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

Imen Kadri, Gabriel Dauphin, Anissa Mokraoui, Zied Lachiri

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

Imen Kadri[a,b], Gabriel Dauphin[a], Anissa Mokraoui[a], Zied Lachiri[b]
[a]L2TI, Institut Galilée,Université Paris 13, Villetaneuse, France ,
[b]LSITI, Ecole Nationale d'Ingénieur de Tunis, Université de Tunis El Manar,Tunis,Tunisie,

Abstract. Stereoscopic images consist of two views rendering a depth sense. The small objects displacements across the two views are interpreted as a relative indication of depth. From a compression viewpoint, these displacements are exploited as specific inter-view redundancies. The classical and still compression scheme in use is called Disparity Compensated Compression scheme. A block-based disparity map modeling these displacements is losslessly compressed. Then the difference, between the original view and its disparity predicted view, is compressed and used by the decoder to compute the compensated view improving therefore 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 analytic expression is embedded into an algorithm, called Fast Disparity Compensated Block Matching algorithm (FDCBM), with a reasonable numerical complexity approaching the performance obtained with the proof-of-concept work. Tested on stereoscopic images, the proposed algorithm provides, at same bitrate, an average performance increase of 0:54dB as compared to the classical stereoscopic image coding scheme, when performance is measured by the Peak Signal to Noise Ratio.

Keywords: Stereoscopic Image Compression, Disparity-Compensation, Block Matching algorithm, JPEG-Distortion.

1 Introduction

Immersive multimedia combines near-to-reality outputs that users can experience through the sense of sight, hearing, touch and even smell. This immersive perception may also be rendered through interactive control of the viewpoint. Applications concern medical intervention, education and live production industry.[1] A hologram is an image that appears to be three dimensional as it changes with the relative position of the observer, being very precise, it can be used to record the complete 3D information of a scene.[2] A stereoscopic image is composed of two views which are perceived as two viewpoints of a single 3D-scene, thanks to a technical device (anaglyph, LC shutter or polarized 3D system), applications concern the entertainment industry (3D cinema), video games, medical field (stereoscopic displays) and cartography (aerial stereo-photography).[3] An integral image is a generalization of stereoscopy in which a large number of different viewpoints is used instead of two. It yields depth perception without using any technical device on the user's side.[4]

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[5] to estimate the depth map, generally assuming that objects look the same when seen from different views, which happens to be not so common.[6] Research in compression aims at reducing that amount of data by exploiting redundancies. In some applications, the depth map is also needed to generate an intermediate virtual view or to adjust the depth perception to improve the quality of experience.

This paper focuses on the stereoscopic images compression.[7–9] In this context, the depth map is *not* by itself an issue and it is needed only in that it explains the differences between the two views (a close object of a 3D-scene appears on both views at two distant locations). The horizontal distance between the two similar points is called the *disparity*. It is inversely related to the depth. Therefore stereoscopic images can be represented as a 2D image and a depth map with which it is possible to recover the other view using view synthesis. Research in this context includes sparse encoding of the depth map, image warping, inpaiting to fill in disoccluded regions.[6] 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 Ref. 6. In Ref. 10, 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 coding standard [11] which has been subjectively evaluated in Ref. 12.

As in Refs. 13, 14, 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. This scheme is implemented in the HEVC and the 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 Refs. 15, 16 and its limited predicting capacity,[17] by using blocks of arbitrary shapes in Ref. 18, and by addressing also the illumination compensation in Ref. 19. Investigating the statistical properties of the residual, reference[20] 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.[21] Reducing the numerical complexity is also a significant research issue. Examples include selecting optimal hyper-parameter values thanks to allocation modeling as opposed to an exhaustive search in Ref. 22 reducing the search area in Ref. 23 and using embedded coding scheme that can be truncated at any point to obtain the best reconstruction for a given bitrate .[20]

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 .[24] 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-distorsion performance of the compressed stereoscopic image.

This paper is organized as follows. Section 2 summarizes the basic concepts of the classical DCC scheme and introduces some notations. Section 3 shows how finding the best performing disparity map can be regarded as solving an optimization problem. The classical approach, called the Block-Matching algorithm (BM), is derived as a suboptimal solution. Section 4 recalls the greedy algorithm solving the optimization problem, it is called the Disparity-Compensated Block-Matching (DCBM) algorithm. Section 5 describes an extension of this algorithm called Fast Disparity-Compensated Block-Matching (FDCBM) algorithm. In Sec. 6, experiments show significant increased performance on some stereoscopic images. Section 7 concludes the paper.

2 Basic concepts and notations

This paper deals with compressing rectified stereoscopic images using the classical DCC scheme. Notations are first given and used in Sec. 3 to set the optimization problem, to which the well-known Block Matching (BM) algorithm is a suboptimal solution.

Notations, used in the following sections, are summarized in Fig. 1. This figure presents the DCC scheme where the dashed line separates the encoder (above) from the decoder (below).

$\mathbf{I}_l$ (upper left corner in Fig. 1) denotes the left view that is here chosen as the reference view. It feeds a lossy encoder denoted C_{q_l} (upper left corner in Fig. 1) where $q_l \in \mathbf{Q}_l$ is its quality factor with $\mathbf{Q}_l$ 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 in Fig. 1) 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 in Fig. 1). $\widehat{\mathbf{I}}_l$ 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 in Fig. 1) 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 (both views may have to be slightly enlarged to cover the last block column and the last block line). 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 $(i_k + \triangle\mathrm{i}, j_k + \triangle\mathrm{j})$ where $(\triangle\mathrm{i}, \triangle\mathrm{j})$ spans $\mathcal{B}$, a set listing all internal-block displacements (including $(0, 0)$). $\mathbf{d}$ 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:

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

where k ranges from 1 to K and $(\triangle\mathrm{i},\triangle\mathrm{j})$ spans $\mathcal{B}$. 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 $\mathbf{I}_p$.

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:

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

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 in Fig. 1) 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 in Fig. 1) 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 in Fig. 1) 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 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 $|\,.\,|$ is the set cardinal number, here it helps counting, above, the number of bits, and below, the number of pixels.

