A Two-Stage Framework for 3D Face Reconstruction from RGBD Images

Kangkan Wang, *Member, IEEE*, Xianwang Wang, *Member, IEEE*, Zhigeng Pan, and Kai Liu, *Senior Member, IEEE*

Abstract—This paper proposes a new approach for 3D face reconstruction with RGBD images from an inexpensive commodity sensor. The challenges we face are: 1) substantial random noise and corruption are present in low-resolution depth maps; and 2) there is high degree of variability in pose and face expression. We develop a novel two-stage algorithm that effectively maps low-quality depth maps to realistic face models. Each stage is targeted toward a certain type of noise. The first stage extracts sparse errors from depth patches through the data-driven local sparse coding, while the second stage smooths noise on the boundaries between patches and reconstructs the global shape by combining local shapes using our template-based surface refinement. Our approach does not require any markers or user interaction. We perform quantitative and qualitative evaluations on both synthetic and real test sets. Experimental results show that the proposed approach is able to produce high-resolution 3D face models with high accuracy, even if inputs are of low quality, and have large variations in viewpoint and face expression.

Index Terms—Face reconstruction, sparse coding, surface modeling, statistical learning, deformation transfer, rigid registration, non-rigid registration, surface tracking

1 Introduction

Human face modeling has many applications including movie special effects, biometric authentication, video conferencing, human-computer interactions, etc. Most current systems address the 3D face reconstruction problem in the form of multiple views or 3D acquisition modalities such as a laser range scanner or structured-light camera/projector with patterns or special textures [1], [2], [3]. The high cost of special hardware and substantial manual post-processing reduce the operational flexibility of such systems and limit their availability in most practical applications. This paper proposes a new low-cost 3D face reconstruction approach that potentially could make 3D face scanning a household routine and enable many new applications that are currently blocked by the prohibitive cost and the cumbersome user interfaces of the current scanning techniques.

Many approaches for 3D face reconstruction introduce a deformable model and recover shapes by optimizing an objective function that measures the fit of the model to the input data. However, these approaches require good initial estimates that must be relatively close to the true shape. Moreover, additional constraints such as linear elasticity [4] or minimization of surface stretching [5], [6] are imposed to resolve the inherent ambiguities. Still, the strong assumption over the input scenes prevents these methods from reconstructing human faces with realistic structures. To relax these constraints, approaches over the years endeavored to make this problem tractable by introducing prior models, e.g., parametric models [7] or machine learning [8], [9], to reduce the deformation solution space. These techniques take advantage of training data in conjunction with dimensionality reduction techniques to learn low-dimensional models. The reconstruction quality depends heavily on the training data sets. While producing some impressive results, these approaches cannot generalize well and lack the modeling precision when the observations cannot be explained by the learned model. Therefore, these approaches have difficulty in capturing facial structures such as facial expressions when the training database does not contain any model that is close to the input.

In this paper, we present a novel approach to reconstruct 3D faces using RGBD images, i.e., the well-registered color images and depth maps captured from a Microsoft's Kinect sensor. The motivation is that depth measurement avoids the ambiguity caused by perspective projection in 2D images, and is invariant to lighting conditions. Nevertheless, using depths and color images for reconstruction is not as easy as it appears. A depth sensor only generates a point cloud with random noise and corruption. The problem is further complicated by variations in viewpoint and expression. To overcome the problems, we take advantage of the fact that the local homogeneous surface follows the same deformation rule and has more constraints in the local surface deformation. Therefore, our solution is to divide the original face surface/depth into regions or patches, to which vertices are attached. The strategy of patch division serves four purposes: first to enforce inter-patch rigidity

- *K. Wang is with the Department of Computer Science, Zhejiang University, Room 410, the State Key Lab of CAD&CG, Hangzhou, Zhejiang 310058, China. E-mail: wangkangkan@gmail.com.*
- *X. Wang is with Hewlett-Packard Company, 1501 Page Mill Road, Palo Alto, CA 94304. E-mail: xianwang.wang@hp.com.*
- *Z. Pan is with Hangzhou Normal University, Cangqian Street, Haishu Road 58, Yuhang, Hangzhou, Zhejiang 311121, China. E-mail: zgpan@hznu.edu.cn.*
- *K. Liu is with the School of Electrical Engineering and Information, Sichuan University, Chengdu Sichuan 610065, China. E-mail: kailiu@scu.edu.cn.*

Manuscript received 25 Sept. 2012; revised 14 July 2013; accepted 12 Nov. 2013. Date of publication 4 Dec. 2013; date of current version 10 July 2014. Recommended for acceptance by T. Cootes.
For information on obtaining reprints of this article, please send e-mail to: reprints@ieee.org, and reference the Digital Object Identifier below.
Digital Object Identifier no. 10.1109/TPAMI.2013.235

0162-8828 © 2013 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission.
See http://www.ieee.org/publications_standards/publications/rights/index.html for more information.

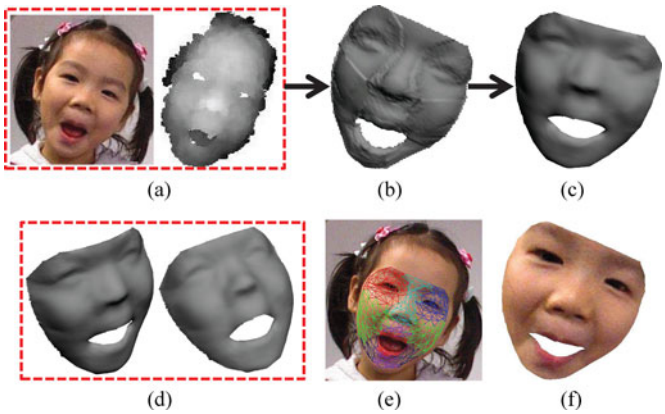


Fig. 1. The top row describes two-stage reconstruction of an input with strong random noise, corruption, and large deformation: Stage 1 (a→b) eliminates most of random noise and corrects the corruption through local sparse coding, while stage 2 (b→c) refines the reconstruction by further reducing the noise on the boundaries between patches and in the interior of patches via our surface refinement. The bottom row shows different presentations of the reconstruction result. It shows that our approach can reconstruct facial structures in expression, eyes, mouth, etc., and the recovered result fits well with the color image. (a) RGBD images from a Kinect sensor. (b) The recovered result from local sparse coding. (c) The reconstructed 3D model corresponding to (a). (d) Different views of (c). (e) 2D projection of (c) on the image. Note that the numbers of vertices (23,469 to 400) and triangles (46,159 to about 500) are reduced for visualization in the whole paper. (f) Texture mapping of the color image in (a) onto (c).

constraints and to eliminate the definition of explicit 3D-3D point correspondences with respect to the reference model in the data set; second to remove the requirement that the database should contain at least one 3D model that is close to the input in shape for a good reconstruction; thus, eliminate the need of building a face database covering all of face expressions and viewpoints, which generally represents a very significant amount of data; third to improve the ability to filter the noise and to fill in the missing regions due to corruption through our proposed patched-based local sparse coding; and fourth to capture local deformations with fewer examples because shapes within a patch have considerably smaller variance. A global face can then be recovered by encouraging its patches to conform to the recovered local models with the technique of our surface refinement. Our approach allows for pose variations by warping depth patches to frontal views.

Our approach, illustrated in Fig. 1, is to first locate facial features by applying active appearance models (AAM) [10] on the color image and segment out facial patches from the depth map. The extracted patches are reconstructed separately using local sparse coding with a limited set of training samples at the first stage. To efficiently handle noisy depth data, the samples are augmented with trivial templates [11]. After that, the locally recovered pieces are merged together to get a global 3D model using the surface refinement, which fits the merged result to a template by finding a trade-off between well-fitting seams, minimization of geometric transformation and distance metrics. Each reconstructed 3D face has the same mesh topology with 23,469 vertices and 46,159 triangles as templates in the training database. Note that AAM is only used for extraction of faces and patch division in our method.

In summary, the main technical contribution of this paper is that we present a novel and robust two-stage framework of building 3D face models from a single depth sensor. The novelty also comes from patch division on depth images and local sparse coding. Experimental results show that both have brought substantial benefits to our approach, including robustness to noise, elimination of the large size requirement of training database in statistical learning-based approaches, strong capability of handling large variations from viewpoint and face expression, and elimination of dependency on explicit 3D-to-3D correspondences for fitting. The sparse coding technique in itself is not technically new, however, it has not been applied to 3D face reconstruction before in our way. Moreover, our surface refinement is a novel way to map these local recoveries onto a base mesh and recover a globally consistent solution. In addition, we are the first to introduce these techniques for processing low-quality depth maps. Therefore, through combining our proposed novel sub-solutions, our approach is able to recover a smooth, high-vertex-count 3D mesh and capture the facial structures (e.g., expressions, nose, eyes, etc.) in a unified way even in the presence of noise, corruption, low resolution, and large deformations, as illustrated in Fig. 1, which is beyond of the current state of the arts.

The rest of the paper is organized as follows. Section 2 reviews the related work on 3D shape recovery. Section 3 introduces the data sources, the preprocessing of the training data set, and the problem that we address. In Section 4, we describe our local sparse coding for local shape recovery and noise elimination. Section 5 presents the reconstruction procedure of the global shape by combining the recovered local surface shapes. The experimental results and analysis are shown in Section 6, and the conclusion is made in Section 7.

