

Progressive Coding of 3-D Objects Based on Overcomplete Decompositions

Ivana Tomic, *Student Member, IEEE*, Pascal Frossard, *Senior Member, IEEE*, and Pierre Vandergheynst, *Member, IEEE*

Abstract—This paper presents a progressive coding scheme for 3-D objects, based on overcomplete signal expansions on the 2-D sphere. Due to increased freedom in the basis construction, redundant expansions have shown interesting approximation properties in the decomposition of signals with multidimensional singularities organized along embedded submanifolds. We propose to map simple 3-D models on 2-D spheres and then to decompose the signal over a redundant dictionary of oriented and anisotropic atoms living on the sphere. The signal expansion is computed iteratively with a matching pursuit algorithm, which greedily selects the most prominent components of the 3-D model. The decomposition therefore inherently represents a progressive stream of atoms, which is advantageously used in the design of scalable representations. An encoder is proposed that compresses the stream of atoms by adaptive coefficient quantization and entropy coding of atom indexes. Experimental results show that the novel coding strategy outperforms state-of-the-art progressive coders in terms of distortion, mostly at low bit rates. Furthermore, since the dictionary is built on structured atoms, the proposed representation simultaneously offers an increased flexibility for easy stream manipulations. We finally illustrate that advantage in the design of a view-dependent transmission scheme.

Index Terms—Matching pursuit, progressive coding, scalable coding, sparse approximations, 3-D model compression, 3-D representation and coding.

I. INTRODUCTION

THE widespread use of 3-D data in many areas like gaming or entertainment, architecture, robotics, or medical imaging, for example, has created an essential need for efficient compression of 3-D models. Simultaneously, the increasingly large variety of decoding engines with heterogeneous capabilities and connectivity imposes a need for multiresolution representations, as well as low-complexity decoders. The most common approaches for 3-D data representation are based on polygonal meshes, which are described by both geometry (i.e., the position of vertices in space) and connectivity information, as well as optional information about normals, colors, and textures. It generally results in models built on arbitrarily defined and nonuniform grids that lead to efficient decoding performance on dedicated hardware. However, these forms of

representations stay quite voluminous and do not provide a lot of flexibility for adaptation to the requirements of specific applications or to the constraints imposed by the decoding engine.

The aim of this paper is to propose a novel coding scheme for 3-D objects, which can provide a progressive representation with flexibility in the stream manipulation, whilst achieving good compression performance. A progressive representation enables the decoder to construct a model at different resolutions, simply by proper stream truncation to meet a well-chosen rate–distortion tradeoff. At the same time, a flexible representation provides the possibility to manipulate the model in the compressed domain, to decode the model at different sizes, or from different viewpoints, for example. Towards this objective, we first propose to move away from restrictive representation techniques on nonuniform grids, by resampling 3-D data on a regular spherical grid, thus reducing the dimension of the input data into a 2-D data set. A spherical representation, defined on a regular grid, further enables the usage of 2-D signal transform coding techniques for 3-D models. A 3-D surface, which can be represented as a function on a 2-D sphere is a genus-zero¹ surface that has only one intersection point with each radial line from the center of the point cloud and thus does not contain any folds. We will refer to these models that can be mapped on the 2-D sphere as *simple genus-zero* or *star-shape* models. We eventually show that the representation of more complex models is feasible by splitting them into several spherical mappings.

Inspired by the good efficiency of discrete wavelet transforms for image compression, the first choice would be to use the spherical wavelet transform for the representation of 3-D models living on a sphere. However, similarly to contours in natural images, 3-D objects often present numerous multidimensional singularities that are organized along embedded submanifolds. It has been shown that wavelets are not optimal in representing such features like contours, because they cannot deal with the geometrical regularity of these characteristics. We therefore propose to represent the 3-D model as a series of oriented and anisotropically refined functions taken from a redundant dictionary of atoms. These atoms are edge-like functions living on the 2-D sphere, which can take arbitrary positions, shapes, and orientations. In order to capture the low-frequency components of the 3-D model, low-frequency atoms built on 2-D Gaussian functions finally complement the dictionary. We propose to use the iterative Matching Pursuit (MP) algorithm to greedily build the signal approximation. MP inherently produces a progressive stream of atoms, which can

Manuscript received October 10, 2005; revised May 14, 2006. This work was supported by the Swiss National Science Foundation, under Grant 20001-107970/1 and Grant 200021-101880/1. This paper was recommended by Associate Editor J.-N. Hwang.

The authors are with the Ecole Polytechnique Fédérale de Lausanne (EPFL), Signal Processing Institute, Lausanne 1015, Switzerland (e-mail: ivana.tomic@epfl.ch; pascal.frossard@epfl.ch; pierre.vandergheynst@epfl.ch).

Color versions of Figs. 3, 4, 6–8, 11, and 13 are available at <http://ieeexplore.org>.

Digital Object Identifier 10.1109/TCSVT.2006.883502

¹A mesh has a genus g , iff one can cut the mesh along $2g$ closed loops without disconnecting the mesh.

be decoded with a reduced complexity. A coefficient quantization step as well as an entropy coding stage for atom parameters are proposed in order to generate a flexible compressed representation of the 3-D model. The proposed encoder offers better compression performance at low rate, compared with classical 3-D compression methods, while still providing interesting scalability properties. Experimental results show that the PSNR gain of the proposed coding scheme over the state-of-the-art schemes even reaches 3 dB at low bit rate. At the same time, the MP encoder produces a completely progressive stream, which can be efficiently truncated at any arbitrary rate. We finally demonstrate the increased flexibility of the proposed representation by presenting a view-dependent coding algorithm, which is typically useful in interactive applications with scarce bandwidth resources.

This paper is organized as follows. In Section II, we give an overview of related work on 3-D model compression schemes and redundant expansions. Section III focuses on the construction of the overcomplete dictionary adapted to 3-D object properties, while in Section IV each step of the proposed 3-D object coding scheme is described in detail. Section V presents experimental results and comparisons with state-of-the-art algorithms. In Section VI, we explain how the proposed scheme is used for view-dependent applications. Finally, Section VII concludes the paper.

II. RELATED WORK

Numerous works have addressed the coding of 3-D models, and we just mention here the most relevant ones in the context of the present paper. The first mesh geometry compression scheme, introduced by Deering [1], is based on triangle strips and triangle fans and implemented in GL [2] and OpenGL [3]. In GL, triangles are ordered to form strips, whose connectivity is defined with a marching bit per triangle; it specifies to which of the two free edges of the current triangle the next triangle has to be attached. In OpenGL, triangles are attached alternatively on left and right edges, and no connectivity information is transmitted. The drawback of this technique is that most meshes have twice as many faces as vertices: each vertex has to be transmitted twice, on average.

Taubin and Rossignac later introduced the topological surgery (TS) scheme [4], which is a single-resolution manifold triangular mesh compression scheme that preserves the connectivity. After extensions to arbitrary manifold meshes, TS has become part of MPEG-4 standard. In TS, faces are interconnected by a face forest, spanning the dual graph of the mesh.² The edges that do not belong to the face forest define a vertex graph and interconnect all of the vertices of the mesh. A simple polygon connectivity mesh is obtained by cutting a mesh through a vertex graph and is eventually encoded along with the vertex graph.

In order to obtain a multiresolution representation with mesh-based coding schemes, several works proposed mesh decimation techniques, which reduce the number of triangles, vertices, and edges. They provide initially a coarse mesh model

²The dual graph of a polygonal mesh is the graph composed of the mesh faces as dual graph nodes, and the internal mesh edges as dual graph edges.

that is progressively refined by insertion of more detailed information. One of the first progressive transmission schemes for multiresolution triangular manifold meshes has been introduced by Hoppe in [5]. A triangular manifold mesh is represented by a base mesh followed by a sequence of successive vertex split refinements. Taubin has introduced the progressive forest split (PFS) scheme [6], which highly reduces the number of levels of detail and thus unnecessary information. A forest split operation is in essence described by a group of consecutive edge split operations. Together with TS, PFS represents the core of 3-D mesh coding in the MPEG-4 standard. Alternatively, Karni and Gotsman [7] have proposed a 3-D mesh compression method based on spectral decompositions, where the mesh geometry is projected onto an orthonormal basis derived from the mesh topology.