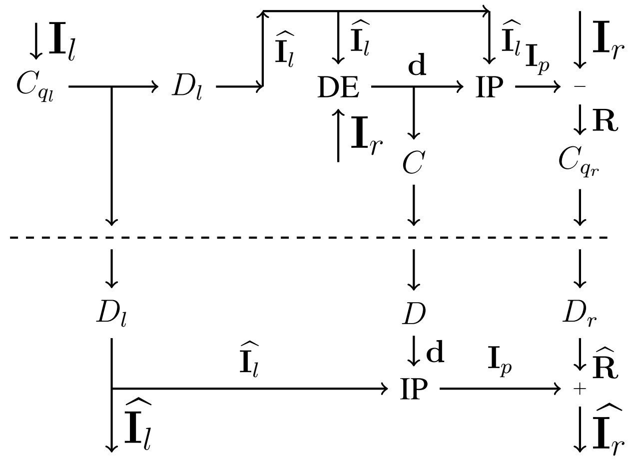

Fig 1 DCC scheme where the encoder (above) is separated from the decoder (below) by a dashed line.

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. As the end user observing the reconstructed stereoscopic image is generally a human being, our focus should be the extent to which the visual experience is being preserved. Regardless of how that visual experience is measured, we have chosen to use $\mathcal{J}$, the mean squared error between $(\widehat{\mathbf{I}}_l, \widehat{\mathbf{I}}_r)$ and $(\mathbf{I}_l, \mathbf{I}_r)$, 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:

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

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:

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

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:

$$\left\{\begin{array}{lcl} \mathbf{d}(q_l, q_r) & = & \underset{\mathbf{d}\in\mathbf{S}^K}{\arg\min}\ J(\widehat{\mathbf{I}}_r(q_l, q_r, \mathbf{d}), \mathbf{I}_r) \\ (q_l, q_r) & = & \underset{q_l\in\mathbf{Q}_l,\ q_r\in\mathbf{Q}_r,\ b\leq b_a}{\arg\min}\ \mathcal{J}\left(\widehat{\mathbf{I}}_l(q_l), \mathbf{I}_l, \widehat{\mathbf{I}}_r(q_l, q_r, \mathbf{d}(q_l, q_r)), \mathbf{I}_r\right) \end{array}\right. \tag{11}$$

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) can be recast into:

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

Algorithm 1 BM algorithm

Input: $\mathbf{I}_l, \mathbf{I}_r, q_l, q_r$

Output: $C_{q_l}(\mathbf{I}_l), C(\mathbf{d}), C_{q_r}(\mathbf{R}), b, \mathcal{J}$

  Compute $C_{q_l}(\mathbf{I}_l)$, $\widehat{\mathbf{I}}_l$ with (1) and $J(\widehat{\mathbf{I}}_l, \mathbf{I}_l)$ with (8) and (9)

  for all $k \in \{1 \dots K\}$ **do**

    for all $d \in \mathbf{S}$ **do**

      Compute the k-block of $\mathbf{I}_p$ with (2) and $J_k(\mathbf{I}_p, \mathbf{I}_r)$ with (8)

    end for

    Select d_k with (12) minimizing $J_k(\mathbf{I}_p, \mathbf{I}_r)$

  end for

  Collect $\mathbf{d} = (d_1, \dots, d_K)$ and compute $C(\mathbf{d})$

  Compute $\mathbf{I}_p$ with (2), and $\mathbf{R}$ and $C_{q_r}(\mathbf{R})$ with (5)

  Compute $\widehat{\mathbf{I}}_r$ with (6) and $J(\widehat{\mathbf{I}}_r, \mathbf{I}_r)$ with (8) and (9)

  Compute $\mathcal{J}$ with (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 (7) using $C_{q_l}(\mathbf{I}_l), C(\mathbf{d}), C_{q_r}(\mathbf{R})$

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

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

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 Eq. (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. Algorithm 1 summarizes the DCC algorithm using BM strategy (to simplify the presentation, the greedy selection of q_l and q_r is not shown here). Note that from then on, this DCC algorithm is referred to as BM algorithm.

4 Disparity compensated block matching algorithm, a greedy algorithm

This section presents the Disparity Compensated Block Matching (DCBM) algorithm already developed in Ref. 24. 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.

The DCBM 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:

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

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 t^{th} block and $d_k(t-1, q_l, q_r)$ for all other blocks:

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

with $(\Delta\mathrm{i}, \Delta\mathrm{j})$ spanning $\mathcal{B}$ 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:

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

The DCBM algorithm is summarized in algorithm 2.

Algorithm 2 DCBM algorithm

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

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.

5 Fast DCBM algorithm, a suboptimal algorithm with reasonable complexity

Due to the interesting performance of the DCBM algorithm in terms of rate-distortion (see [24]), 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. This section is organized as follows. Considering first the case of blocks having a size of 8×8, subsection 5.1 shows that only 8×8 pixel-values of $\mathbf{R}$ are to be taken into account. Subsection 5.2 derives from the JPEG-codec an explicit formula using these 8×8 pixel-values. Subsection 5.3 derives the FDCBM algorithm. Subsection 5.4 extends this algorithm to some larger blocks.

5.1 FDCBM algorithm underlying idea

This section considers that the size of $\mathcal{B}$ 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:

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

So as to be consistent with notations defined in Sec. 2, indexes of these 8×8 matrices start from 0: $\Delta\mathrm{i}, \Delta\mathrm{j} \in \{0, \dots 7\}$. 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 $\mathbf{R}_k$:

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

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 *Using the JPEG-codec strategy*

Of the JPEG-codec, this section is only interested in what causes distortions, namely the quantization of the DCT-components:

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

where Q_{q_r} is the $8{\times}8$-JPEG-quantizier.

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

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

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

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

The explicit formula uses the following information extracted from the JPEG-codec (see[25]).

