1-2012

Modeling and Compressing 3-D Facial Expressions Using Geometry Videos

Jiazhi XIA
Nanyang Technological University

Dao T. P. QUYNH
Nanyang Technological University

Ying HE
Nanyang Technological University

Xiaoming CHEN
Nanyang Technological University

Steven C. H. HOI
Singapore Management University, CHHOI@smu.edu.sg

DOI: https://doi.org/10.1109/TCSVT.2011.2158337

Citation

XIA, Jiazhi; QUYNH, Dao T. P.; HE, Ying; CHEN, Xiaoming; and HOI, Steven C. H.. Modeling and Compressing 3-D Facial Expressions Using Geometry Videos. (2012). *IEEE Transactions on Circuits and Systems for Video Technology (IEEE TCSVT)*. 22, (1), 77-90. Research Collection School Of Information Systems.
Available at: https://ink.library.smu.edu.sg/sis_research/2276

Published in IEEE Transactions on Circuits and Systems for Video Technology, Volume: 22 , Issue: 1 , Jan. 2012), Pages 77-90
https://doi.org/10.1109/TCSVT.2011.2158337

Modeling and Compressing 3-D Facial Expressions Using Geometry Videos

Jiazhi Xia, Dao Thi Phuong Quynh, Ying He, *Member, IEEE,* Xiaoming Chen, and Steven C. H. Hoi, *Member, IEEE*

***Abstract*—In this paper, we present a novel geometry video (GV) framework to model and compress 3-D facial expressions. GV bridges the gap of 3-D motion data and 2-D video, and provides a natural way to apply the well-studied video processing techniques to motion data processing. Our framework includes a set of algorithms to construct GVs, such as hole filling, geodesic-based face segmentation, expression-invariant parameterization (EIP), and GV compression. Our EIP algorithm can guarantee the exact correspondence of the salient features (eyes, mouth, and nose) in different frames, which leads to GVs with better spatial and temporal coherence than that of the conventional parameterization methods. By taking advantage of this feature, we also propose a new H.264/AVC-based progressive directional prediction scheme, which can provide further 10%–16% bitrate reductions compared to the original H.264/AVC applied for GV compression while maintaining good video quality. Our experimental results on real-world datasets demonstrate that GV is very effective for modeling the high-resolution 3-D expression data, thus providing an attractive way in expression information processing for gaming and movie industry.**

***Index Terms*—3-D facial expression, expression-invariant parameterization, feature correspondence, geometry video (GV), H.264/AVC, video compression.**

I. Introduction

Over the past decade, we have witnessed a revolution in movie and game industries resulting from the use of motion data. Nowadays, it is very common that actors work in front of a blue screen and interact with invisible computer-animated characters which are added later, trying to fit into a computer-animated world. The movements of actors are recorded using a motion capture (or mocap) system, by which complex movement, realistic physical interactions, and exchange of forces can be recreated in a physically accurate manner. Despite the great success in movies and gaming, the current mocap requires the subject to wear calibrated markers. The output of motion capture is just the approximate motion of a skeleton representing the rigid parts of the subject, rather than its precise geometry. Therefore, much editing work is often needed to map the skeletal movement to a virtual character. Furthermore, artifacts may occur when applying the recorded motion to a virtual model with proportions different than the captured subject.

The latest 3-D image sensing technology provides an alternative way to capture the moving and deforming objects. For example, the structured light system consists of a structured light source (such as a digital projector) and a high-speed digital video camera, and can be set up easily in an everyday environment. By encoding phase information of the light, it can provide the depth information of a 3-D scene in real time. Compared to the traditional marker-based mocap system, the 3-D camera allows us to capture the moving objects in a less restrictive manner, i.e., without placing any markers on the subject, and it can provide more accurate geometry data of the objects. However, the current structured light-based 3-D camera has several serious drawbacks that inhibits its use in broader applications.

1) First, the scanned motion data is usually bulky. For example, the latest high-resolution 3-D camera [1] is able to capture 30 f/s with a resolution of 512 × 512 of each frame, resulting in approximately 5 MB raw data per frame as shown in Fig. 1. This imposes a significant challenge for compressing the captured video efficiently while maintaining the video reconstruction quality.
2) Second, the captured raw data may contain noise and/or holes due to various reasons, such as camera occlusion, specular reflection, shadows, light interference, depth discontinuity, and others. Thus, much efforts are needed to clean and repair the datasets.
3) Third, each frame of the captured motion data is in the reference system of the scanner, and it is not registered in the object space. Thus, the correspondences between points in different frames are not available. However, from the modeling point of view, it is highly desirable to have such correspondence among frames. Furthermore, as we will demonstrate later, the correspondence is helpful to improve the compression ratio.

Geometry images [2] are a novel concept that intelligently encodes the 3-D geometry into an image format, in which each pixel $\{r, g, b\}$ represents a 3-D vertex $\{x, y, z\}$. To process 3-D motion data, it is natural to extend geometry images to geometry videos (GVs) which bridges the gap of 3-D motion data and 2-D video, and provides a way to apply the well-studied

Manuscript received September 23, 2010; revised February 20, 2011 and April 27, 2011; accepted April 29, 2011. Date of publication May 31, 2011; date of current version January 6, 2012. This work was supported by the National Research Foundation (NRF), under Grant NRF2008IDM-IDM004-006. This paper was recommended by Associate Editor H. Wang.

The authors are with the School of Computer Engineering, Nanyang Technological University, 639798, Singapore (e-mail: xiaj0002@e.ntu.edu.sg; daot0006@e.ntu.edu.sg; yhe@ntu.edu.sg; xmchen@ntu.edu.sg; chhoi@ntu.edu.sg).

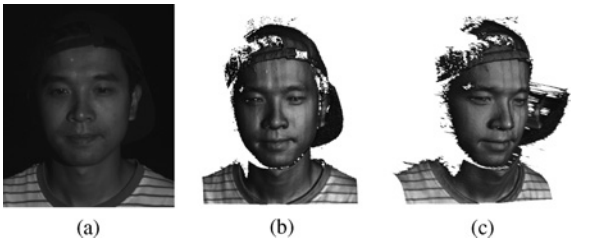


Fig. 1. 3-D camera is capable of capturing high-resolution motion data at 30 f/s. Both geometry (vertex coordinate) and texture (grayscale color) are encoded in a quadrilateral mesh with approximately 250 K vertices. (a) Image captured by conventional 2-D camera. (b) 3-D mesh captured by 3-D camera. (c) Another view of the 3-D mesh.

video processing techniques to motion data compression and processing. However, the existing GV techniques [3] applied only to datasets that are created by the animators, of which the correspondences among frames are available and the data are usually simple and noise-free. As a result, the user only needs to parameterize one frame, and the remaining frames can be easily induced by the given correspondence. Unfortunately, as mentioned above, such correspondence is not available for the 3-D motion data acquired by 3-D cameras. Thus, it is usually very challenging to construct GVs for real-world datasets.

To solve the aforementioned challenges and promote GV to real-world applications, this paper presents a novel framework that can capture high-resolution 3-D facial expressions in a less restrictive manner, store the recorded data in a compact way, and allow users to manage, manipulate, and render the data easily. Given the captured expression data, GV first analyzes the geometry and detects salient features, and then parameterizes the motion data to a rectangular domain such that the detected features in all frames can be mapped consistently. Finally, the parameterized motion data are converted into a video format such that the well-developed video compression techniques can be used to compress the motion data. Specifically, the GV compression task in this paper is accomplished by the state-of-the-art video coding standard—H.264/AVC [4], together with our proposed progressive directional prediction scheme.

The specific contributions of this paper include the following.

1) We develop a GV framework to model 3-D facial expressions. Our framework consists of a set of algorithms, including hole filling, geodesic-based face segmentation, and expression-invariant parameterization (EIP). Our algorithms are efficient and robust, and can guarantee the exact correspondence of the salient features (eyes, mouth, and nose) among frames.
2) We present a comprehensive analysis on GV and show that GV is fundamentally different from natural videos (NV) in that the GV pixels are highly correlated in both spatial domain and temporal domain, and adhere to an organized distribution.
3) By taking advantage of the strong coherence of GV, we propose a new intra prediction scheme incorporated to H.264/AVC. The proposed prediction scheme can improve the rate-distortion performance for not only intra frames but also inter frames. Our experimental results show that the proposed scheme can achieve further 10–16% bitrate reductions over original H.264/AVC applied to real-world 3-D facial expressions.
4) Through a quantitative comparison to some existing methods, such as discrete Ricci flow (RF) and harmonic map (HM), we demonstrate that the proposed approach can guarantee the exact feature correspondence among frames, which is very helpful to compress GVs. As a result, our method can produce GVs of better quality and smaller size.

The rest of this paper is organized as follows. Section II briefly reviews the related previous work. Section III presents the 3-D motion data acquisition and pre-processing. Section IV details the algorithm to parameterize the motion data. Section V presents our proposed prediction scheme for GV compression. Experimental results and discussions are presented in Section VI. Finally, we conclude our work in Section VII.

II. Related Work

GV bridges two different research fields, geometry processing, and video processing. This section briefly reviews related work in motion data acquisition and processing, 3-D motion data compression, geometry images/videos, and video compression.

A. 3-D Motion Data Acquisition and Processing

In recent years, we have witnessed the significant advances in developing high-speed shape acquisition devices. Using range scanning techniques, such as phase-shifting structure light [1], [5], [6] and spacetime stereo [7], [8], it is possible to scan high-resolution 3-D geometry and/or texture of moving and deforming objects at video speeds.

Wang *et al.* [9] presented a data-driven approach for accurate facial tracking and expression retargeting. Wang *et al.* [10] simplified the 3-D human face registration problem to a 2-D image matching problem by conformal parameterization. Mitra *et al.* [11] proposed an algorithm to register large sets of unstructured point clouds of moving and deforming objects without computing correspondences. Chang and Zwicker [12] presented an unsupervised algorithm that aligns a pair of articulated shapes with significant motion and missing data. Sharf *et al.* [13] developed a volumetric space-time technique to reconstruct the moving and deforming objects from point clouds. Wang *et al.* [14] developed an efficient non-rigid 3-D motion tracking algorithm to establish inter frame correspondences that facilitate the temporal study of subtle motions in facial expressions.

