ORIGINAL PAPER

Lossless contour coding using elastic curves in multiview video plus depth

MARCO CALEMME, MARCO CAGNAZZO AND BEATRICE PESQUET-POPESCU

Multiview video plus depth is emerging as the most flexible format for three-dimensional video representation, as witnessed by the current standardization efforts by ISO and ITU. In particular, in depth representation, arguably the most important information lies in object contours. As a consequence, an interesting approach consists in performing a lossless coding of object contours, possibly followed by a lossy coding of per-object depth values. In this context, we propose a new technique for lossless coding of object contours, based on the elastic deformation of curves. Using the square-root velocity representation for the elements of the space of curves, we can model a continuous evolution of elastic deformations between two reference contour curves. An elastically deformed version of the reference contours can be sent to the decoder with a reduced coding cost and used as side information to improve the lossless coding of the actual contour. Experimental results on several multiview video sequences show remarkable gains with respect to the reference techniques and to the state of the art.

I. INTRODUCTION

The video-plus-depth representation for multiview video sequences (MVDs) consists of several views of the same scene with their associated depth information, which is the distance from the camera for every point in the view. The MVD representation allows functionalities like three-dimensional (3D) television and free-viewpoint video [1–3], but it generates large volumes of data that need to be compressed for storage and transmission. As a consequence, MVD compression has attracted a huge amount of research effort in the last years, while ISO and ITU are jointly developing an MVD coding standard [4]. Compression should exploit all kinds of statistical dependencies present in this format: spatial, temporal, and inter-view, but also inter-component dependencies, i.e. between color (or texture) and depth data [5, 6].

We focus in particular on depth images compression through contour-based coding means. The techniques developed for texture images are not well suited for depth images, since the latter have different properties and they are not meant to be visualized but only used for rendering of virtual views. Objects within a depth map are usually arranged along planes in different perspectives; as a consequence, there are areas of smoothly varying levels,

Telecom ParisTech, 75634 Paris, France

Corresponding author:
M. Cagnazzo
Email: cagnazzo@telecom-paristech.fr

separated by sharp edges corresponding to object boundaries. These characteristics call for an accurate encoding of contour information. Some of the proposed approaches include modeling the depth signal as a piecewise polynomials (wedgelets and platelets [7]), where the smooth parts are separated by the object contours. However, it is generally recognized that a high-quality view rendering at the receiver side is possible only by preserving the contour information [8, 9], since distortions on edges during the encoding step would cause a sensible degradation on the synthesized view and on the 3D perception. In other words, a lossless or quasi lossless coding of the contour is practically mandatory.

For these reasons, in this paper we focus on lossless coding of contours, and we target as natural application the compression of depth maps. However, we observe that an effective contour coding algorithm may prove beneficial to other applications such as object-based image and video coding. This promising coding framework has not been able to replace traditional block-based approaches because, among other things, lossless coding of contours was too expensive [10], other causes being the absence of reliable segmentation algorithms.

On the other hand, depth coding segmentation is eased by the nature of the depth signal, and the extraction of the contours can be achieved with specifically designed segmentation techniques [11]. In conclusion, lossless contour coding is very relevant in the context of depth map coding and it is the main focus of this paper. Moreover, it has potential applications also for object-based coding of natural video.

The relevance of depth contour lossless coding has been recognized in previous works. Gautier *et al.* [8] use JBIG [12] to encode the object contours and a diffusion-based inpainting algorithm to fill in the interior of the objects, starting from a subsampled version of the image and the boundary values. In [8], it is underlined the importance of a lossless representation of contours for depth maps, even though the problem of efficient lossless coding is not in its scope: authors use JBIG when more adapted tools are available, such as JBIG2 [13], chain coding [14] and differential chain coding [15, 16]. Recently the problem of contour coding for depth maps has been explored by Daribo *et al.* [17]: their method tries to predict the next edge direction within the context of previously transmitted symbols, under the assumption that pixel boundaries exhibit a linear trend. This approach allows us to achieve large improvement with respect to the state of the art. However, even this high-performing technique can be improved by considering better models for object contours and by exploiting temporal correlation: these elements are at the basis of the proposed technique [18].

Indeed, we achieve relevant results by using elastic deformation of curves [19] to provide more effective context information for encoding the current curve: after computing the portion of the deformed curve corresponding to the current symbol, we use the shape of this deformed curve and the past samples of the encoded curve to estimate the most probable direction of the contour. This direction is in turn used to parametrize the probability distribution of the next symbol in the curve representation. Finally, this symbol is encoded with a context-based arithmetic encoder. Experimental results show remarkable rate reductions with respect to standards (about -65% with respect to JBIG2), to commonly used algorithms (about -20% with respect to arithmetic coding plus differential chain coding), and to the state-of-the-art method in [17] (about -6.5%). We observe that this use of the elastic deformation tool is quite different from what it was proposed in the past. The only previous paper using elastic curves (ECs) in compression is [18]. However, in that paper, ECs are used in the context of distributed video coding, in order to improve the motion-compensated fusion of background and foreground objects, while here it is used for the lossless coding of contours.

The outline of the paper is as follows. We first introduce basic notions on elastic deformations of curves and the basics of arithmetic edge coding using the method of [17] in Section II. The proposed technique is then described in Section III, experimental results and conclusions are presented in Sections IV and V, respectively.

II. BACKGROUND NOTIONS

A) Elastic curves

Srivastava *et al.* [19] introduced a framework to model a continuous evolution of elastic deformations between two reference curves. The referred technique interpolates between shapes and makes the intermediary curves retain the global shape structure and the important local features such as corners and bends.

In order to achieve this behavior, a variable speed parameterization is used, specifically square-root velocity (SRV), so that it is possible to bend one shape into another as well as stretch or compress a certain part of it. Let us introduce some notation. We call p the curve defining the shape, and $t \in [0, 1]$ the curve parameter, leading to:

$$p : [0, 1] \rightarrow (x, y) \in \mathbb{R}^2, \tag{1}$$

where (x, y) are the coordinates of each point in the contour. Then, p is represented in the SRV space by q:

$$q : [0, 1] \rightarrow (x, y) \in \mathbb{R}^2, \tag{2}$$

$$q(t) = \frac{\dot{p}}{\sqrt{||\dot{p}||}}, \tag{3}$$

where $|| \cdot ||$ is the Euclidean norm in $\mathbb{R}^2$ and $\dot{p} = \frac{dp}{dt}$. This transformation is reversible (up to a translation): the curve p can reversely be obtained from q by:

$$p(t) = \int_0^t q(s)||q(s)||ds. \tag{4}$$

Introducing the SRV representation is very interesting because it can be shown that the simple $\mathcal{L}^2$ metric in this space corresponds to an "elastic" metric for the original curve space [20], i.e. a metric that measures the amount of "stretching" and "bending" between two curves, independently from a translation, scale, rotation, and parametrization. Moreover, using the SRV it is also relatively easy to compute the geodesic between the two curves: according to the interpretation of the elastic metric, this geodesic consists in a continuous set of deformations that transforms one curve into another with a minimum amount of stretching and blending, and independently from their absolute position, scale, rotation, and parametrization [19]. Classical applications of elastic deformations of curves are related to shape matching and shape recognition. An example of the geodesic connecting two curves is shown in Fig. 1. We show in black two contours extracted (with the Canny edge detector) from the depth of the video sequence "ballet". These depths correspond to the views 1 and 8, at time instant 2. In red we show the extracted contours of intermediary views, whereas in dashed blue we show a sampling of the elastic geodesic computed between the two extreme curves: it is evident that the elastic deformations along the geodesic reproduce very well the deformations related to a change of viewpoint. Similar results where obtained in the temporal domain: elastic deformations are able to represent the temporal deformation of an object contours, given the initial and final shapes.