The DCT of an 8×8 matrix is:

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

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

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

The JPEG quantization table is:

$$\mathcal{Q} = \begin{bmatrix} 16 & 11 & 10 & 16 & 24 & 40 & 51 & 61 \\ 12 & 12 & 14 & 19 & 26 & 58 & 60 & 55 \\ 14 & 13 & 16 & 24 & 40 & 57 & 69 & 56 \\ 14 & 17 & 22 & 29 & 51 & 87 & 80 & 62 \\ 18 & 22 & 37 & 56 & 68 & 109 & 103 & 77 \\ 24 & 35 & 55 & 64 & 81 & 104 & 113 & 92 \\ 49 & 64 & 78 & 87 & 103 & 121 & 120 & 101 \\ 72 & 92 & 95 & 98 & 112 & 100 & 103 & 99 \end{bmatrix}. \tag{27}$$

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

$$Q_{q_r}(\mathbf{I}) = \left[\text{Round}\left(\frac{\mathbf{I}(\Delta\mathrm{i},\Delta\mathrm{j})}{\mathcal{Q}(\Delta\mathrm{i},\Delta\mathrm{j})\alpha(q_r)}\right)\mathcal{Q}(\Delta\mathrm{i},\Delta\mathrm{j})\alpha(q_r)\right]_{\Delta\mathrm{i},\Delta\mathrm{j}} \tag{28}$$

using a nonlinear mapping transforms q_r into a scaling factor (see[26]):

$$\alpha(Q) = \begin{cases} \frac{50}{Q} & \text{if } Q \leq 50 \\ 2 - \frac{Q}{50} & \text{if } Q > 50 \end{cases} \tag{29}$$

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

$$\tilde{J}_k\left(q_l, q_r, s\right) = \|Q_{q_r}\left(\mathbb{DCT}\left[\mathbf{R}_k\right]\right) - \mathbb{DCT}\left[\mathbf{R}_k\right]\|^2. \tag{30}$$

Finally the k-block disparity is selected as:

$$d_k(q_l, q_r) = \underset{s \in \mathbf{S}}{\arg\min}\ \tilde{J}_k(q_l, q_r, s). \tag{31}$$

5.3 Derived FDCBM algorithm

The FDCBM algorithm is summarized in algorithm 3. It is very similar to the DCBM algorithm, the difference is inside the double loop. Instead of computing large scale images, only $8{\times}8$-matrices are computed and yield $\tilde{J}_k(q_l, q_r, s)$, an approximation of $J_k(\widehat{\mathbf{I}}_r, \mathbf{I}_r)$ using Eq. (30). 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 definitely much lower than that of DCBM. It remains higher than BM algorithm, not only because of the complexity of Eq. (30) but also because it takes into account q_l and q_r, whereas BM takes into account only q_l.

Algorithm 3 FDCBM algorithm

Input: $\mathbf{I}_l, \mathbf{I}_r, q_l, q_r$
Output: $C_{q_l}(\mathbf{I}_l), C(\mathbf{d}), C_{q_r}(\mathbf{R}), b, \mathcal{J}$
  Compute $C_{q_l}(\mathbf{I}_l)$, $\widehat{\mathbf{I}}_l$ with (1) and $J(\widehat{\mathbf{I}}_l, \mathbf{I}_l)$ with (8) and (9)
  for all $k \in \{1 \ldots K\}$ **do**
    for all $s \in$ **S do**
      Compute $\mathbf{R}_k$ using $\widehat{\mathbf{I}}_l$ and $\mathbf{I}_r$ with (19), (2) and (5)
      Compute $\tilde{J}_k(q_l, q_r, s)$ with (30)
    end for
    Select d_k with (31) using all s-values of $\tilde{J}_k(s)$
  end for
  Collect $\mathbf{d} = (d_1, \ldots, d_K)$ and compute $C(\mathbf{d})$
  Compute $\mathbf{I}_p$ with (2) using $\mathbf{d}$
  Compute $\mathbf{R} = \mathbf{I}_r - \mathbf{I}_p$ and $C_{q_r}(\mathbf{R})$ with (5)
  Compute $\widehat{\mathbf{I}}_r$ with (6) and $J(\widehat{\mathbf{I}}_r, \mathbf{I}_r)$ with (9)
  Compute $\mathcal{J}$ with (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 (7) using $C_{q_l}(\mathbf{I}_l), C(\mathbf{d}), C_{q_r}(\mathbf{R})$

5.4 *Extending the FDCBM algorithm to larger blocks*

This section considers the case when the block decomposition yielding the disparity map is not the same than the JPEG-block decomposition. To distinguish them, the former is denoted $\mathcal{B}_k$, ($1 \le k \le K$, $\mathcal{B}$ as the set of internal displacements), the latter is denoted $\mathcal{B}'_{k'}$, ($1 \le k' \le K'$, $\mathcal{B}'$ as the set of internal displacements). In general, a block $\mathcal{B}_k$ is likely to have common pixels with several blocks $\mathcal{B}'_{k'}$ and each of these blocks may have common pixels with other blocks $\mathcal{B}_{k''}$. In such a situation, the optimal choice of a disparity d_k depends on the choice of disparities of neighboring blocks, and adapting the FDCBM algorithm seems difficult. Here we assume that each block $\mathcal{B}_k$ can be divided exactly in a finite number of blocks $\mathcal{B}'_{k'}$, and show how FDCBM can easily be extended. For instance when an image is decomposed into 16×16-blocks, each of them covers exactly four 8×8-blocks. And when an image is decomposed into 32×32-blocks, each of them covers exactly sixteen 8×8-blocks.

Let $\mathcal{K}_k$ be the set of indexes indicating the blocks $\mathcal{B}'_{k'}$ that are exactly covering $\mathcal{B}_k$:

$$\mathcal{B}_k = \bigcup_{k' \in \mathcal{K}_k} \mathcal{B}'_{k'}. \tag{32}$$

The size of $\mathcal{K}_k$ can be computed:

$$\begin{cases} K|\mathcal{B}_k| & = & K'|\mathcal{B}'_{k'}| \\ |\mathcal{B}_k| & = & |\mathcal{K}_k||\mathcal{B}'_{k'}| \end{cases} \Rightarrow |\mathcal{K}_k| = \frac{K'}{K}, \tag{33}$$

where $|\ |$ indicates the size of a set. The first equation is derived from the fact that each view of the considered stereoscopic image is divided into a set of non-overlapping blocks. The second equation is derived from Eq. (32).