2 Related Work

Over the years, many approaches have been proposed to recover surfaces from color and/or depth images using deformation or geometric cues. To make this problem tractable, these approaches introduce the prior knowledge about the observed surface to limit the space of possible deformations. The prior knowledge is usually presented in the form of template-based models, reconstruction models from non-rigid structure-from-motion (SFM) algorithms, and models learned from statistical learning techniques.

Template-based models: Template-based methods recover the deformation surface by fitting a template or reference model to the input with non-convex optimization, which uses a regularization term over vertex displacements as prior knowledge. To obtain an unambiguous solution, geometry or deformation priors, e.g., constraints in stretching [5], [6], [12], [13], geodesic distances [14] and elasticity [4], are proposed to limit the search space of solution. However, these methods require a search for correspondences between the input and reference views, which is computationally expensive. Moreover, noise and corruption in depths poses a great challenge to linear or nonlinear optimizations.

Non-rigid SFM: Non-rigid SFM methods rely on tracking of feature points to simultaneously recover 3D surface

points and the deformation models [3], [15] through image sequences. The advantage of these approaches is that they require less priori knowledge. However, these methods suffer from two major drawbacks. First, a sufficient number of feature points are required to be tracked throughout the whole sequence to learn both shape and motion, which limits their applicability. Second, similar to template-based model techniques, Non-rigid structure from motion techniques prove effective only for relatively small deformations or smooth deformations, because they oversimplify the motion of surface by modeling deformations as either a linear combination of online learned [16], constant basis vectors [17], or several piecewise rigid objects independently moving with respect to one another [18].

Statistical learning: Due to the complexity of modeling the true physical properties of surfaces, statistical learning techniques therefore have been increasing over the years to capture the non-linear physics of large deformations. They take advantage of training data in conjunction with dimensionality reduction techniques to learn low-dimensional models [7], [19]. Most of these models currently in use trace their roots to the early Active Appearance Models [10] in 2D case, followed by Morphable Models [20]. These linear models can capture the true variability more than modal analysis, because they are learned from training examples. However, these methods share the same restriction of smooth constraints as non-rigid SFM. Due to many degrees of freedom from highly deformable surfaces, learning of the models would be tractable only when sufficient training data are available. Thus, the challenge of building database with enough examples has limited the spread of non-linear model-based approaches [21]. Furthermore, nonlinear learning generally involves optimization of complex objective function that may be difficult to resolve because of non-convexity. Through combination of template-based modeling and statistical learning, our approach is able to recover 3D facial models from the data with large appearance variations without the requirement of sufficient training data and the restriction of smooth constraints.

3 Overview

We have data sets from two different sources: the input data of well-registered RGBD images from a Kinect sensor and a pre-generated training database of depth maps in the frontal view and the corresponding 3D face models.

Input data: A depth sensor provides real-time scene scanning, in which each pixel contains intensity and range information for scene points. We aim to automatically recover 3D face surfaces that best fit the observed low-resolution color images and depth maps from a single depth sensor. We employ a Kinect camera which gives a 640×480 image at 30 frames per second with depth resolution of a few millimeters. Kinect has several advantages over traditional intensity sensors including simplification of background subtraction, independence of lighting conditions, and avoidance of projection ambiguity. Unfortunately, the captured data contains the inherent characteristics like low resolution, strong random noise, distortion, and corruption, which pose a great challenge to 3D facial reconstruction.

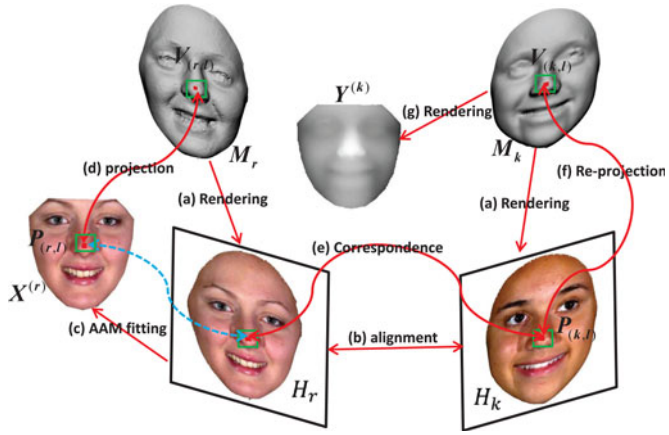


Fig. 2. The procedure of generating training database. By repeating (a)∼(g), we generate all the training models $\{\mathbf{V}_X^{(k)}\}_{k=1}^N$ and their frontal depth images $\{\mathbf{Y}^{(k)}\}_{k=1}^N$ respectively. Note that all $\{\mathbf{V}_X^{(k)}\}_{k=1}^N$ have the same number of vertices and triangles, and vertex-to-vertex correspondences.

Training database. We use the BU-4DFE [22] to generate our face training database. The original data set consists of 101 subjects (58 females and 43 males). Each subject has 6 model sequences, each of which respectively exhibits different expression (e.g., anger, disgust, fear, happiness, sadness, and surprise). Each expression sequence contains 100 frames, resulting in a total of $101 * 6 * 100 = 60{,}600$ 3D face expression models in the database. Each model has a resolution of approximately 35,000 vertices. To conveniently describe the procedure of the database building, we introduce the following notations.

We select one of the models from BU-4DFE to be the *reference model* $\mathbf{M}_r$. We also select $N = 2{,}500$ face models as *template models* $\{\mathbf{M}_k\}_{k=1}^N$ totally for our training database. That is, we select $4 \sim 5$ models for each expression of each subject. Fig. 2 shows the data processing pipeline from which we can build all template models $\{\mathbf{V}_X^{(k)}\}_{k=1}^N$, which have vertex-to-vertex correspondences. The pipeline first uses the standard computer graphics techniques to respectively render the frontal images from texture mapped $\mathbf{M}_r$ and $\{\mathbf{M}_k\}_{k=1}^N$ (Fig. 2a). Each frontal image $\mathbf{H}_k$ from $\mathbf{M}_k$ is then automatically aligned to the frontal image $\mathbf{H}_r$ from $\mathbf{M}_r$ using the method in [23] (Fig. 2b). Thus, we have pixel-to-pixel correspondences between $\mathbf{H}_k$ and $\mathbf{H}_r$. Following that, the AAM fitting algorithm [10] is then run on the frontal image $\mathbf{H}_r$ to segment out the face region $\mathbf{X}^{(r)}$ (Fig. 2c). For each vertex $\mathbf{V}_{r,l}$ of $\mathbf{M}_r$, if the projected pixel $\mathbf{P}_{r,l}$ of $\mathbf{V}_{r,l}$ (Fig. 2d) is located in the interior of $\mathbf{X}^{(r)}$, we can find the correspondence $\mathbf{P}_{k,l}$ on $\mathbf{X}^{(k)}$ (Fig. 2e) due to the known pixel-to-pixel correspondences between $\mathbf{H}_r$ and $\mathbf{H}_k$. $\mathbf{P}_{k,l}$ is then reprojected to 3D space and the projection ray intersects $\mathbf{M}_k$ at some front-most point $\mathbf{V}_{k,l}$ (Fig. 2f). Connection of all $\mathbf{V}_{k,l}$ using the topology of $\mathbf{M}_r$ corresponding to the region in $\mathbf{X}^{(r)}$ results in the model $\mathbf{V}_X^{(k)}$. From $\mathbf{V}_X^{(k)}$, we can render the synthesized frontal depth image $\mathbf{Y}^{(k)}$ (Fig. 2g). By repeating the procedure, we can obtain all the models $\{\mathbf{V}_X^{(k)}\}_{k=1}^N$ that have the same number of vertices and triangles (in our experiments, 23,469 vertices and 46,159 triangles for each $\mathbf{V}_X^{(k)}$). Furthermore, we segment all $\{\mathbf{Y}^{(k)}\}_{k=1}^N$ into $N_s = 11$

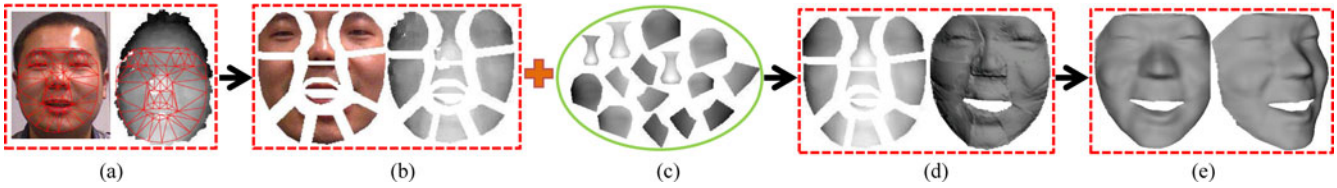


Fig. 3. The pipeline of 3D face reconstruction. (a) AAM fitting on an input pair of color image and depth map. (b) Patch division. (c) The training database with different patches such as left eyes, noses, etc. (d) Local sparse coding. (e) Surface refinement.

patches based on the AAM shape model as illustrated in Fig. 3b.

