Patch-Based Conditional Context Coding of Stereo Disparity Images

Ioan Tabus, *Senior Member, IEEE*

***Abstract*—This letter proposes a method for lossless coding the left disparity image, L, from a stereo disparity image pair (L,R), conditional on the right disparity image, R, by keeping track of the transformation of the constant patches from R to L. The disparities in R are used for predicting the disparities in L, and the locations of the pixels where the prediction is erroneous are encoded in a first stage, conditional on the patch-labels of R image, allowing the decoder to already reconstruct with certainty some elements of the L image, e.g., the disparity values at certain pixels and parts of the contours of left image patches. Second, the contours of the patches in L image that are still unknown after first stage are conditionally encoded using a mixed conditioning context: the usual causal current context from the contours of L and a noncausal context extracted from the contours in the correctly estimated part of L obtained in the first stage. The depth values in the patches of L image are finally encoded, if they are not already known from the prediction stage. The new algorithm, dubbed conditional crack-edge region value (C-CERV), is shown to perform significantly better than the non-conditional coding method CERV and than another existing conditional coding method, over the Middlebury corpus. C-CERV is shown to reach lossless compression ratios of 100-250 times for those images that have a high precision of the disparity map.**

***Index Terms*—Arithmetic coding, context tree coding, inter-coding, lossless disparity image compression.**

I. Introduction

THE compression of a pair (L, R) of stereo disparity images has the potential of reaching very high compression ratios, since each disparity image contains information about how its pixels should be translated in order to obtain a partial reconstruction of the other disparity image of the pair. In this paper R is considered to be known and we concentrate on the efficient encoding of L, conditional on R.

The stereo matching problem, of finding the left and right disparity images given a pair of two color stereo images is one of the classical problems in image processing and computer vision, for which numerous algorithms were proposed along several decades. The site http://vision.middlebury.edu/stereo/eval/ ranks the performance of 150 stereo matching algorithms over the Middlebury database [1], which provides also ground truth for the (L, R) pairs.

The lossless compression of a single disparity image or of a depth map has received many solutions: bit-plane decomposition was combined with binary shape coding in [2]; an "intra" method based on bitplane decomposition was proposed in [3]; encoding the contours of large regions, followed by predictive coding inside each region was used in [4]; modifications of the lossless modes of existing standards were proposed in [5][6]. The "intra" method that is used here for encoding the right image is CERV [7], which serves also as the starting point for the newly proposed conditional coding. We use most of notations and part of the encoding strategy from CERV. The same representation of the image into piecewise constant regions (called here patches) and coding of the patch contours as in CERV was first used in [8], for indexed palette images.

The lossy coding of depth maps was studied even more intensively. The early good performers were based on platelets [9] and region based coding [10]. Especially related with the method proposed in this letter are the encoding with piecewise linear approximations of the contours [11] and the lossy methods using region merging and splitting [12]. Scalable lossy coding was introduced based on a scalable description of the geometry and subband coding [13][14]. Other lossy coding methods for depth maps were presented in [15]–[19] [20][21].

The problem studied here, that of lossless encoding one image in a pair of disparity images conditional on the other image, has received one solution earlier in [3], utilizing a prediction stage based on horizontal translation of the pixels in the known disparity image and additionally using median filter for filling some small gaps, while the entropy coding stage involves the decomposition of the images in binary planes.

In the method C-CERV presented in this letter, a warped image is obtained by horizontal translation, similarly to the prediction in [3], however median prediction is not used. A binary image is transmitted in a first encoding stage, to signal the location of pixels where prediction is correct. In the second encoding stage, the contours of the patches in the L image are transmitted by mixing the prediction from the contours encoded so far with the prediction from the contours of the patches in the warped image.

The C-CERV method is described in Section II. Section III presents experimental results and Section IV draws conclusions.