The mean squared error defined in Eq. (8) has now two definitions depending on the considered block-decomposition:

$$\begin{array}{rcl} J_k\left(\widehat{\mathbf{I}}, \mathbf{I}\right) & = & \frac{1}{|\mathcal{B}|}\sum_{(i,j)\in\mathcal{B}_k}\left(\widehat{\mathbf{I}}(i,j) - \mathbf{I}(i,j)\right)^2, \\ J'_{k'}\left(\widehat{\mathbf{I}}, \mathbf{I}\right) & = & \frac{1}{|\mathcal{B}'|}\sum_{(i,j)\in\mathcal{B}'_{k'}}\left(\widehat{\mathbf{I}}(i,j) - \mathbf{I}(i,j)\right)^2. \end{array} \tag{34}$$

Equation (32) yields a relationship between the two mean squared error functions:

$$J_k\left(\widehat{\mathbf{I}}, \mathbf{I}\right) = \frac{1}{|\mathcal{K}_k|}\sum_{k' \in \mathcal{K}_k} J'_{k'}\left(\widehat{\mathbf{I}}, \mathbf{I}\right). \tag{35}$$

Subsections 5.1 and 5.2 are applied to the JPEG-block decomposition, resulting in the following approximation:

$$J'_{k'}\left(\widehat{\mathbf{I}}_r, \mathbf{I}_r\right) \approx \tilde{J}_{k'}(q_l, q_r, s), \tag{36}$$

where s is the k'-block disparity in the sense of the JPEG-block decomposition.

Eqs. (35) and (36) yield the k-block disparity:

$$d_k(q_l, q_r) = \underset{s \in \mathbf{S}}{\arg\min} \frac{1}{|\mathcal{K}_k|} \sum_{k' \in \mathcal{K}_k} \tilde{J}_{k'}(q_l, q_r, s). \tag{37}$$

This is the proposed extended FDCBM algorithm.

6 Performance evaluation of the proposed FDCBM algorithm

This section analyzes and discusses the simulation results of the developed FDCBM algorithm using synthetic data and stereoscopic image datasets.

6.1 Impact on the performance of the proposed explicit formula

First of all, this section proposes to discuss the relevance of the derived Eq. (30) on which the proposed FDCBM algorithm is based. To do so, experiments are conducted on synthetic data to measure the ability of this equation to reduce distortions more than the BM algorithm.

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

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

Moreover, the experiment shows that most often:

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

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

$$\rho(q_r) = \frac{1}{200}\sum_{\omega=1}^{200} \frac{J_k(q_r, d_{\text{BM}}(q_r,\omega),\omega) - J_k(q_r, d_{\text{FDCBM}}(q_r,\omega),\omega)}{J_k(q_r, d_{\text{BM}}(q_r,\omega),\omega) - J_k(q_r, d_{\text{DCBM}}(q_r,\omega),\omega)}. \tag{38}$$

Fig. 2 illustrates the behaviour of $\rho(q_r)$ when q_r ranges from 1 to 100. It shows that 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.

6.2 *Simulation results on stereoscopic images*

Simulation results are performed on stereoscopic images downloaded from the Middleburry dataset[5] and 3D LIVE dataset.[27] A performance comparison of the FDCBM algorithm with BM algorithm is discussed. Measuring the true performance of an algorithm means evaluating the average visual experience provided by the compressed stereoscopic image at a given bitrate. Subjective evalu-

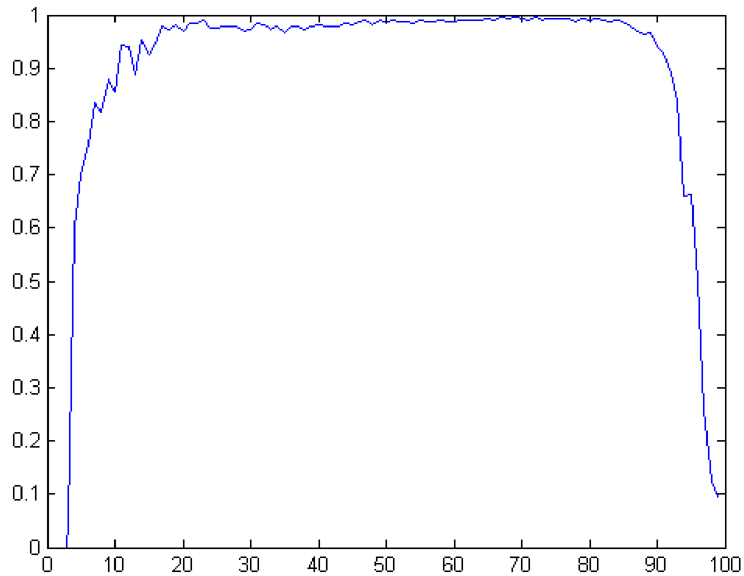

Fig 2 Average distortion reduction ratio of BM-FDCBM as compared to BM-DCBM on synthetic data as a function of q_r.

ation is the most accurate technique and the most demanding. Evaluating with objective quality metrics is a much easier and the design of such metrics is an existing research field as exemplified in Ref. 28. As of now, no objective quality metric has proven to be completely reliable when applied to stereoscopic images.

In this paper, the distortion is measured using the Peak Signal to Noise Ratio (PSNR) using a dB scale. To simplify the experiment, the left view is not compressed, the rate-distortion is measured only on the right view as follows:

$$\begin{aligned} PSNR &= 10\log_{10}\left(\frac{255^2}{J(\widehat{\mathbf{I}}_r,\mathbf{I}_r)}\right), \\ b &= \frac{|C(\mathbf{d})|+|C_{q_r}(\mathbf{R})|}{|\mathbf{I}_r|}, \end{aligned} \tag{39}$$

where b is in bits per pixel (bpp); and pixel-values, on both views, are ranging from 0 to 255.

The lossless coder, C is here an arithmetic coder (see[29]). To reduce the numerical complexity,

Fig 3 Original right view of the "Art" stereoscopic image.

the set of quality factor values is reduced to $\mathbf{Q}_r = \{5, 10, 15, \dots, 90\}$. The set of all available disparities is $\mathbf{S} = \{0, \dots, 120\}$. This choice (on the selected dataset) ensures better performance than when each of the views is encoded in an independent way.