Notations	
N	the number of templates in the training database
$\mathbf{M}_r$	the reference model
$\mathbf{M}_k$	the k^{th} template in the training database
$\mathbf{H}_k$	the rendered frontal image from $\mathbf{M}_k$
$\mathbf{H}_r$	the rendered frontal image from $\mathbf{M}_r$
$\mathbf{X}^{(r)}$	the extracted AAM face region from $\mathbf{H}_r$
$\mathbf{V}_{r,l}$	the l^{th} vertex of $\mathbf{M}_r$
$\mathbf{P}_{r,l}$	the projected pixel of $\mathbf{V}_{r,l}$ on $\mathbf{H}_r$
$\mathbf{P}_{k,l}$	the correspondence to $\mathbf{P}_{r,l}$ on $\mathbf{H}_k$
$\mathbf{X}^{(k)}$	the extracted face region from $\mathbf{H}_k$ corresponding to $\mathbf{X}^{(r)}$
$\mathbf{V}_{k,l}$	the vertex corresponding to $\mathbf{P}_{k,l}$ on $\mathbf{M}_k$
$\mathbf{V}_X^{(k)}$	the built 3D model from $\mathbf{M}_k$ that has the same topology as the one from $\mathbf{M}_r$
$\mathbf{Y}^{(k)}$	the rendered frontal-view depth from $\mathbf{V}_X^{(k)}$
N_s	the number of patches for division in each face

Problem definition: We are given a set of face models $\{(\mathbf{Y}^{(1)}, \mathbf{V}_X^{(1)}), \ldots, (\mathbf{Y}^{(N)}, \mathbf{V}_X^{(N)})\}$ for training. We also know an input pair of well-registered color image $\mathbf{X}$ and depth map $\mathbf{Y}$ from Kinect, which may be different from ones in the database in terms of subject, viewpoint, face expression, etc. Note that $\{\mathbf{Y}^{(k)}\}_{k=1}^{N}$ are only of frontal views. The goal of our algorithm is to reconstruct the 3D face model $\mathbf{V}_X$. An outline of the processing pipeline for our approach is given in Fig. 3. We first remove the background by segmenting out the face region of the depth map using the AAM outline (Fig. 3a) as described above. We then divide the depth map into N_s patches (Fig. 3b). The depth map is quite noisy and contains holes at arbitrary positions. Following that, we employ the proposed local sparse coding (Fig. 3d) to eliminate the noise on these local patches respectively and fill up holes by taking advantage of all $\mathbf{Y}^{(i)}$ and the corresponding template models $\mathbf{V}_X^{(i)}$ in the training database (Fig. 3c). Because this step only captures piece-wise deformations and local models, and reduces the local noise, subsequently the shape is globally refined with our surface refinement (Fig. 3e). In the following, we will discuss the two stages of our approach in details.

4 Local Sparse Coding

The depth patches of the same position (e.g., left eye) under different subjects lie in a low dimensional subspace. That is, any new input depth patches can be approximately spanned by a set of template depth patches. Let $\mathbf{Y}_j^{(k)}$ represent the jth patch from kth template depth. Given target instance set $\mathbf{A}_j = [\mathbf{y}_j^{(1)} \ldots \mathbf{y}_j^{(N)}] \in \mathbb{R}^{d \times N} (d \ll N)$, the jth patch of the input $\mathbf{Y}$, $\mathbf{Y}_j$, can be approximated by a linear combination of target instances in $\mathbf{A}_j$

$$\mathbf{y}_j = \mathbf{A}_j\omega + \varepsilon = \omega_1\mathbf{y}_j^{(1)} + \omega_2\mathbf{y}_j^{(2)} + \cdots + \omega_N\mathbf{y}_j^{(N)} + \varepsilon, \quad (1)$$

where $\mathbf{y}_j \in \mathbb{R}^d$ is an 1D vector formed by stacking depth patch $\mathbf{Y}_j$, $\mathbf{y}_j^{(k)} \in \mathbb{R}^d$ is vectorized depth patch of $\mathbf{Y}_j^{(k)}$, $\omega = [\omega_1, \omega_2, \ldots, \omega_N]^T$ is the coefficient vector, and ε is a noise term that models the effect of noise and corruption. The nonzero entries of ε indicate the pixels corrupted by noise or distortion. To explicitly capture the positions of the corrupted or occluded pixels, we adopt the technique of trivial templates [11] here, such that each trivial template has only one nonzero element. The trivial templates $I = [I_1, I_2, \ldots, I_d] \in \mathbb{R}^{d \times d}$ are augmented into $\mathbf{A}_j$ as the following:

$$\mathbf{y}_j = [\mathbf{A}_j, I, -I] \begin{bmatrix} \omega \\ \boldsymbol{e}^+ \\ \boldsymbol{e}^- \end{bmatrix} = \mathbf{B}_j\boldsymbol{\alpha}, \qquad s.t.\ \boldsymbol{\alpha} \geq 0, \quad (2)$$

where $I(-I)$ denotes positive (negative) trivial templates, $I_i \in \mathbb{R}^d$ is a vector with only one nonzero entry in the position i, $\mathbf{e}^+(\mathbf{e}^-)$ is called a positive (negative) trivial coefficient vector, $B_j = [\boldsymbol{A}_j, \boldsymbol{I}, -\boldsymbol{I}] \in \mathbb{R}^{d \times (N+2d)}$, and $\boldsymbol{\alpha}^T = [\omega, \boldsymbol{e}^+, \boldsymbol{e}^-] \in \mathbb{R}^{(N+2d)}$ is a non-negative coefficient vector. The argument for enforcing nonnegativity constraints on $\boldsymbol{\alpha}$ comes from their ability to fill up the holes on $\mathbf{y}_j$ and filter out corruption, which is often happened in captured stream as shown in Fig. 11. This problem can be avoided by enforcing nonnegativity constraints as shown in Figs. 11c, 11d, and 11e.

There is a sparse solution to Eq. (2). The reason is twofold. First, we intend to choose the most similar templates from $\mathbf{A}_j$ for matching $\mathbf{y}_j$. So the coefficient ω should be as sparse as possible. Second, the corrupted pixels by noise and distortion are typically a fraction of the depth map. Therefore, there are only a limited number of nonzero elements in e^+ and e^-. The minimization problem to obtain $\boldsymbol{\alpha}$ can be defined as follows,

$$\min_{\boldsymbol{\alpha}} \|\boldsymbol{B}_j\boldsymbol{\alpha} - \mathbf{y}_j\|_2 + \lambda\|\boldsymbol{\alpha}\|_1, \quad (3)$$

where $\|\cdot\|_1$ and $\|\cdot\|_2$ denote the ℓ_1 and ℓ_2 norms, respectively.

$\mathbf{y}_j$ is subject to pose variation or misalignment in real cases. Instead of observing $\mathbf{y}_j$, we observe the warped image $\widetilde{\mathbf{y}}_j = \mathbf{y}_j \circ \tau_j^{-1}$, where τ_j is a transformation applied on the depth domain. The problem in Eq. (3) is transferred to the one as follows by seeking a deformation τ_j and $\boldsymbol{\alpha}$ that allows the sparsest representation,

$$\min_{\boldsymbol{\alpha}, \tau_j} \|\mathbf{B}_j\boldsymbol{\alpha} - \widetilde{\mathbf{y}}_j \circ \tau_j\|_2 + \lambda\|\boldsymbol{\alpha}\|_1. \quad (4)$$

This is a difficult nonconvex optimization problem due to the simultaneous optimization over the coefficients $\boldsymbol{\alpha}$ and transformation τ_j. It can be solved via iterative ℓ_1-minimization. The initial transformation τ_0 of τ_j is first estimated using

Procrustes analysis [24] with correspondences that are established in the first stage. The initialization can be refined to an estimate of the true transformation by repeatedly linearizing about the current estimate of τ_j and seeking a deformation step $\triangle\tau$ that best satisfies of the registration error in terms of ℓ_1-norm:

$$\min_{\boldsymbol{\alpha},\triangle\tau} \|\boldsymbol{B}_j\boldsymbol{\alpha} - \widetilde{\mathbf{y}}_j \circ \tau_j^{(i)} - \boldsymbol{J} \triangle \tau\|_2 + \lambda\|\boldsymbol{\alpha}\|_1, \tag{5}$$

where $\boldsymbol{J} = \frac{\partial}{\partial\tau_j}\widetilde{\mathbf{y}}_j \circ \tau_j$ is the Jacobian of $\widetilde{\mathbf{y}}_j \circ \tau_j$ with respect to the transformation τ_j, and $\triangle\tau$ is the step in τ_j. Our implementation solves the ℓ_1-regularized least square problem via an interior-point method based on [25]. It can be seen from Eq. (5) that noise, corruption, and pose variations in depths are well considered. The depth map can be therefore refined by the optimization procedure summarized as Algorithm 1. From the experiments, we can see that the proposed local sparse coding performs much better than PCA on depth patches with small size of training database in handling depth appearance variations due to noise and occlusion.