These observations lead us to conceive a lossless coding technique for object contours: supposing that the encoder and the decoder share a representation of the initial and final shapes, they can reproduce the exactly same geodesic
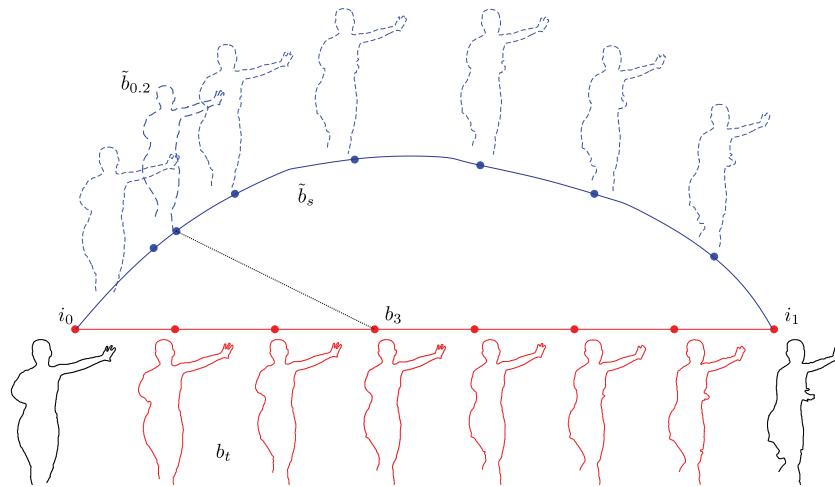
Fig. 1. Geodesic path of elastic deformations $\tilde{b}_s$ from the curve i_0 to i_1 (in dashed blue lines). b_3 is one of the contours b_t extracted from the intermediate frames between the two reference ones, a good matching EC $\tilde{b}_{0.2}$ along the path is highlighted.
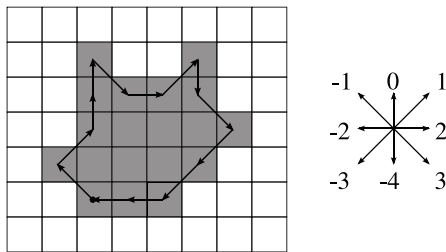

Fig. 2. Differential chain code: the arrows represent the symbols of the differential chain code if the previous symbol is "Up". We can code the contour of the object, from the starting point, as: "Up-Left", $2, -1, 0, 3, -1, -1, 3, -1, 2, 0, 1, 0$.

path between them. Then, the decoder will conveniently decide which point of the geodesic (one of the blue curves in Fig. 1) shall be used as context information to encode an intermediary contour (one of the red curve in the same figure). The encoder will only have to send a value $s^{\star} \in [0, 1]$ to identify this curve. If this curve is actually similar to the one to be encoded, it is possible to exploit this information to improve the lossless coding of the latter. However, how to do this is not obvious, and the solution to this problem is one of the main contributions of this paper.

B) Lossless contour coding

Various techniques have been developed to represent and code the boundaries of objects, like polygon approximation [21] or chain coding [14]. The latter is the most common method to losslessly encode boundary pixels. A chain code follows the contour of an object and encodes the direction of the next boundary pixel with respect to the current one. Since an object tends to have a quite regular contour it is usually more convenient to code the change of direction with respect to the previous one, thus leading to differential chain codes. An example of differential chain coding of the 8-connected contour of an object is given in Fig. 2. The sequence of symbols produced by a chain code is fed into

an entropic encoder, such as a variable length encoder or an arithmetic encoder, possibly using contexts to improve its performance [22].

Recently Daribo *et al.* [17] introduced a new technique aimed at the lossless coding of edges that resulted to be the best among the others. However, it is conceived in a block-based coding environment, we retain it as a base to develop our contour-based coding technique. Their main idea is that contours of a physical object possess geometrical structures that can be exploited to predict the direction of the next symbol, given a window of consecutive previous samples. A chain code is used to represent a contour, and each symbol is encoded with an arithmetic encoder that uses a probability distribution adaptively computed using the previous symbols. The probability distribution is centered around the estimated most probable direction, which in turn is obtained using linear regression (LR) on a suitable window of previous samples: the underlying assumption is that contours exhibit linear trends. The prediction of directions and the assignment of the probability values can be reproduced at the decoder, provided that some parameter is transmitted as side information.

This method has much better performance than other popular techniques [17], but further improvements are possible. On the one hand, the LR can be replaced with a more effective estimator for the direction of the next symbol. On the other hand, the encoding algorithm does not take advantage from temporal correlation of contours and the prediction of the most probable direction cannot cope with sudden changes of direction.

Our proposed method uses temporal correlation of contours by producing an elastic estimation of the current curve; this curve is subsequently employed for improving the statistical model of the current symbol of the chain code, resulting in a remarkable improvement of the coding rate. The two reference curves used to produce the elastic deformation are encoded using our improved version of the technique described by Daribo *et al.* [17].

III. PROPOSED TECHNIQUE

We propose a technique for encoding the contour of an object in a single view video sequence, in a multiview set of images or in an MVD, and in any context where two reference curves are available. The targeted application is depth coding in MVD applications, and this for two reasons: first, contour information is extremely important for depth, and its lossless representation is necessary for obtaining a good subjective quality for synthesized views; second, extracting contours from a depth map is relatively easy, since they are typically made up of smooth regions separated by sharp discontinuities. However, in this paper, we do not investigate the contour extraction from depth images, but uniquely their lossless coding; moreover, we point out that our method can also be applied on contours extracted from natural videos, as we show in the experimental results. In this last case, contours are typically extracted from precise segmentation maps: our method efficiently encodes both contours automatically extracted from depths and contours obtained by segmentation.