As for the sizes of blocks fixed in the disparity prediction process, we use 8×8, 16×16 and 32×32 blocks. Note that both views are always decomposed into non-overlapping blocks of same size.

The rate-distortion curves, in Fig. 4 confirm the results stated above (subsection 6.1) using "Art" stereoscopic image (original right view is provided by Fig. 3) of 2006 Middlebury-dataset and blocks of size 8×8. Indeed the performance of the proposed FDCBM algorithm is similar to that of DBCM algorithm, which is however better than that of the classical BM algorithm.

Fig. 5 provides the original right view of the "Aloe" stereoscopic image extracted from the 2006 Middlebury-dataset. Fig. 6 presents the compressed and decompressed "Aloe" right view using BM algorithm on the left 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

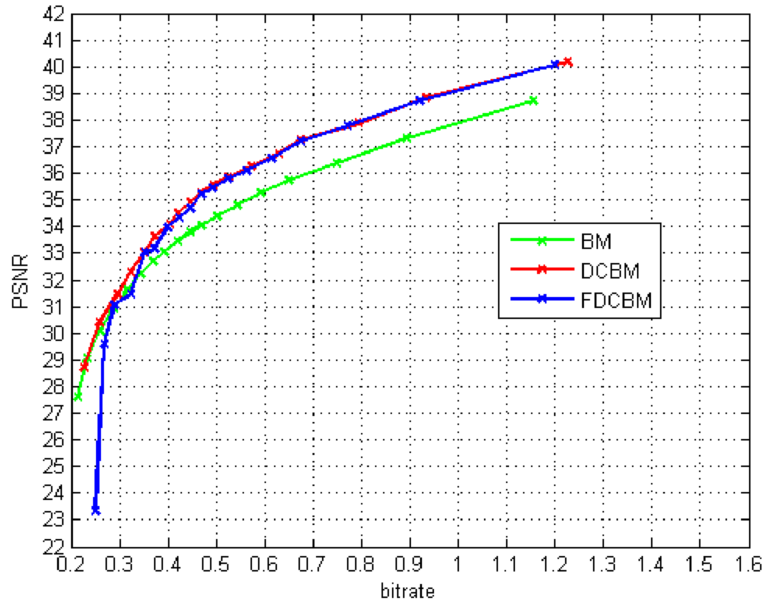

Fig 4 Performance comparison of BM, DCBM and FDCBM algorithms using "Art" stereoscopic image of 2005 Middlebury-dataset.

with the original, it appears that the background cloth on right neighborhoods of each vertical leaf is wrongly drawn. The reason may be that these neighborhoods are occluded in the left view. The BM algorithm yields 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 (30.14dB) than the BM-reconstructed view (29.5dB).

Fig. 7 shows the histograms of, on the left side, the BM-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 proved 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

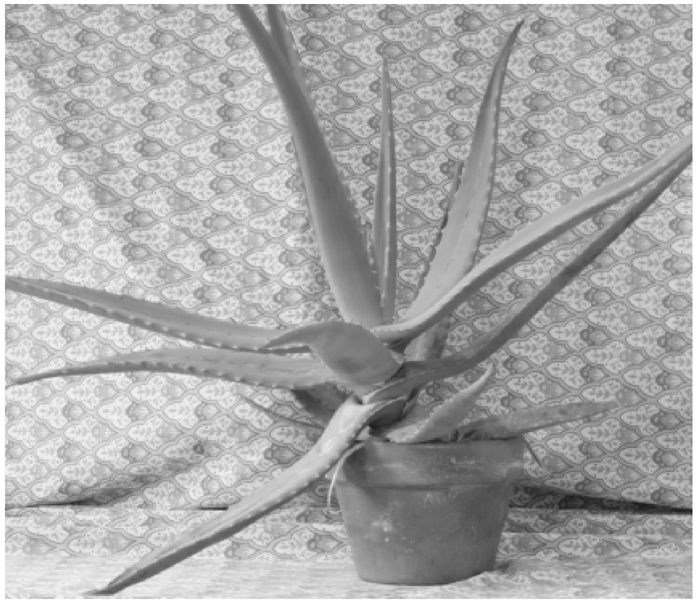

Fig 5 Original right view of the ”Aloe” stereoscopic image.

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 algorithm.

As for numerical complexity, FDCBM algorithm (consuming 17 seconds) is 3 388 times quicker than DCBM algorithm (consuming 4 hours) and 6.8 times slower than BM algorithm (consuming 2.5 seconds). This was 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 3.7GHz.

Simulation results provided in Fig. 8 have been conducted on ”Art” stereoscopic image (original right view is shown in Fig. 3). FDCBM algorithm is implemented with different q_r values yielding three different rate-distortion curves shown on the left side. The green and right-most curve is obtained with $8{\times}8$ blocks. The red and middle curve is obtained with $16{\times}16$ blocks. The blue and left-most curve is obtained with $32{\times}32$ blocks. As we can see, using blocks of greater size, reduces the bitrate and tends to move the rate-distortion curve leftwards and downwards.

When addressing a specific need, in terms of expected PSNR or of allowed bitrate, it makes