A common characteristic of multiresolution mesh-based compression schemes mentioned above is that most of the geometry information of a coarse mesh is embedded within a finer mesh, except for a set of vertices or edges that result from vertex or edge split operations. This kind of surface sampling does not necessarily lead to the best approximation at a given resolution. On the other side, by representing a 3-D model as a continuous function on a 2-D surface, positions of vertices are determined by uniform sampling of this function, hence they are different from one resolution to another. This results in equal approximation enhancement over the 3-D object surface, which is an important advantage of 2-D surface methods versus mesh-based methods. Moreover, the mapping of a 3-D object in the continuous space enables the use of various signal transformation techniques towards building fully progressive representations.

Schröder and Sweldens [8] proposed one of the earliest works that represent 3-D models as functions defined on the surface of a sphere, as an alternative to mesh-based approaches. They introduced a lifting scheme to construct bi-orthogonal spherical wavelets with customized properties. Shape compression using spherical wavelets has become recently an active area of research. The progressive coding scheme introduced by Khodakovskiy *et al.* [9] uses wavelet transform, zerotree coding and subdivision-based reconstruction to improve the compression ratio. Hoppe and Praun [10] have described a shape compression technique using spherical geometry images, which represent the surface remeshed into a regular 2-D grid. In comparison with ordinary image wavelets, spherical wavelets are shown to provide better compression performance for surfaces that can be nicely parametrized on the sphere. However, the related compression techniques suffer from rippling artifacts for surfaces with long extremities.

III. REDUNDANT REPRESENTATIONS ON THE 2-D SPHERE

A. Preliminaries

Redundant expansions have shown interesting approximation properties in the decomposition of signals with multidimensional singularities organized along embedded submanifolds, like images [11], [12]. Redundant expansions provide a lot of freedom in the design of the bases or dictionaries. In particular, it is possible to include rotation or anisotropy in the basis

functions. These two properties are keys to the development of efficient algorithms for the approximation of multidimensional signals. In such a context, separable orthogonal bases like wavelets have shown their limitations in terms of approximation rate, whilst they are optimal for 1-D continuous signals with point-like singularities.

Since 3-D models are signals composed of multidimensional features, we propose to extend the work on redundant image expansions to the overcomplete decomposition of signals defined on spherical manifolds. We decompose 3-D models as a series of atoms, taken from a redundant dictionary of functions defined on the 2-D sphere. Dictionaries are in general constructed as a set of different waveforms, where each waveform is defined by a generating function. Each generating function can serve as a base for building the overcomplete dictionary, simply by changing the function parameters or indexes (e.g., position or scale indexes). While there is *a priori* no restriction on the construction of the dictionary, the usage of generating functions advantageously leads to structured dictionaries, whose indexes directly correspond to atom characteristics. Furthermore, the storage or transmission of the dictionary become unnecessary, since atoms can be reconstructed only from their indexes.

The construction of the dictionary is certainly the most important step towards efficient approximation algorithms. Increasing the number of functions generally augments the redundancy of the dictionary, and thus the approximation performance: there is indeed an increasingly high probability that prominent signal features can be efficiently captured by a single atom. At the same time, it also increases the size of the dictionary, and most probably augments the coding rate, and the search complexity. We now discuss in more detail the construction of the overcomplete dictionary that we propose for expansions on the 2-D sphere. It involves the three following steps:

- definition of the generating function(s) on the sphere;
- definition of the motion of atoms on the sphere, and their rotation around their axis;
- implementation of the anisotropic scaling of atoms.

Since the signal to be approximated is defined in the space of square-integrable functions on a unit two-sphere S^2 , denoted as $L^2(S^2)$ (i.e., $f(\theta, \varphi) \in L^2(S^2)$), the atoms have obviously to live in the same space. Let g denote a generating function on the 2-D sphere. By combining motion, rotation, and scaling, we form an overcomplete set of atoms g_γ , where $\gamma = (\theta, \varphi, \psi, a_1, a_2) \in \Gamma$ is the atom index. This index is described by five parameters that respectively represent the position of the atom on the sphere (θ, φ) , its orientation (ψ) , and the scaling parameters (a_1, a_2) . In order to finally map the atoms on the sphere, we use an inverse stereographic projection from the complex plane to the 2-D sphere. The stereographic projection [13] at the North pole can be expressed as $\Phi : S^2 \rightarrow \mathbb{C}$, where \mathbb{C} represents the complex plane (see Fig. 1). This can be written as

$$\Phi(\omega) = \vec{v} = \rho e^{j\varphi} = 2 \tan\left(\frac{\theta}{2}\right) e^{j\varphi} \quad (1)$$

where $\omega \equiv (\theta, \varphi)$ and $0 \leq \theta \leq \pi$, $-\pi \leq \varphi \leq \pi$. Since the stereographic projection is bijective, any point with polar coordinates (ρ, φ) and represented by a vector $\vec{v} = (\rho \cos \varphi, \rho \sin \varphi)$

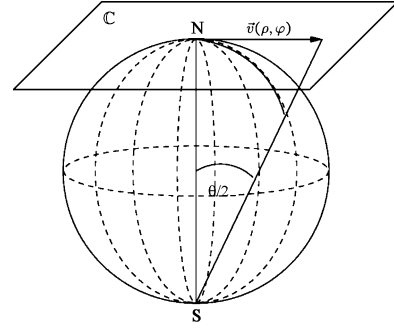


Fig. 1. Stereographic projection. A point on the 2-D sphere can be uniquely mapped on the plane tangent to the North Pole.

on the tangent plane can be uniquely mapped back onto the 2-D sphere. We use that property in the design of the dictionary, as presented below.

B. Generating Functions

Under the assumption that simple 3-D models are mostly composed of smooth surfaces and singularities aligned on pieces of great circles, we propose to build the dictionary over two generating functions. First, in order to efficiently capture the singularities, we use a generating function that resembles to a piece of contour on the sphere. In the space $L^2(\mathbb{R}^2)$, it can be written as

$$g_{\text{rect}}(\vec{v}) = -\frac{1}{K}(4x^2 - 2) \exp\left(-\frac{(x^2 + y^2)}{4}\right) \quad (2)$$

where $\vec{v} = (x, y)$ is a vector in \mathbb{R}^2 , and K is a normalization factor. Note that this function is very similar to the one that has been efficiently used for image coding in [11], [12], which is a Gaussian function in one direction and its second derivative in the orthogonal direction

$$g_{\text{image}}(\vec{v}) = \frac{1}{K}(4x^2 - 2) \exp(-(x^2 + y^2)). \quad (3)$$

The motivation for the choice of a Gaussian kernel lies in its optimal joint spatial and frequency localization. On the other side, the second derivative in the orthogonal direction is used to filter out the smooth polynomial parts of the signal and capture the signal discontinuities. However, the function defined in (2) differs from (3), in the sense that it generates longer atoms (slower decay) in the direction of Gaussian, but with the same sharp decay in the direction of its derivative. This leads to improved approximation of singularities on the 2-D sphere. The generating function from (2) can further be expressed in polar coordinates, as

$$g_{\text{rect}}(\rho, \varphi) = -\frac{1}{K}(4\rho^2 \cos^2 \varphi - 2) \exp\left(-\frac{\rho^2}{4}\right). \quad (4)$$

By inverse stereographic projection $\Phi^{-1} : \mathbb{R}^2 \rightarrow S^2$, the generating function is mapped on the sphere, and can be written as

$$g_{\text{HF}}(\theta, \varphi) = -\frac{1}{K_1} \left(16 \tan^2 \frac{\theta}{2} \cos^2 \varphi - 2\right) \cdot \exp\left(-\tan^2 \frac{\theta}{2}\right) \quad (5)$$

where K_1 is a normalization constant. The generating function g_{HF} defines an edge-like atom that is centered exactly on the North pole.

Second, in order to represent efficiently the smooth areas in the 3-D models corresponding to low-frequency (LF) components, we propose to use a second generating function for the construction of the dictionary. The second function is built on a 2-D Gaussian function in $L^2(S^2)$

$$g_{\text{LF}}(\theta, \varphi) = \exp\left(-\tan^2 \frac{\theta}{2}\right). \quad (6)$$

Equation (6) represents an isotropic function, centered at the North Pole. The extension of the dictionary to contain atoms built on two generating functions actually improves the approximation rate, but does not increase the search complexity. In our implementation, the dictionary is indeed divided into two distinct parts, one with LF atoms (LF part) and the other of oscillating or high-frequency atoms (main part), which are used successively to form the signal expansion.

C. Motion on the Sphere

Now that the generating functions have been defined, we form the redundant dictionary by applying geometrical transformations to these functions, on the 2-D sphere. In other words, the dictionary is constructed by moving the generating functions on the sphere, by rotation of the functions around their axis, and by anisotropic scaling.