Observing that the human facial expressions are isometric, Bronstein *et al.* [15] developed an algorithm to embed human faces into spherical domain, by which the canonical spherical coordinates induce an EIP. In this paper, we also present an EIP algorithm. Our method is different from [15] in the following aspects: 1) our algorithm guarantees exact correspondence of the salient features (eyes, mouth and nose), and 2) the parameterization distortion is much less than that of [15]. To our best knowledge, this is the first work that can parameterize the 3-D facial expressions with guaranteed feature correspondence.

B. 3-D Motion Data Compression

Time-varying meshes (TVM) has been introduced in 3-D motion data compression by Han *et al.* [16]. TVM is a 3-D motion representation which is generated from multiple cameras [17]. Since TVM are generated from multiple viewpoint images, frame by frame independently, TVM do not have the correspondence among frames. The generated data is bulky and noisy as the data is captured by the structured light-based 3-D camera. TVM cannot afford correspondences between frames either. Although it is natural to model 3-D motion data with TVM, there are few papers on TVM compression due to the challenges. Han *et al.* [16] proposed an extended block matching algorithm for TVM compression. By extending the block matching algorithm from 2-D video to 3-D mesh data, they could achieve 10–18% compression. But only inter frame coding is used in their work. By considering both spatial and temporal redundancies of TVM, Han *et al.* [18] achieved compression of 1.9–16%. Yamasaki *et al.* [19] also compress connections and color textures. Instead of marching cubes, they used a patch-based method to describe the model. However, none of the above works has taken the correspondence between frames into consideration.

C. Geometry Images and Videos

The concept of geometry images was pioneered by Gu *et al.* [2], who parameterized the 3-D mesh into a square domain and then encoded the normalized vertex coordinates (x, y, z) as a pixel value (r, g, b) of a 2-D image. Therefore, geometry images naturally bridge 3-D shape compression and 2-D image compression algorithms, e.g., [20]. Along this direction, Lin *et al.* [21] presented JEPG2000 for compression and streaming of geometry images. Peyré and Mallat [22] presented geometric bandlets to compress geometry images and normal maps. They showed that bandeletization algorithm outperforms the wavelet-based compression by removing the geometric redundancy of orthogonal wavelet coefficients.

Geometry images are an elegant representation of static shape. To model motion data, it is a natural idea to extend geometry images to GVs. In [3], Briceño *et al.* [3] parameterized the animated mesh sequence onto a rectangular domain and then formed GV. However, their method [3] applied only to synthetic data, of which the correspondence among frames are available. They also used 2-D wavelet-based video compression techniques. In contrast to [3], our proposed parameterization algorithm works for real-world datesets which may contain artifacts such as holes and noise, and do not have the correspondence between adjacent frames. Furthermore, our parameterization method matches the salient features among frames in a consistent manner. As a result, the generated GVs are highly correlated in both spatial and temporal domains. This feature enables us to exploit the potential of H.264/AVC, which is incorporated with many advanced video compression techniques, for heavier compression of GV.

D. Video Compression and H.264/AVC Intra Prediction

Traditional 2-D video compression techniques can be categorized as prediction, transformation, quantization, and entropy coding. Sullivan and Wiegand [23] provided a comprehensive review on these techniques. H.264/AVC, the state-of-the-art video coding standard [24], has employed many advanced compression techniques.

H.264/AVC has provided a means for spatial prediction, namely, intra prediction. Intra prediction allows the video encoder to predict pixel values of the current block (to be encoded) from its previously reconstructed upper and left neighbor pixels. As a result, intra prediction is particularly effective for encoding video pictures with a high degree of spatial correlations. H.264/AVC has adopted a block-size adaptive intra prediction scheme, i.e., the prediction can be performed at 16×16, 8×8, or 4×4 block basis for obtaining smallest prediction errors. This scheme has provided four prediction modes for 16×16 blocks and 8×8 blocks, and nine prediction modes for 4×4 blocks including vertical prediction, horizontal prediction, diagonal prediction, and others [4]. These prediction modes, however, simply "copy" and "paste" neighbor pixels (or their weighted/unweighted averages) to the predicted block, showing an insufficient respect to the local pixel changes, i.e., they assume little changes between two adjacent pixels. As a result, they cannot well model the pixel-varying trend that inherently appeared in GVs (see Section V-A for details). There are also many other intra frame prediction schemes proposed, e.g., [25]–[28], focusing on reducing the prediction complexity, and [29]–[32] aiming at reducing the prediction errors. However, all the existing schemes are mainly designed for NV compression instead of GV compression. In this paper, we will present a dedicated and better intra frame prediction scheme for our GV framework in Section V.

Wavelet-based compression is used in JPEG-2000 for still image compression, which supersedes the original discrete cosine transform-based JPEG standard by improving the compression performance and offering significant flexibility of the codestream [33]. However, wavelet-based coding is not available within the H.264/AVC standard. In this paper, we choose H.264/AVC-based compression algorithm for the efficiency of implementation.

III. 3-D Motion Data Acquisition and Pre-Processing

We employ the structured light-based 3-D camera system [1] to capture the moving objects in real time. The system contains a video camera and a structured light projector. The projector projects digital fringe patterns that are composed of vertical straight stripes to the object. The stripes are deformed due to the surface profile. Then a high-speed charge-coupled device camera synchronized with the projector captures the distorted fringe image. Finally, by analyzing the fringe images, the 3-D information is obtained based on the deformation using triangulation. The system is able to capture the geometry and texture of the moving objects in real time. Despite the high speed, the 3-D camera system is not robust due to various reasons, such as ambient light interference, occlusions,

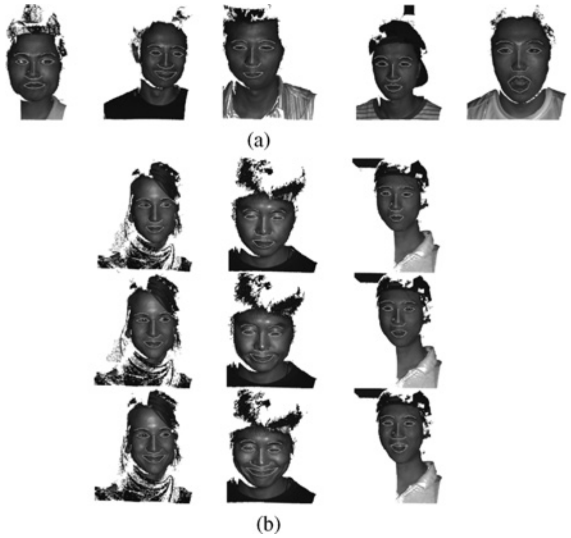


Fig. 2. Training data set and the detecting results of AAM. The salient features are marked by red points and yellow lines. (a) Training data set. (b) Testing data set.

shadows, and depth discontinuity. Therefore, much efforts are needed to pre-process the captured raw data.

A. Feature Tracking

We first project the captured 3-D expressions to 2-D images, and then use active appearance model (AAM) [34] to automatically detect the feature points. Finally, the detected 2-D feature points are mapped back to the original 3-D meshes. In our experiment, we collect 3-D facial expressions from five subjects and choose one frame from each sequence to form the training set. Then we manually label the salient features, including nose tip, eyes, eyebrows, mouth, on the selected five training frames. Next the trained AAM is applied to detect the salient features in the new expression sequences. Note that we only need to manually label on a few training frames, the tracking on new sequences are fully automatic. Fig. 2 shows the training frames and several examples of the automatic detected features on non-training frames.

B. Hole Filling

The captured raw data is a genus-0 open surface M. Let $\partial M = \gamma_0 \cup \gamma_1 \cup \cdots \gamma_k$ denote boundaries where γ_0 is the outer boundary and γ_i, $i \geq 1$ the interior holes. To fill the hole γ_i, we construct a minimal surface H_i that satisfies the Laplacian equation [35] as follows:

$$\triangle v = 0, \quad \forall v \notin \partial H_i$$

with boundary conditions as follows:

$$v|_{\partial H_i} = v|_{\gamma_i}$$
$$\triangledown v|_{\partial H_i} = \triangledown v|_{\gamma_i}.$$

The boundary conditions guarantee that the filled surface is of C^1 continuity along γ_i, thus, leads to visually pleasing results. Note that we can also fill the colors using the same equation except that the vertex position (x, y, z) is replaced by the color (r, g, b). Fig. 3 shows the hole filling results.

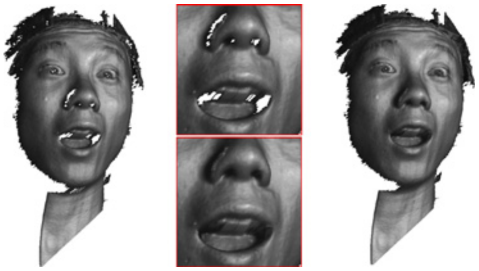


Fig. 3. Captured raw data usually contains holes due to the occlusions. We fill both the geometry and texture of the holes by constructing a minimal surface which has C^1 continuity along the hole boundaries.

C. Face Segmentation

The captured raw data contains not only the 3-D faces but also some unnecessary information, such as cloth, hair, and background. Observe that human expressions are approximate isometry, in other words, the arc length in mesh surface is preserved during mesh deformation, thus, the intrinsic properties, such as Gaussian curvature, first fundamental form, geodesic, conformal factor, and others, which are invariant under isometry, can be used to segment the face. In our framework, we adopt the geodesic since it is fairly easy to compute and highly robust to the mesh resolution and triangulation. With the eyes and mouth as source points, we compute the “multiple-sources all-destinations” geodesic using the modified Xin and Wang’s algorithm [36] which takes only a few seconds for each frame. Fig. 4(a) and (d) shows the computed geodesic fields. The blue lines are the iso-lines of the geodesic fields. And the green lines serve as the base iso-line which are generated by connecting the feature points. Then we segment the facial expressions using the user-specified radius which is the distance from the green line. In our experiment, we choose the radius of 60–65 mm for male and 45–50 mm for female faces. Finally, we remove the eyes and mouth by specified feature points. As shown in Fig. 4, our method leads to highly consistent segmentation results.

IV. Expression-Invariant Parameterization

In each frame of the captured motion data, the geometry is given in the reference system of the scanner, and it is not registered in object space, and correspondences between points in different frames are not available. From the analysis and editing point of view, it is highly desirable to find the correspondence among the captured data. Motion data parameterization serves this purpose by mapping all frames to a parametric domain and then re-sample the data on the domain.

Although there are large amount of literatures in surface parameterization [37], [38], there is little work on the motion data parameterization. The key challenging in motion data parameterization is that it must take the temporal coherence into consideration, i.e., the features in all frames should be mapped consistently to the parametric domain.