Algorithm 1. Local Sparse Coding for Noise and Corruption Correction

1: **Input:** j^{th} patches from all template depth maps $\mathbf{A}_j = [\mathbf{y}_j^{(1)} \dots \mathbf{y}_j^{(N)}] \in \mathbb{R}^{d\times N}$, and the j^{th} patch of the input $\mathbf{y}$.
2: Generate $\boldsymbol{B}_j$ with $\mathbf{A}_j$ and trivial templates $\boldsymbol{I}$
3: $\tau_j^{(0)} = \tau_0$.
4: **do**
5: $\overline{\mathbf{y}}_j(\tau_j) \leftarrow \frac{\widetilde{\mathbf{y}}_j\circ\tau_j}{\|\widetilde{\mathbf{y}}_j\circ\tau_j\|_2}$; $\boldsymbol{J} \leftarrow \frac{\partial}{\partial\tau_j}\overline{\mathbf{y}}_j(\tau_j)|_{\tau_j^{(i)}}$
6: Estimate $\boldsymbol{\alpha}$ and $\triangle\tau$ using Eq. 5;
7: $\tau_j^{(i+1)} \leftarrow \tau_j^{(i)} + \triangle\tau$;
8: **while** $\|\tau_j^{(i+1)} - \tau_j^{(i)}\| \geq \delta$.
9: **Output:** Compute the refined depth patch $\widehat{\mathbf{y}}_j \leftarrow \mathbf{A}_j\boldsymbol{\omega} \circ (\tau_j^{(i+1)})^{-1}$.

5 Surface Refinement

Applying the local sparse coding significantly reduces distortion and corruption caused by noise and occlusion. However, the recovered depth still includes unstructured and random noise on each patch $\widehat{\mathbf{y}}_j$, as shown in Figs. 1b and 3d. Combination of $\widehat{\mathbf{y}}_j$ will result in the whole recovered depth $\widehat{\mathbf{y}}$. Accordingly, we obtain the estimated 3D model $\widehat{\mathbf{V}}_X$. In addition, different patches are independently modeled in our patch-based face model, which results in the local misalignment on $\widehat{\mathbf{y}}$. The goal of surface refinement is to unveil the unstructured noise and enforce the consistency on the boundary vertices between regions of $\widehat{\mathbf{V}}_X$ by the deformation of $\mathbf{V}_X^{(k)}$ onto $\widehat{\mathbf{V}}_X$.

We represent the deformation as a collection of affine transformations, i.e., assign an affine transformation to each triangle on $\mathbf{V}_X^{(k)}$. An affine transformation can be defined as the 3×3 matrix $\mathbf{T}$ and 3×1 displacement vector $\mathbf{d}$. However, the affine transformation cannot be determined by the three vertices of a triangle before and after deformation. Similar to [26], we add a fourth vertex in the direction perpendicular to the triangle as

$$\mathbf{v}_4 = \mathbf{v}_1 + (\mathbf{v}_2 - \mathbf{v}_1) \times (\mathbf{v}_3 - \mathbf{v}_1)/\sqrt{|(\mathbf{v}_2 - \mathbf{v}_1) \times (\mathbf{v}_3 - \mathbf{v}_1)|}, \tag{6}$$

where $\mathbf{v}_i, i \in 1\dots 3$, are the undeformed vertices of the triangle on $\mathbf{V}_X^{(k)}$. Then, the transformation of the triangle can be written as

$$\mathbf{T}\mathbf{v}_i + \mathbf{d} = \widetilde{\mathbf{v}}_i \quad i \in 1\dots 4, \tag{7}$$

where $\widetilde{\mathbf{v}}_i, i \in 1\dots 3$, are the deformed vertices of the triangle on $\mathbf{V}_X$. By subtracting the first equation from the others, we can get a closed form expression for $\mathbf{T}$,

$$\mathbf{T} = \widetilde{\mathbf{Q}}\mathbf{Q}^{-1}, \tag{8}$$

where $\mathbf{Q}$ and $\widetilde{\mathbf{Q}}$ are defined as follows:

$$\begin{aligned}\mathbf{Q} &= [\mathbf{v}_2 - \mathbf{v}_1 \quad \mathbf{v}_3 - \mathbf{v}_1 \quad \mathbf{v}_4 - \mathbf{v}_1],\\ \widetilde{\mathbf{Q}} &= [\widetilde{\mathbf{v}}_2 - \widetilde{\mathbf{v}}_1 \quad \widetilde{\mathbf{v}}_3 - \widetilde{\mathbf{v}}_1 \quad \widetilde{\mathbf{v}}_4 - \widetilde{\mathbf{v}}_1].\end{aligned} \tag{9}$$

Therefore, the elements of $\mathbf{Q}^{-1}$ are coordinates of the known original vertices of the template model $\mathbf{V}_X^{(k)}$, while the elements of $\widetilde{\mathbf{Q}}$ are coordinates of the unknown deformed vertices of $\mathbf{V}_X$. From this definition, we see that the elements of $\mathbf{T}$ are linear combinations of the coordinates of the unknown deformed vertices. Given these definitions, we formulate our surface refinement with two constraints as follows:

$$\begin{aligned}\{\widetilde{\mathbf{v}}_k\}_{k=1}^n = \min_{\mathbf{v}_1\dots\mathbf{v}_n} & w_1 \sum_{i=1}^{|T|} \sum_{j\in adj(i)} \|\mathbf{T}_i - \mathbf{T}_j\|_F^2\\ & + w_2 \sum_{i=1}^{n} \|\mathbf{v}_i - \mathbf{c}_i\|^2,\end{aligned} \tag{10}$$

where $|T|$ is the number of triangles in $\mathbf{V}_X^{(k)}$, $\{\mathbf{T}_i\}_{i=1}^{|T|}$ are affine transformations defined in terms of target vertices, $adj(i)$ denotes the set of triangles adjacent to triangle i, $\mathbf{c}_i$ is the closest valid point on $\widehat{\mathbf{V}}_X$ to target vertex i. The first item enforces the change smoothness in deformation by minimizing the difference of the transformations between adjacent triangles. The purpose of the second item is to impose the constraint that the position of each vertex of $\mathbf{V}_X^{(k)}$ should be equal to the closest valid point on $\widehat{\mathbf{V}}_X$. By replacing all transformations with Eq. (8) and (9), the problem can be rewritten in the matrix form

$$\min_{\widetilde{\mathbf{v}}_1\dots\widetilde{\mathbf{v}}_n} \|\mathbf{b} - \mathbf{G}\widetilde{\mathbf{x}}\|_2^2, \tag{11}$$

where $\widetilde{\mathbf{x}}$ is a vector of unknown deformed vertex coordinates, $\mathbf{b}$ is a vector containing entries from the source transformations, and $\mathbf{G}$ is a large, sparse matrix, and its entries depend only on the target model's undeformed vertex locations. Therefore, all the vertices of the target shape $\mathbf{V}_X$ can be solved in the least-square sense.

6 Experiments

In this section we describe the experiments performed to evaluate our method. Both qualitative and quantitative results are shown with several challenging data sets, and are compared to the state of the art. Please refer to the accompanying video for more results. Unless otherwise specified, parameters below are used in the experiments: $N = 300$, $N_s = 11$, $\lambda = 0.05$ in Eq. (5), and $w_1 = 1.0$, and $w_2 = 1.0$ in Eq. (10).

Test data. We use both synthetic and real test data in our experiments. For quantitative analysis on noise tolerance,

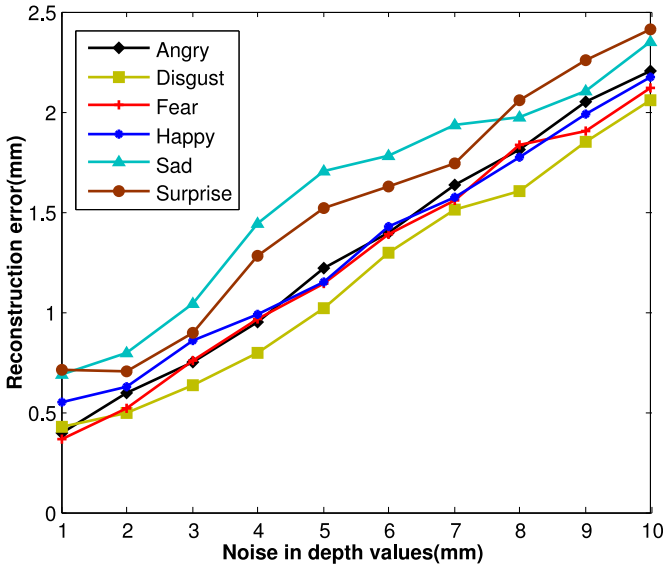


Fig. 4. Reconstruction accuracy (mAP) of our method on six sequences of synthetic depths with six different expressions from five subjects in terms of levels of random noise.

we employ certain 3D scans [22] that are held out of our training models to generate synthetic test data. The synthetic data includes 20 subjects. In addition, we perform qualitative analysis on our real test set with large variations in pose and expression, which are captured over 15 subjects including adults and children from Kinect. Note that there are not public data sets available as our inputs. For all the experiments we limit user pose variations to $\pm 45^\circ$ (e.g., the frontal view is 0°) in depth data in order that the whole face can be seen.

Error metrics. We quantify reconstruction accuracy as mean average precision (mAP) over all vertices in millimeter (mm), which is measured as

$$\epsilon = \frac{1}{N_f}\sum_{k=1}^{N_f}\frac{1}{N_m}\sum_{i=1}^{N_m}\|\mathbf{v}_i - \widetilde{\mathbf{v}}_i\|, \tag{12}$$

where N_f and N_m are numbers of frames and vertices, respectively. $\mathbf{v}_i$ is measured ground truth of vertex location and $\widetilde{\mathbf{v}}_i$ is our estimation.