Motion and rotation belong to the group of affine transformations of the unit 2-D sphere S^2 . They are both realized by a single rotation $\varrho \in SO(3)$, where $SO(3)$ is the rotation group in \mathbb{R}^3 . It is equivalent to apply the unitary operator Π_ϱ on the matrix of Cartesian coordinates (x, y, z) of the unit sphere, denoted as P

$$P_r = \Pi_\varrho P = R(\psi)U(\theta)R(\varphi)P, \quad \varrho \in SO(3) \quad (7)$$

where $\{P\}_{3 \times N}$ is the matrix of (x, y, z) coordinates of the non-transformed unit sphere, and $\{P_r\}_{3 \times N}$ is the matrix of (x, y, z) coordinates of the transformed unit sphere. Three rotation matrices $R(\varphi)$, $U(\theta)$ and $R(\psi)$ realize the rotation given by Euler angles (φ, θ, ψ) , which respectively describe the motion of the atom on the sphere by angles θ and φ and the rotation of the atom around its axis with an angle ψ . These rotation matrices are given by

$$\begin{aligned} R(\varphi) &= \begin{pmatrix} \cos \varphi & \sin \varphi & 0 \\ -\sin \varphi & \cos \varphi & 0 \\ 0 & 0 & 1 \end{pmatrix} \\ U(\theta) &= \begin{pmatrix} \cos \theta & 0 & \sin \theta \\ 0 & 1 & 0 \\ -\sin \theta & 0 & \cos \theta \end{pmatrix} \\ R(\psi) &= \begin{pmatrix} \cos \psi & \sin \psi & 0 \\ -\sin \psi & \cos \psi & 0 \\ 0 & 0 & 1 \end{pmatrix}. \end{aligned}$$

The generating function, as defined in (5), is therefore transformed into an atom that can be moved to the particular point (θ, φ) on the sphere and rotated.

D. Anisotropic Refinement of Atoms on the Sphere

In order to approximate the elongated characteristics of 3-D models, we further deform atoms by anisotropic refinement that scales the generating function differently in each orthogonal directions, with scale factors a_1 and a_2 . We perform the scaling operation on the plane tangent to the North pole and then map the resulting atom on the sphere S^2 by inverse stereographic projection as described before. Let $\vec{v} = (x, y)$ denote a vector in the tangent plane; the anisotropic scaling operator is then expressed as

$$D(a_1, a_2)g(\vec{v}) = Cg(a_1x, a_2y) \quad (8)$$

where the constant C is a normalization factor. The coordinates of the vector after scaling \vec{v}_s become

$$\begin{aligned} x_s &= a_1x = a_1\rho \cos \varphi \\ y_s &= a_2y = a_2\rho \sin \varphi. \end{aligned} \quad (9)$$

In polar coordinates, this translates into

$$\begin{aligned} \rho_s &= \sqrt{x_s^2 + y_s^2} = \rho \sqrt{a_1^2 \cos^2 \varphi + a_2^2 \sin^2 \varphi} \\ \varphi_s &= \arctan \frac{y_s}{x_s} = \arctan \frac{a_2 \sin \varphi}{a_1 \cos \varphi}. \end{aligned} \quad (10)$$

Anisotropic refinement of high-frequency atoms, as given in (2), is obtained by substitution of the polar coordinates with the ones obtained after scaling. They can be written as

$$\begin{aligned} g_{\text{rect}}(\rho, \varphi) &= -\frac{1}{K} (4a_1^2\rho^2 \cos^2 \varphi - 2) \\ &\cdot \exp\left(-\frac{\rho^2 (a_1^2 \cos^2 \varphi + a_2^2 \sin^2 \varphi)}{4}\right). \end{aligned} \quad (11)$$

By inverse stereographic projection $\Phi^{-1} : \mathbb{R}^2 \rightarrow S^2$, the reshaped atom is mapped on the sphere and can be written as

$$\begin{aligned} g_{\text{HF}}(\theta, \varphi) &= -\frac{1}{K_2} \left(16a_1^2 \tan^2 \frac{\theta}{2} \cos^2 \varphi - 2\right) \\ &\cdot \exp\left(-\tan^2 \frac{\theta}{2} (a_1^2 \cos^2 \varphi + a_2^2 \sin^2 \varphi)\right) \end{aligned} \quad (12)$$

where K_2 is a normalization factor. On the other hand, the low-frequency atoms after anisotropic refinement, can be written as

$$g_{\text{LF}}(\theta, \varphi) = \exp\left(-\tan^2 \frac{\theta}{2} (a_1^2 \cos^2 \varphi + a_2^2 \sin^2 \varphi)\right). \quad (13)$$

In summary, the dictionary is obtained by anisotropic scaling, rotation, and displacement of atoms, which are implemented by first transforming the unit sphere with respect to rotations and then evaluating the generating function on the transformed sphere. Anisotropic scaling is finally applied on the displaced atom, as explained above. Such a process is performed for both

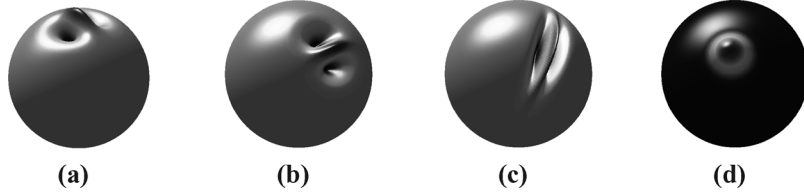


Fig. 2. Anisotropic atoms (a) on the North pole ($\theta = 0, \varphi = 0$), $\psi = 0, a_1 = 8, a_2 = 8$, (b) $\theta = (\pi/4), \varphi = (\pi/2), \psi = 0, a_1 = 8, a_2 = 8$, (c) $\theta = (\pi/4), \varphi = (\pi/2), \psi = (\pi/4), a_1 = 16, a_2 = 4$. (d) Low-frequency atom: $\theta = (\pi/4), \varphi = (\pi/4), \psi = (\pi/4), a_1 = 8, a_2 = 8$.

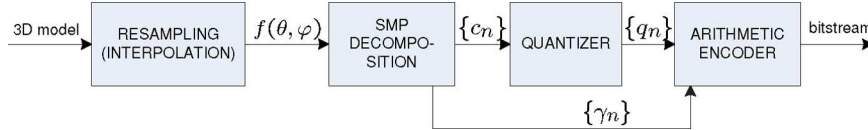


Fig. 3. 3D-SMP encoding scheme.

oscillating atoms as expressed in (12) and for low-frequency ones given in (13). By a proper choice of the transformation parameters, one finally obtains an overcomplete dictionary of functions, which is used to represent simple 3-D models. Sample atoms are illustrated in Fig. 2.

IV. MATCHING PURSUIT ENCODER FOR 3-D OBJECTS

A. Matching Pursuit Overview

Finding the sparsest representation of a simple 3-D model, with functions taken from a redundant dictionary, as the one described before, is in general an NP-hard³ problem. MP [14] provides a solution to this approximation problem, with a tractable computational complexity. Under its generic form, MP is an algorithm that iteratively decomposes a signal into a linear combination of waveforms or atoms. Interestingly, very few restrictions are imposed on the dictionary construction, besides the fact that it should at least span the space of the signal to represent. In other words, the dictionary is defined as a set of vectors $\mathcal{D} = \{g_\gamma\}_{\gamma \in \Gamma}$ in a Hilbert space H . In order to be able to represent each vector in H as a linear combination of unit norm vectors in \mathcal{D} , the dictionary must satisfy the completeness property (i.e., \mathcal{D} spans H).

Let $f \in H$ denote a function that we want to approximate with a linear expansion over \mathcal{D} . With MP, an M -term linear expansion is obtained by successive approximations of $R^n f$ through orthogonal projections on dictionary vectors

$$f = \sum_{n=0}^M \langle R^n f, g_{\gamma_n} \rangle g_{\gamma_n} + R^{M+1} f \quad (14)$$

where $R^n f$ is the residue after $n - 1$ iterations of the algorithm ($R^0 f = f$). One must choose, at each iteration, the atom that best approximates $R^n f$, with the maximal projection $|\langle R^n f, g_{\gamma_n} \rangle|$ over the dictionary

$$|\langle R^n f, g_{\gamma_n} \rangle| = \sup_{\gamma \in \Gamma} |\langle R^n f, g_\gamma \rangle|. \quad (15)$$

³A problem is NP-hard if an algorithm for solving it can be translated into an algorithm for solving any other NP (nondeterministic polynomial time) problem.