This section presents a novel algorithm to parameterize the 3-D facial expression data. The proposed algorithm is

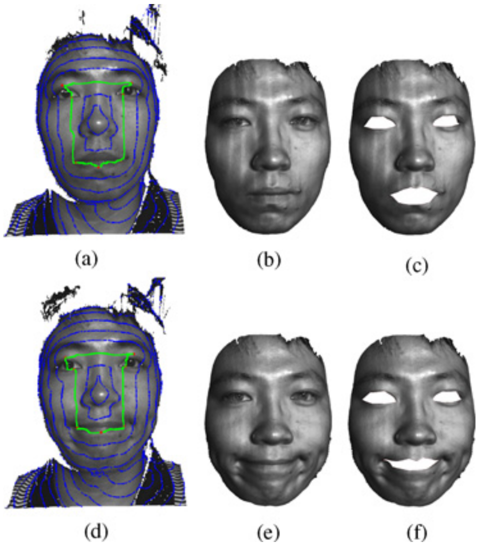


Fig. 4. Face segmentation using geodesic mask. Human expressions are approximate isometry, thus, the geodesic distance is independent of the expressions. (a), (d) We first compute a geodesic mask from the detected features on mouth and eyes, (b) and (e) then segment the front face by the user-specified radius. Finally, we (c), (f) remove the mouth and eyes.

guaranteed to be bijective and the salient facial features (such as mouth, nose, and eyes) are mapped consistently onto the parametric domain.

The input of our algorithm is a sequence of genus-0 meshes with four boundaries. Let us denote by M the mesh and $\partial M = \gamma_0 \cup \cdots \cup \gamma_3$ the boundaries, where γ_0 is the boundary of the human face, γ_1 and γ_2 are the two eyes and γ_3 is the mouth. We define the parametric domain $D \in \mathbb{R}^2$ as a rectangle with three holes, as a result, D has the same topology of M.

We first compute the geodesic c between two eyes, i.e., γ_1 and γ_2. Then we compute the geodesic d from the middle point of c to the mouth γ_3. Note that geodesic is an intrinsic property, thus, independent of the expressions which are approximate isometry. By slicing the mesh along the geodesics c and d, the number of boundaries is reduced to 2. The resulted mesh $\overline{M}$ is a genus-0 mesh with two boundaries, i.e., γ_0 and $\gamma_1 \cup \gamma_2 \cup \gamma_3 \cup c \cup d$. In the following, we use $\partial\overline{M}_0$ and $\partial\overline{M}_1$ to denote the two boundaries of $\overline{M}$.

Then we compute the harmonic function $f : \overline{M} \to \mathbb{R}$, $\triangle f(v) = 0, \quad \forall v \notin \partial\overline{M}$, with Dirichlet boundary condition as follows:

$$\begin{aligned} f(v) &= 0, \quad \forall v \in \partial\overline{M}_0 \\ f(v) &= 1, \quad \forall v \in \partial\overline{M}_1. \end{aligned}$$

Since the function f is harmonic, all its local extrema are on the boundaries. Furthermore, the mesh $\overline{M}$ is of genus-0 with two boundaries. According to Morse theory, f has no critical point (the point with vanishing gradient) inside $\overline{M}$. Therefore, the gradient vector field ∇f has no singularity. The integration curve of ∇f is a curve such that the tangent vector to the curve at any point v along the curve is precisely the vector $\nabla f(v)$. Xia *et al.* [39] show that each integral curve has unique ending points, one on $\partial\overline{M}_0$, and the other on $\partial\overline{M}_1$. Furthermore, any two integral curves do not intersect.

Algorithm 1 Expression-invariant 3-D face parameterization

```
Input:
$M \in \mathbb{R}^3$, the input 3-D facial expression of genus-0 mesh with four boundaries, $\partial M = \gamma_0 \cup \cdots \cup \gamma_3$;
$D \in \mathbb{R}^2$, the parametric domain with the same topology of $M$;
Output:
The one-to-one map $\phi : M \to D$ such that the salient features (eyes, mouth, and nose) are mapped to the corresponding features on $D$.
1. Compute the geodesic $c$ between $\gamma_1$ and $\gamma_2$.
2. Compute the geodesic $d$ from the middle point of $c$ to $\gamma_3$.
3. Cut $M$ along $c$ and $d$, the resulted mesh $\overline{M}$ is of genus-0 with 2 boundaries.
4. Process the parametric domain $D$ in the similar way (as steps 1–3). Let $\overline{D}$ denote the processed mesh of genus-0 with two boundaries.
5. Compute the harmonic function $f : \overline{M} \to \mathbb{R}$ with Dirichlet boundary condition, $\triangle f = 0$, $f|_{\partial\overline{M}_0} = 0$, $f|_{\partial\overline{M}_1} = 1$
6. Compute the harmonic function $g : \overline{D} \to \mathbb{R}$ with Dirichlet boundary condition, $\triangle g = 0$, $g|_{\partial\overline{D}_0} = 0$, $g|_{\partial\overline{D}_1} = 1$
7. Parameterize $\partial\overline{M}_1$ and $\partial\overline{D}_1$ by the arc length parameterization, $h : \partial\overline{M}_1 \to \partial\overline{D}_1$.
8. For each point $v \in \partial\overline{M}_1$
    8.1 Trace the integral curve $\alpha \in \overline{M}$ of the gradient vector field $\nabla f$.
    8.2 Trace another integral curve $\beta \in \overline{D}$ starting from $h(v) \in \partial\overline{D}_1$ and following the vector field $\nabla g$.
    8.3 Construct the one-to-one map $\overline{\phi} : \overline{M} \to \overline{D}$ as $\overline{\phi}(\alpha) = \beta$
9. The parameterization $\phi : M \to D$ is induced from $\overline{\phi} : \overline{M} \to \overline{D}$.
```

We process the parametric domain D in the same way and let $\overline{D}$ denote the sliced mesh with two boundaries. We compute the harmonic function $g : \overline{D} \to \mathbb{R}$ with the same boundary condition as f. We also construct a bijective map between two boundary curves $h : \partial\overline{M}_1 \to \partial\overline{D}_1$ by arc-length parameterization.

Then the parameterization $\phi : \overline{M} \to \overline{D}$ is constructed as follows: for each vertex $v \in \partial\overline{M}_1$, trace the integral curve $\alpha \in \overline{M}$ following the gradient ∇f. Then, starting from $h(v) \in \partial\overline{D}_1$, trace another integral curve $\beta \in \overline{D}$. Thus, we build a one-to-one map between two integral curves α and β. By going through every point $v \in \partial\overline{M}_1$, we build the one-to-one map between $\overline{M}$ and $\overline{D}$ which in turn induces a one-to-one map $\phi : M \to D$.

Remark: In the parameterization algorithm, we cut the 3-D face along the geodesics connecting the three holes (i.e., eyes and mouth). So the resulted mesh is of genus-0 with two boundaries. The inner boundary $\gamma_1 \cup \gamma_2 \cup \gamma_3 \cup c \cup d$ is invariant to the expressions, thus, highly consistent among all frames.

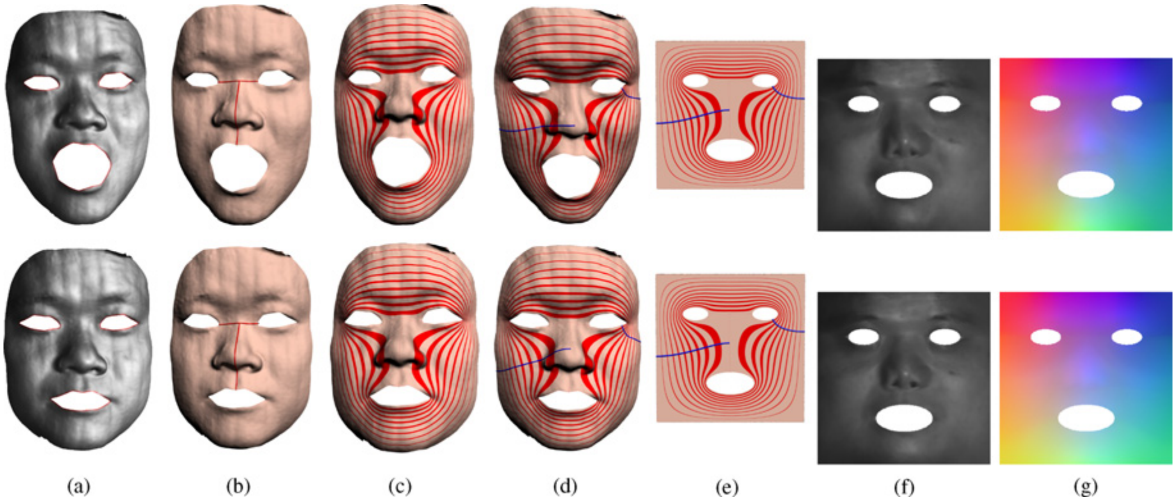


Fig. 5. EIP. (a) Input mesh M. (b) Geodesics connecting the eyes, nose, and mouth. (c) Harmonic function with Dirichlet boundary condition. As human facial expressions are approximate isometric transformation, the computed harmonic functions are insensitive to expressions. (d) Integral curves follow the gradient of the harmonic function. (e) Integral curves on the parametric domain. (f) Integral curves induce the parameterization between M and D, which guarantees the exact correspondence of the eyes, mouth, and nose. As a result, different expressions have very similar parameterization. (g) Converting the parameterized mesh to geometry image, in which the pixel color (r, g, b) encodes the vertex position (x, y, z).

Furthermore, the outer boundary is determined by a geodesic mask with the user-specified radius applied to all expressions. Thus, the outer boundary is also invariant to the expression. Observe that the harmonic function is intrinsic to the geometry and independent of the expressions. As a result, the proposed parameterization is invariant to the expression. Furthermore, as proven in [39], the inner boundary of $\overline{M}$ is mapped to the inner boundary of $\overline{D}$ precisely, thus, guarantees exact correspondence of salient features, such as eyes, mouth, and nose. As shown in Fig. 5, two expressions are parameterized consistently using our approach.

V. GV Compression

In our previous work [39], we proposed a fundamental "tailored intra prediction" (TIP) scheme for GV compression. The TIP scheme employs a single-stage prediction, i.e., all pixels in a block are predicted in the same way. Also, the TIP can only predict pixels in horizontal and vertical directions for 4×4 block size. In this section, we present the progressive directional prediction (PDP) scheme, which employs more accurate multistage predictions for variable block sizes in horizontal, vertical, diagonal down-left, and down-right directions. PDP is also based on the H.264/AVC intra prediction (HIP). However, it can provide benefits to not only intra frames but also inter frames. In the remaining of this section, we first outline the GV structure and features, and then describe the PDP scheme in details.

