: Stereo image compression with disparity compensation using the MRF model. Vis. Commun. Image Process. **2727**, 1–14 (1996)
11. Flierl, M., Mavlankar, A., Girod, B.: Motion and disparity compensated coding for multi-view video. IEEE Trans. Circuits Syst. Video Technol. **17**, 1474–1484 (2007)
12. Kadaikar, A., Dauphin, G., Mokraoui, A.: Sequential block-based disparity map estimation algorithm for stereoscopic image coding. Signal Process. Image Commun. **39**, 159–172 (2015)
13. Kadaikar, A., Dauphin, G., Mokraoui, A.: Joint disparity and variable size-block optimization algorithm for stereoscopic image compression. Signal Process. Image Commun. **61**, 1–8 (2017)
14. Dauphin, G., Kaaniche, M., Mokraoui, A.: Block dependent dictionary based disparity compensation for stereo image coding. In: IEEE International Conference on Image Processing, ICIP, pp. 1–5. Québec City (Sept 2015)
15. Shen, G., Kim, W., Ortega, A., Lee, J., Wey, H.: Edge-aware intra prediction for depth-map coding. In: 2010 IEEE International Conference on Image Processing, pp. 3393–3396 (Sept 2010)
16. Chen, Y., Hannuksela, M.M., Zhu, L., Hallapuro, A., Gabbouj, M., Li, H.: Coding techniques in multiview video coding and joint multiview video model. In: 2009 Picture Coding Symposium, pp. 1–4 (May 2009)
17. Frajka, Tamás, Zeger, Kenneth: Residual image coding for stereo image compression. Opt. Eng. **42**, 182–189 (2003)
18. Wallace, G.K.: The JPEG still picture compression standard. IEEE Trans. Consum. Electron. **38**(1), 18–34 (1992)
19. Hachicha, W., Kaaniche, M., Beghdadi, A., Cheikh, F.A.: Efficient inter-view bit allocation methods for stereo image coding. IEEE Trans. Multimed. **17**(6), 765–777 (2015)
20. Kadaikar, Aysha, Dauphin, Gabriel, Mokraoui, Anissa: Sequential block-based disparity map estimation algorithm for stereoscopic image coding. Signal Process. Image Commun. **39**, 159–172 (2015)
21. Pan, R., Hou, Zh-X, Liu, Y.: Fast algorithms for inter-view prediction of multiview video coding. J. Multimed. **6**, 191–201 (2011)
22. Kadri, I., Dauphin, G., Mokraoui, A.: Stereoscopic image coding performance using disparity-compensated block matching algorithm. In: IEEE International Conference on Signal Processing: Algorithms, Architectures, Arrangements, and Applications, SPA, pp. 1–5 (Sept 2019)
23. CCIT: Information Technology-Digital Compression and Coding of Continuous-Tone Still Images-Requirements and Guidelines. Technical Report T.81, The International Telegraph and Telephone Consultative Committee CCIT (Sept 1992)
24. Alam, L., Dhar, P.K., Hasan, M.R., Bhuyan, M.G., Daiyan, G.M.: An improved JPEG image compression algorithm by modifying luminance quantization table. Int. J. Comput. Sci. Netw. Secur. (IJCSNS) **17**(1), 200–208 (2017)
25. Howard, Paul G., Vitter, Jeffrey: Arithmetic coding for data compression. Proc. IEEE **82**, 857–865 (1994)
26. Kadaikar, A., Dauphin, G., Mokraoui, A.: Modified block matching algorithm improving rate-distortion performance for stereoscopic image coding. In: 2015 IEEE International Symposium on Signal Processing and Information Technology (ISSPIT), pp. 478–483 (Dec 2015)
27. Bjøntegaard, G.: Calculation of average PSNR differences between RD-curves. In: Document VCEG-M33, ITU-T VCEG Meeting, Austin, Texas, USA (2001)