When $M \rightarrow \infty$, under the assumption that the dictionary is complete, it can be shown that

$$f = \sum_{n=0}^N \langle R^n f, g_{\gamma_n} \rangle g_{\gamma_n}. \quad (16)$$

Under the same assumption, it has also been proven that the residue decays exponentially in a finite dimensional space [14]. The decay rate depends on the correlation between the residue and the dictionary elements, so that the construction of an efficient dictionary, adapted to the structure of the signal f , is a crucial step.

Overall, MP offers a suboptimal solution to the optimal (sparsest) signal representation problem, since it iteratively approximates the signal, in a totally greedy manner. However, it allows for an efficient approximation of the signal by rapidly capturing its most important components, which is an interesting property towards the design of a scalable coder. At the same time, it does not impose any condition on the dictionary construction, and the complexity at decoder is kept small.

B. 3D-SMP Encoder

Our objective is to build on the nice approximation properties offered by redundant expansions in order to obtain compressed versions of 3-D models. The block diagram of the proposed encoder, which is a 3-D model encoder based on spherical MP (3D-SMP), is represented in Fig. 3. MP selects a series of atoms from the dictionary described above with their relative magnitude of their coefficients. Atoms are then sorted along the decreasing magnitude of their coefficients. The coefficients are then uniformly quantized, with a decaying quantization range. It takes advantage of the property that the energy of MP coefficients is limited by an exponentially decaying upper bound. A piecewise linear approximation of that upper bound is used for quantization, inspired from the scheme proposed in [15]. Quantized coefficients and discrete atom indexes are finally encoded with an arithmetic coder [16] in order to obtain a compact representation. Interested readers are referred to [12] for more details about quantization and entropy coding of MP atoms.

The initial block of the codec extracts a set of vertices $p_i = (x_i, y_i, z_i) \in I, I \subset \mathbb{R}^3$, which represents a point cloud of a 3-D

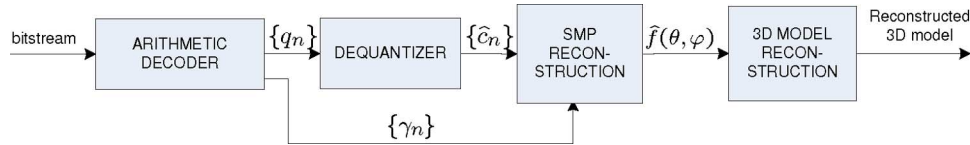


Fig. 4. 3D-SMP decoding scheme.

model. It generally describes a set of nonuniformly spaced samples on the 2-D sphere S^2 and defines a function $f : S^2 \rightarrow \mathbb{R}$ as $R_i = f(\theta_i, \varphi_i)$. Since the proposed coding scheme requires the spherical data to be sampled on an equiangular (θ, φ) grid, an interpolation step may be needed. Because of its low complexity, we have chosen a simple nearest-neighbor interpolation method, where each value on the equiangular spherical grid R_{int} is interpolated as an average of its four nearest neighbors. In addition to enabling the use of processing algorithms like the fast spherical Fourier transform, an equiangular spherical grid has a regular structure that can be exploited at the decoder for regenerating the mesh connectivity removed by the encoding process.

C. 3D-SMP Decoder

On the other end, the decoder, as represented in Fig. 4, first performs the entropy decoding and inverse quantization. It then reconstructs the approximated 3-D model as a spherical function by linear combination of atoms whose relative weights are given by the MP coefficients. The reconstruction step at the decoder side has a quite low computational complexity, roughly proportional to the number of atoms (and hence the coding rate).

The decoder then generates the decoded 3-D object in the form of a standard polygonal mesh, as accepted by all modern computer graphics application and hardware. Since the encoder has completely discarded the mesh connectivity information of the original 3-D model, the decoder has to generate new connectivity. This problem can be formulated as a surface reconstruction problem from an unorganized point cloud, which is still an active area of research, and many surface reconstruction algorithms already exist (e.g., [17]). Since we are primarily dealing with simple models parameterized as one spherical function, we can use the *a priori* knowledge of the (θ, φ) coordinates for each vertex on the spherical grid and construct a *semi-regular* connectivity structure. A mesh with semi-regular connectivity has almost all vertices of valence 6 (i.e., six incident edges), except for a few isolated extraordinary vertices. The connectivity matrix is defined with indexes of three incident vertices for each face. In order to obtain a semi-regular triangular mesh, we can divide the spherical grid into rings limited by two successive values of θ and then triangulate each ring to produce a triangular strip. Such a mesh construction is illustrated in Fig. 5(a), which shows the triangular subdivision of the sphere. Fig. 5(b) represents the same grid but applied to the Venus model. All vertices are of valence 6, except for the two poles, so that the resulting mesh is indeed semi-regular.

For more complex models whose representation requires multiple spheres, the method explained above is not directly applicable, since the boundary between two neighboring spheres does not necessarily coincide with a great circle on the sphere. In these cases, a simpler solution would be to use a

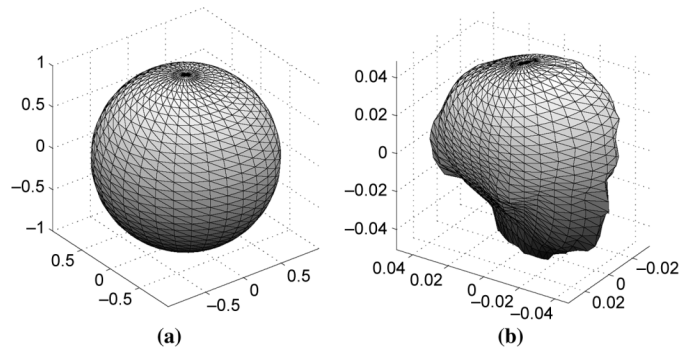


Fig. 5. Generating connectivity matrix. (a) Sphere connectivity. (b) Connectivity on the Venus model.

more generic surface reconstruction algorithm. The proposed 3D-SMP scheme uses the algorithm explained in [18].⁴

V. EXPERIMENTAL CODING RESULTS

A. Preliminaries

Two models are used in our experiments: Venus and Rabbit.⁵ Venus satisfies the assumption of a simple genus-zero model; thus, it is represented via one spherical function. Rabbit is not a simple model, and we decompose it into three spheres separated by two parallel planes: one below the head and the other below the arms of the Rabbit. Each spherical function is obtained by interpolation within the point cloud on the corresponding part of the model. Afterwards, SMP is independently run on each of these three spheres and finally gathered into a single decomposition by ordering the atoms in a decaying order of their coefficient values. One additional atom parameter is introduced to denote the sphere that the atom belongs to. The quantization and entropy coding steps are the same as for the one-sphere decompositions. Finally, three spherical functions are reconstructed at the decoder, and point clouds are merged using a surface reconstruction algorithm, as explained in Section IV-C. The original and interpolated models are shown in Fig. 6.

It should be noted that the PSNR values obtained after interpolation are actually the upper bounds for the overall quality, since the interpolated models are given as inputs to the 3D-SMP compression scheme. The interpolation error is expressed with relative L^2 error and in PSNR (dB), as computed with the MESH software.⁶ The relative L^2 error is actually a ratio of the rms, which measures the squared symmetric distance between two surfaces averaged over the first surface, relative to

⁴The reconstruction server is available at <http://cgai.inria.fr/Reconstruction/submit.html>.

⁵The models have been downloaded from <http://www.cyberware.com>.

⁶MESH is available at <http://mesh.epfl.ch>.