A. Analysis of GVs

Using the proposed EIP algorithm, each frame of the motion data is mapped to a rectangular domain that can be easily converted into a 2-D geometry image representation as described in Section II. We define such an image as a GV picture (GVP). Then a GV is constructed by combining a sequence of successive GVPs. Before presenting our PDP scheme, we first analyze GVs in both temporal and spatial domains.

1) *Temporal Coherence:* The temporal feature of GV is similar to that of NV, i.e., a block in a GVP usually closely matches another block locating at the same or close position in the neighbor GVP. However, the degree of temporal correlation of GV is even stronger than that of NV. As a result, the temporal redundancy in a GV sequence can be significantly removed by using well-developed video compression algorithms, e.g., motion estimation in H.264/AVC [4]. For example, Fig. 6 shows that, for a NV (grayscale) and its corresponding GV (RGB), the mean square errors (MSEs) produced after a 32×32 Full Search motion estimation of H.264/AVC (quantization parameter, QP = 8). It has been observed that, after the motion estimation, the MSE of GV in each color channel is significantly smaller than that of the corresponding NV. In other words, the temporal redundancy in GV has been significantly removed by motion estimation. Therefore, we have adopted the Full Search algorithm of H.264/AVC in our paper for temporal compression of GV.

2) *Spatial Coherence:* The spatial feature of GV is different from that of NV. Obviously, a GVP has a peculiar appearance as shown in Fig. 5(g). It is apparent that the GVP pixels are strongly correlated in spatial domain. In particular, the neighbor pixels share very similar (but not exactly same) pixel values. This is because that, since the pixels represent the vertex coordinates of the 3-D face surface, a GVP inherently has a special pixel distribution. In a small local region of a GVP, e.g., a 4×4, 8×8, or 16×16 block region, the pixels in that region only correspond to a very small region of the 3-D face surface, which we can assume to be with smoothly varying vertex coordinates. In this case, the corresponding pixels in that GVP region will also adhere to a smoothly varying trend. For example, Fig. 7(a) shows an enlarged portion of a GVP (R color channel corresponding to x coordinate), and Fig. 7(b) shows an 8×8 block extracted from

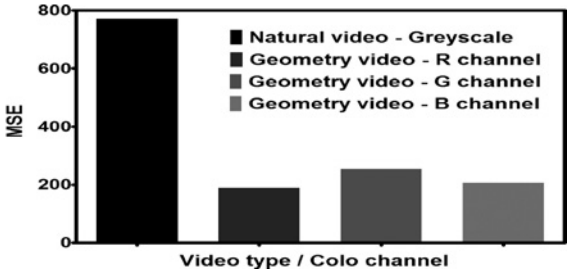


Fig. 6. MSE comparison for NV and GV after a 32×32 Full Search motion estimation of H.264/AVC.

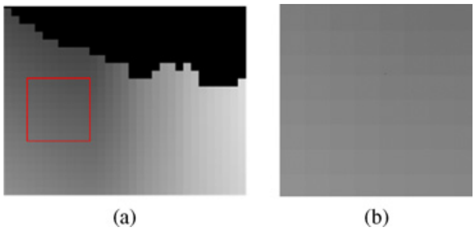


Fig. 7. Pixel-varying trend of a typical GVP. (a) Enlarged portion for R color channel. (b) 8×8 block showing the smoothly varying trend of pixels.

that portion. In this example, it is noticeable that the pixels of the block are varying smoothly, because the coordinates of that small portion in the face model gradually change.

3) *GVs Versus NVs:* We next provide quantitative statistics for GV in comparison to NV. Our statistics is based on an extension of Prewitt operator. Note that our purpose to use Prewitt operator in this paper is for modeling the directional varying trend of pixels instead of edge detections. Specifically, we applied the Prewitt operator in horizontal and vertical directions, and also extended it to diagonal down-left and down-right directions. Then we work out the histograms based on the Prewitt operator for different color channels of a GV in the four directions, respectively. For example, the histograms for a 2-D NV and its corresponding 3-D GV are shown in Fig. 8. The histogram values are divided into 255 buckets, which are indexed in the range of ±127.

Fig. 8(a) shows that: 1) the histograms of NV are very similar in four directions, and 2) the histograms essentially follow a zero-centered distribution, and the (small) differences between adjacent pixels are almost equally distributed in positive and negative buckets. In this context, the current HIP, which assumes pixel invariant (as we introduced in Section II), can compress this kind of NV very well. On the other hand, Fig. 8(b) and (c) shows the following features of GV: 1) the histograms are dissimilar in four directions, and 2) the histograms of some color channels are not zero-centered, e.g., the R color in horizontal direction and G color in vertical direction, and this demonstrates the nature of stronger directional pixel-varying trends of GV. Fig. 8(d) compares the histograms in horizontal and vertical directions of GV and NV. It is noticeable that the histograms of GV are centered at bucket -1, indicating that most GV pixels follow a single-direction and consistent pixel-varying trend. The above analytical results show that GV inherently appears to have more organized pixel distributions with stronger directional pixel-varying trends.

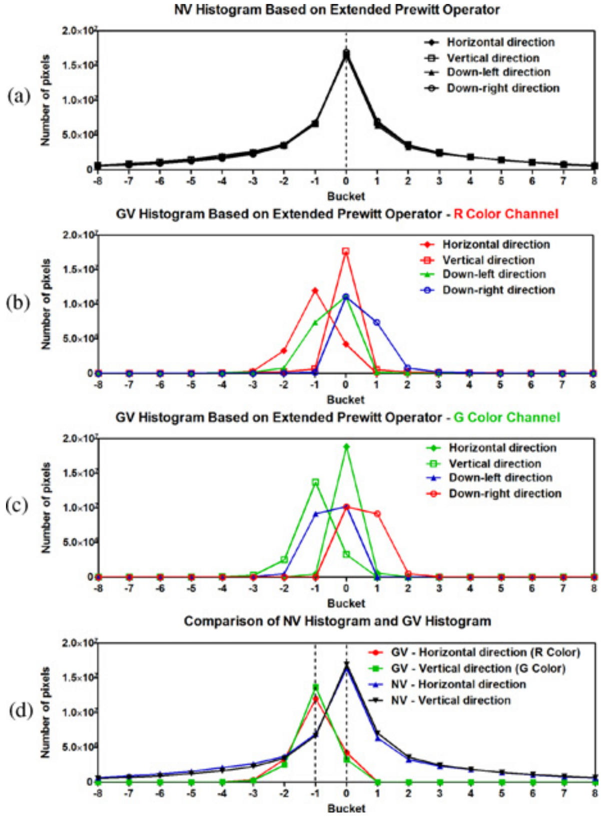


Fig. 8. Histograms for NV and GV based on extended Prewitt operator in four directions. (a) NV: grayscale. (b) GV: R channel. (c) GV: G channel. (d) Comparisons of GV to NV.

B. Progressive Directional Prediction for GV Compression

To take advantage of the GV features we analyzed in previous section, we have devised a new PDP scheme correspondingly, including four new intra prediction modes designed for the horizontal, vertical, diagonal down-left, and diagonal down-right directions, respectively. These four modes are incorporated into HIP. Specifically, the prediction process of PDP is divided into the following three steps: 1) pixel-varying trend (PVT) estimation; 2) pixel-varying map (PVM) construction; and 3) progressive pixel prediction (PPP). We have implemented the PDP scheme for 4×4, 8×8, and 16×16 block sizes. Please note that the PDP scheme operates in pixel domain similar to the original intra prediction scheme in H.264/AVC.

1) *PVT Estimation:* The purpose of introducing PVT is to estimate the overall pixel-varying trend of a block in four directions by using the available neighbor boundary pixels (NBPs) of that block. As an example, the PVT calculation process for a 4×4 block is illustrated in Fig. 9. In this figure, the NBPs are marked with letters such as A, B, A2, B2, and so on. The dashed arrows indicate the PVT calculation between two adjacent NBPs in different directions.

Let (i, j) denote the location of the current block, e.g., $(0, 0)$ refers to the top-left block in the GVP, n denotes the size of the block, e.g., $n = 4$ refers to a 4×4 block, (x_0, y_0) denote the location of the top-left pixel in the current block to be encoded, and $P(x, y)$ denotes a pixel locates at (x, y). Then the PVTs for

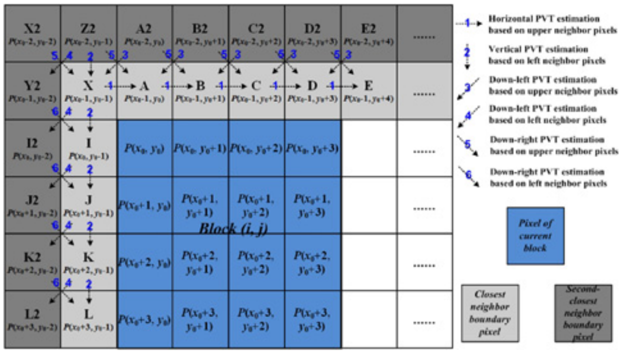


Fig. 9. PVT calculations based on NBPs for 4 × 4 block.

the block (i, j) in horizontal and vertical directions, denoted by $PVT^{hor}_{(i,j)}$ and $PVT^{ver}_{(i,j)}$, are calculated by using the immediate upper or left NBPs, shown in Fig. 9, as follows:

$$PVT^{hor}_{(i,j)} = \frac{1}{n}\sum_{k=0}^{n+1}(P(x_0-1, y_0-1+k) - P(x_0-1, y_0+k))$$

$$PVT^{ver}_{(i,j)} = \frac{1}{n}\sum_{k=0}^{n}(P(x_0-1+k, y_0-1) - P(x_0+k, y_0-1)).$$

For down-left and down-right directions, we calculate separate PVTs when using different NBPs, denoted by $PVT^{dl1}_{(i,j)}$ and $PVT^{dr1}_{(i,j)}$ (when using the upper NBPs, e.g., A, B, C, and D), and $PVT^{dl2}_{(i,j)}$ and $PVT^{dr2}_{(i,j)}$ (when using the left NBPs, e.g., I, J, K, and L) as follows:

$$PVT^{dl1}_{(i,j)} = \frac{1}{n}\sum_{k=0}^{n+1}(P(x_0-1, y_0+k-2) - P(x_0-2, y_0+k-1))$$

$$PVT^{dl2}_{(i,j)} = \frac{1}{n}\sum_{k=0}^{n}(P(x_0-2+k, y_0-1) - P(x_0-1+k, y_0-2))$$

$$PVT^{dr1}_{(i,j)} = \frac{1}{n}\sum_{k=0}^{n+1}(P(x_0-1, y_0+k-1) - P(x_0-2, y_0+k-2))$$

$$PVT^{dr2}_{(i,j)} = \frac{1}{n}\sum_{k=0}^{n}(P(x_0-1+k, y_0-1) - P(x_0-2+k, y_0-2)).$$

In practice, the PVTs are good to be used in predicting pixels that are spatially close to NBPs (see Section V-B3 for details).