II. Description of the Method

This section presents the encoding procedure structured in three stages, $\mathcal{P}$, $\mathcal{C}$, and $\mathcal{Y}$. The detailed pseudocode of the encoder and decoder is presented in the file with additional figures in Figs. 11–15. The right and left disparity images R and L are

Manuscript received May 08, 2014; revised June 07, 2014; accepted June 12, 2014. Date of current version June 18, 2014. The associate editor coordinating the review of this manuscript and approving it for publication was Prof. Yao Zhao.

The author is with the Department of Signal Processing, Tampere University of Technology, Tampere, Finland (e-mail: ioan.tabus@tut.fi; http://www.cs.tut.fi/~tabus/).

Supplemental multimedia files for this paper are available online at http://ieeexplore.ieee.org.

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/LSP.2014.2331107

assumed to be rectified so that the corresponding points in the images have the same i coordinate. The axes of coordinates are shown in 1a). Each image has size $n_r \times n_c$.

Several pixels from the R image, translated by their disparity, may end up on a same pixel (i, j) in the L image. The disparity at (i, j) in L will record only one correspondence, namely to the pixel of coordinates (i, j^r) in R having the largest disparity, i.e. largest difference $j - j^r$. Hence the prediction $L^w(i, j)$ will be the maximum disparity in the set of all pixels from R translating in $L(i, j)$, i.e., $L^w(i, j) = \max\{R(i, j^r) | j^r + R(i, j^r) = j\}$. The predicted image L^w is called here also warped image.

A. Stage $\mathcal{P}$: Encoding the Locations of Correctly Predicted Disparities

The locations of warping errors are encoded, so that the decoder knows the warped disparities that are correct. The binary error image I_L has ones at the location of warping errors and zeros in rest. For transmitting the error locations it is not needed to encode the whole binary $n_r \times n_c$ image I_L, but only the part where warping was possible. In Fig. 1(f) the "inactive" areas marked in blue do not need to be encoded.

Next is presented a method for encoding the black and white parts of the image I_L that takes into account the partition of the warped image into patches. A patch is a set of pixels forming a connected region (in 4-connectivity) having the same disparity value for all pixels in the region (see more formal definitions in [7]). The key for efficient coding of the image I_L is notticing in Fig. 1(g) that the locations of the warping errors are at the boundary of the patches of L^w most of the time. Hence, a more informative context can be obtained about a white pixel of I_L in Fig. 1(f), by considering separately the restriction of I_L over each patch, as in Fig. 1(h).

When encoding the value $I_L(i, j)$ at a pixel from Patch 1, the context is formed of two parts: first, the values of I_L in the causal template of six pixels outlined in cyan in Fig. 1(i) (where all pixels outside Patch 1 are considered to be zero) and then the second part is formed of the six pixels outlined in cyan in Fig. 1(j), extracted from the image indicator of Patch 1. The indicator image of any patch, known also to the decoder, has ones at each pixel of the patch and is zero everywhere else. This 12-pixel context brings information about the location of (i, j) relative to the contour of Patch 1 and about the values of the image I_L at the causal neighbours of (i, j). The context so defined will be used in the optimal context tree pruning process described in SubSection II-C.

A very efficient way to encode big parts of the binary image I_L is to transmit the indices of the patches of R (warped in L^w) which are errorless, by scanning through the string of patch indices k and transmitting $\gamma_k = 0$ for errorless patches and $\gamma_k = 1$ for a patch having at least one error. The binary string is transmitted using arithmetic coding, where the necessary coding distribution is estimated from the counts collected adaptively at the leaves of a the balanced tree having tree-depth equal to one (a first order Markov model).