As already mentioned, our method apply to the case where we have a set of contours related to the same object, be them representing the temporal evolution of the object borders or their deformation related to the change of viewpoint: the example in Fig. 1 refers to the latter case. However, without loosing generality, we consider the following use case. We have a set of $K \geq 3$ contours ($K = 8$ in Fig. 1). We refer to the parametric representation of the first contour as $i_0[n] = (x_{i_0}[n], y_{i_0}[n])$, with $n \in \{1, 2, \ldots, N_{i_0}\}$; likewise, $i_1[n]$ with $n \in \{1, 2, \ldots, N_{i_1}\}$ is the parametric representation of the last contour, and $b_t[n]$ with $n \in \{1, 2, \ldots, N_t\}$ the one of the generic t-th intermediate curve, with $t \in \{1, \ldots, K - 1\}$. We propose an "intra method" (i.e. without temporal prediction) to encode i_0 and i_1, and a "bidirectional method" (i.e. with prediction from two already encoded curves) for the curves b_t. We will refer to the intra-coded contours as "I-contours" and to the $K - 2$ intermediate contours, to which our prediction-based method is aimed, as "B-contours".

Indeed, the I-contours are supposed to be available at the encoder and at the decoder before the B-contours in order to develop a prediction method to code the latter. To encode the I-contours i_0 and i_1 we propose a small yet effective modification to the technique described in [17]. The difference between the modified and the original version lies in the predictor for the direction of the next symbol, as shown in Section IIIB.

Let us now describe the proposed method for the B-contours, the basic idea is independent from the value of K and from the structure of dependencies (or, borrowing the terms from the classical hybrid video coding paradigm, from the GOP structure). More precisely, in order to encode b_t, we consider the geodesic path between i_0 and i_1. The elastic deformation tool allows us to easily generate any intermediate curve on the geodesic (dashed blue curves in Fig. 1), simply by specifying a position parameter $s \in [0, 1]$. Let us refer to $\widetilde{b}_s[n]$ the parametric representation of a curve

on the geodesic. We observe that $\widetilde{b}_0 = i_0$ and $\widetilde{b}_1 = i_1$. Since i_0 and i_1 are available at the encoder and the decoder, they both can produce the same curve $\widetilde{b}_s$, for any s, and use it as side information for the lossless encoding of b_t. Intuition suggests that the encoder and the decoder could agree in using $s = \frac{t}{K-1}$ as the position on the geodesic used to predict b_t. However, as we will show in the experimental section, a significant coding gain can be obtained if we let the encoder select a suitable value for s, let it be $s^\star$, and use it for the encoding. Of course, $s^\star$ should be transmitted to the decoder using a suitable number of bits. The choice of $s^\star$ and the number of coding bits will be discussed in the experimental section. Likewise, the problem of the optimal "GOP structure" will be discussed in Section IVD. In the rest of the current section, in order to simplify the discussion, we will consider a simple IBIBIB... configuration. This is equivalent to having $K = 3$: in other words, we encode an object contour knowing the previous and the future curves. The proposed method can however be applied to any value of K without major modifications. Other coding structures are shown in the experimental part at Section IVD.

In the rest of this section, we will explain how to select a suitable part of the EC $\widetilde{b}_s$ to be used as context to encode the current edge symbol; how to use this information to determine the most probable direction for the next symbol on the contour; and which side information needs to be sent to the decoder such that it can replicate the same behavior as the encoder.

A) Correspondence function

In this subsection, we consider the encoding of a single curve b_t using the elastic representation $\widetilde{b}_{s^\star}$, with $s^\star$ suitably selected by the encoder. For the encoding of i_0 and i_1 we do not use the elastic deformation.

The curves are sampled respectively on N_b and $N_{\widetilde{b}}$ points. To simplify the notation, we will drop the subscript t and $s^\star$ where this does not give rise to ambiguities.

To use the suitable portion of the synthetic curve $\widetilde{b}$ as side information to code the current point on b, it is essential to have a function that associates each point of b to the corresponding point of $\widetilde{b}$. This function is generated at the encoder and has to be transmitted to the decoder.

In order to establish an association between the two curves, we use dynamic time warping (DTW) [23]. First, it is needed to establish a feature space $\mathcal{F}$ for the curves: a distance in this space will then be used to create the DTW function. Since we want to link the parts of the curve that have the same characteristics in terms of lobes, spikes and such, we used the direction of the tangent vector as feature[1]. Let us use a complex representation that associates to every point (x, y) the complex number $p = x + jy$, and

---

[1]Other features we tested are: curvature and first-order derivative of the direction of the tangent vector.

let us refer to the sequence of tangent vectors of the curve $c = (x_c, y_c)$ as φ_c:

$$\forall n \in \{1, \ldots, N_c\}, \varphi_c[n] = \arg\left(p_c[n] - p_c[n-1]\right) \in \mathcal{F}, \tag{5}$$

where $\mathcal{F} = [-\pi, \pi[$. Computing equation (5) for b and $\widetilde{b}$ we obtain the sequences of features φ_b and $\varphi_{\widetilde{b}}$ defined on N_b and $N_{\widetilde{b}}$ points, respectively.

A typical behavior of two feature sequences is shown in Fig. 3; while the two sequences have similar shapes, they are not aligned. To perform the alignment we need a local distance measure (or local cost measure), defined as $d : \mathcal{F} \times \mathcal{F} \rightarrow \mathbb{R}^+$. The local distance measure $d(\varphi_b[n], \varphi_{\widetilde{b}}[m])$ should be small when the two features are similar, large otherwise. Since in our case $\mathcal{F} \subset \mathbb{R}$, we can use as distance the square of the direction difference: $\left(|\varphi_b - \varphi_{\widetilde{b}}| \mod \pi\right)^2$. By evaluating the local cost measure for every couple of elements in the sequences, we obtain the cost matrix $C \in \mathbb{R}^{N \times M}$, where the generic element is defined as follows:

$$\begin{aligned} C(n, m) &= d(\varphi_b[n], \varphi_{\widetilde{b}}[m]) \\ &= \left(\left|\varphi_b[n] - \varphi_{\widetilde{b}}[m]\right| \mod \pi\right)^2. \end{aligned}$$

We have to find now the sequence ψ, defined as a sequence of couples:

$$\psi[\ell] = (v_\ell^\star, \mu_\ell^\star) \in \{1, \ldots, N\} \times \{1, \ldots, M\},$$

such that:

$$(v^\star, \mu^\star) = \arg\min_{(v, \mu)} \sum_{l=1}^{L} C\left(v[\ell], \mu[\ell]\right)$$

under the conditions:

- boundary condition, $\psi[1] = (1, 1)$ and $\psi[L] = (N, M)$;
- monotonicity of $v^\star[\ell]$ and $\mu^\star[\ell]$;
- step size, $\psi[\ell] - \psi[\ell-1] \in \{(0, 1), (1, 0), (1, 1)\}$.

In practice, ψ is a sequence of indices $(v^\star[\ell], \mu^\star[\ell])$ of the curves φ_b and $\varphi_{\widetilde{b}}$, such that $\varphi_b[v^\star[\ell]]$ and $\varphi_{\widetilde{b}}[\mu^\star[\ell]]$ are best matched under the aforementioned constraints. The association by DTW of the two sequences is shown in dotted black lines in Fig. 3.