2) *PVM Construction:* To model the relationship in pixel-varying trends between successive GVPs, we also construct a temporal PVM based on the PVTs obtained from previous section at 4×4 block basis. Specifically, there are four PVMs constructed for the four directions, including PVM^{hor}, PVM^{ver}, PVM^{dlf}, and PVM^{drt}. For example, the PVM in the down-left direction is given by

$$PVM^{dlf}=\{\tfrac{1}{2}(PVT^{dl1}_{(0,0)}+PVT^{dl2}_{(0,0)}),\dots,\tfrac{1}{2}(PVT^{dl1}_{(m,n)}+PVT^{dl2}_{(m,n)})\}$$

where m is the number of 4×4 blocks in one row of a GVP while n is the number of 4×4 blocks in one column. The PVMs are a compact representation that describes the overall pixel varying trend at picture level. For example, for a 512×512 GVP, the size of each PVM is 128×128. The PVMs can be used as reference for predicting the pixels that are far away from the spatial NBPs. For example, when predicting $P(x_0+3, y_0+3)$ in the current GVP (Fig. 9), we can look up the constructed PVMs of the previous GVP to decide the way of prediction (see Section V-B3 for details). After encoding the current GVP, the PVMs are updated accordingly. It is to be noted that we only need to keep PVMs for one GVP at an instance. Therefore, the memory overhead by PVMs is negligible.

3) *Progressive Pixel Prediction:* In this section, we detail the pixel prediction process of the proposed PDP scheme. We only present the prediction process of 4×4 blocks as an example. The prediction process for 8×8 and 16×16 blocks are very similar to that of 4×4 blocks.

The basic idea of PDP is to partition a block into a few regions according to the spatial distances between the region and the NBPs in specific directions, as shown in Fig. 11. For horizontal direction, our partitioning only considers the distance between the pixels in the current block and the left NBPs of the block, as shown in Fig. 11(a). Similarly, the partitioning in vertical direction only considers the distance between block pixels and the upper NBPs, as shown in Fig. 11(b). For diagonal down-left and down-right directions, the partitioning takes into account both the left and upper NPBs, as shown in Fig. 11(c). During the progressive prediction process, we particularly use two kinds of information for prediction: 1) the spatial correlation exploited from the NBPs and PVTs, and 2) the temporal correlation obtained from PVMs. From partition 1 to partition 3, we expect that the spatial correlation shared by the NBPs and the pixels in the partition is increasingly reduced. Accordingly, the progressive prediction will also reduce the usage of spatial correlation, e.g., PVT, while increasing the usage of temporal correlation, e.g., PVM, for prediction from partition 1 to partition 3. To demonstrate the prediction process clearer, we show the predictions for horizontal and down-left directions in Fig. 10 as an example.

a) *Partition* 1 *Prediction*: The pixels in partition 1 are very close to the NBPs of the block and they are very correlated in pixel values. Therefore, these pixels, e.g., $P(x_0, y_0)$, are predicted based on the immediate upper or left NBPs of the block, and by using the residue between two adjacent NBPs in a specific direction. For example, according to Fig. 10(a), $P(x_0, y_0)$ can be predicted by I+(I-I2) in horizontal direction. In some cases, it is possible to predict a single pixel from more than one neighbor pixels in a direction, e.g., as shown in Fig. 10(b), we can predict $P(x_0, y_0)$ by either B+(B-C2) or J+(J-K2) in down-left direction (Δ_1=B-C2, Δ_2=J-K2). In this case, we would take the average of the two predictions. The partition 1 prediction process for the block locating at (i, j) can be formulated in Table I.

b) *Partition* 2 *Prediction*: Compare to the partition 1 pixels, the remaining pixels in the current block are less spatially correlated to the upper and left NBPs. Our investigations show that the partition 1 prediction would

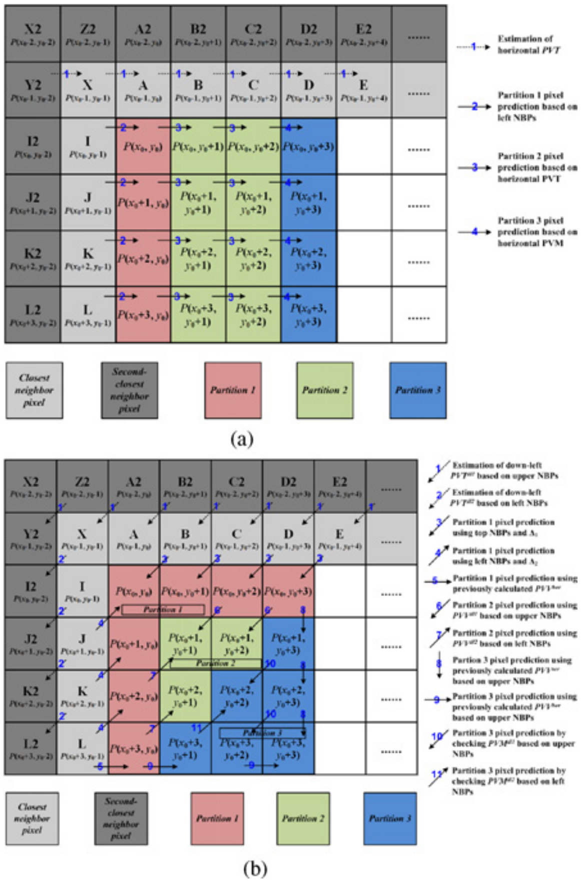

X2
Z2
A2
B2
C2
D2
E2
Y2
X
A
B
C
D
E
I2
I
J2
J
K2
K
L2
L
$P(x_0, y_0)$
$P(x_0, y_0+1)$
$P(x_0, y_0+2)$
$P(x_0, y_0+3)$
$P(x_0+1, y_0)$
$P(x_0+2, y_0)$
$P(x_0+3, y_0)$
Estimation of horizontal PVT
Partition 1 pixel prediction based on left NBPs
Partition 2 pixel prediction based on horizontal PVT
Partition 3 pixel prediction based on horizontal PVM
Closest neighbor pixel
Second-closest neighbor pixel
Partition 1
Partition 2
Partition 3
(a)
Estimation of down-left PVT^{dl1} based on upper NBPs
Estimation of down-left PVT^{dl2} based on left NBPs
Partition 1 pixel prediction using top NBPs and Δ_1
Partition 1 pixel prediction using left NBPs and Δ_2
Partition 1 pixel prediction using previously calculated PVT^{hor}
Partition 2 pixel prediction using PVT^{dl1} based on upper NBPs
Partition 2 pixel prediction using PVT^{dl2} based on left NBPs
Partion 3 pixel prediction using previously calculated PVT^{ver} based on upper NBPs
Partition 3 pixel prediction using previously calculated PVT^{hor} based on upper NBPs
Partition 3 pixel prediction by checking PVM^{dl1} based on upper NBPs
Partition 3 pixel prediction by checking PVM^{dl2} based on left NBPs
(b)

Fig. 10. PPP process. (a) Horizontal direction. (b) Diagonal down-left directions.

not obtain good results in partition 2 due to the increased distance (between partition 2 and the NBPs). In this case, predicting the pixels by using the PVTs yields much better results, because that the PVTs reflect the overall pixel-varying trend of the block so that they can remove the side effect caused by individual NBPs. Specifically, the partition 2 pixel prediction is given in Table I.

c) *Partition* 3 *Prediction*: The remaining pixels in partition 3 are even further away from the NBPs. In this case, it would not be suitable to use the varying trend calculated based on the spatial NBPs (as for partition 1) or the spatial PVTs (as for partition 2) for predictions. Instead, PDP will look up the PVMs constructed from the previous GVP and decide the pixel-varying trend and the way of pixel interpolations from a temporal prospective. For a particular prediction direction, PDP will look up at corresponding PVMs for taking advantage of the established link between successive GVPs in terms of pixel-varying trend (as explained in Section V-B2). In this case, a predicted pixel is formed by interpolating the pixel changes according to the PVM in the specific direction, e.g., PVM^{ver} for vertical direction. Specifically, the partition 3 pixel predictions in horizontal, vertical, and down-right directions are given in Table I.

For the down-left direction, in some cases, the neighbor pixels required by prediction may not be available. For example in Fig. 10(b), $P(x_0+3, y_0+3)$ cannot be predicted in the down-left direction since $P(x_0+2, y_0+4)$ and $P(x_0+4, y_0+1)$ are not available. In this case, we can look up the horizontal PVM and predict $P(x_0+3, y_0+3)$ using $P(x_0+3, y_0+2)$ as indicated by the arrow marked with number 9 in Fig. 10(b). In many other cases, similar operations are performed to form the predictions.

Please note that the PDP scheme is dedicatedly designed for GV compression. Hence, it would not provide benefits to NV compression.

VI. Experiments

A. Experimental Setups

1) *Test Sequence Capturing:* In this paper, we have captured eight test video sequences containing human facial expressions from eight different subjects, indexed as "GV1" to "GV8." Each sequence consists of 200 frames with the resolution of 512×512. During the video capturing, the subjects were asked to make various face expressions, e.g., laughing, shouting, and others.

2) *Experimental Setups:* The video compression tool used in this paper is JM14.2 H.264/AVC reference software [24]. Each test sequence is encoded at 30 f/s. For each video sequence, we encoded it with original H.264/AVC and our proposed schemes based on H.264/AVC at different bitrates by changing the QPs in the range of -12 to 28.

We have particularly adopted two H.264/AVC FRExt [5] profiles that designed for high-fidelity video applications, including the "CAVLC 4:4:4 Intra" profile and the "High 4:4:4" profile (group of picture, GOP = 30 and IPBPB structure). The peak signal-to-noise ratio (PSNR) is used to measure the quality of reconstructed GV.