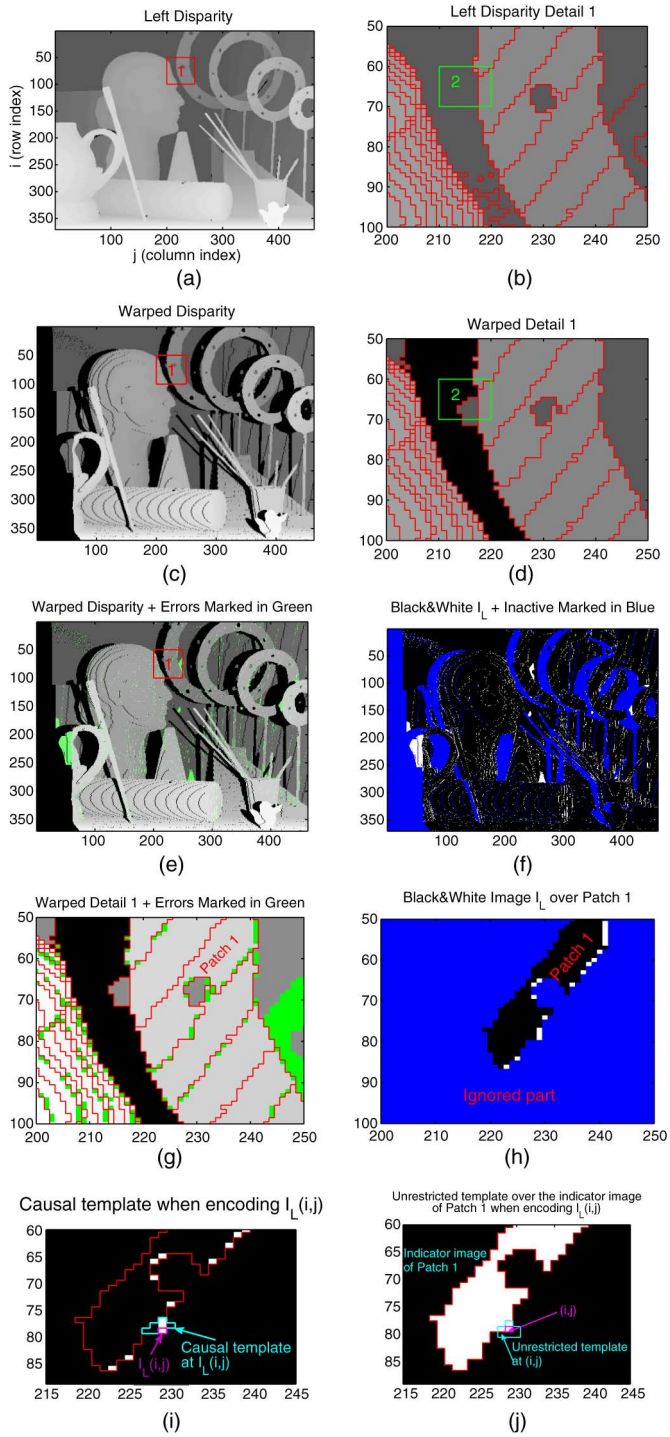

Fig. 1. Images illustrating the processing in Stage $\mathcal{P}$ of the encoding algorithm: (a) and (b) the left disparity L; (c) and (d) the warped image L^w; (e) and (g) marking in green the pixels (i, j) where the warped image is incorrect; (f) The binary image I_L, which is encoded in Stage $\mathcal{P}$, is set to one at the coordinates of the warping errors. The coordinates where no warping occured are marked in blue, they do not need to be encoded; (h) image I_L restricted over one patch from L^w; (i) and (j) the two parts of the context to be used when encoding $I_L(i, j)$: in (i) the causal context is extracted from I_L. In (j) the unrestricted context is extracted from the indicator image of the Patch 1.

B. Stage $\mathcal{C}$: Encoding the Patch Contours in the Left Image

The second encoding stage has the goal of encoding the contours of the patches of the left disparity image L, making use of all information available after Stage $\mathcal{P}$, i.e the warped image L^w and the binary error image I_L. At this stage we set to zero any value $L^w(i, j)$ for which $I_L(i, j) = 1$. Incorrect warping becomes equivalent to non-existent warping.