6.1 Quantitative Evaluation

We first demonstrate the robustness of our approach in terms of noise level. The performance of our approach is also analyzed on viewpoint and database dependency. We then investigate the effect of different number of patches on the reconstruction results. Finally, we evaluate the effectiveness of our approach at each stage, and also compare its performance to 3D scanners and other state of the arts.

Resilience to noise. The test is performed to investigate the robustness of the algorithm to various levels of input noise.

TABLE 1
Reconstruction Accuracy on Real Data with Different Viewpoints

Viewpoint	-45°	-30°	-15°	0°	15°	30°	45°
mAP(mm)	0.1665	0.1651	0.1671	0.0	0.1672	0.1635	0.1658

The recovered results from the frontal view are considered as the ground truth.

In this experiment, we select 5*6 200-frames sequences of synthetic depths from six different expressions of five subjects. We simulate the noise on Kinect data using the same scheme as [27]. Each depth image is perturbed with random noise in the interval $[-\alpha, \alpha]$, where α ranges from 1.0 to 10.0 mm with 1.0 mm interval. Note that the noise level in depth images captured by the Kinect sensor is about 3.0 mm. The reconstruction errors are reported in Fig. 4 with different levels of noise. It can be seen that the reconstruction error of our method is always below 1.0 mm in all expressions around the level of noise in real data 3.0 mm. However, the average distance between two neighboring points in the point cloud of both synthetic and real depth data is about 1.5 mm. That is, the mean deviation for all points in the reconstruction result compared to their ground truth positions is below the resolution of the depth data. As the noise increases, our algorithm gets progressively less accurate, but it still works well with a high accuracy.

Viewpoint independency. To verify view independency, we capture Kinect data in the same condition except for the pose. The pose changes horizontally from -45° to $+45^\circ$ at 15 degree increments. 200 frames are used in each pose of each subject, totally 10 subjects. We consider the recovered results of the frontal view (0°) as the references. The reconstruction errors to the references measured by mAP in other poses are shown in Table 1, which shows that our algorithm is quite insensitive to pose variations and achieves almost the same level of accuracy in all poses.

Database dependency. In this test, we aim at quantitatively analyzing the relationship between our reconstruction result and the number of database samples using a child sequence (500 frames). Table 2 studies the effectiveness of our approach when varying the amount of training data used for the reconstruction. Originally our database contains around $N = 2{,}500$ samples. We sub-sample it with different ratios up to 100. We consider the recovered results of the full samples ($N = 2{,}500$) as the references. It can be seen that the reconstruction accuracy do not differ substantially even when the size of training database changes approximately exponentially. Difference between reconstruction errors using 100 samples and the full samples is about 0.5 mm which is quite small compared to the resolution of depth data. Overall our method is quite insensitive to size of the training database.

Number of patches. Table 3 shows how the number of patches used in local sparse coding affects the reconstruction accuracy. The test is performed over three 500-frame sequences with pose and expression variations

TABLE 2
Reconstruction Accuracy on Real Data with Different Sizes of the Training Database

Database size	25	100	250	500	750	1000	1250	1500	1750	2000	2250	2500
mAP(mm)	0.921	0.531	0.363	0.289	0.262	0.203	0.139	0.128	0.113	0.061	0.031	0.0

The recovered results of the full samples ($N = 2{,}500$) are considered as the ground truth.

TABLE 3
Reconstruction Accuracy on Real Data in Terms of Numbers of Patches with $N = 150$

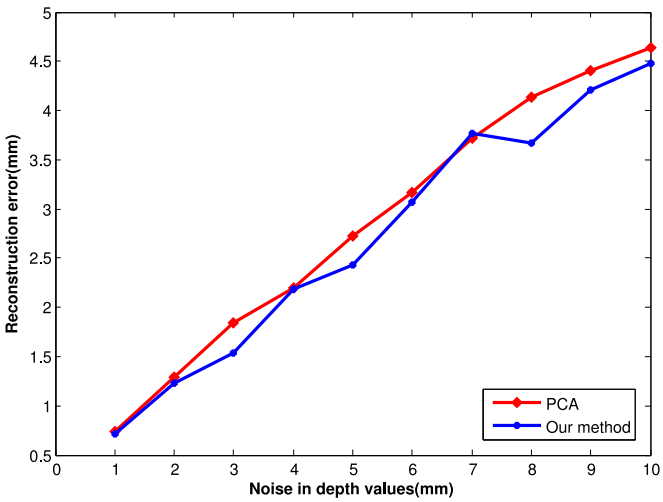
Patch number	1	2	4	6	9	11	17	20	25	28
mAP(mm)	0.863	0.643	0.227	0.196	0.165	0.0	0.204	0.271	0.315	0.323

The reconstruction results using $N_s = 11$ are considered as the ground truth.

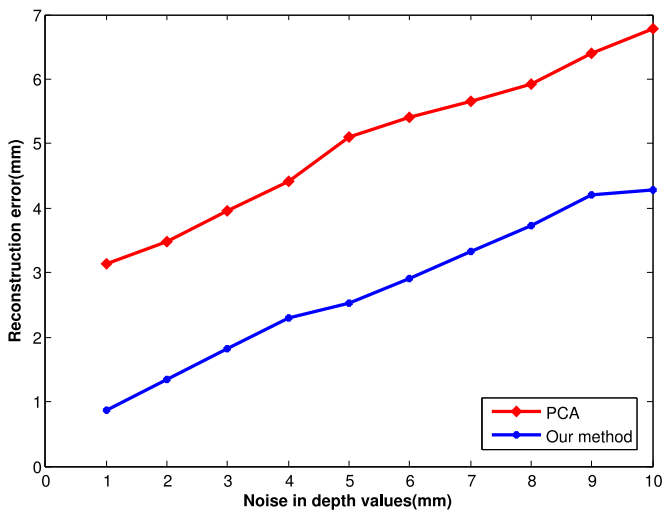
from three subjects. The reconstruction results using $N_s = 11$ are considered as the references. The results are listed in Table 3. We can see that the number of patches appears to have significant effect on the quality of reconstruction, as it directly impacts the capacity of removing the noise. Using fewer patches would conflict with linearity assumption of local regions and decrease the power of denoising in local sparse coding. On the other hand, the reconstruction error would also increase when too many patches are used. More patches mean more boundaries between patches, which would lead to lower accuracy of reconstruction in surface refinement.

Evaluation of local sparse coding. PCA has been used for face reconstruction in previous work [29]. In this test, we investigate the functionality of local sparse coding by replacing it with PCA in our approach. Each patch is applied PCA independently as local sparse coding. The test data is composed of 5*6 100-frame of synthetic depths from 6 different expressions of five subjects so that the surfaces in the test cover various deformations. Besides adding random noise, we also add some occlusion into synthetic depth images because occlusion often occurs in real data. The random noise setting is the same as the one in the experiment of resilience to noise. The occlusion is added into the nose patch of depth images and the occlusion fraction is about one fourth of the nose patch size. Therefore, the comparison is performed in two cases. One is random noise, and the other is random noise and occlusion. The reconstruction error is calculated on vertices of the nose patch in order to demonstrate the robustness of our approach. From results of PCA showed in Fig. 5, we see that PCA can remove the random noise in the raw depth patch. However, adding occlusion into the data has dramatic effects for PCA. PCA fails to fill the depth occlusion. This is because the missing parts due to corruption and occlusion, and the details in eyes, nose, etc., can only be recovered when the coefficient of each training sample is nonnegative. PCA represents the test sample using a linear combination of the principal components of the training set and the principal components are the linear combination of training samples, so some linear combination coefficients of the principal components may be negative. In particular, even under the case of random noise the error of our result is lower than that of PCA due to better details recovered. It is the advantage of our local sparse coding over PCA to fill occlusion and recover details in handling random noise and occlusion.

Evaluation of surface refinement. After random noise is removed and occlusion is corrected in each depth patch during the local sparse coding, there still exists noise on the boundaries between patches and some noise in the interior of patches. The second stage of our proposed algorithm, surface refinement, is to remove the noise and reconstruct the global shape by combining local patches. To demonstrate the effectiveness of surface refinement in our algorithm, we compare the error results with surface refinement and without surface refinement in this experiment. The test data in the experiment of evaluation of local sparse coding is used. The results are reported in Fig. 7. We see that there is a remarkable decrease in the reconstruction error after applying the surface refinement in our algorithm. This proves the second stage of our approach can refine the result after the local sparse coding by further reducing some noise on depth. The noise mainly arises from the boundaries between patches. Also, the inherited smoothness from surface refinement is able to filter the noise inside depth patches which the local sparse coding fails to remove thoroughly.



(a) Reconstruction error on the nose patch under the case of random noise in different levels.



(b) Reconstruction error on the nose patch under the case of occlusion + random noise in different levels.

Fig. 5. Evaluation of local sparse coding through comparison between local sparse coding + surface refinement and local PCA + surface refinement.

TABLE 4
Comparison between Our Local Sparse Coding and RMS under Case of Random Noise

Patch index	1	2	3	4	5	6	7	8	9	10	11
RMS	1.082	0.806	1.101	0.764	2.517	1.361	0.426	0.824	0.713	0.754	1.808
Local sparse coding	1.005	0.429	1.119	0.624	2.374	0.826	0.373	0.562	0.513	0.294	0.688

Reconstruction error(mm) is reported on the each patch under 5 mm random noise.