In our application, the correspondence function obtained with DTW on the direction of the tangent vector is very close to a straight line and it can be approximated with a
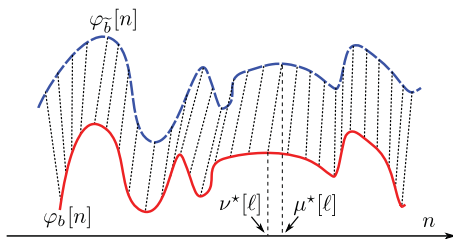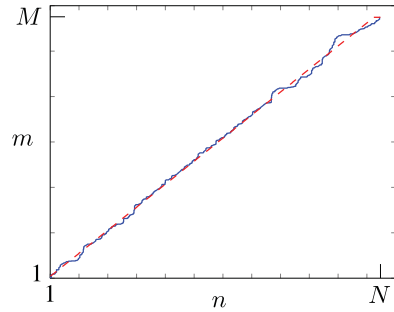

Fig. 4. *Ballet*: correspondence function. In blue the association of the two curves using the DTW of the direction of the tangent vector, in dashed red the approximation with a first-order polynomial, whereas n and m are the indices of samples on the curves b and $\widetilde{b}$, respectively.

first-order polynomial. The approximation has two main effects: first it reduces the number of bits needed to code the function; moreover it prevents sudden variations on the correspondence function that are rather related to outliers than actual values. However, one may wonder whether it is worth computing the exact DTW only for approximating at a first order: maybe a simple rescaling of the "temporal" axis from N points to M points could be as effective as the approximated DTW, without needing to compute the correspondence function. We have dealt with this issue with a simple heuristic approach: we compared the coding rate of our algorithm in two cases: in the first, we use the first-order approximation of the DTW function; in the second we use a rescaling of N/M. We observed that using the DTW gives an average rate reduction of 5.33%. For this reason, we kept the DWT in our system.

In Fig. 4, there is an example of DTW and its linear approximation. The resulting correspondence between the points of the two curves is shown in Fig. 5. We see that all the main features of the curves are located and put in correspondence, so that for each point of the actual curve b there is an associated point on the EC $\widetilde{b}$, whose neighborhood is the side information we want to use to enhance the coding of b.

B) Context

The correspondence function allows us to associate the current point to a portion of the EC, centered in the corresponding point. This information is used as side information to have more accurate probability values for the next symbol. We called this information the *context*, and for each point of the curve b, it is composed by:

- $\mathbf{v}_0$, a vector of N_0 points of the curve b transmitted so far (in red in Fig. 6);
- $\mathbf{v}_{1p}$: the "past" on the EC (in dashed blue in Fig. 6), a vector of N_p points of $\widetilde{b}$ corresponding to $\mathbf{v}_0$; more precisely, $\mathbf{v}_{1p}$ is constituted of the points between those corresponding to the terminal points of $\mathbf{v}_0$;
- $\mathbf{v}_{1f}$: the "future" on the EC, a vector of N_f points of the EC $\widetilde{b}$ following the current correspondent point on $\widetilde{b}$ (in dark blue in Fig. 6).
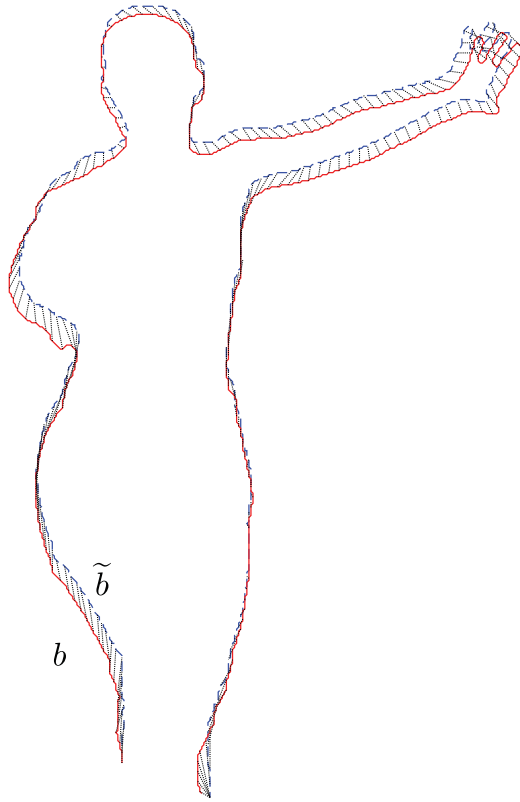

Fig. 3. Example of association of two sequences by DTW.

Fig. 5. *Ballet*: correspondences between the EC $\widetilde{b}$ (dashed blue) and the curve to code b (red).
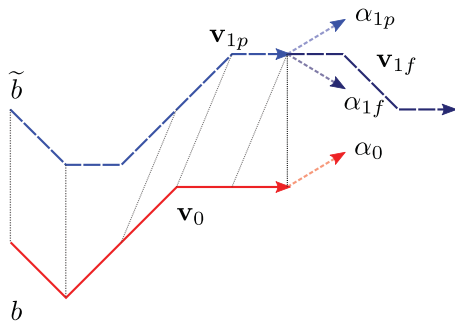

Fig. 6. Extracts from the curves b (red) and $\widetilde{b}$ (dashed blue). The correspondences between the two curves are indicated with thin dotted black lines. The dashed lines represent the extracted direction for the vectors of points $\mathbf{v}_0$, $\mathbf{v}_{1p}$, and $\mathbf{v}_{1f}$.

Of course $\mathbf{v}_{1p}$ and $\mathbf{v}_{1f}$ are only available for B-contours, for I-contours only $\mathbf{v}_0$ is available. We use the context to obtain the most probable direction for the next symbol, then we use this result to define a distribution using the von Mises statistical model [24].

1. DIRECTION EXTRACTION
For all the set of points of the different curves, we have to estimate the direction of the next symbol. In the case of the I-contours, only the set of points $\mathbf{p}_0$ are used, whereas in the case of B-contours all the three sets of points $\mathbf{v}_0$, $\mathbf{v}_{1p}$, and $\mathbf{v}_{1f}$ are used.

Several approaches can be used to extract a direction from a set of ordered points. E.g. in [17] an LR on $\mathbf{v}_0$ is used. We found that a simple average direction (AD) is even more effective. Using the same complex representation as in IIIA, the estimated direction α can be obtained using the following formula:

$$\alpha\,(\mathbf{v}) = \arg\left(\frac{1}{N-1}\sum_{n=2}^{N} (\mathbf{v}[n] - \mathbf{v}[n-1])\right) \in [-\pi, \pi], \tag{6}$$

where $\mathbf{v}$ is the complex representation of a generic vector of N points, and is defined as $\mathbf{v} = [x_1 + jy_1, x_2 + jy_2, \ldots, x_N + jy_N]$; arg is the argument of a complex number.

2. MOST PROBABLE DIRECTION
Applying α on the complex representation of $\mathbf{v}_0$, we obtain α_0, the angle of the AD based solely on the previously transmitted samples. Likewise, α_{1p} and α_{1f} are the ADs based on the vectors $\mathbf{v}_{1p}$ and $\mathbf{v}_{1f}$ of the curve $\widetilde{b}$. We develop a method to estimate θ, the most likely direction of next symbol of b: we use α_{1f} to adjust the direction α_0. In this way, we manage to seize the sudden changes and go along the long trends.