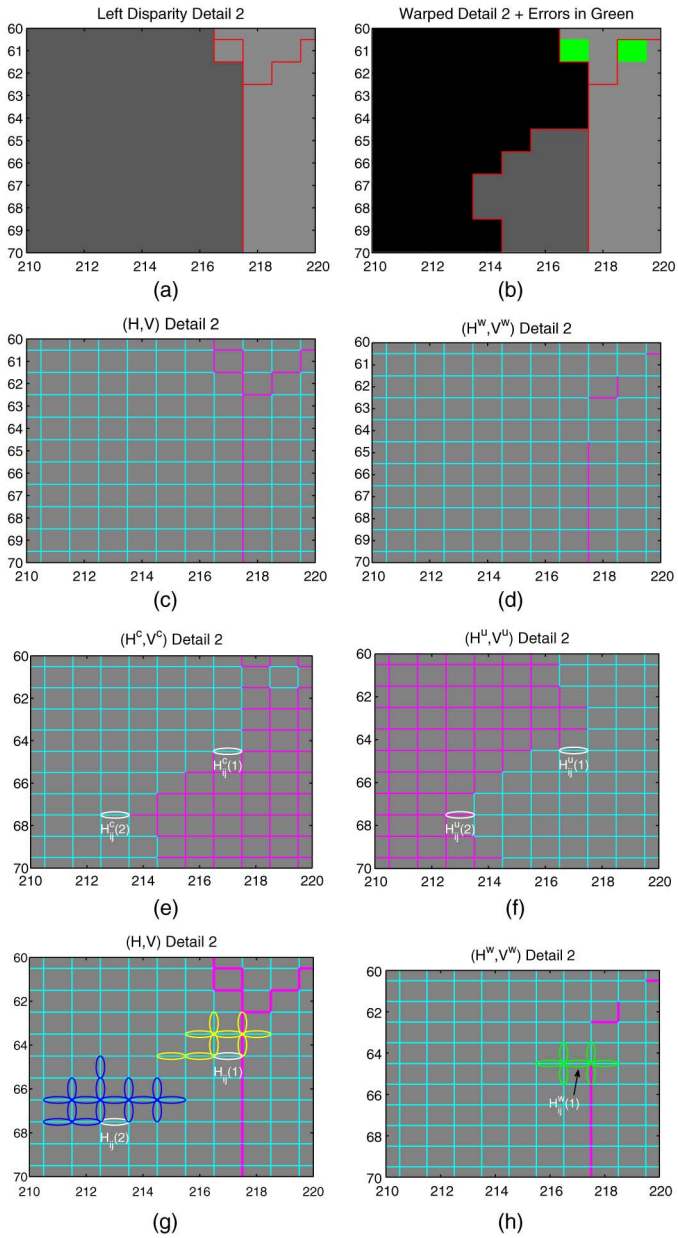

Fig. 2. Stage $\mathcal{C}$. (a) - (b) Detail 2 of L and L^w; (c) - (f) the crack-edge images (H, V), (H^w, V^w), (H^c, V^c), and (H^u, V^u) used in Stage $\mathcal{C}$. When encoding (H, V), the context tree encoding algorithm builds the contexts from a causal template of (H, V), and additionally from a template in the known image (H^w, V^w), controlled by the information from (H^c, V^c), and (H^u, V^u); (g) - (h) context selection examples: the two parts of the context used for encoding the horizontal crack-edges $H_{i,j}(1)$ are marked in yellow and green, while the context for encoding $H_{i,j}(2)$ is marked in blue.

The most efficient way to encode the contours of the patches for a disparity image L was introduced in CERV [7], by considering two matrices of crack-edges, H and V, which are marking the places where neighbor pixels in L are different. In the horizontal crack-edge binary image, $H(i,j) = 1$ indicates that $L(i,j) \neq L(i-1,j)$ and in the vertical crack-edge binary image, $V(i,j) = 1$ indicates that $L(i,j) \neq L(i,j-1)$. The image in Fig. 2(c) is obtained by representing the binary values of the crack-edges with cyan lines for 0 and with magenta lines for 1. The same partition of pixels into patches can be extracted from either of the Figs. 2(a) and 2(c).