Accuracy relative to scanners. To test the accuracy of our approach using a Kinect sensor versus a scanner, we first use a high quality structured light illumination (SLI) [30] to capture reference scans of five subjects with four different expressions. The scanner produces 3D point clouds in real-time with techniques of a lookup-table (LUT) and a dual-frequency pattern that combines a high-frequency sinusoid component with a unit-frequency sinusoid component. The technical details can be found in [30]. We also recover face models from Kinect data captured in the same poses of the same subjects using our algorithm. The recovered models are then fitted to the scans using ICP for accuracy measurement, and the differences between them measured by mAP are 8.29 mm (maximum), 1.59 mm (mean), 1.37 mm (median), 1.46 mm (average deviation). It can be seen that our algorithm can reconstruct the face model with high accuracy.

Comparison with RMS [31]. Root mean square (RMS) optimization aims to minimize the ℓ_2 norms between an input sample and linear representation of the input sample using training samples. In this experiment, we compare the performance of our local sparse coding with RMS. The test data is the same as that used in evaluation of local sparse coding. Comparison between RMS optimization and our local sparse coding is reported in terms of reconstruction error in each local patch. Without constraints of sparse and nonnegative coefficients, the linear representation of RMS has a larger space, and it can represent the input as close as possible. Therefore, RMS is very sensitive to noise because it treats the noise as input details. Furthermore, RMS cannot efficiently handle occlusion. From the results showed in Tables 4 and 5, we can see that RMS performs similar to PCA. Under the random noise case, our local sparse coding performs better than RMS optimization on noise handling. Under case of occlusion and random noise, the error in the nose part is larger using RMS optimization since RMS optimization fails to correct the depth occlusion.

Comparison with [4] and [28]. We compare our algorithm with two template-based methods: F-PSR [4] and N-ICP-A [28]. Both reconstruct the face model by fitting a template model to the input scan formulated as a cost function with three linear items. The first term, feature distance term, indicates that the 68 key points of the template model are equal to those of the face scan. The second term is surface distance term that fits the correspondences between the template model and the face scan. The correspondences are established through the closest point method. The purpose of the third term, smoothness term, is to enforce smoothness of surface deformation without a drastic change. F-PSR achieves smoothness by minimizing the change of the distance between neighboring vertices, while N-ICP-A ensures smoothness through minimizing the difference between the transformations of the neighboring vertices.

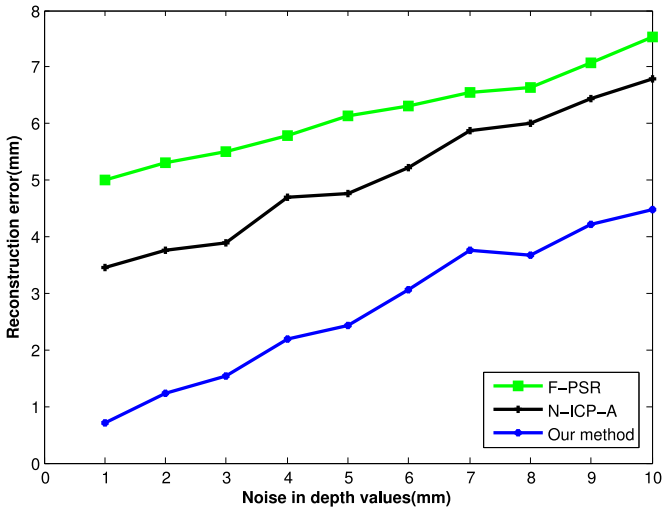
The template model is initially aligned to the input depth using the rigid registration of ICP [32] for two template-based methods. During each iteration, the correspondences are found through the approximate nearest neighbor searching [33]. The template is then deformed toward the target by minimizing the cost function. A new set of correspondences are searched based on the new deformed positions and are used in the next iteration. This deformation process is the same for two methods and is repeated until a stable state is found. The norm of the difference between the parameter vectors is computed from two successive iterations. The condition of convergence is that the average value over all vertices of the norm is smaller than a threshold. F-PSR aims to optimize each vertex location, while N-ICP-A refines the transformation for each vertex. Through parameter tuning with different settings for F-PSR and N-ICP-A respectively, we got the best reconstruction results on our data set with the following particular parameters settings for them in our experiment. The convergence threshold of F-PSR is $1e^{-3}$, and that of N-ICP-A is $1e^{-4}$. The feature weight α, distance weight β, and smoothness weight γ of F-PSR is set to 0.5, 0.5, 1.0, respectively. In N-ICP-A, the weight matrix of distance term W is set to identity matrix I, the landmark weight β is set to 1, and the stiffness weight α descends from 5 to 1 with 1 interval.

The experiment is performed over the same test data as the one in the evaluation of local sparse coding. From the result shown in Fig. 6, we can see that there is a remarkable improvement of our method over F-PSR and N-ICP-A respectively in reconstruction accuracy and smoothness. This can be explained by the fact that the input depths contain remarkable noise, corruption, and occlusion, which results in the increase of the reconstruction error. More precisely, noise is introduced into the vertices of the input depth, which brings the positional error of the feature points, and in turn leads to the increase of the registration error in the feature distance term. In addition, remarkable noise poses a great challenge to establish reliable correspondences between the template model and the depth scan. Wrong

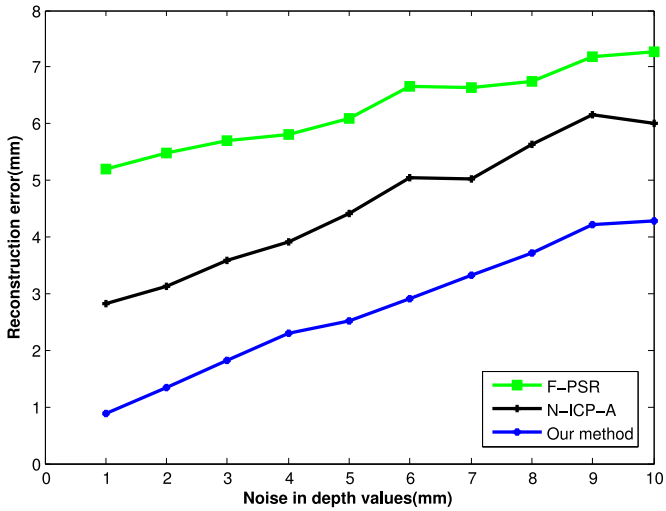
TABLE 5
Comparison between Our Local Sparse Coding and RMS under Case of Occlusion + Random Noise

Patch index	1	2	3	4	5	6	7	8	9	10	11
RMS	0.987	0.730	1.040	0.723	4.637	1.227	0.379	0.828	0.577	0.671	1.550
Local sparse coding	1.022	0.453	1.052	0.523	2.470	0.831	0.385	0.558	0.375	0.300	0.653

Reconstruction error(mm) is reported on the each patch under 5 mm random noise + occlusion on the nose patch. The nose patch (patch 5) has a large error using RMS minimization due to occlusion added on the nose patch.



(a) Reconstruction error on the nose patch under the case of random noise in different levels.



(b) Reconstruction error on the nose patch under the case of occlusion + random noise in different levels.

Fig. 6. Comparison results of our approach to two template-based methods, F-PSR [4] and N-ICP-A [28].

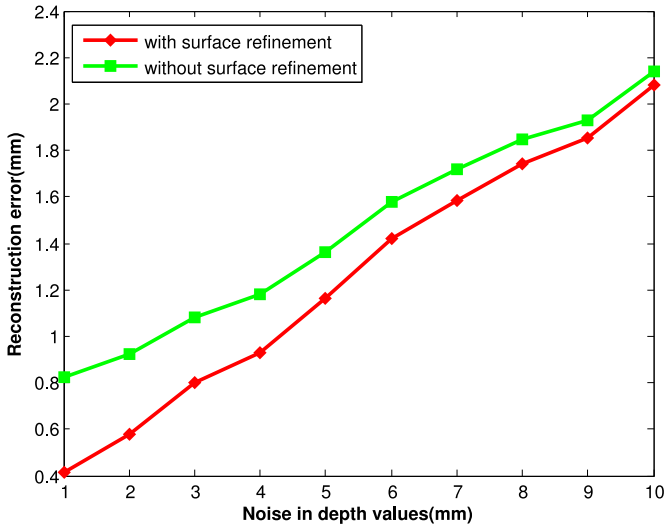


Fig. 7. Reconstruction errors with and without surface refinement (mAP) in different levels of random noise.