A simple and intuitive formula to take into account the context for the prediction of the most probable direction is:

$$\theta = (1-q)\alpha_0 + q\alpha_{1f}, \tag{7}$$

with $q \in [0, 1]$. This way we can weight the directions of the past of the curve b and the future of the curve $\widetilde{b}$. To decide the weight q, we observe that the side information of the EC is not essential when the curves are regular, while it becomes fundamental next to the occurrence of a sudden change. So q should be small if for the current point the directions extracted from the two curves are similar (so that θ is close to α_0), and close to 1 if they are not (so that θ close to α_{1f}):

$$q = \frac{\max\{|\alpha_0 - \alpha_{1p}|, |\alpha_0 - \alpha_{1f}|\}}{\pi}. \tag{8}$$

The value of q is related to the modulus of the difference of directions, and it is large if α_0 is very different from α_{1p} or α_{1f} because a dissimilarity in the neighborhood of the current point suggests a change which is not predicted solely from α_0.

3. ADAPTIVE STATISTICAL MODEL
We retain the statistical model described in [17] to assign values to the symbols for the next edge. It is based on the von Mises distribution and the distribution parameters are set according to the information extracted from the curves b and $\widetilde{b}$. The von Mises distribution is the Gaussian distribution for angular measurements and it is defined as [24]:

$$p(\beta|\mu,\kappa) = \frac{e^{\kappa \cos(\beta-\mu)}}{2\pi\,I_0(\kappa)}, \tag{9}$$

where $I_0(\cdot)$ is the modified Bessel function of order 0, μ is the mean and in this case it coincides with the estimated

direction θ, $1/\kappa$ is the variance of the distribution. As in [17], we set κ as a function of the predicted direction θ: when θ is aligned with the axis of the pixel connection grid, intuition tells us that it is more convenient to give a higher probability value to the symbol that represents that direction. So κ is set to:

$$\kappa = \rho \cos(2\hat{\theta}), \qquad (10)$$

where $\hat{\theta} = \min\{|\theta - \gamma_i|\}$, and γ_i are the angles of the pixel connection grid ($\{0, \frac{\pi}{8}, \frac{2\pi}{8}, \dots\}$ in the case of an eight-connected grid). The parameter ρ represents a "confidence level" of the prediction: as it grows it makes the distribution more unbalanced, so the more precise is the prediction based on the context, the larger it should be to achieve higher coding gains.

C) Coding

The curve b, represented with a differential chain code, is encoded with an arithmetic coder which for each symbol uses the probability vector assigned by the adaptive statistical model. The encoder needs to transmit to the decoder the parameters involved in the coding process:

- $s^\star$, the selected point on the geodesic path;
- the correspondence function, approximated by a first-order polynomial, so two parameters;
- the parameter ρ;
- N_0 and N_f.

With this information the decoder can reproduce the behavior of the encoder and it will compute the same probability values for each point of the curve b. We observe that N_p does not need to be sent, since it is deduced by applying the correspondence function to $\mathbf{v}_0$.

IV. EXPERIMENTAL RESULTS

The proposed method has been evaluated using the multiview sequences *ballet* (provided by Microsoft Research), *mobile* (Philips), *lovebird* (ETRI/MPEG Korea Forum), and *beergarden* (Philips). We encoded the curves corresponding to the main object in the depth sequences for a fixed time instant or view. We used to test our algorithm also masks extracted from the standard monoview sequences *stefan* and *foreman*.

The curves for *ballet*, *mobile*, *lovebird*, and *beergarden* depths were obtained using the Canny edge detector [25]. Depth maps are not as complex as texture images and the use of the Canny edge detector produced very good results in our test cases. We extracted the contours of very precise segmentation maps for the sequences *stefan* [26, 27] and *foreman* [28]. This latter case can be seen as the ideal test case.

The coding scheme concentrates almost all the computational load at the encoder side. In particular, for the B-contours the choice of the values of (N_p, N_f, ρ, s) can be simply made trying out every possible combination and

Table 1. Coding results (in bits) for the different contributions of the developed tools to the technique proposed in [17], applied to object contours. Two different methods to extract the probable direction from a set of points: linear regression (LR), and average direction (AD), without and with elastic curve (EC) context.

	LR	AD	LR + EC context	AD + EC context
Ballet	1428.20	1375.20	1386.94	1338.78
Beergarden	1687.67	1610.33	1656.67	1575.58
Lovebird	1424.82	1370.73	1382.08	1341.73
Mobile	711.71	642.29	697.15	639.12
Foreman	411.73	396.82	398.15	390.20
Stefan	745.33	701.00	738.33	700.00
Average	1068.24	1016.06	1043.22	997.57

selecting the one that gives the best outcome. This full search method, however optimal, is extremely costly in terms of time and complexity. We thus introduced a greedy algorithm that optimizes one variable at a time to find a sub-optimal solution.

A) Coding of I- and B-contours

To code the I-contours we replaced the LR with the more effective AD in the technique described in [17], whereas for the coding of the B-contours we can take advantage of the context provided by the ECs, thus achieving better coding gains. In Table 1, are shown the results for a set of images taken from the test sequences. We notice that just altering the direction extraction method from LR to AD the average coding cost is reduced by 4.88%, while passing from AD to AD with EC context leads to a reduction of 1.82%, for a total reduction of 6.53%. If on the other hand we use the LR with the EC context, then the rate reduction is 2.25%.

B) Side information cost

We will now account for the cost to transmit the four parameters $s^\star$, $N_p^\star$, $N_f^\star$, and $\rho^\star$, as well as the correspondence function.

The correspondence function is the result of a linear approximation and experiments show that using 10 bits to code the two parameters of the straight line leads to the same performance as coding them with double precision.

For $N_p^\star$ and $N_f^\star$ we decided to use 1 bit for the range of N_p ($\{5, 6\}$), and 2 bits for the range of N_f ($\{6, 7, 9, 11\}$), once again based on experimental results. On the other hand, the optimal ρ has a wider range of values, and we observed that typical values can be represented by the set $\{6.6 + k\Delta\}$, with $\Delta = 0.1$ and $k = 0, \dots, 31$, for a cost of 5 bits.

We observe that the accuracy of the representation of the position on the geodesic $s^\star$ has a quite large influence on the coding efficiency. Using for example 2 bits to code the possible positions we have four curves to use as side information to code the curve b, using 3 bits leads to eight curves, in

Table 2. Average coding cost (in bits) for different ways of coding $s^\star$: fixed length coding up to 10 bits and Exp-Golomb.