In Stage $\mathcal{C}$ of the currently proposed algorithm, encoding of H and V is done in interleaved manner, similar to CERV, by encoding one row from V, and then one row from H, which was shown to be the key for efficient coding in CERV [7]. Additionally the encoding of H and V is done conditionally on the crack-edge images (H^w, V^w), constructed from the warped image L^w. The images (H^w, V^w) are defined similarly to the way (H, V) is computed from L, with the additional constraint that a crack-edge is set to one only if the pixels it separates in L^w are known to be defined and correct, e.g., $H^w(i,j) = 1$ if $L^w(i,j) \neq L^w(i,j-1)$ and $L^w(i,j) > 0$ and $L^w(i,j-1) > 0$. Fig. 2(d) is obtained in this way from 2b).

The Stage $\mathcal{P}$ of the algorithm has operated with the patches of R, which were tracked as contours in L^w after warping R into L^w. These contours of patches in L^w are in general different of the contours of the patches in L, see Fig. 2(a) and Fig. 2(b). Consequently the locations set to one in the crack-edge images (H, V) may be different than the locations set to one in the crack-edge images (H^w, V^w) (compare Figs. 2(c) and 2(d)). However, it pays off to encode (H, V) conditioning on (H^w, V^w), through a mechanism of context tree coding which is the main body of Stage $\mathcal{C}$ and is described next.

Two auxiliary crack-edge *flag* images (H^c, V^c) and (H^u, V^u), built from L^w and I_L, are used for controlling the encoding algorithm. A crack-edge set to one in (H^c, V^c), say $H^c(i,j) = 1$ will signify that the crack-edge $H(i,j)$ needs to be skiped from encoding since $H(i,j) = H^w(i,j)$. In Fig. 2(e) are shown in magenta the "certain" labels, where no encoding is needed and in cyan the locations where the values of (H, V) will have to be encoded.

A crack-edge set to one in (H^u, V^u), say $H^u(i,j) = 1$ will signify that the context from L^w and (H^w, V^w) is not to be trusted in the vicinity of (i,j) when encoding $H(i,j)$, and only a causal context extracted from (H, V) can be used. The flag, $H^u(i,j)$, is taken here to be set "on" when $L^w(i-1,j) = L^w(i,j) = 0$, in which case only the causal context from (H, V) is used. When at least one of $L^w(i-1,j)$ and $L^w(i,j)$ is nonzero, the flag $H^u(i,j)$ is set to "off", and then the mixed context from both (H, V) and (H^w, V^w) is used. Fig. 2(f) shows the image of flags (H^u, V^u).

The contexts used for encoding a vertical and a horizontal crack-edge are different, similar to [7]. In Fig. 2(g) the context at a horizontal crack-edge $H(i,j)$ where $H^u(i,j) = 1$ is shown by surrounding with blue ellipses the crack-edges of the context; this context is the same as in [7] at this location, situated in the large black region from Fig. 2(b), where no warping was available for a large neighborhood.

However, when the flag $H^u(i,j) = 0$, the context for encoding $H(i,j)$ is built differently, including the crack-edges marked by yellow ellipses in Fig. 2(g) and the crack-edges marked in green from the images (H^w, V^w), shown in Fig. 2(h). The contexts for the vertical crack-edges are defined in a similar way, see Fig. 10 in additional figure file.

The continuity of the boundaries between patches imposes redundancy relations between the four crack-edges that have one end in common, leading to *deterministic cases* where no encoding is needed at a horizontal crack-edge, as explained in Fig. 15 in the file with additional figures.