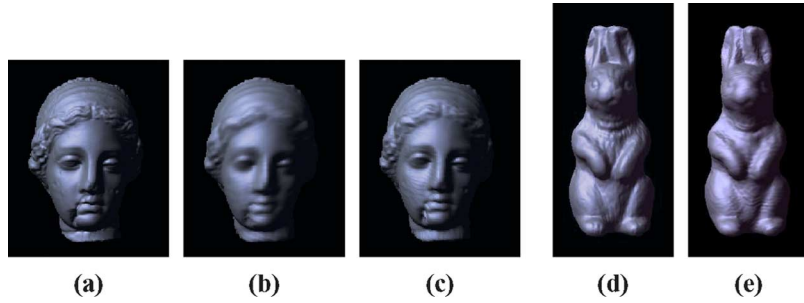


Fig. 6. (a) Original Venus model. (b) Interpolated Venus model at resolution $N_\theta \times N_\varphi = 128 \times 128$, PSNR = 65.7983 dB, $L^2 = 5.1296 \cdot 10^{-4}$. (c) Interpolated Venus model at resolution $N_\theta \times N_\varphi = 256 \times 256$, PSNR = 70.1930 dB, $L^2 = 3.09278 \cdot 10^{-4}$. (d) Original rabbit model. (e) Interpolated Rabbit model on three spheres at resolution $N_\theta \times N_\varphi = 128 \times 128$, PSNR = 64.1790 dB, $L^2 = 6.18091 \cdot 10^{-4}$.

a bounding box diagonal d . The PSNR for 3-D meshes is thus expressed as

$$\text{PSNR (dB)} = 20 \log \left(\frac{d}{\text{RMS}} \right) = 20 \log \left(\frac{1}{L^2} \right). \quad (17)$$

B. Implementation

In the dictionary presented in Section III, the atom indexes, which represent the parameters of the generating functions, obviously take discrete values. In general, a fine granularity in atom indexes leads to high redundancy and, most likely, to a high approximation rate. At the same time, it leads to a large dictionary with possibly high coding cost. The design of an optimal dictionary is still an open problem that is beyond the scope of this paper. Here, we propose to use a dictionary built mostly on empirical choices for atom parameter values. First, we use the equiangular spherical grid to drive the values of the position parameters θ and φ ; both parameters are uniformly distributed on the interval $[0, \pi]$ and $[-\pi, \pi)$, respectively, with a resolution that is identical to the input signal. The rotation parameter ψ is uniformly sampled on the interval $[-\pi, \pi)$, with the same resolution as θ and φ . This choice is mostly due to the use of spherical convolution in the MP algorithm, as explained below. Finally, scaling parameters are distributed in a logarithmic manner, from 1 to half of the resolution of the input signal, with a granularity of one third of octave. Due to the definition of the atoms [see (12)], scaling parameters are inversely proportional to the size of the atoms. The largest atom has a scale 1 and it covers half of the sphere. For low-frequency atoms, the maximal value for the scaling parameters is chosen to be 1/16 of the signal resolution. Motion and rotation parameters are discretized in the same way as for anisotropic atoms.

In our implementation, the full dictionary is divided into low-frequency atoms g_{LF} , and high-frequency ones. During the first iterations, the MP algorithm uses the low-frequency subdictionary, and later switches to the anisotropic subdictionary when the energy of the coefficients starts to saturate or, more precisely, when

$$\frac{|C^n|}{\|R^n\|_2} \rightarrow \text{const} \quad (18)$$

where C^n denotes a projection after $n - 1$ iterations. In each of these subdictionaries, the MP algorithm performs a full search

to determine the highest energy atom. Our implementation uses the fast spherical transform [19] to compute the convolution of atoms on the sphere. In particular, we used in our implementation the *SpharmonicKit* library,⁷ which is part of the *YAW toolbox*.⁸ Such a transform allows to identify the position and rotation of an atom on the sphere, which has the best correlation with the signal. One spherical convolution allows to determine the parameters (θ, φ, ψ) for each atom with given scale parameters. Therefore, our implementation iterates over the scale parameters: for each couple (a_1, a_2) , it computes the spherical convolution between the corresponding atom on the North Pole and the residual signal $R^n f$. The convolution coefficient with the largest magnitude corresponds to the position and rotation parameters (θ, φ, ψ) of the best matching atom for that pair of scales. The coefficient of the corresponding atom is computed with the inner product of two functions defined on the sphere given by

$$\langle R^n f, g \rangle = \int_{\theta} \int_{\varphi} R^n f(\theta, \varphi) g(\theta, \varphi) \sin \theta d\theta d\varphi. \quad (19)$$

Finally, the algorithm selects, among all pairs of scales, the atom with the largest coefficient, removes its contribution from the residual signal and repeats the whole procedure until a stopping criteria is met (e.g., a predefined number of atoms, or an energy threshold). The search algorithm is described in Algorithm 1.

Algorithm 1: Full search of the dictionary

for all scale couples $(a_1(j), a_2(k))$ **do**

$$C = \text{convspsh}(R^n f, g(0, 0, 0, a_1(j), a_2(k)))$$

$$C_{\max} = \max_{1 < m < N_\theta, 1 < n < N_\varphi, 1 < p < N_\psi} C(m, n, p)$$

$$\theta = \theta(m); \varphi = \varphi(n); \psi = \psi(p)$$

$$P(j, k) = \langle R^n f, g(\theta, \varphi, \psi, a_1(j), a_2(k)) \rangle$$

end for

$$P_{\max} = \max_{j,k} (P(j, k))$$

$$a_1 = a_1(j); a_2 = a_2(k)$$

⁷[Online]. Available: <http://www.cs.dartmouth.edu/~geelong/sphere/>.

⁸[Online]. Available: <http://fyma.fyma.ucl.ac.be/projects/yawtb/>.

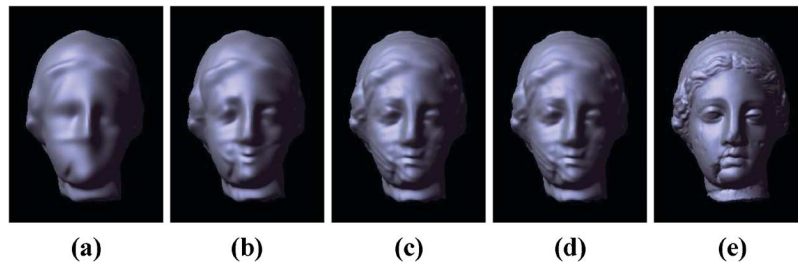


Fig. 7. Venus reconstructed after decoding (resolution 256×256). (a) 100 coefficients (0.58 kB). (b) 200 coefficients (1.03 kB). (c) 300 coefficients (1.5 kB). (d) 400 coefficients (1.94 kB). (e) Input model.

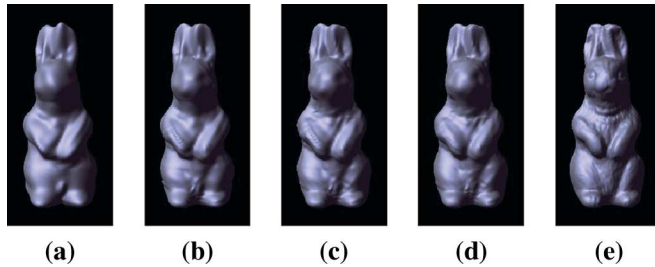


Fig. 8. Rabbit reconstructed after decoding (three spheres, resolution 128×128). (a) 100 coefficients (0.64 kB). (b) 200 coefficients (1.05 kB). (c) 300 coefficients (1.45 kB). (d) 400 coefficients (1.84 kB). (e) Input model.

Note that a full-search MP algorithm can become quite complex when the dictionary size is large. However, the search complexity can be significantly reduced by efficient arrangement of atoms [20] or parallelization [21], possibly at the expense of a small decrease in the approximation rate. The decoding complexity is however very low and is performed in linear time.

C. Numerical Results

Venus and Rabbit models, as reconstructed by the decoder, are represented in Figs. 7 and 8, respectively, for different numbers of atoms. It can be seen that MP rapidly captures the most important features of the 3-D model and progressively refines the representation with finer details. The type of coding artifacts is quite different than the degradations observed in mesh-based coders and is visually less annoying at low rates. It can be seen also that the gain in representation accuracy is less important when the number of iterations increases. As expected, MP is mostly efficient for low-bit-rate compression.

Figs. 9 and 10 present the rate–distortion performance of the proposed 3D-SMP algorithm for the Venus and Rabbit models in terms of both (a) PSNR and (b) L^2 error. Distortion values are evaluated with respect to the original models, meaning that they take into account both the interpolation and the decoding error. These figures compare the 3D-SMP encoder performance with the following state-of-the-art encoders: 1) TG: Touma–Gotsman nonprogressive coding [22]; 2) Alliez–Desbrun progressive coding [23]; and 3) PGC: progressive coding scheme by Khodakovskiy *et al.* [9]. Due to the differences in input formats and coding approaches, we use the following method to obtain fair performance comparisons between these four very different approaches. As PGC uses its own mesh format, the input models are downloaded from

the PGC website.⁹ The base mesh for the PGC is encoded using TG with 8 bits per vertex. The input models for TG and Alliez–Desbrun methods are the models that have been used to obtain the interpolated version for 3D-SMP encoder,¹⁰ but decimated to 1400 faces (using the Qslim software¹¹), in order to have a comparison in the same rate region. Different rates for the TG algorithm are obtained by changing the number of bits per vertex for encoding. Note that the rate is actually given by the filesize, or the total number of bits, rather than in bits per vertex, since the proposed 3D-SMP coding scheme uses one single mesh (256×256 or 128×128 vertices in the current implementation).