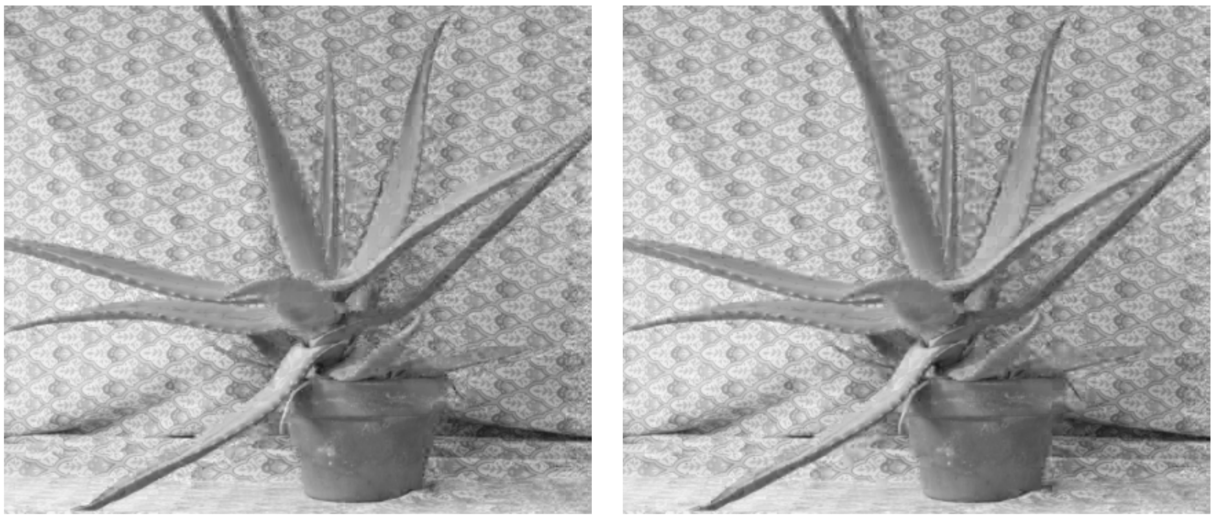

Fig 6 On the left side: reconstructed "Aloe" right view with BM algorithm at $b = 0.3$bpp ($PSNR = 29.5$dB); On the right side: reconstructed "Aloe" right view with FDCBM algorithm at $b = 0.3$bpp ($PSNR = 30.14$dB).

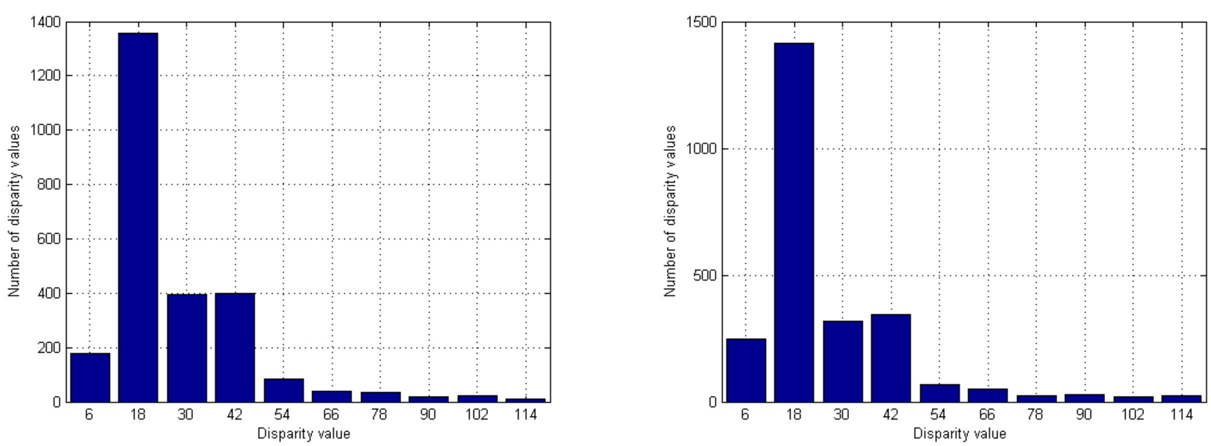

Fig 7 Histogram of the disparity map yielded at $b = 0.3$bpp using: on the left side, the BM-algorithm; and, on the right side, the FDCBM-algorithm.

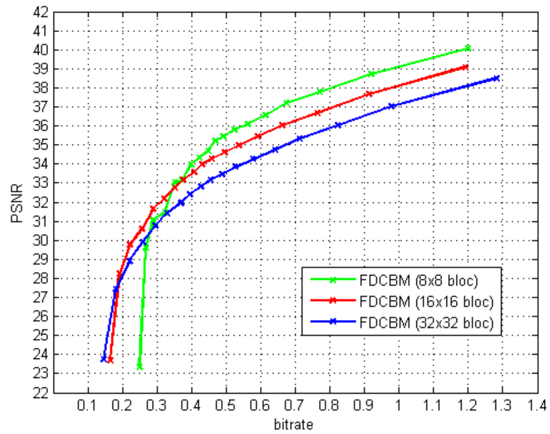

Fig 8 FDCBM rate-distortion curves for the ”Art” stereoscopic image, using 8×8-blocks (green and left-most curve), 16×16-blocks (red and middle curve), 32×32-blocks (blue and right-most curve); and BM (red and bottom curve).

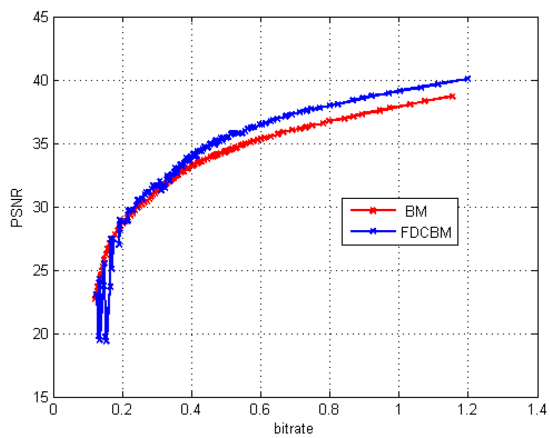

Fig 9 Higher convex envelop curve generated with the three-block-size, rate-distortion curves, using FDCBM (blue and top curve).

sense to use the most appropriate block size to meet the request. So when comparing BM with FDCBM, it is more relevant to compare, the two higher convex envelopes generated by each three rate-distortion curves, than to compare the rate-distortion curves one by one. These two envelopes are illustrated in Fig. 9 where the blue top curve is obtained with FDCBM algorithm and the red bottom curve is obtained with BM algorithm. At low bitrate (below 0.2bpp), BM is better performing, whereas at higher bitrate, FDCBM algorithm is better performing.

The Bjøntegaard metric[30] is used here to quantify the increase in performance of FDCBM as compared to BM. 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 0.78dB.