C. Semi-Adaptive Context Tree Coding

The main encoding steps in Stage $\mathcal{P}$ and Stage $\mathcal{C}$ utilize arithmetic coding based on distributions extracted from context tree models [22], [23], [24] in semi-adaptive version. In the first step, a balanced tree of tree-depth equal to the size of the context (e.g. 12 in Stage $\mathcal{P}$ and 16 in Stage $\mathcal{C}$) is used as a model, and counts are collected at all its nodes (interior and leaves) in a first pass through data. In the second step the balanced tree is pruned so that the best codelength is achieved, by using dynamic programming, and then the optimal tree structure is encoded. In the last step the data is encoded, going through data points in a second pass through data, using as contexts the leaves of the optimally pruned tree, see [7].

D. Stage $\mathcal{Y}$: Encoding the Depth Value of Each Patch in the Image L

After Stage $\mathcal{C}$, the encoder and decoder both know the images (H, V), out of which one can find a patch label image C_L, which specifies the patches $P_1^L, \ldots, P_{N_L}^L$ in the left disparity image L. By overlapping the correctly warped locations with the patch label image C_L one can allocate the correct disparity values in L^w to the patches $P_1^L, \ldots, P_{N_L}^L$. Algorithm Y from [7] is used to encode the disparity only for the patches that do not overlap with any pixel at which $L^w(i,j) > 0$. The disparity values at these patches are encoded efficiently using prediction and exclusion based on the known disparity values of the neighbor patches.

III. Experimental Results

The program used in experiments was implemented as a collection of matlab and C functions, using the arithmetic coding routines from [25]. For each reported encoded file it was checked that the encoded file can be decoded losslessly.

The images used in experiments are 81 pairs (L, R) of disparity images from the Middleburry data set [1], consisting of 27 scenes represented at 3 different resolutions: full, one half, and one third resolution. The results of the new scheme, C-CERV, are compared in Fig. 3 to those obtained by CERV, which was shown to be better than all other lossless compressors evaluated in [7] over the Middleburry data set. The results of the lossless conditional coding scheme from [3] are also compared to C-CERV over all images used in [3].

The best compression is obtained for the full size images, where the disparity images are very precise and allow an accurate reconstruction through the predicted L^w, resulting in an improvement of about 3 times over the non-conditional compression of CERV. The results of conditional coding from [3] are not as good as the results of the new method. More results, including plots and numerical values, are given in the file with additional figures and tables.

The obtained results for conditional coding for the full size are varying between compressing 70 to 300 times the L image conditional on R; the results are in a similar range when encoding conditionally R based on L. For any given file the values $CR(R|L)$ and $CR(L|R)$ are in general quite different (see Fig. 21 in the file with additional figures). However, the pair codelength $\mathcal{L}(L|R) + \mathcal{L}(R)$ and the inversely ordered pair codelength $\mathcal{L}(R|L) + \mathcal{L}(L)$ are remarkably close, as seen in Fig. 22 (additional figures file). The encoding of a pair of images at full-resolution achieves a compression between 40 and 100

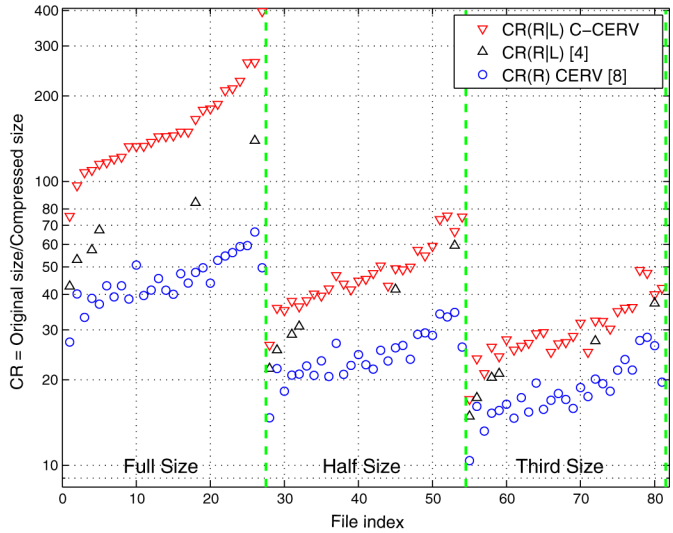

Fig. 3. Compression results (higher is better) for encoding the right view in 81 disparity image pairs from the Middlebury dataset. (Red): encoding the right view condtional on left view using the new method, C-CERV; (Black): encoding conditionally using the method from [3]; (Blue) encoding non-conditionally using CERV [7]. The order of the files was selected so that the results in the full resolution for the new method are increasingly ordered.

times (i.e., 0.08-0.2 bits per pixel). It is remarkable that *lossless* compression of a pair of disparity images reaches the range of compressions at which typically the *lossy* depth compression methods are operating.