Sequence	2 bits	3 bits	4 bits	5 bits	6 bits	7 bits	8 bits	9 bits	10 bits	Exp-Golomb
Ballet	1338.20	1329.00	1323.80	1323.80	1323.40	1321.40	1319.80	1318.20	1316.20	1320.80
Beergarden	1573.33	1565.33	1559.33	1560.33	1559.00	1559.67	1554.67	1554.33	1554.00	1559.00
Lovebird	1370.45	1347.18	1336.73	1334.18	1331.27	1328.73	1328.18	1324.73	1323.91	1332.09
mobile	638.29	635.29	635.43	635.57	634.57	634.86	635.71	635.00	634.71	635.71
Foreman	390.27	388.27	388.45	388.91	387.00	386.82	387.09	386.55	387.09	388.09
Stefan	705.00	695.67	693.00	693.00	691.67	689.33	688.33	689.33	690.00	694.00
Average	1002.59	993.46	989.46	989.30	987.82	986.80	985.63	984.69	984.32	988.28

general using a fixed length representation with b_s bits permits us to choose among 2^{b_s} different curves. The chance of finding a good matching curve to use as a side information increases with b_s, but for every bit added the complexity doubles, and we are increasing the cost of the representation too. If some values are more probable than others it is worth to consider a variable length code to reduce the cost of the representation.

We thus used an experimental approach and compared two different ways to code $s^\star$: a fixed length coding, with length from 2 to 10, and an Exponential-Golomb code. In Table 2, we see that, performing the average on different curves and on different sequences, a fixed length coding leads to better performance if the number of bits used for the representation of $s^\star$ approaches to 6 or more. To compare the proposed method to other techniques we choose the best performing 10 bits fixed length coding.

For the I-contours we only have to transmit as side information the parameters $N_p^\star$ and $\rho^\star$, corresponding to an overall cost of 6 bits. On the other hand, the decoding of the B-contours needs the correspondence function as well as the four parameters $N_p^\star$, $N_f^\star$, $\rho^\star$, and $s^\star$, corresponding to an overall cost of 28 bits.

C) Greedy algorithm (GA)

Resting on experimental results for each variable we selected a range of typical values. We initialize the GA with a starting point $(N_{p0}, N_{f0}, \rho_0, s_0)$, corresponding to the solution in which every value is the closest to the center of the coded range, and then the GA optimizes the variables in the order N_p, N_f, ρ, s. Keeping fixed N_{f0}, ρ_0, and s_0 the GA runs the proposed technique to select the N_p that minimizes the bit rate. Once $N_p^\star$ has been found the algorithm starts again to optimize N_f from the point $(N_p^\star, N_{f0}, \rho_0, s_0)$. Then again for ρ and s, until it reaches the solution $(N_p^\star, N_f^\star, \rho^\star, s^\star)$.

The search order for the parameters is set according to our observations of the optimum parameter distributions after the full search for many sequences: $N_p^\star$ has a very peaked distribution, so the selected value after the first step of the GA should actually be the best one. $N_f^\star$, on the contrary, has an almost uniform distribution, thus making difficult to locate the best value; it is however required to select it before $\rho^\star$, because the best value of ρ is influenced

Table 3. Average coding cost (in bits) for the full search and the greedy algorithm (GA).

Sequence	Full Search	GA
Ballet	1309.60	1316.20
Beergarden	1548.00	1554.00
Lovebird	1305.18	1323.91
Mobile	631.43	634.71
Foreman	380.82	387.09
Stefan	684.00	690.00
Average	976.51	984.32

by the selected values of N_p and N_f. Still based on our observations, the last parameter to select is s.

As we can see in Table 3, the average loss with respect to the full search is 0.82%, but the number of calculations the encoder has to do is approximately reduced by a factor of 250.

Regarding the overall time needed for the coding of a curve, we can distinguish fixed and variable time contributions. The elastic estimation is fixed but we can decide how many frames to leave in between the two reference ones. On the other hand, the execution time for the choice of the parameters can vary greatly: a full search is very costly, and even if the GA speeds up the whole process, one can also decide to keep the same parameters (or a subset of the parameters) for a certain number of frames.

D) Comparisons

We compare our technique to various methods to code the differential chain code of the contours: Adaptive Arithmetic Coder (AAC), Context Based Arithmetic Coder (CBAC), and the technique proposed in [17].

In the compression of B-contours, we achieve gains up to 10% compared to the method of [17], but to make a fair comparison we have to consider for our technique both I-contours and B-contours of the GOP structure. To study the influence of the GOP structure on the coding performance we used the following:

- IBIBIB … is very effective for the B-contours, since the two reference frames are very close and the prediction is very accurate, but the I-contours have a non negligible

Table 4. Average coding cost (in bits) for various sequences in the view domain (*ballet*) and in the time domain (*mobile, lovebird, beergarden, stefan*). The tested methods are: JBIG2, Adaptive Arithmetic Coder (AAC), Context Based Arithmetic Coder (CBAC) with 1 symbol context, the one proposed in [17], and the proposed technique (all the side information cost accounted). In the last column are reported the gains of the proposed technique over the other best performing one in the group.

Sequence	# symbols	JBIG2	AAC	CBAC	Method [17]	Proposed	Gain (%)
Ballet	1125.00	3968.00	1715.40	1585.60	1428.20	**1338.78**	6.26
Beergarden	1369.00	4226.67	2052.33	1882.67	1687.67	**1575.58**	6.64
Lovebird	1302.09	3153.45	1886.55	1740.36	1418.82	**1341.73**	5.43
Mobile	661.86	1915.43	837.14	696.00	711.71	**639.12**	10.20
Foreman	330.00	1776.73	517.18	522.91	411.73	**390.20**	5.23
Stefan	565.00	2106.67	859.33	877.33	745.33	**700.00**	6.08
Average	*892.16*	*2857.82*	*1311.32*	*1217.48*	*1067.24*	*997.57*	*6.53*

cost on the final outcome. Using this GOP structure produced in our experiments an average bit rate of 1000,19 bits per contour;

- IBBIBB … and IBBBIBBB … are flat structures with fairly distant I-contours, they produced an average bit rate of 1004.03 and 1010.28 bits per contour, respectively;
- $I_1B_2B_3I_2$ … is a hierarchical structure with 5 frames in the GOP, in which B_2 is predicted using I_1 and I_2, B_1 using I_1 and B_2, and so on. It produced an average bit rate of 997,57 bits per contour.

The hierarchical structure proved to be the most effective GOP structure: the cost of the I-contours is low and the elastic prediction for B_2 is just slightly less accurate than the ones for B_1 and B_3. This result is expected, given the previous study on depth map compression with traditional hybrid techniques that shows the importance of the prediction order [29].

In Table 4, the average results for the test sequences are shown, and in every case our technique performs better than the others. The overall average gain with respect to the second best coding technique in the group, the one proposed in [17], is 6.53%. If we consider instead the gain of the proposed technique with respect to JBIG2, which is not optimized for this kind of data but has been chosen to code the boundary information in [8], it is 65.09%. Over other standard techniques, such as AAC and CBAC (with one symbol context, the best choice in our tests) the average gains are of 23.93 and 18.06%, respectively.

E) Object-based depth maps coding technique

We underline here that the goal of this paper is not to propose a complete system for MVD coding, but only to show the potential of the lossless contour coding based on elastic deformation. In this respect, we show that a relatively simple codec based on this tool may be competitive with (or even better than) the state of the art and this may be considered a validation of the proposed approach.