All 30 stereoscopic images extracted from the 2005 and 2006 Middlebury-database are compressed and decompressed using BM and FDCBM algorithms and using blocks of sizes 8×8, 16×16 and 32×32. For each stereoscopic image, two corresponding higher convex envelopes are computed. Four rate-distortions points from each envelope are extracted and used by the Bjøntegaard metric to provide an average PSNR increase, or, an average bitrate decrease. These measures are shown in Table 1. To simplify its reading, the stereoscopic images have been sorted by their increase in PSNR-performance. This table shows that, on average, for all stereoscopic images, FDCBM is better performing than BM, the difference ranges from 0.17dB up to 1.28dB. 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 (+0.17dB), namely ”Plastic”, is having a rather important bitrate decrease (-15.7%). Table 2 provides other simulation results performed on 3D LIVE database .[27]

The increase in performance of all 48 stereoscopic images of FDCBM as compared to BM is also shown in Fig. 10. More specifically, 10 bins have been considered ranging from $+0.25$dB up to $+1.25$dB in terms of increase in terms of PSNR performance of FDCBM as compared to BM.

Table 1 Performance comparison between FDCBM and BM algorithms using the Bjøntegaard metric and 2005 and 2006 Middlebury-database.

Image	ΔPSNR (dB)	bpp (%)
Plastic	+0.17	-15,73
Cloth3	+0.31	-7,44
Midd1	+0.31	-4,79
Cloth1	+0.34	-9,08
Laundry	+0.37	-5.66
Computer	+0.39	-6.3
Baby1	+0.39	-8.17
Baby2	+0.42	-9.35
Wood1	+0.43	-8.9
Rocks2	+0.44	-9.89
Books	+0.46	-7.93
Aloe	+0.53	-11.7
Lampshade1	+0.54	-5.09
Rocks1	+0.56	-12.43
Bowling2	+0.57	-9.86
Midd2	+0.58	-10.51
Drumsticks	+0.58	-8.85
Dolls	+0.59	-10.65
Moebius	+0.65	-11.44
Cloth2	+0.66	-12.68
Baby3	+0.67	-12.74
Wood2	+0.72	-9.34
Monopoly	+0.75	-13.56
Cloth4	+0.76	-17.12
Art	+0.78	-11.36
Bowling1	+0.89	-14.47
Dwarves	+1.06	-17.48
Flowerpots	+1.12	-14.52
Lampshade2	+1.23	-22.76
Mean	+0.62	-11.38

Each bar is associated to a specific bin, and its height indicates the number of stereoscopic images, having an increase in PSNR performance of roughly the amount indicated on the bin. The average increase in PSNR performance is 0.54dB.

Table 2 Performance comparison between FDCBM and BM algorithms using the Bjøntegaard metric and 3D LIVE-database.

Image	ΔPSNR (dB)	bpp (%)
im20	+0.18	-7,32
im8	+0.21	-5,85
im29	+0.22	-7.06
im13	+0.25	-7.43
im22	+0.29	-9.48
im18	+0.34	-7.52
im3	+0.36	-11.14
im14	+0.37	-12.23
im16	+0.38	-11.47
im21	+0.42	-15.13
im7	+0.45	-9.47
im17	+0.51	-17.75
im15	+0.53	-15.92
im10	+0.55	-14.22
im24	+0.58	-14.22
im26	+0.61	-17.23
im5	+0.62	-18.26
im27	+0.73	-19.38
Mean	+0.42	-12.74

7 Conclusion

A new block-based disparity estimation technique called FDCBM algorithm for Fast Disparity Compensated Block Matching strategy is proposed. 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.

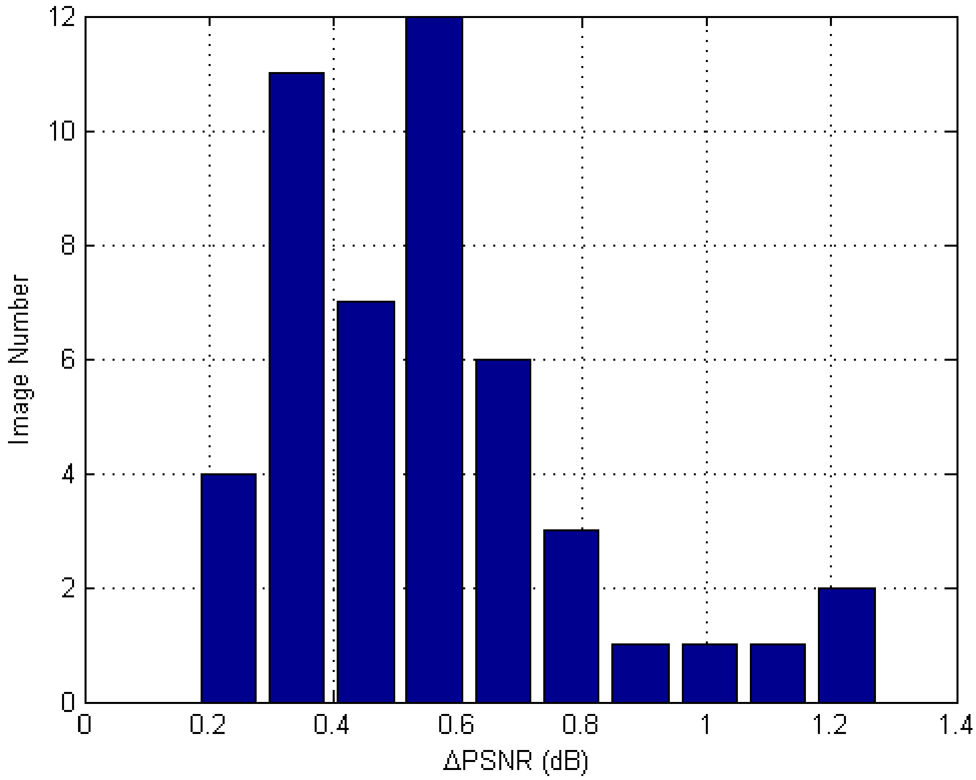

Fig 10 Histogram of the increase in average PSNR-performance of FDCBM as compared to BM, in terms of the number of stereoscopic images among the 30 extracted from the 2005 and 2006 Middlebury-database.

Tested on the 48 stereoscopic images extracted from the 2005-2006 Middelbury-dataset and 3D LIVE-dataset, FDCBM algorithm is performing better than the classical Disparity Compensated Compression algorithm using a Block-Matching disparity estimation technique. As compared to the latter, the increase in performance, at same bitrate, is ranging, depending on the stereoscopic image, from 0.18dB up to 1.3dB with an average of 0.54dB, (performance being here measured using the Peak Signal to Noise Ratio).