The evaluations were based on our proposed EIP method, and employed the test conditions described in the previous section. For compression purpose, we tested the original HIP, our previously proposed tailored intra frame prediction (TIP for short) in [39], and the newly proposed PDP scheme based on H.264/AVC. The tested methods were then denoted as "EIP+HIP," "EIP+TIP," and "EIP+PDP" when applied to EIP-generated GVs.

B. Experimental Results

1) *Improvement for Intra Frames:* Our proposed prediction schemes are able to reduce the size of encoded intra frames while maintaining the video quality. To prove this, we used context-adaptive variable-length coding (CAVLC) 4:4:4 Intra profile in this evaluation. As an example, Fig. 12(a) compares the resulted bitrates of tested schemes for GV1 to GV8 (when QP = 4 and very close PSNRs are achieved). It is apparent that our EIP+PDP outperforms both EIP+HIP and EIP+TIP, and a bitrate reduction of 8%–15% can be usually achieved for intra frames.

2) *Improvement for Inter Frames:* The proposed prediction schemes can also provide significant benefits to inter frame encodings. For instance, Fig. 12(b) shows the percentage of successfully intra coded blocks in inter frames (by using HIP, TIP, or PDP) with rate-distortion optimization (RDO)

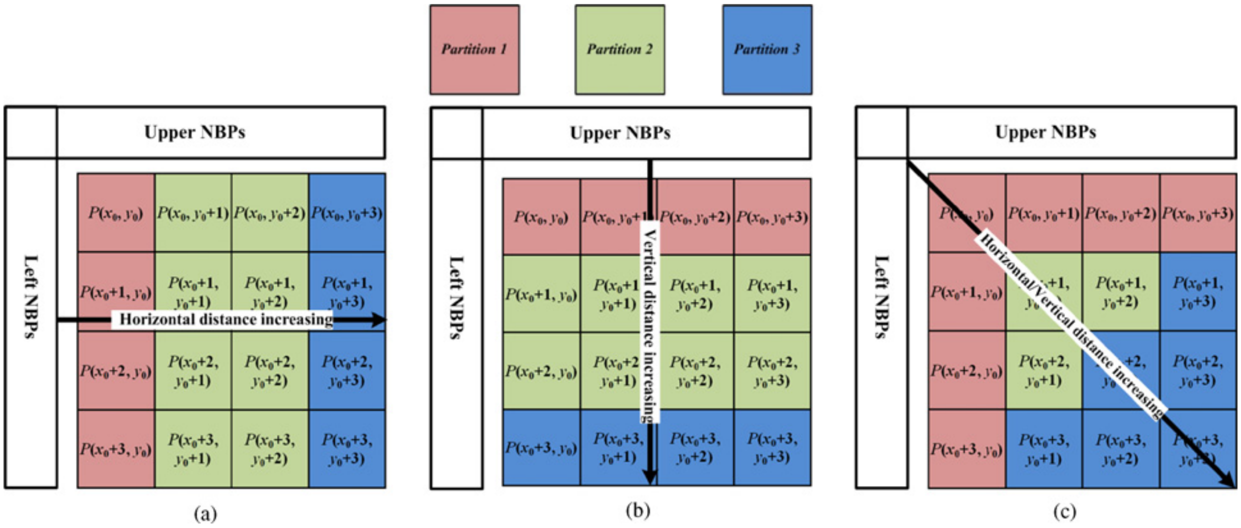


Fig. 11. 4 × 4 block partitions for PPP. (a) Horizontal direction. (b) Vertical direction. (c) Down-left and down-right directions.

TABLE I

PROGRESSIVE PIXEL PREDICTION

Partition 1 Prediction	
Horizontal direction	$P(x, y) = P(x, y-1) + P(x-1, y) - P(x-1, y-1)$
Vertical direction	$P(x, y) = P(x-1, y) + P(x, y-1) - P(x-1, y-1)$
Down-left direction	$P(x, y) = \begin{cases} \frac{1}{2}(P(x-1, y+1) + \Delta_1) + \frac{1}{2}(P(x+1, y-1) + \Delta_2), & \text{if } x{=}x_0,\ y{=}y_0 \\ P(x-1, y+1) + \Delta_1, & \text{if } x{=}x_0,\ y{\neq}y_0 \\ P(x, y) = P(x+1, y-1) + \Delta_2, & \text{if } y{=}y_0,\ x{\neq}x_0 \end{cases}$ $\Delta_1 = P(x-1, y+1) - P(x-2, y+2)$ $\Delta_2 = P(x+1, y-1) - P(x+2, y-2)$
Down-right direction	$P(x, y) = P(x-1, y-1) + P(x-1, y-1) - P(x-2, y-2)$
Partition 2 Prediction	
Horizontal direction	$P(x, y) = P(x, y-1) + PVT^{hor}_{(i,j)}$
Vertical direction	$P(x, y) = P(x-1, y) + PVT^{ver}_{(i,j)}$
Down-left direction	$P(x, y) = \frac{1}{2}(P(x-1, y+1) + PVT^{dl1}_{(i,j)} + P(x+1, y-1) + PVT^{dl2}_{(i,j)})$
Down-right direction	$P(x, y) = \frac{1}{2}(P(x-1, y-1) + PVT^{dr1}_{(i,j)} + P(x-1, y-1) + PVT^{dr2}_{(i,j)})$
Partition 3 Prediction	
Horizontal direction	$P(x, y) = P(x, y-1) + PVT^{hor}_{(i,j)} + (PVM^{hor}_{(i,j+1)} - PVM^{hor}_{(i,j)})$
Vertical direction	$P(x, y) = P(x-1, y) + PVT^{ver}_{(i,j)} + (PVM^{ver}_{(i+1,j)} - PVM^{ver}_{(i,j)})$
Down-right direction	$P(x, y) = P(x-1, y-1) + \frac{1}{2}(PVT^{dr1}_{(i,j)} + PVT^{dr2}_{(i,j)}) + (PVM^{drt}_{(i+1,j+1)} - PVM^{drt}_{(i,j)})$

and High 4:4:4 profile. The figure shows that our prediction schemes, especially PDP, can lead to a considerably increased percentage of well intra coded blocks. In fact, due to the better utilization of GV features, those well intra coded blocks (as selected by RDO) can achieve even better rate-distortion performance than motion estimation that is usually used in inter frames encodings (see next section for more details).

3) *Improvement for the Entire Sequence:* We show in this section the rate-distortion performance of tested schemes with High 4:4:4 profile, which encodes a GOP of 1 intra frame followed by 29 inter frames. As an example, the results for GV1 and GV2 sequences are shown in Fig. 13 (while we have observed similar performance for other test sequences). From the figure, it is clear to observe that our prediction schemes consistently outperform EIP+HIP at various bitrates. In particular, EIP+PDP provides an evident improvement over both EIP+HIP and EIP+TIP in rate-distortion performance, which demonstrated the effectiveness of the more accurate multi-stage predictions of PDP. In the high PSNR range, e.g., when PSNR is greater than 65 dB, we have observed that our EIP+PDP is usually able to achieve bitrate reductions of 10–16% over EIP+HIP (i.e., the original H.264/AVC) while maintaining good video quality. For example, referring to Fig. 13(a), when the PSNR is around 68.7 dB, EIP+PDP is able to achieve a bitrate reduction of 16.1% over EIP+HIP at the same PSNR, as well as a further bitrate reduction of 7.4% over our previous EIP+TIP.

It is to be noted that, the improvement achieved by EIP+PDP does not merely come from better encoding of intra frames. For instance, when comparing Fig. 12(b) to Fig. 13, we have discovered that the rate-distortion performance of the tested schemes are proportional to the percentage of successfully intra coded blocks of inter frames, i.e., the EIP+PDP results in an increased percentage of intra coded

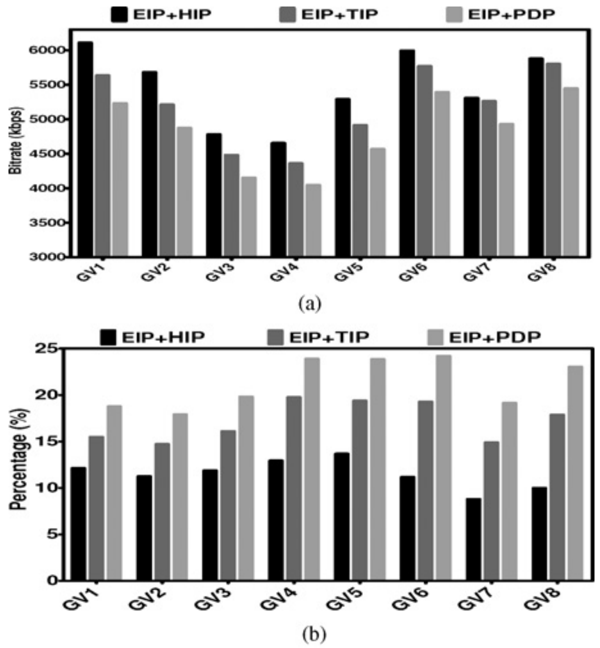


Fig. 12. (a) Comparisons on encoded bitrates for GV1 to GV8 with CAVLC 4:4:4 Intra profile. (b) Percentage of intra coded blocks in inter frames with High 4:4:4 profile.

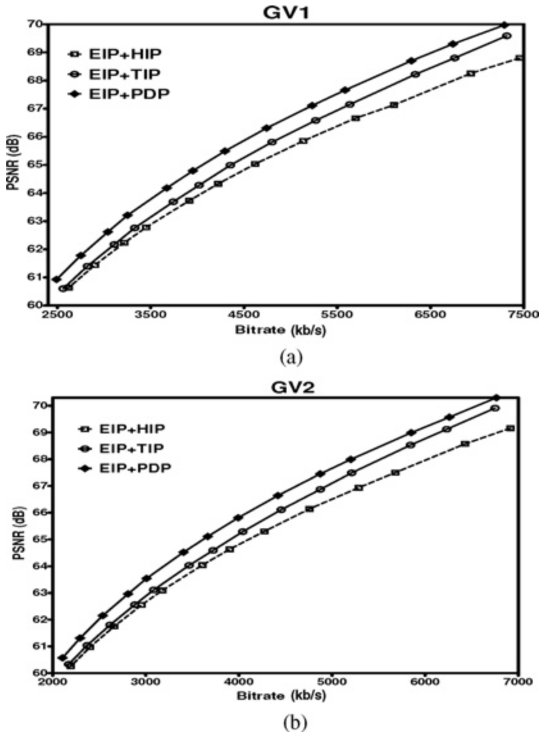


Fig. 13. Rate-distortion performance of tested schemes with High 4:4:4 profile. (a) GV1. (b) GV2.