In order to compare C-CERV and CERV methods over disparity images other than the ground truth disparity provided in [1], we considered the *estimated* disparity images for the scenes Teddy and Cones provided in the additional files of [26]. The estimated disparities are about 4-5 times more compressible, by both CERV and C-CERV, when using estimated images, than when using the ground truth images, showing that the estimation "noise" consists in removing fine details from the true disparity images and in fact favors better compression. The method C-CERV is better than CERV for all combinations of images (see Table VI in additional tables file).

The most complex operations in C-CERV are: encoding of the $n_r \times n_c$ binary images I_L, H, and V, by using semi-adaptive context coding, finding and encoding the structure of two optimal context trees, encoding the string $\gamma_1, \ldots, \gamma_{N_R}$, and encoding the depth values in the N_L patches. The encoding complexity of CERV is about 1.5 times lower, since CERV will not require encoding the $n_r \times n_c$ binary image I_L. The conditional coding method from [3] encodes a sequence of $n_r \times n_c$ binary planes using context tree coding, where the template is optimized differently than here or in [7], making a precise comparison of complexity difficult. At decoder, C-CERV is about 1.5 times slower than CERV and also slower than the method in [3], where the context trees are balanced, while in C-CERV they have an optimal, but irregular shape.

IV. Conclusion

The method for conditional coding C-CERV provides high lossless compression ratios, encoding one disparity image at a fraction from the cost of non-conditionally coding, when the other disparity image is known. The compression ratio of the method is highest for precise disparity images in high resolution.

REFERENCES

[1] H. Hirschmuller and D. Scharstein, "Evaluation of cost functions for stereo matching," in *Proc. IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, Minneapolis, MN, USA, Jun. 2007.

[2] K. Kim, G. Park, and D. Suh, "Bit-plane-based lossless depth-map coding," *Opt. Eng.*, vol. 49, no. 6, pp. 067 403–1–10, 2010.

[3] M. Zamarin and S. Forchhammer, "Lossless compression of stereo disparity maps for 3D," in *Proc. IEEE Int. Conf. Multimedia and Expo*, 2012, pp. 617–622.

[4] I. Schiopu and I. Tabus, "Depth image lossless compression using mixtures of local predictors inside variability constrained regions," in *Proc. Int. Symp. Communications, Control and Signal Processing*, Rome, Italy, May 2012, pp. 1–4.

[5] J. Heo and Y.-S. Ho, "Improved context-based adaptive binary arithmetic coding over H.264/AVC for lossless depth map coding," *IEEE Signal Process. Lett.*, vol. 17, no. 10, pp. 835–838, 2010.

[6] J. Heo and Y.-S. Ho, "Improved CABAC design in H.264/AVC for lossless depth map coding," in *Proc. IEEE Int. Multimedia and Expo Conf.*, 2011, pp. 1–4.

[7] I. Tabus, I. Schiopu, and J. Astola, "Context coding of depth map images under the piecewise-constant image model representation," *IEEE Trans. Image Process.*, vol. 22, pp. 4195–4210, Nov. 2013.

[8] P. Ausbeck, "The piecewise-constant image model," *Proc. IEEE*, vol. 88, no. 11, pp. 1779–1789, Nov. 2000.

[9] Y. Morvan, D. Farin, and P. de With, "Depth-image compression based on an R-D optimized quadtree decomposition for the transmission of multiview images," in *Proc. IEEE Int. Conf. Image Processing*, Oct. 2007, vol. 5, pp. V-105–108.