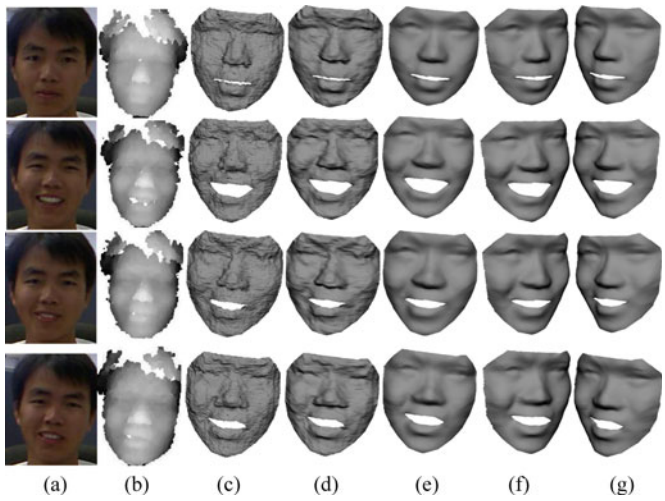


Fig. 8. Reconstructed examples of F-PSR [4], N-ICP-A [28], and our approach. (a) Input color images. (b) Input depth images. (c) Reconstructed results of F-PSR. (d) Reconstructed results of N-ICP-A. (e), (f), (g) Reconstructed results of our approach in different views.

correspondences may increase the registration error in the surface distance term. Some reconstructed examples are showed in Fig. 8. It can be seen that the smoothness term of F-PSR fails to make the surface smooth during deformation, because both registration errors from noise break the smoothness constraint and lead to rough surfaces in the results. N-ICP-A performs better than F-PSR in recovery accuracy and smoothness due to enforcement of the stronger constraint on smoothness. However, both registration errors still introduce the distortion in the reconstruction results. On the contrary, through patch division and local sparse coding, we remove the strong random noise, correct the corruption in the depth images, and also establish the reliable correspondences, which significantly reduce the possibility of the registration error at the stage of the surface refinement.

Timing results. Our algorithm mainly consists of three steps: data preprocessing, local sparse coding and surface refinement. We measure the mean computational time needed for each step in the experiment above. The time results in second are showed in Table 6. The data preprocessing stage includes the AAM fitting, segmentation of face region and the raw data preprocessing. Since several patches need handle in the local sparse coding stage, the consumed time is longer than other steps. Without any parallel computations, our current implementation in Matlab does make our approach far from being real-time. The main bottleneck is the optimization in Eqs. (5) and (11). However, some parts of our algorithm, such as local sparse coding on each local patch, are well fitted for parallelization. We believe that a GPU implementation of our method would drastically

TABLE 6
Timing Results of Our Approach in Three Steps, Respectively

Step	Time consumed(s)
Data preprocessing	4.83
Local Sparse coding	50.74
Surface refinement	13.55

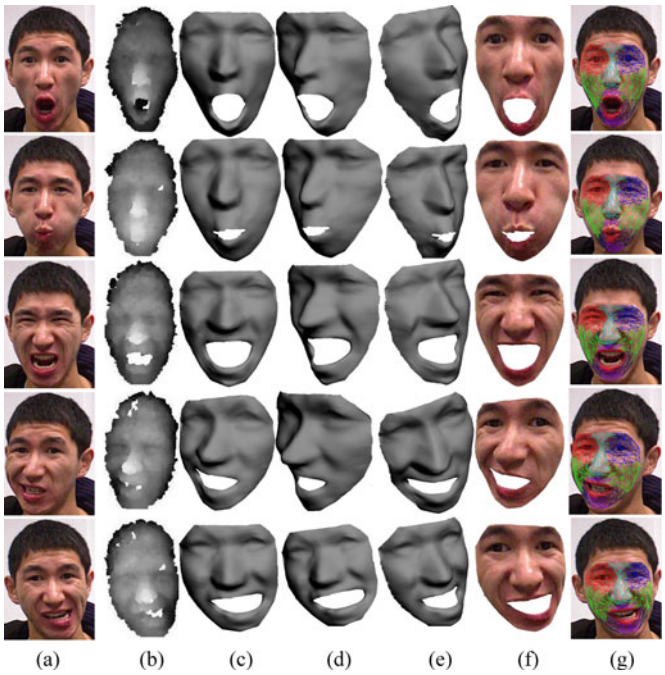


Fig. 9. Face reconstruction results from inputs with large variations in expression using our approach. (a) Input color images. (b) Input depth images. (c), (d), (e) Different views of the reconstructed 3D models. (f) The textured faces. (g) 2D projections on (a).

reduce the computational time. As an extension of this work, implementation on GPU which would offer at least near real-time performance is within our plan.

6.2 Qualitative Evaluation

We evaluate the performance of our approach on real inputs from Kinect. The results in the accompanying video demonstrate that our method can correctly recover 3D faces despite

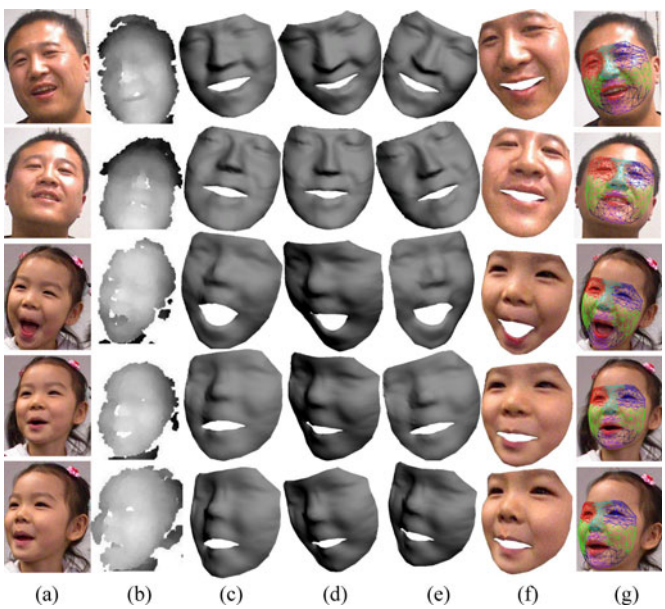


Fig. 10. Face reconstruction results from relatively large variations in pose using our approach. (a) Input color images. (b) Input depth images. (c), (d) Different views of the reconstructed 3D models. (e) The textured faces. (f) 2D projections on (a).

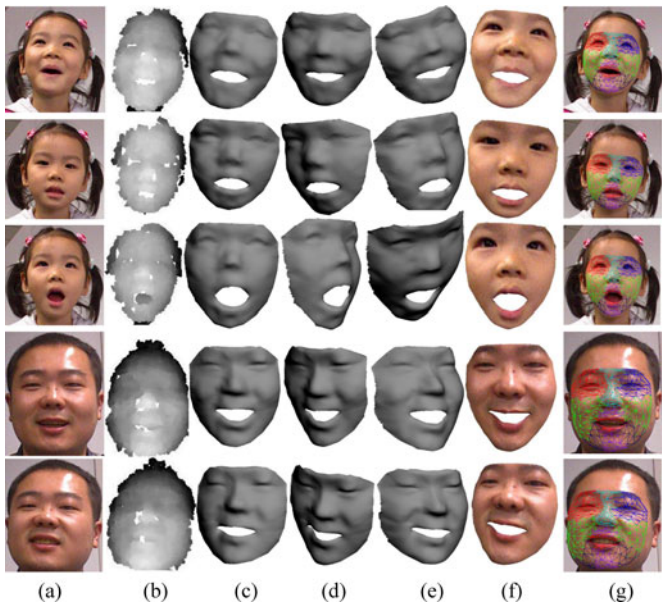


Fig. 11. Face reconstruction results from depth maps with considerable corruption using our approach. (a) Input color images. (b) Input depth images. (c), (d), (e) Different views of the reconstructed 3D models. (f) The textured faces. (g) 2D projections on (a).

of large variations from pose, expression, corruption, and subject. It can be seen from Fig. 9 that our approach can recover complex expressions, which is significant as our template database from BU-4DFE [22] has no similar expression, or even close to the displayed expression. Fig. 10 shows the robustness of the proposed algorithm in non-frontal views. Both demonstrate that our method can handle pose and expression variations. That is very important, because the template database can only contain frontal views and the limited number of models with both variations.

Our approach can deal with challenging scenes with corruption as demonstrated in the results of Fig. 11. Figs. 10 and 11 also show that our method has the capability of handling face variations across subjects. Moreover, all the examples of the paper and the accompanying video demonstrate that our approach can capture the facial details like eyes' shapes and

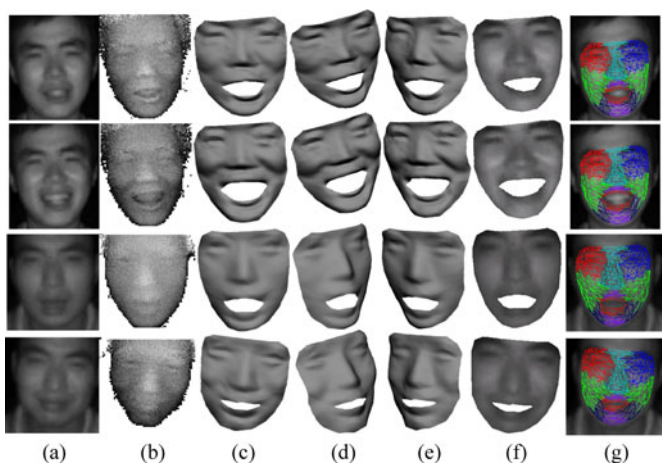


Fig. 12. Reconstructed examples of TOF data using our approach. (a) Input color images. (b) Input depth images. (c), (d), (e) Different views of the reconstructed 3D models. (f) The textured faces. (g) 2D projections on (a).

the underlying dynamics of 3D shapes such as facial expressions. Note that all these are achieved automatically without any user interaction and careful placement of makers.