It can be seen that 3D-SMP significantly outperforms the state-of-the-art compression methods TG and Alliez–Desbrun, as well as the PGC wavelet-based coder at low bit rates. MP then tends to saturate towards high bit rates, as was observed earlier. For the Venus model, the performance is slightly better than that for the Rabbit model. This behavior is actually expected since the resolution of the employed SMP for Rabbit is smaller. Therefore, we can certainly expect better performance for the Rabbit model at higher resolutions. It has to be noted also that the input model for the SMP is an interpolated version of the original model, and this introduces a distortion that is independent of the coding method. Fig. 11 shows a visual comparison of Venus encoded with 3D-SMP using 250 coefficients and resolution 256×256 , and encoded with the the PGC algorithm, for the same filesize of 1287 B. It can be seen that both coders offer similar performance, but the coding artifacts are quite different. The 3D-SMP coder generally provides a smoother approximation of the model, but fails to capture the highly textured regions like the hair, for example. The proposed encoder offers an interesting alternative to classical approaches, with excellent compression performance at low bit rates and, at the same time, an inherently progressive representation. Additionally, it offers a great flexibility in the stream construction, which can be advantageously exploited in adaptive applications, like view-dependent rendering, which is presented in Section VI.

VI. VIEW-DEPENDENT RENDERING

This section proposes a simple application that uses the increased flexibility offered by the use of a structured dictionary of functions, in addition to the progressive nature of the stream

⁹[Online]. Available: <http://www.multires.caltech.edu/software/pgc/>.

¹⁰[Online]. Available: <http://www.cyberware.com/samples/index.html>.

¹¹[Online]. Available: <http://graphics.cs.uiuc.edu/~garland/software/qslim.html>.

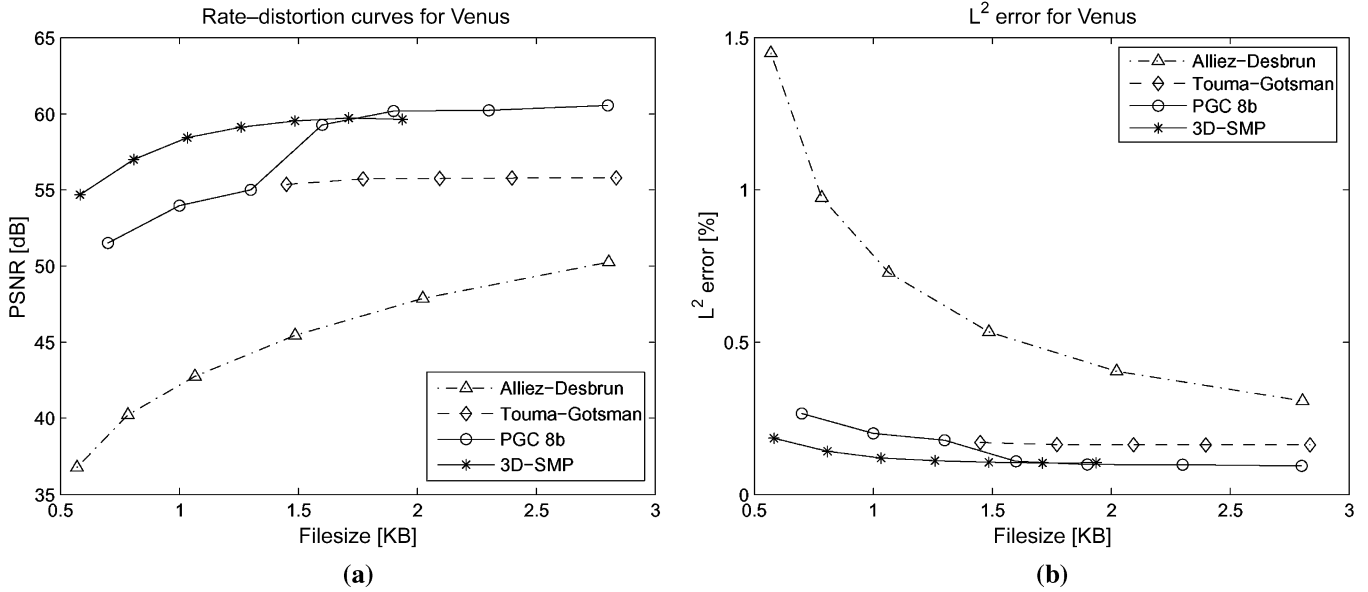


Fig. 9. Rate-distortion performance for the Venus model (resolution 256 × 256). (a) PSNR. (b) L² error.

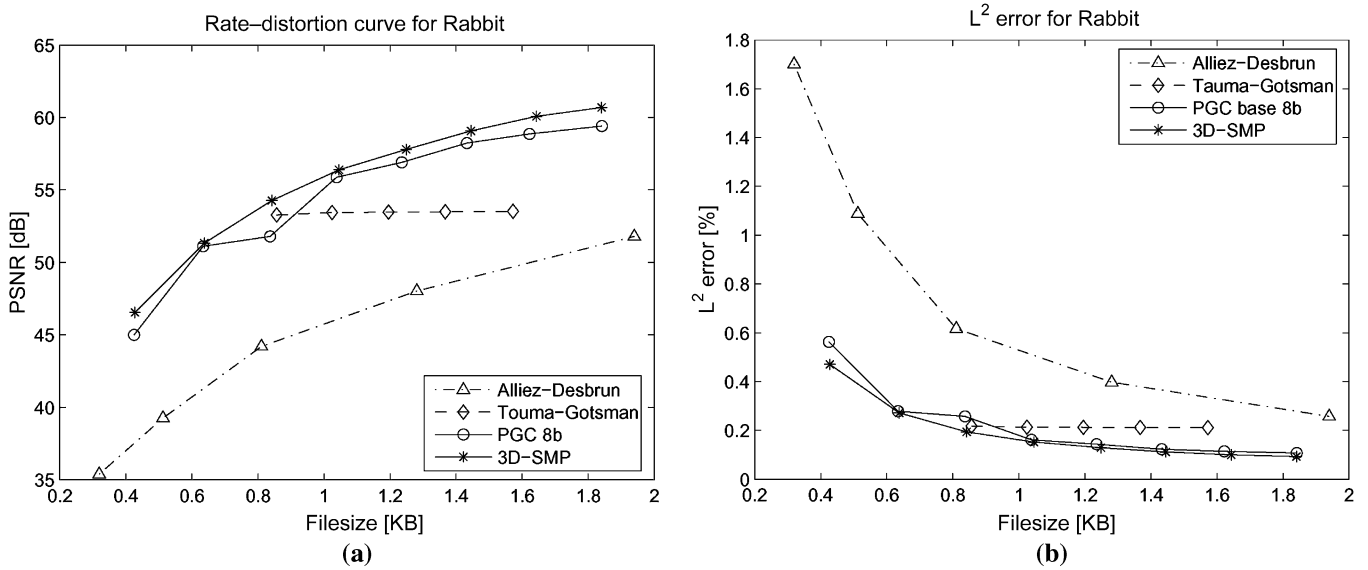


Fig. 10. Rate-distortion performance for the Rabbit model (resolution 128 × 128). (a) PSNR. (b) L² error.

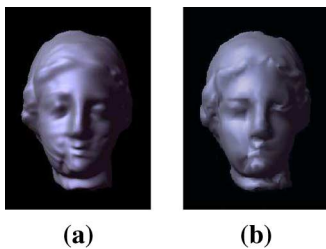


Fig. 11. Venus with a filesize of 1287 B. (a) 3D-SMP. (b) PGC.

generated by the MP decomposition. In scenarios where very high compression is necessary or when the transmission channel represents an important bottleneck, it becomes interesting to use view-dependent rendering algorithms. In such cases, the 3-D model does not need to be completely downloaded and decoded

before being displayed, and the rendering can be made to be dependent on the viewpoint. The server transmits in priority the parts of the model that are visible while cutting off the invisible parts. View-dependent progressive transmission and rendering can therefore significantly improve the performance of 3-D graphics streaming applications.