[10] S. Milani and G. Calvagno, "A depth image coder based on progressive silhouettes," *IEEE Signal Process. Lett.*, vol. 17, no. 8, pp. 711–714, 2010.

[11] F. Jäger, "Contour-based segmentation and coding for depth map compression," in *Proc. IEEE Visual Communications and Image Processing Conf.*, Nov. 2011, pp. 1–4.

[12] I. Schiopu and I. Tabus, "Lossy depth image compression using greedy rate-distortion slope optimization," *IEEE Signal Process. Lett.*, vol. 20, no. 11, pp. 1066–1069, Nov. 2013.

[13] R. Mathew, P. Zanuttigh, and D. Taubman, "Highly scalable coding of depth maps with arc breakpoints," in *Proc. Data Compression Conf.*, Apr. 2012, pp. 42–51.

[14] R. Mathew, D. Taubman, and P. Zanuttigh, "Scalable coding of depth maps with R-D optimized embedding," *IEEE Trans. Image Process.*, vol. 22, no. 5, pp. 1982–1995, May 2013.

[15] S. Milani, P. Zanuttigh, M. Zamarin, and S. Forchhammer, "Efficient depth map compression exploiting segmented color data," in *Proc. IEEE Int. Conf. Multimedia and Expo*, Barcelona, Jul. 2011.

[16] M. Zamarin, M. Salmistraro, S. Forchhammer, and A. Ortega, "Edge-preserving intra depth coding based on context-coding and H.264/AVC," in *Proc. IEEE Int. Conf. Multimedia and Expo*, Jul. 2013, pp. 1–6.

[17] G. Carmo, M. Naccari, and F. Pereira, "Binary tree decomposition depth coding for 3D video applications," in *Proc. IEEE Int. Conf. Multimedia and Expo*, Barcelona, Jul. 2011.

[18] Y.-H. Lin and J.-L. Wu, "Rendering lossless compression of depth image," in *Proc. Data Compression Conf.*, Mar. 2011, pp. 467–467.

[19] I. Daribo, G. Cheung, and D. Florencio, "Arithmetic edge coding for arbitrarily shaped sub-block motion prediction in depth video compression," in *Proc. IEEE Int. Conf. Image Processing*, Orlando, USA, Oct. 2012.

[20] S. Li, J. Lei, C. Zhu, L. Yu, and C. Hou, "Pixel-based inter prediction in coded texture assisted depth coding," *IEEE Signal Process. Lett.*, vol. 21, no. 1, pp. 74–78, 2014.

[21] K. Muller, H. Schwarz, D. Marpe, C. Bartnik, S. Bosse, H. Brust, T. Hinz, H. Lakshman, P. Merkle, F. Rhee, G. Tech, M. Winken, and T. Wiegand, "3D high-efficiency video coding for multi-view video and depth data," *IEEE Trans. Image Process.*, vol. 22, no. 9, pp. 3366–3378, 2013.

[22] J. Rissanen, "A universal data compression system," *IEEE Trans. Inf. Theory*, vol. 29, Sep. 1983.

[23] J. Rissanen, M. Weinberger, and R. Arps, "Applications of universal context modeling to lossless compression of grey-scale images," *IEEE Trans. Image Process.*, vol. 5, Apr. 1996.

[24] B. Martins and S. Forchhammer, "Tree coding of bilevel images," *IEEE Trans. Image Process.*, vol. 7, pp. 517–528, Apr. 1998.

[25] I. H. Witten, R. M. Neal, and J. G. Cleary, "Arithmetic coding for data compression," *Commun. ACM*, vol. 30, pp. 520–540, 1987.

[26] B. Ham, D. Min, C. Oh, M. Do, and K. Sohn, "Probability-based rendering for view synthesis," *IEEE Trans. Image Process.*, vol. 23, no. 2, pp. 870–884, Feb. 2014.