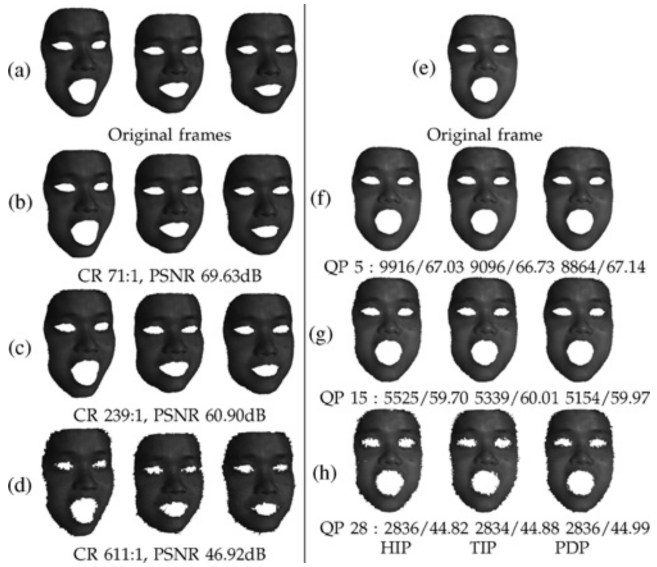


Fig. 14. Experimental results. (a) Sample frames of the original video sequence. (b)–(d) Reconstructed videos at various compression ratios by our H.264/AVC-based PDP scheme. (e) One sample frame of the original video sequence. (f)–(h) Reconstructed videos at various compression ratios by HIP, TIP, and PDP, respectively. The statistics of each video are bitrates and PSNR, respectively.

blocks than other schemes and as a result it achieved the best rate-distortion performance.

4) *Subjective Compression Results:* Fig. 14(a)–(d) shows sample frames of original GV3 (male face), and the reconstructed GVs by EIP+PDP with different compression ratios. We have observed that our scheme will result in no visual distortion (PSNR around 69–70 dB) at a reasonable compression ratio, e.g., 71:1 for GV3, as shown in the figure. At higher compression ratios, e.g., 239:1 for GV3, the reconstructed frames yet maintain good quality (PSNR around 59–61 dB), which still can reasonably reflect accurate human facial expressions, although slight deformations can be found in the boundaries of the human face. However, at very high compression ratios, e.g., over 600:1, the decoded frames are visually distorted.

The right part of Fig. 14 compares the same frame coded by HIP, TIP, and PDP, respectively, at the same QPs. We show the results for a small QP (=5), a medium QP (=15), and a large QP (=28) as an example. For the small and medium QPs, the figure shows that PDP requires considerably lower bitrates for encoding the frame while achieving similar or better PSNR than HIP and TIP. For the large QP, all the prediction schemes result in similar PSNR and similar bitrates due to the significant distortions caused by this QP, i.e., PDP would not provide gains under this large QP. However, within the scope of this paper, we only care about high-quality video applications with small or medium QPs, which can benefit from our proposed PDP scheme.

5) *Comparisons to Other Parameterization Methods:* We present the evaluation results on comparing our EIP method to some other parameterization algorithms, e.g., the HM method [14] and the discrete RF method [40]. Note that both HM and RF do not consider the salient feature correspondence of the input model, thus, the eyes, nose, and mouth are mapped to different locations for different frames, as shown in Fig. 15(b) and (c). In other words, HM and RF are not invariant to the expressions. Our method, in sharp contrast to HM and RF, is invariant to the expressions and can guarantee the exact feature correspondence among different

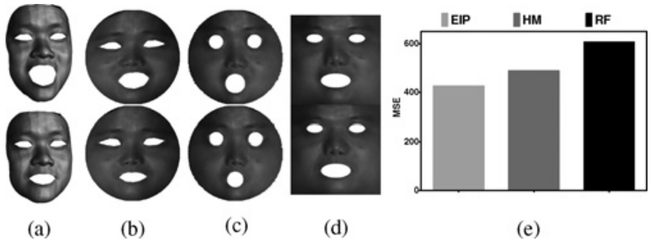


Fig. 15. Compared to other parameterization techniques. (a) Input Facial expression. (b) HM [14]. (c) RF [40]. (d) Our method. (e) MSE. Our algorithm leads to the results that are highly consistent and insensitive to the expressions. Thus, the constructed GVs have better spatial and temporal coherence than that of other methods. As shown in (e), EIP leads to stronger temporal coherence evidenced by smaller MSE produced after a 32×32 Full Search motion estimation (QP = 12).

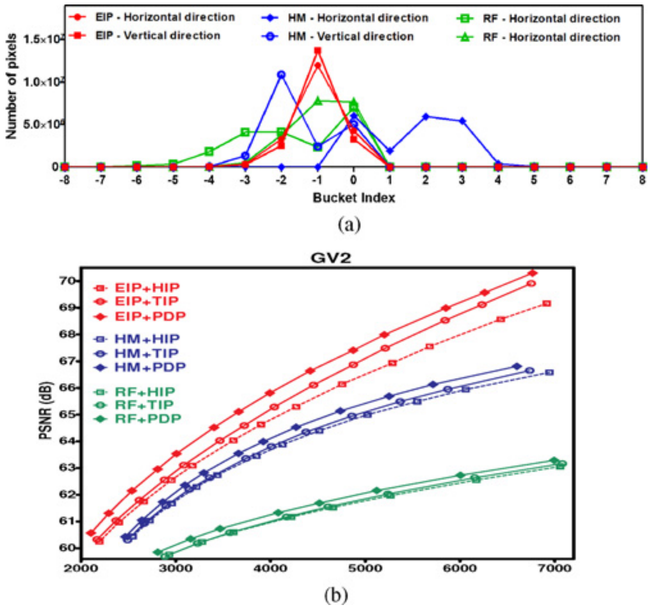


Fig. 16. Comparison of EIP to other parameterization methods (for GV2). (a) EIP leads to stronger spatial coherence evidenced by the GV histograms (horizontal component of R channel and vertical component of G channel). (b) EIP leads to significantly better rate-distortion performance than other methods.

frames. As shown in Fig. 15(d), our EIP can lead to results that are highly consistent and insensitive to the expressions. Thus, the constructed GVs by EIP have stronger temporal and spatial coherence than other methods, which results in better compression ratios.

From the temporal perspective, Fig. 15(e) compares the average MSE produced by EIP to those produced by HM and RF after a 32×32 Full Search motion estimation of H.264/AVC (for GV2, three color channels, and QP = 12 as an example). The results show that our EIP can achieve smaller MSE than other methods, and this demonstrates that EIP can lead to a stronger temporal coherence in the resulted GVs. From the spatial perspective, Fig. 16(a) compares the histograms of GV2 generated by our EIP to those generated by HM and RF (by using the extended Prewitt operator as we described in Section V-A). The histograms show that the resulted pixel distributions of EIP are much more concentrated than other methods, and the pixels are apparent to have stronger spatial coherence with an organized directional pixel-varying trend. Consequently, when combined to our PDP scheme (i.e., EIP+PDP), our EIP method can achieve better compression results over other methods. Fig. 16(b) compares the rate-distortion performance of all the tested methods for GV2. The figure shows that our EIP, which provides stronger temporal and spatial coherence of GV, can achieve significantly better performance over other methods.

TABLE II

COMPARISON OF 3-D GEOMETRY COMPRESSION METHODS

	Object	Feature Alignment	Compression Ratio (%)	Geometry Distortion (RMS/ML) (%)
Han *et al.* [16]	General 3-D model	No	10–18	0.16
Han *et al.* [18]	General 3-D model	No	1.9–16	0.07–0.13
Our method	Facial expression	Yes	0.41	0.10

6) *Comparisons to Other 3-D Motion Data Compression Methods:* At last, we present the comparison of our method to other 3-D motion data compression methods, such as TVM by Han *et al.* [16], [18], as shown in Table II. The statistics of TVM are from the paper [16] and [18], respectively, where the ratio root mean square/max length (RMS/ML) measures the normalized geometry distortion. More details of this measure can be found in [16]. To make a fair comparison, we chose our compression results with a PSNR of 60.9 dB and RMS/ML 0.10%, which is of similar quality as the TVM results. However, our method results in a much better compression ratio than TVM [16], [18]. This reason is as follows. Our GVs work for 3-D expressions, and can guarantee the exact correspondence of salient features among frames. Consequently, the temporal redundancy is largely eliminated. TVM methods, however, work for general 3-D models, which do not have such correspondence. Considering the significant gain of compression ratio, our method is more practical and effective for 3-D expressions. Again, this justifies that the strong temporal and spatial coherence of GVs facilitate the compression algorithm. As a future direction, we will investigate techniques to extend our expression-invariant parameterization algorithm to general 3-D models with guaranteed feature correspondence.

VII. CONCLUSION

This paper presented a novel framework to model and encode 3-D facial expressions using GVs. Within our framework, we parameterized the 3-D expressions with *guaranteed feature correspondence* and stored them into a video format, allowing the 3-D data being significantly compressed by well-studied video compression techniques. Compared to other parameterization methods, our method can lead to results that are highly consistent and insensitive to the expressions, and a higher degree of coherence of constructed GVs, which is highly desirable for video compression. Our experimental results on real-world datasets showed that our framework was very effective for modeling 3-D facial motion data, and our predictive compression scheme can lead to considerably improved rate-distortion performance over the original H.264/AVC without any extra costs thus allowing better GV compression.

Limitations and Future Work: Our feature correspondence-based parameterization does not work for partial 3-D faces, e.g., when the subject rotate his/her face away from the camera. In the future, we will investigate on finding the correspondence between partial 3-D faces. We will also quantitatively evaluate other compression techniques (e.g., wavelet, bandelet, and others) applied to GVs. It would also be interesting to extend the current GV framework to model 3-D human motions.

Acknowledgment

The authors would like to thank the reviewers for their constructive comments and suggestions.

References

[1] S. Zhang and P. Huang, "High-resolution, real-time 3D shape acquisition," in *Proc. CVPRW*, vol. 3. 2004, p. 28.

[2] X. Gu, S. J. Gortler, and H. Hoppe, "Geometry images," in *Proc. SIGGRAPH*, 2002, pp. 355–361.