All of the previous work done in this area uses classical 3-D multiresolution coding techniques based on mesh simplification algorithms by vertex split operations. For example, Yang *et al.* [24] split the 3-D model into progressively encoded partitions, which are reorganized so that the visible ones are transmitted with higher priority. Zach *et al.* [25] proposed a view-dependent mesh connectivity encoding scheme which reorganizes the vertex tree, while Wang and Li [26] employed an absolute path coding of vertices. Recently, Bischoff and Rossignac introduced the *TetStreamer* [27], which performs

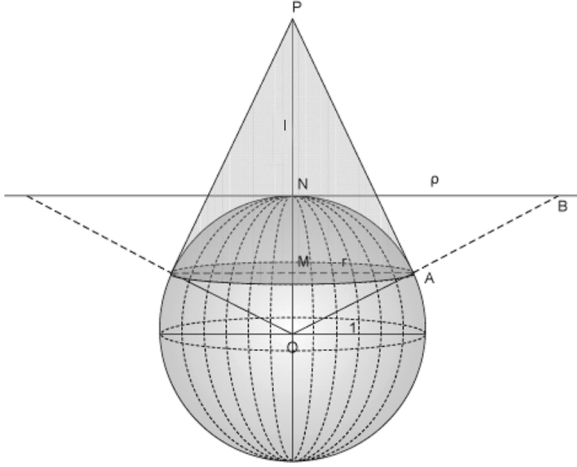


Fig. 12. Intersection of the unit sphere and the conical surface originating from the viewpoint P .

a view-dependent front-to-back (in the visibility order) encoding of 3-D models. However, we note that none of the mesh geometry compression schemes based on wavelets [8]–[10] has addressed this issue, even if some solutions inspired from region-of-interest coding in images could be envisaged.

The use of a structured dictionary in the proposed MP encoder presents a great advantage for manipulation of the compressed stream. In particular, it allows for a quite simple extension toward view-dependent transmission and rendering. Since a 3-D model is decomposed into a linear combination of atoms specified by their position and scale parameters, it can be partially transmitted by selecting the atoms visible only from a given viewpoint. Assume that the viewpoint can be associated with a window visibility defined on the unit sphere (i.e., $w(\theta, \varphi) \in L^2(S^2)$). Such a window function on the sphere is obtained by an inverse stereographic projection of a circle on the tangent plane, with the center at the North Pole and the radius ρ . The window function can be expressed as

$$w(\theta, \varphi) = \begin{cases} 1, & \text{if } 2 \tan\left(\frac{\theta}{2}\right) \leq \rho, -\pi \leq \varphi \leq \pi; \\ 0, & \text{otherwise.} \end{cases}$$

This is illustrated in Fig. 12, where the tangent plane is shown as a line passing through the North Pole N . The part of the sphere that is visible from the viewpoint is delimited by a conical surface formed by the union of all of the straight lines that pass through the viewpoint and are tangent to the unit sphere. This part is shown in Fig. 12 as the darker shaded part of the sphere. Let r denote the radius of the circle obtained by intersecting the cone and the sphere ($r = \overline{MA}$). From the similarity of triangles $\triangle OPA$ and $\triangle OAM$, it follows that

$$r = \frac{\sqrt{l^2 - 1}}{l} \quad (20)$$

where $l = \overline{OP}$. From the similarity of triangles $\triangle OAM$ and $\triangle OBN$, we obtain

$$\rho = \frac{r}{\sqrt{1 - r^2}} = \sqrt{l^2 - 1}. \quad (21)$$

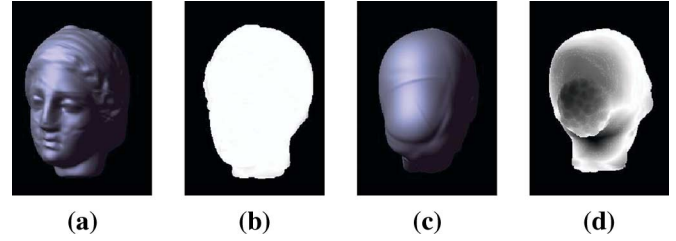


Fig. 13. View-dependent coding of Venus, $\sigma = r$; $\delta = 1\%$; 42.8% of the model is transmitted (214 out of 500 atoms). (a) Visible part of the decoded model. (b) Difference between the original model and the decoded model for the visible part, (c) Invisible part of the decoded model. (d) Difference between the original model and the decoded model for the invisible part.

The radius ρ is therefore directly determined by the distance between the viewpoint and the 3-D model.

To generate a view-dependent reconstruction of the 3-D model, each atom g_{γ_i} , $i = 1, \dots, M$, in the signal decomposition is simply multiplied with the window function. This results in a windowed atom $g_{\gamma_i}^w$, which represents the part of the atom visible from the viewpoint P

$$g_{\gamma_i}^w(\theta, \varphi) = g_{\gamma_i}(\theta, \varphi) \cdot w(\theta, \varphi), \quad i = 1, \dots, M. \quad (22)$$

Only the atoms that keep a significant contribution after windowing are finally considered for view-dependent transmission and rendering. The selection of atoms is based on the comparison of their maximal value, with a predefined relative threshold δ . If there exists a pair (θ, φ) such that

$$g_{\gamma_i}^w(\theta, \varphi) \max_{(\theta', \varphi')} g_{\gamma_i}(\theta', \varphi') > \delta$$

then the atom g_{γ_i} is transmitted.

It has to be noted that the uniform window proposed above obviously attributes the same importance to all of the atoms that are visible from the viewpoint P . However, other windows are straightforward to implement. For example, if one wants to emphasize the importance of atoms closer to the view direction (\overline{OP}), a Gaussian window can be used, as given by

$$w_g(\theta, \varphi) = \begin{cases} \exp\left(\frac{-4 \tan^2 \frac{\theta}{2}}{2\sigma^2}\right), & \text{if } 2 \tan \frac{\theta}{2} \leq \rho, -\pi \leq \varphi \leq \pi \\ 0, & \text{otherwise.} \end{cases}$$

Fig. 13 presents the results of view-dependent coding and transmission, applied to the Venus model. The threshold δ is set to 1% of the nonwindowed atom peak value, and the variance of the Gaussian window function is set to $\sigma = r$. It can be seen that the difference between the original model and the decoded model is very small in the visible parts of the model, resulting mostly from the error due to lossy coding. On the other hand, the difference between the original model and the decoded model is much bigger in the parts of the model which are not visible from the given viewpoint. Hence, the emphasis is correctly put on the part of the model that is visible, as expected. Furthermore, the view-dependent transmission of atoms allows for a significant decrease in required bit rate, since only 42.8% of the model has been transmitted in the scenario under consideration.

By increasing σ , more precise view-dependent encoded models are obtained as more atoms are transmitted. This of

course implies that the bit rate is also increased. A change of σ can be therefore used for the coarse tuning of the rate–distortion tradeoff. For a finer adaptation, the threshold value δ can be changed. As the viewpoint may change during the interactive 3-D model rendering, the rest of the atoms can be progressively transmitted in lower priority, until the complete scene is received. Finally, the view-dependent rendering by atom windowing is a simple algorithm that allows to prioritize atoms from the MP decomposition. Obviously, more sophisticated methods could be proposed with application-specific requirements, yet taking advantage of the structured nature of the redundant dictionary used in the proposed MP encoder.

VII. CONCLUSION AND FUTURE WORK

This paper proposed a novel approach for the coding of 3-D objects. The models are mapped on a 2-D sphere and decomposed over a redundant dictionary of multidimensional atoms, which is built in order to efficiently capture the most prominent features of the signal. An encoding algorithm based on MP has been used to generate progressive representations of the 3-D objects. The proposed encoder has been shown to outperform state-of-the-art progressive coders, especially at low bit rates. At the same time, it offers a truly progressive representation with increased flexibility in the stream manipulation. The structured nature of the dictionary has been advantageously used in the design of a simple view-dependent rendering application, which may prove to be useful in services with important transmission bottlenecks. The proposed algorithm still leaves numerous possibilities for future improvements. The interpolation step, which uses a very basic nearest-neighbor method, has an important influence that penalizes the compression performance. We are also currently working on better parametrization of input models and on generalizing the multiple spheres decomposition approach for the coding of more complex 3-D models. Finally, since it has been shown that redundant decompositions are mostly beneficial at low rates, the proposed scheme can offer an efficient coding solution for the base layer in scalable applications. Similarly to hybrid image coding schemes [28], enhancement layers based on spherical wavelets, for example, could nicely complement the proposed scheme for high-bit-rate coding.

ACKNOWLEDGMENT

The authors would like to thank Dr. N. Aspert for his valuable help in implementation of the 3D-SMP and Dr. R. Maria Figueras i Ventura for helpful discussions on the Matching Pursuit coding scheme.