Coming to the implementation, we observe that it is very difficult to integrate our object-based technique into a 3DVC coder. We resort to an existing object-based technique since it can immediately benefit from an improved contour coding method, even though it is not the best option from an rate-distortion (RD) perspective. Despite the simplicity of the approach, as we will show, we have satisfying results, due to the nature of the data we want to compress. In summary, the new object-based compression technique is composed of:

- the proposed technique for the contours of the object with a hierarchical GOP structure $I_1B_1B_2B_3I_2$, as described in Section IVD. This part provides a lossless coding with inter-frame prediction;
- for the inner part of the objects we use the Shape Adaptive (SA) Wavelet Transform, followed by SA Set Partitioning In Hierarchical Trees (SPIHT), followed by an arithmetic coder [30]. We remark that for the inner part of the objects we thus have an entirely "Intra" technique.

We have chosen this technique because it is reasonable and simple, and it complements perfectly with our lossless coding technique.

To make a meaningful comparison we used HEVC Intra to compress the depth maps. We believe that the comparison is fair because in the case of HEVC Intra the arithmetic coder for the lossless coding part take advantage of the context updating, and there is no temporal prediction. Likewise, in our technique, the lossless coding part exploits the temporal redundancy, while the object coding is totally "Intra". Moreover, it would not be fair to make a comparison with HEVC Inter because we have no temporal prediction for the objects, neither would be easy to develop an object-based coder with temporal prediction.

Once defined the compression technique, we use the decoded depths to synthesize new views and make a comparison with the images generated by the uncompressed depth maps. Given any two adjacent views of the multiview sequence, we generated three equally spaced synthetic intermediate views. To test our depth maps compression technique we used the multiview sequences *ballet*, *beergarden*, *lovebird* and *mobile*, synthesizing 6, 15, 30, and 54 frames, respectively. In order to assess the quality of the virtual

views, we compared them to synthesized images obtained by applying the same DIBR algorithm to the uncompressed depths and views, and thus we obtained the RD points related to our techniques and to HEVC Intra.

Finally, we computed the Bjontegaard metrics [31] on these RD points: we observed that our technique outperforms the reference for *ballet*, *beergarden* and *mobile*, achieving respectively +2.6 dB, +0.16 dB and +0.88 dB, while the PSNR was practically identical for *lovebird*. These results, however, pertinent to quite specific test conditions (PSNR of synthesized images, limited range), show that our technique can perform at least as well as HEVC Intra, or even better depending on the sequence: these results imply that the proposed approach is worth considering. This is even more relevant in sight of the fact that in general, object-based coding techniques achieve not very good results in image compression. It has been shown that the cost of lossless contour coding is one of the elements that undermine these techniques the most [10].

V. CONCLUSIONS

In this paper, we have proposed a new technique for lossless coding of object contours for MVD. Using elastic deformation between two reference contour curves, we obtained a useful side information for the coding of the actual contour. The price payed with the coding cost for the side information is fully rewarded with significant gains with respect to the reference techniques and to the state of the art. We have also improved the technique described in [17] by substituting its prediction method with the AD method.

Moreover, a simple object-based depth map coding technique has been set up, showing that this approach can give interesting results, even if compared to state-of-the-art techniques, such as HEVC Intra.

So far, only a monodimensional elastic interpolation has been considered, but we expect a more precise estimation if we can take into account four or eight reference curves from different views and different times, thus leading to further improvements of the technique. This will be the subject of further study.

REFERENCES

[1] Marpe, D.; Wiegand, T.; Sullivan, G.J.: The H.264/MPEG4 advanced video coding standard and its applications. IEEE Commun. Mag., **44** (8) (2006), 134–143.

[2] Vetro, A.; Wiegand, T.; Sullivan, G.: Overview of the stereo and multiview video coding extensions of the H.264/MPEG-4 AVC standard. Proc. IEEE, **99** (4) (2011), 626–642 (invited paper).

[3] Guillemot, C.; Pereira, F.; Torres, L.; Ebrahimi, T.; Leonardi, R.; Ostermann, J.: Distributed monoview and multiview video coding: Basics, problems and recent advances. IEEE Signal Process. Mag., **22** (5) (2007), 67–76.

[4] Call for proposals on 3D video coding technology: ISO/IEC JTC1/SC29/WG11, Geneva, Switzerland, Technical Report, March 2011, doc. N12036.

[5] Merkle, P.; Smolic, A.; Müller, K.; Wiegand, T.: Multi-view video plus depth representation and coding. in *IEEE Int. Conf. on Image Processing*, vol. 1, 2007, 201–204.

[6] Dufaux, F.; Pesquet-Popescu, B.; Cagnazzo, M. (eds.): Emerging Technologies for 3D Video: Content Creation, Coding, Transmission and Rendering, *Wiley*, New York, 2013.

[7] Merkle, P.; Morvan, Y.; Smolic, A.; Farin, D.; Mueller, K.; Wiegand, T. *et al.*: The effects of multiview depth video compression on multiview rendering. Signal Process., Image Commun. (Elsevier Science), **24** (1) (2009), 73–88.

[8] Gautier, J.; Le Meur, O.; Guillemot, C.: Efficient depth map compression based on lossless edge coding and diffusion, in *Proc. of Picture Coding Symp.*, May 2012, 81–84.

[9] Shen, G.; Kim, W.-S.; Narang, S.K.; Ortega, A.; Lee, J.; Wey, H.C.: Edge-adaptive transforms for efficient depth map coding, in *Proc. of Picture Coding Symp.*, 2010.

[10] Cagnazzo, M.; Parrilli, S.; Poggi, G.; Verdoliva, L.: Costs and advantages of object-based image coding with shape-adaptive wavelet transform. EURASIP J. Image Video Process., **2007**, (2007), Article ID 78 323, 13 pp., doi:10.1155/2007/78323.

[11] Zhao, D.; Ren, Y.; Sun, J.; Liu, W.; Liu, J.: Depth map extraction based on geometry, in *Proc. of IEEE Southeastcon*, 2012.

[12] Joint Bi-level Image Experts Group: Information technology – progressive lossy-lossless coding of bi-level images, ISO/IEC JTC1 11544, ITU-T Recommendation T.82., 1993.

[13] Joint Bi-level Image Experts Group: Information technology – lossy-lossless coding of bi-level images, ISO/IEC JTC1 14492, ITU-T Recommendation T.88., 2000.

[14] Katsaggelos, A.; Kondi, L.P.; Meier, F.W.; Ostermann, J.; Schuster, G.M.: MPEG-4 and rate-distortion-based shape-coding techniques. Proc. IEEE, **86** (6) (1998), 1126–1154.

[15] Kim, J.I.; Bovik A.C.; Evans, B.L.: Generalized predictive binary shape coding using polygon approximation. Signal Process.: Image Commun. (Elsevier Science), **15** (7–8) (2000), 643–663.