[3] H. Briceño, P. Sander, L. McMillan, S. Gortler, and H. Hoppe, "Geometry videos: A new representation for 3D animations," in *Proc. SCA*, 2003, pp. 136–146.

[4] T. Wiegand, G. Sullivan, G. Bjöntegaard, and A. Luthra, "Overview of the H.264/AVC video coding standard," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 13, no. 7, pp. 560–576, Jul. 2003.

[5] S. Rusinkiewicz, O. Hall-Holt, and M. Levoy, "Real-time 3D model acquisition," *ACM Trans. Graphics*, vol. 21, no. 3, pp. 438–446, Jul. 2002.

[6] T. P. Koninckx and L. V. Gool, "Real-time range acquisition by adaptive structured light," *IEEE Trans. Patt. Anal. Mach. Intell.*, vol. 28, no. 3, pp. 432–445, Mar. 2006.

[7] D. Nehab, R. Ramamoorthi, J. Davis, and S. Rusinkiewicz, "Spacetime stereo: A unifying framework for depth from triangulation," *IEEE Trans. Patt. Anal. Mach. Intell.*, vol. 27, no. 2, pp. 296–302, Feb. 2005.

[8] L. Zhang, N. Snavely, B. Curless, and S. M. Seitz, "Spacetime faces: High resolution capture for modeling and animation," in *Proc. SIGGRAPH*, 2004, pp. 548–558.

[9] Y. Wang, X. Huang, C. S. Lee, S. Zhang, Z. Li, D. Samaras, D. Metaxas, A. Elgammal, and P. Huang, "High resolution acquisition, learning and transfer of dynamic 3-D facial expressions," in *Proc. CGF*, 2004, pp. 677–686.

[10] S. Wang, Y. Wang, M. Jin, X. Gu, and D. Samaras, "3D surface matching and recognition using conformal geometry," in *Proc. CVPR*, vol. 2. 2006, pp. 2453–2460.

[11] N. J. Mitra, S. Flöry, M. Ovsjanikov, N. Gelfand, L. Guibas, and H. Pottmann, "Dynamic geometry registration," in *Proc. SGP*, 2007, pp. 173–182.

[12] W. Chang and M. Zwicker, "Automatic registration for articulated shapes," *Comput. Graphics Forum*, vol. 27, no. 5, pp. 1459–1468, 2008.

[13] A. Sharf, D. A. Alcantara, T. Lewiner, C. Greif, A. Sheffer, N. Amenta, and D. Cohen-Or, "Space-time surface reconstruction using incompressible flow," in *Proc. SIGGRAPH Asia*, 2008, pp. 1–10.

[14] Y. Wang, M. Gupta, S. Zhang, S. Wang, X. Gu, D. Samaras, and P. Huang, "High resolution tracking of non-rigid motion of densely sampled 3D data using harmonic maps," *Int. J. Comput. Vision*, vol. 76, no. 3, pp. 283–300, Mar. 2008.

[15] A. M. Bronstein, M. M. Bronstein, and R. Kimmel, "Expression-invariant representations of faces," *IEEE Trans. Image Process.*, vol. 2, no. 3, pp. 537–547, Sep. 2007.

[16] S.-R. Han, T. Yamasaki, and K. Aizawa, "Time-varying mesh compression using an extended block matching algorithm," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 17, no. 11, pp. 1506–1518, Nov. 2007.

[17] T. Matsuyama, X. Wu, T. Takai, and T. Wada, "Real-time dynamic 3-D object shape reconstruction and high-fidelity texture mapping for 3-D video," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 14, no. 3, pp. 357–369, Mar. 2004.

[18] S.-R. Han, T. Yamasaki, and K. Aizawa, "Geometry compression for time-varying meshes using coarse and fine levels of quantization and run-length encoding," in *Proc. ICIP*, 2008, pp. 1045–1048.

[19] T. Yamasaki and K. Aizawa, "Patch-based compression for time-varying meshes," in *Proc. ICIP*, 2010, pp. 3433–3436.

[20] H. Hoppe and E. Praun, "Shape compression using spherical geometry images," in *Advances in Multiresolution for Geometric Modelling*. New York: Springer-Verlag, 2005, pp. 27–46.

[21] N.-H. Lin, T.-H. Huang, and B.-Y. Chen, "3D model streaming based on a JPEG 2000 image," *IEEE Trans. Consumer Electron.*, vol. 53, no. 1, pp. 182–190, Feb. 2007.

[22] G. Peyré and S. Mallat, "Surface compression with geometric bandelets," in *Proc. SIGGRAPH*, 2005, pp. 601–608.

[23] G. Sullivan and T. Wiegand, "Video compression: From concepts to the H.264/AVC standard," *Proc. IEEE*, vol. 93, no. 1, pp. 18–31, Jan. 2005.

[24] *The H.264/AVC standard*, document ITU-T Rec. H.264, ITU-T, 2008.

[25] A.-C. Tsai, A. Paul, J.-C. Wang, and J.-F. Wang, "Intensity gradient technique for efficient intra-prediction in H.264/AVC," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 18, no. 5, pp. 694–698, May 2008.

[26] M. Parlak, Y. Adibelli, and I. Hamzaoglu, "A novel computational complexity and power reduction technique for H.264 intra prediction," *IEEE Trans. Consumer Electron.*, vol. 54, no. 4, pp. 2006–2014, Nov. 2008.

[27] A.-C. Tsai, J.-F. Wang, J.-F. Yang, and W.-G. Lin, "Effective subblock-based and pixel-based fast direction detections for H.264 intra prediction," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 18, no. 7, pp. 975–982, Jul. 2008.

[28] D.-Y. Kim, K.-H. Han, and Y.-L. Lee, "Adaptive single-multiple prediction for H.264/AVC intra coding," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 20, no. 4, pp. 610–615, Apr. 2010.

[29] Y. Zheng, P. Yin, O. Escoda, X. Li, and C. Gomila, "Intra prediction using template matching with adaptive illumination compensation," in *Proc. ICIP*, 2008, pp. 125–128.

[30] J. Zhou, H. Zhou, and X. Yang, "An interpolation method by predicting the direction of pixel texture changing trend for H.264/AVC intra prediction," in *Proc. IITA*, vol. 1. 2008, pp. 884–888.

[31] D. Liu, X. Sun, F. Wu, and Y.-Q. Zhang, "Edge-oriented uniform intra prediction," *IEEE Trans. Image Process.*, vol. 17, no. 10, pp. 1827–1836, Oct. 2008.

[32] L. Wang, L.-M. Po, Y. Uddin, K.-M. Wong, and S. Li, "A novel weighted cross prediction for H.264 intra coding," in *Proc. ICME*, 2009, pp. 165–168.

[33] M. W. Marcellin, M. J. Gormish, A. Bilgin, and M. P. Boliek, "An overview of JPEG-2000," in *Proc. DCC*, 2000, pp. 523–544.

[34] T. F. Cootes, G. J. Edwards, and C. J. Taylor, "Active appearance models," *IEEE Trans. Patt. Anal. Mach. Intell.*, vol. 23, no. 6, pp. 681–685, Jun. 2001.

[35] X. Gu and S.-T. Yau, *Computational Conformal Geometry*. Somerville, MA: International Press of Boston, 2008.

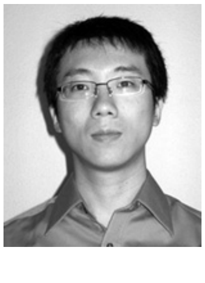
[36] S.-Q. Xin and G.-J. Wang, "Improving Chen and Han's algorithm on the discrete geodesic problem," *ACM Trans. Graphics*, vol. 28, no. 4, pp. 104:1–104:8, Aug. 2009.

[37] M. S. Floater and K. Hormann, "Surface parameterization: A tutorial and survey," in *Proc. AMGM*, 2005, pp. 157–186.

[38] A. Sheffer, E. Praun, and K. Rose, "Mesh parameterization methods and their applications," *Found. Trends. Comput. Graph. Vis.*, vol. 2, no. 2, pp. 105–171, 2006.

[39] J. Xia, Y. He, D. T. P. Quynh, X. Chen, and S. Hoi, "Modeling 3D facial expressions using geometry videos," in *Proc. ACM MM*, 2010, pp. 591–600.

[40] X. Gu, S. Wang, J. Kim, Y. Zeng, Y. Wang, H. Qin, and D. Samaras, "Ricci flow for 3D shape analysis," in *Proc. ICCV*, 2007, pp. 1–8.



Jiazhi Xia received the B.S. and M.S. degrees from Zhejiang University, Zhejiang, China, in 2005 and 2008, respectively. He is currently pursuing the Ph.D. degree from the School of Computer Engineering, Nanyang Technological University, Singapore, under the supervision of Dr. Y. He.

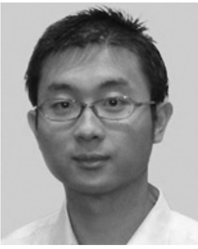
His current research interests include computer graphics and human–computer interaction.

Dao Thi Phuong Quynh received the B.S. degree in applied mathematics and computer science from Moscow State University, Moscow, Russia. She is currently a Ph.D. student with the School of Computer Engineering, Nanyang Technological University, Singapore.

Her current research interests include computational geometry and computer graphics.

Ying He (M'08) received the B.S. and M.S. degrees in electrical engineering from Tsinghua University, Beijing, China, and the Ph.D. degree in computer science from the State University of New York, Stony Brook.

He is currently an Assistant Professor with the School of Computer Engineering, Nanyang Technological University, Singapore. His current research interests include broad areas of visual computing, with a focus on the problems that require geometric computation and analysis.



Xiaoming Chen received the B.S. degree from the Royal Melbourne Institute of Technology, Melbourne, Australia, in 2003, and the M.Info.Tech. and Ph.D. degrees from the University of Sydney, Sydney, Australia, in 2005 and 2009, respectively.

He has been with the National University of Singapore, Singapore, and Nanyang Technological University, Singapore.

Steven C. H. Hoi (M'06) received the Bachelors degree from Tsinghua University, Beijing, China, in 2002, and the Ph.D. degree in computer science and engineering from the Chinese University of Hong Kong, Shatin, Hong Kong, in 2006.

He is currently an Assistant Professor with the School of Computer Engineering, Nanyang Technological University, Singapore. His current research interests include machine learning and its applications to multimedia information retrieval (image and video retrieval), web and social media search and mining, pattern recognition, and computational finance.