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 to 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. Of course, blocks of variable size give better performance and will be investigated in the near future.

References

1 C. Soares and E. Simao, *Immersive Multimedia in Information Revolution. Trends, Experiences, and Perspectives in Immersive Multimedia and Augmented Reality*, IGI Global, Pennsylvanie, United States (2019).

2 R. Corda, D. Giusto, A. Liotta, *et al.*, "Recent advances in the processing and rendering algorithms for computer-generated holography," *Electronics* **8**(5), 1–17 (2019).

3 K. Nam, P. Anh-Hoang, E. Munkh-Uchral, *et al.*, "3D Display Technology," *Display and Imaging* **1**, 73–95 (2013).

4 Y. Frauel, E. Tajahuerce, O. Matoba, *et al.*, "Comparison of passive ranging integral imaging and active imaging digital holography for three-dimensional object recognition," *Applied Optics* **43**, 452–462 (2004).

5 D. Scharstein and C. Pal, "Learning conditional random fields for stereo," in *IEEE Conference on Computer Vision and Pattern Recognition*, 1–8 (2007). http://vision.middlebury.edu/stereo/data/.

6 F. Dufaux, B. Pesquet-Popescu, and M. Cagnazzo, *Emerging Technologies for 3D Video: Creation, Coding, Transmission and Rendering*, Wiley Publishing, 1st ed. (2013).

7 P. F. M. B. Schenkel, C. Luo and F. Wu, "Joint decoding of stereo jpeg image pairs," in *IEEE International Conference on Image Processing*, 2633–2636 (2010).

8 A. Ortis and S. Battiato, “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, 2015*, 93930K (2015).

9 U. Ahlvers, U. Zölzer, and S. Rechmeier, “Fft-based disparity estimation for stereo image coding,” *Proceedings 2003 International Conference on Image Processing (Cat. No.03CH37429)* **1**, I–761 (2003).

10 H. Schwarz, C. Bartnik, S. Bosse, *et al.*, “3D video coding using advanced prediction, depth modeling, and encoder control methods,” in *Picture Coding Symposium*, 1–4 (2012).

11 Shiping Li, Mei Yu, Gangyi Jiang, *et al.*, “Approaches to h.264-based stereoscopic video coding,” in *Third International Conference on Image and Graphics (ICIG’04)*, 365–368 (2004).

12 P. Hanhart, M. Rerabek, P. Korshunov, *et al.*, “Subjective evaluation of HEVC intra coding for still image compression,” tech. rep., [JCT-VC contribution] AhG4 (2013).

13 W. Woo and A. Ortega, “Stereo image compression with disparity compensation using the MRF model,” in *Visual Communications and Image Processing*, **2727**, 1–14 (1996).

14 M. Flierl, A. Mavlankar, and B. Girod, “Motion and disparity compensated coding for multiview video,” *IEEE Transactions on Circuits and Systems for Video Technology* , 1474–1484 (2007).

15 A. Kadaikar, G. Dauphin, and A. Mokraoui, “Sequential block-based disparity map estimation algorithm for stereoscopic image coding,” *Elsevier journal, Signal Processing: Image Communication* (2015).

16 A. Kadaikar, G. Dauphin, and A. Mokraoui, “Joint disparity and variable size-block opti-

mization algorithm for stereoscopic image compression," *Elsevier journal, Signal Processing:Image Communication* (2017).

17 G. Dauphin, M. Kaaniche, and A. Mokraoui, "Block dependent dictionary based disparity compensation for stereo image coding," in *IEEE International conference on image processing, ICIP*, 1–5 (2015, Québec City Canada).

18 G. Shen, W. Kim, A. Ortega, *et al.*, "Edge-aware intra prediction for depth-map coding," in *2010 IEEE International Conference on Image Processing*, 3393–3396 (2010).

19 Y. Chen, M. M. Hannuksela, L. Zhu, *et al.*, "Coding techniques in multiview video coding and joint multiview video model," in *2009 Picture Coding Symposium*, 1–4 (2009).

20 T. Frajka and K. Zeger, "Residual image coding for stereo image compression," *Optical Engineering* **42**, 182–189 (2003).

21 G. K. Wallace, "The jpeg still picture compression standard," *IEEE Trans. on Consum. Electron.* **38**, xviii–xxxiv (1992).

22 W. Hachicha, M. Kaaniche, A. Beghdadi, *et al.*, "Efficient inter-view bit allocation methods for stereo image coding," *IEEE Transactions on Multimedia* **17**, 765–777 (2015).

23 R. Pan, Z.-X. Hou, and Y. Liu, "Fast algorithms for inter-view prediction of multiview video coding," *Journal of Multimedia* **6**, 191–201 (2011).

24 I. Kadri, G. Dauphin, and A. Mokraoui, "Stereoscopic image coding performance using disparity-compensated block matching algorithm," in *IEEE International Conference on Signal Processing: Algorithms, Architectures, Arrangements, and Applications, SPA*, 1–5 (2019).

25 CCIT, “Information technology-digital compression and coding of continuous-tone still images-requirements and guidelines,” Tech. Rep. T.81, The International Telegraph and Telephone Consultative Committee CCIT (1992).

26 L. Alam, P. K. Dhar, M. R. Hasan, *et al.*, “An improved JPEG image compression algorithm by modifying luminance quantization table,” *International Journal of Computer Science and Network Security (IJCSNS)* **17**, 200–208 (2017).

27 http://live.ece.utexas.edu/research/quality/live_3dimage_phase1.html.

28 A. Benoit, P. L. Callet, P. Campisi, *et al.*, “Using disparity for quality assessment of stereoscopic images,” in *15th IEEE International Conference on Image Processing*, *15th IEEE International Conference on Image Processing* , 389 – 392 (2008).

29 P. G. Howard and J. Vitter, “Arithmetic coding for data compression,” *Proceedings of the IEEE* **82**, 857 – 865 (1994).

30 G. Bjøntegaard, “Calculation of average PSNR differences between RD-curves,” document VCEG-M33, ITU-T VCEG Meeting, Austin, Texas, USA (2001).