REFERENCES

- [1] M. Deering, "Geometric compression," in *Proc. ACM Siggraph*, Aug. 1995, pp. 13–20.
- [2] "GL Programming Guide," Silicon Graphics Inc., 1991.
- [3] J. Neider, T. Davis, and M. Woo, *OpenGL Programming Guide*. Reading, MA: Addison-Wesley, 1997.
- [4] G. Taubin and J. Rossignac, "Geometry compression through topological surgery," *ACM Trans. Graphics*, vol. 17, no. 2, pp. 84–115, Apr. 1998.
- [5] H. Hoppe, "Progressive meshes," in *Proc. ACM Siggraph*, Aug. 1996, pp. 99–108.
- [6] G. Taubin, A. Guéziec, and W. Horn, "Progressive forest split compression," in *Proc. ACM Siggraph*, Jul. 1998, pp. 123–132.

- [7] Z. Karni and C. Gotsman, "Spectral compression of mesh geometry," in *Proc. ACM Siggraph*, Jul. 2000, pp. 279–286.
- [8] P. Schröder and W. Sweldens, "Spherical wavelets: Efficiently representing functions on the sphere," in *Proc. ACM Siggraph*, Aug. 1995, pp. 161–172.
- [9] A. Khodakovsky, P. Schröder, and W. Sweldens, "Progressive geometry compression," in *Proc. ACM Siggraph*, Jul. 2000, pp. 271–278.
- [10] H. Hoppe and E. Praun, "Shape compression using spherical geometry images," in *Proc. Symp. Multiresolution Geometric Modeling*, Cambridge, U.K., Sep. 2003, pp. 27–46.
- [11] P. Vanderghyest and P. Frossard, "Efficient image representation by anisotropic refinement in matching pursuit," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process.*, Salt Lake City, UT, May 2001, vol. 3, pp. 1757–1760.
- [12] R. M. Figueras i Ventura, P. Vanderghyest, and P. Frossard, "Low rate and flexible image coding with redundant representations," *IEEE Trans. Image Process.*, vol. 15, no. 3, pp. 726–739, Mar. 2006.
- [13] J. Antoine and P. Vanderghyest, "Wavelets on the 2-sphere: a group theoretical approach," *Appl. Computat. Harmonic Anal.*, vol. 7, no. 3, pp. 1–30, Nov. 1999.
- [14] S. G. Mallat and Z. Zhang, "Matching pursuits with time-frequency dictionaries," *IEEE Trans. Signal Process.*, vol. 41, no. 12, pp. 3397–3415, Dec. 1993.
- [15] P. Frossard, P. Vanderghyest, R. M. Figueras i Ventura, and M. Kunt, "A posteriori quantization of progressive matching pursuit streams," *IEEE Trans. Signal Process.*, vol. 52, no. 2, pp. 525–535, Feb. 2004.
- [16] I. H. Witten, R. M. Neal, and J. G. Cleary, "Arithmetic coding for data compression," *Commun. ACM*, vol. 30, no. 6, pp. 520–540, Jun. 1987.
- [17] H. Hoppe, T. DeRose, T. Duchamp, J. McDonald, and W. Stuetzle, "Surface reconstruction from unorganized points," in *Proc. ACM Siggraph*, Jun. 1992, pp. 71–78.
- [18] D. Cohen-Steiner and F. Da, "A Greedy Delaunay Based Surface Reconstruction Algorithm" INRIA, ECG Tech. Rep. ECG-TR-124202-01, Dec. 2001.
- [19] J. R. Driscoll and D. M. Healy, Jr., "Asymptotically fast algorithms for spherical and related transforms," in *Proc. 30th Annu. Symp. Foundations Comput. Sci.*, Oct. 1989, pp. 344–349.
- [20] P. Jost, P. Vanderghyest, and P. Frossard, "Tree-based pursuit: Algorithm and properties," *IEEE Trans. Signal Process.*, to be published.
- [21] A. Rahmoune, P. Vanderghyest, and P. Frossard, "The M-term pursuit for image representation and progressive compression," in *Proc. IEEE Int. Conf. Image Process.*, Sep. 2005, vol. 1, pp. 73–76.
- [22] C. Touma and C. Gotsman, "Triangle mesh compression," in *Proc. Graphics Interface*, Vancouver, BC, Canada, Jun. 1998, pp. 26–34.
- [23] P. Alliez and M. Desbrun, "Progressive compression for lossless transmission of triangle meshes," in *Proc. ACM Siggraph*, Aug. 2001, pp. 195–202.
- [24] S. Yang, C.-S. Kim, and C.-C. Jay Kuo, "View-dependent progressive mesh coding for graphic streaming," in *Proc. SPIE ITCOM*, Aug. 2001, vol. 4518, pp. 154–165.
- [25] C. Zach, M. Grabner, and K. Karner, "Improved Compression of Topology for View-Dependent Rendering," Tech. Rep. VRVis, 2004.
- [26] H. Wang and J. Li, "Octmesh—Interactive mesh browsing over the internet," in *Proc. Int. Conf. Inf. Technol.: Coding Computing*, Mar. 2000, pp. 104–108.
- [27] U. Bischoff and J. Rossignac, "TetStreamer: Compressed back-to-front transmission of Delaunay tetrahedra meshes," in *Proc. Data Compression Conf.*, Mar. 2005, pp. 93–102.
- [28] L. Peotta, L. Granai, and P. Vanderghyest, "Image compression using an edge adapted redundant dictionary and wavelets," *Signal Process.*, vol. 86, no. 3, pp. 444–456, Mar. 2006.



Ivana Tosić (S'04) received the Dipl.-Ing. degree in telecommunications from the University of Nis, Nis, Serbia, in 2003. She graduated from the Doctoral School in Information and Communication Sciences, EPFL, in 2004, where she is currently working toward the Ph.D. degree.

She joined the Signal Processing Institute, EPFL, in 2004, as a Research and Teaching Assistant. Her research interests include representation and coding of visual information, 3-D objects compression, distributed source coding, and plenoptic sampling.



Pascal Frossard (S'96–M'01–SM'04) received the M.S. and Ph.D. degrees from the Swiss Federal Institute of Technology (EPFL), Lausanne, Switzerland, in 1997 and 2000, respectively, both in electrical engineering.

From 1998 to 2000, he was with the Signal Processing Laboratory, EPFL, as a Research and Teaching Assistant under a grant from Hewlett-Packard. Between 2001 and 2003, he was a Member of the Research Staff with the IBM T. J. Watson Research Center, Yorktown Heights, NY, where he worked on media compression and streaming technologies. Since 2003, he has been an Assistant Professor with EPFL, supported by the Swiss National Science Foundation. His research interests include image representation and coding, nonlinear representations, visual information analysis, joint source and channel coding, multimedia communications, and multimedia content distribution.

Dr. Frossard has been the General Chair of IEEE ICME 2002 (Lausanne, Switzerland), and member of the organizing or technical program committees of numerous conferences. He has served as Guest Editor of special issues on Streaming Media (IEEE TRANSACTIONS ON MULTIMEDIA), on Media and Communication Applications on General Purpose Processors: Hardware and Software Issues (*Journal of VLSI SPSS*), and on Image and Video Coding Beyond Standards (*Journal of Signal Processing*). He is an Associate Editor of the IEEE TRANSACTIONS ON MULTIMEDIA (2004-) and of the IEEE TRANSACTIONS ON CIRCUITS AND SYSTEMS FOR VIDEO TECHNOLOGY (2006-), and he served as a

member of the Editorial Board of the *EURASIP Journal of Signal Processing* (2003–2005). Since 2004, he has served as Vice-Chair of the IEEE Multimedia Communications Technical Committee, as a member of the IEEE Multimedia Signal Processing Technical Committee and of the IEEE Multimedia Systems and Applications Technical Committee. He was a recipient of the Swiss NSF Professorship Award in 2003 and the IBM Faculty Award in 2005.



Pierre Vanderghyest (M'01) received the M.S. degree in physics and the Ph.D. degree in mathematical physics from the Université Catholique de Louvain, Louvain-la-Neuve, Belgium, in 1995 and 1998, respectively.

From 1998 to 2001, he was a Postdoctoral Researcher with the Signal Processing Laboratory, Swiss Federal Institute of Technology (EPFL), Lausanne, Switzerland. He is now an Assistant Professor with EPFL, where his research focuses on harmonic analysis, information theory, sparse approximations,

and mathematical image processing with applications to higher dimensional and complex data processing.

Dr. Vanderghyest was Co-Editor-in-Chief of the *Signal Processing* (2002–2006).