[16] Tabus, I.; Schiopu, I.; Astola, J.: Context coding of depth map images under the piecewise-constant image model representation. IEEE Trans. Image Process., **22** (11) (2013), 4195–4210.

[17] Daribo, I.; Cheung, G.; Florencio, D.: Arithmetic edge coding for arbitrarily shaped sub-block motion prediction in depth video compression, in *Proc. IEEE Int. Conf. Image Processing*, 2012.

[18] Abou-Elailah, A.; Dufaux, F.; Farah, J.; Cagnazzo, M.; Srivastava, A.; Pesquet-Popescu, B.: Fusion of global and local motion estimation using foreground objects for distributed video coding. IEEE Trans. Circuits Syst. Video Technol., (2015), (to appear), 1–14.

[19] Srivastava, A.; Klassen, E.; Joshi, S.H.; Jermyn, I.H.: Shape analysis of elastic curves in euclidean spaces. IEEE Trans. Pattern Anal. Mach. Intell., **33** (7) (2010), 1415–1428.

[20] Mio, W.; Srivastava, A.; Joshi, S.: English on shape of plane elastic curves. English Int. J. Comput. Vis., **73** (3) (2007), 307–324, [Online], Available at: http://dx.doi.org/10.1007/s11263-006-9968-0

[21] O'Connell, K.J.: Object-adaptive vertex-based shape coding method. IEEE Trans. Circuits Syst. Video Technol., **7** (1) (1997), 251–255.

[22] Cagnazzo, M.; Antonini, M.; Barlaud, M.: Mutual information-based context quantization. Signal Process.: Image Commun. (Elsevier Science), **25** (1) (2010), 64–74.

[23] Müller, M.: Information Retrieval for Music and Motion, *Springer*, New York, 2007.

[24] Mardia, K.V.; Jupp, P.E.: Directional Statistics, *Wiley*, New York, 1999.

[25] Canny, J.: A computational approach to edge detection. IEEE Trans. Pattern Anal. Mach. Intell., PAMI–8 (6) (1986), 679–698.

[26] Chen, Y.-M.; Bajić, I.V.; Saeedi, P.: Coarse-to-fine moving region segmentation in compressed video, in *Proc. IEEE WIAMIS'09*, 2009.

[27] Chen, Y.-M.; Bajić, I.V.: Compressed-domain moving region segmentation with pixel precision using motion integration, in *Proc. IEEE PacRim'09*, 2009.

[28] MPEG4 Segmented Masks: [Online]. Available at: ftp://ftp.tnt.uni-hannover.de/pub/MPEG/mpeg4_masks/

[29] Maugey, T.; Pesquet-Popescu, B.: Side information estimation and new schemes for multiview distributed video coding. Elsevier J. Vis. Commun. Image Represent., **19** (8) (2008), 589–599.

[30] Cagnazzo, M.; Poggi, G.; Verdoliva, L.; Zinicola, A.: Region-oriented compression of multispectral images by shape-adaptive wavelet transform and SPIHT, in *Proc. IEEE Int. Conf. Image Processing*, vol. 4, Singapore, October 2004, 2459–2462.

[31] Bjontegaard, G.: Calculation of average PSNR differences between RD-curves, in *VCEG Meeting*, Austin, USA, April 2001.

Marco Calemme received a BSc degree (2007) and a MSc degree (2012) in Telecommunication Engineering from Federico II University, Napoli, Italy. Since 2012 he is pursuing a Ph.D. within the multimedia group at Telecom Paristech in Paris, France. His current research interests include video coding, compression and multiview synthesis.

Marco Cagnazzo (SM'11) obtained the Laurea (equivalent to the M.S.) degree in Telecommunication Engineering from Federico II University, Napoli, Italy, in 2002, and the Ph.D. degree in Information and Communication Technology from Federico II University and the University of Nice-Sophia Antipolis, Nice, France in 2005. He was a post-doc fellow at I3S Laboratory (Sophia Antipolis, France) from 2006 to 2008. Since February 2008 he has been Maître de Conférences (roughly equivalent to Associate Professor) at Institut Mines-Telecom, Telecom ParisTech (Paris), within the Multimedia team. He is author of more than 90 contributions in peer-reviewed journals, conferences proceedings, book chapters and standardization documents. His research interests are three-dimensional video communication and coding, distribued video coding, multiple description coding, video delivery over MANETs, network coding. Dr. Cagnazzo is an Area Editor for Elsevier Signal Processing: Image Communication and Elsevier Signal Processing. Moreover he is a reviewer for major international scientific reviews (IEEE Trans. Multimedia, IEEE Trans. Image Processing, IEEE Trans. Signal Processing, IEEE Trans. Circ. Syst. Video Tech., Elsevier Signal Processing, Elsevier Sig. Proc. Image Comm., and others) and conferences (IEEE ICIP, IEEE ICASSP, IEEE ICME, IEEE MMSP, EUSIPCO, and others).

Béatrice Pesquet-Popescu (SM'06, F'13) received the engineering degree in telecommunications from the "Politehnica" Institute in Bucharest in 1995 (highest honours) and the Ph.D. degree from the Ecole Normale Supérieure de Cachan in 1998. In 1998, she was a Research and Teaching Assistant with Université Paris XI, Paris. In 1999, she joined Philips Research France, Suresnes, France, where she worked for two years as a Research Scientist, then as a Project Leader, in scalable video coding. Since Oct. 2000 she is with Telecom ParisTech (formerly, ENST), first as an Associate Professor, and since 2007 as a Professor, Head of the Multimedia Group. She is the Head of the UBIMEDIA common research laboratory between Alcatel-Lucent and Institut Telecom. Her current research interests are in source coding, scalable, robust and distributed video compression and sparse representations. Dr. Pesquet-Popescu was an EURASIP BoG member (2003-2010), and an IEEE Signal Processing Society IVMSP TC member and MMSP TC associate member. She serves as an Associate Editor for IEEE Signal Processing Magazine, IEEE Trans. on Image Processing, IEEE Trans. on Multimedia, IEEE Trans. on CSVT, Elsevier Image Communication, and Int. J. Digital Multimedia Broadcasting journals and was till 2010 an Associate Editor for Elsevier Signal Processing. She was a Technical Co-Chair for the PCS2004 conference, and General Co-Chair for IEEE SPS MMSP2010, EUSIPCO 2012, and IEEE SPS ICIP 2014 conferences. Beatrice Pesquet-Popescu is a recipient of the "Best Student Paper Award" in the IEEE Signal Processing Workshop on Higher-Order Statistics in 1997, of the Bronze Inventor Medal from Philips Research and in 1998 she received a "Young Investigator Award" granted by the French Physical Society. She holds 23 patents in wavelet-based video coding and has authored more than 250 book chapters, journal and conference papers in the field. In 2006, she was the recipient, together with D. Turaga and M. van der Schaar, of the IEEE Trans. on Circuits and Systems for Video Technology "Best Paper Award". In 2012, she was "Associate Editor of the Year" for IEEE Trans. on Circuits and Systems for Video Technology.