Our approach can be generalized to handle noisy data from other depth sensors. We test our method on the data from another type of Time-Of-Flight sensor, SwissRanger SR-4000. The captured gray images have low resolution, 176×144, which is about one fourth of that from Kinect. We increase the resolution of the data by three times through simple linear interpolation. Some reconstructed examples are showed in Fig. 12. Although data from TOF have much lower resolution and are noisier than that from Kinect, our method is able to recover impressive 3D models.

7 Conclusion

In this paper, we have proposed a novel approach to reconstruct 3D faces from a depth sensor. Our approach can generate face deformations with the personalized details. The key insight is to use local sparse coding to locally remove the random noise and correct the corruption by taking advantage of template priors. Moreover, our approach globally refines the full face model by model fitting with the proposed surface refinement. The experimental results demonstrated that our method is quite insensitive to database and viewpoint dependency, generalizes well with a wide variety of variations from subject, pose, and expression, and has the ability of handling noise and corruption.

The main limitation of our approach is that it is unable to reconstruct subtle face details of the subject such as wrinkles. This is due to noise suppression operations in local sparse coding in depth and the inherited smoothness from surface refinement. In addition, the low resolution of a depth sensor can lead to the sparse correspondences between the templates and the input, which further aggravates this limitation. In the near future, we will improve the face details using higher resolution of color images or benefit from a large volume of samples.

AAM is employed for detecting the facial features in our algorithm. However, when the face pose is large, the AAM fitting fails. Fortunately, the most recent work [34] presents a robust approach for facial feature location under largely varying pose which takes advantage of intensity and depth information simultaneously. We think this method is promising and can provide us more stable and accurate feature points even under challenging poses.

Acknowledgments

This research was supported by NSFC No. 61332017 and No. 61170318.

References

[1] L. Zhang, N. Snavely, B. Curless, and S.M. Seitz, "Spacetime Faces: High-Resolution Capture for Modeling and Animation," *Proc. ACM SIGGRAPH*, pp. 548-558, Aug. 2004.

[2] T. Beeler, B. Bickel, P. Beardsley, B. Sumner, and M. Gross, "High-Quality Single-Shot Capture of Facial Geometry," *ACM Trans. Graphics*, vol. 29, no. 3, article 40, 2010.

[3] D. Bradley, W. Heidrich, T. Popa, and A. Sheffer, "High Resolution Passive Facial Performance Capture," *ACM Trans. Graphics*, vol. 29, no. 3, article 41, 2010.

[4] Y.S. Kim, H. Lim, B. Kang, O. Choi, K. Lee, J. Kim, and C.-Y. Kim, "Realistic 3D Face Modeling Using Feature-Preserving Surface Registration," *Proc. IEEE Int'l Conf. Image Processing (ICIP)*, pp. 1821-1824, Sept. 2010.

[5] M. Perriollat, R.I. Hartley, and A. Bartoli, "Monocular Template-Based Reconstruction of Inextensible Surfaces," *Int'l J. Computer Vision*, vol. 95, no. 2, pp. 124-137, 2011.

[6] M. Zollhöfer, M. Martinek, G. Greiner, M. Stamminger, and J. Süßmuth, "Automatic Reconstruction of Personalized Avatars from 3D Face Scans," *Computer Animation Virtual Worlds*, vol. 22, pp. 195-202, Apr. 2011.

[7] V. Blanz, K. Scherbaum, and H.-P. Seidel, "Fitting a Morphable Model to 3D Scans of Faces," *Proc. IEEE 11th Int'l Conf. Computer Vision (ICCV)*, pp. 1-8, Oct. 2007.

[8] A. Weiss, D. Hirshberg, and M.J. Black, "Home 3D Body Scans from Noisy Image and Range Data," *Proc. IEEE 11th Int'l Conf. Computer Vision (ICCV)*, Nov. 2010.

[9] T. Weise, S. Bouaziz, H. Li, and M. Pauly, "Realtime Performance-Based Facial Animation," *ACM Trans. Graphics*, vol. 30, no. 4, article 77, July 2011.

[10] T.F. Cootes, G.J. Edwards, and C.J. Taylor, "Active Appearance Models," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 23, no. 6, pp. 681-685, June 2001.

[11] X. Mei and H. Ling, "Robust Visual Tracking Using L1 Minimization," *Proc. IEEE 12th Int'l Conf. Computer Vision (ICCV)*, pp. 1436-1443, 2009.

[12] J. Zhu, S.C. Hoi, Z. Xu, and M.R. Lyu, "An Effective Approach to 3D Deformable Surface Tracking," *Proc. 10th European Conf. Computer Vision (ECCV)*, pp. 766-779, 2008.

[13] A. Ecker, A.D. Jepson, and K.N. Kutulakos, "Semidefinite Programming Heuristics for Surface Reconstruction Ambiguities," *Proc. European Conf. Computer Vision (ECCV)*, pp. 127-140, 2008.

[14] M. Salzmann and P. Fua, "Reconstructing Sharply Folding Surfaces: A Convex Formulation," *Proc. IEEE Conf. Computer Vision and Pattern Recognition (CVPR)*, pp. 1054-1061, 2009.

[15] M. Brand, "Morphable 3D Models from Video," *Proc. Conf. Computer Vision and Pattern Recognition (CVPR)*, pp. 456-463, June 2001.

[16] L. Torresani, A. Hertzmann, and C. Bregler, "Learning Non-Rigid 3D Shape from 2D Motion," *Proc. Advances in Neural Information Processing Systems (NIPS)*, 2003.

[17] T. Weise, H. Li, L. Van Gool, and M. Pauly, "Face/Off: Live Facial Puppetry," *Proc. ACM SIGGRAPH/Eurographics Symp. Computer Animation*, pp. 7-16, Aug. 2009.

[18] J. Xiao and T. Kanade, "Uncalibrated Perspective Reconstruction of Deformable Structures," *Proc. 10th IEEE Int'l Conf. Computer Vision (ICCV)*, pp. 1075-1082, Oct. 2005.

[19] P. Breuer, K.-I. Kim, W. Kienzle, B. Schölkopf, and V. Blanz, "Automatic 3D Face Reconstruction from Single Images or Video," *Proc. Int'l. Conf. Automatic Face Gesture Recognition*, pp. 1-8, Sept. 2008.

[20] V. Blanz and T. Vetter, "A Morphable Model for the Synthesis of 3D Faces," *Proc. ACM SIGGRAPH*, pp. 187-194, 1999.

[21] M. Salzmann, V. Lepetit, and P. Fua, "Deformable Surface Tracking Ambiguities," *Proc. IEEE Conf. Computer Vision and Pattern Recognition (CVPR)*, pp. 1-8, June 2007.

[22] L. Yin, X. Chen, Y. Sun, T. Worm, and M. Reale, "A High-Resolution 3D Dynamic Facial Expression Database," *Proc. Int'l Conf. Automatic Face Gesture Recognition*, pp. 1-6, Sept. 2008.

[23] H. Wang, L. Dong, J. O'Daniel, R. Mohan, A.S. Garden, K.K. Ang, D.A. Kuban, M. Bonnen, J.Y. Chang, and R. Cheung, "Validation of an Accelerated 'Demons' Algorithm for Deformable Image Registration in Radiation Therapy," *Physics in Medicine and Biology*, vol. 50, pp. 2887-2905, 2005.

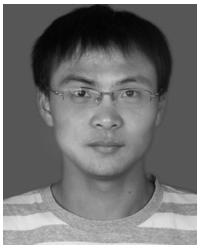
[24] D.G. Kendall, "A Survey of the Statistical Theory of Shape," *Statistical Science*, vol. 4, no. 2, pp. 87-99, May 1989.

[25] S.J. Kim, K. Koh, M. Lustig, S. Boyd, and D. Gorinevsky, "An Interior-Point Method for Large-Scale L1-Regularized Logistic Regression," *J. Machine Learning Research*, vol. 8, no. 3, pp. 1519-1555, 2007.

[26] R.W. Sumner and J. Popović, "Deformation Transfer for Triangle Meshes," *Proc. ACM SIGGRAPH*, pp. 399-405, 2004.

[27] D. Thomas, Y. Matsushita, and A. Sugimoto, "Robust Simultaneous 3D Registration via Rank Minimization," *Proc. Second Int'l Conf. 3D Imaging, Modeling, Processing, Visualization and Transmission (3DIMPVT)*, pp. 33-40, 2012.

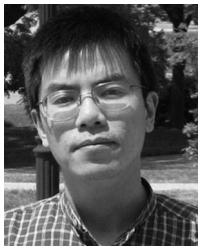
[28] B. Amberg, S. Romdhani, and T. Vetter, "Optimal Step Nonrigid ICP Algorithm for Surface Registration," *Proc. IEEE Conf. Computer Vision and Pattern Recognition (CVPR)*, pp. 1-8, June 2007.
[29] F. De la Torre, J.R. Tena, and I. Matthews, "Interactive Region-Based Linear 3D Face Models," *Proc. ACM SIGGRAPH*, July 2011.
[30] K. Liu, Y. Wang, D.L. Lau, Q. Hao, and L.G. Hassebrook, "Dual-Frequency Pattern Scheme for High-Speed 3-D Shape Measurement," *Optics Express*, vol. 18, no. 5, pp. 5229-5244, 2010.
[31] J. Wright, A. Yang, A. Ganesh, S. Sastry, and Y. Ma, "Robust Face Recognition via Sparse Representation," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 31, no. 2, pp. 210-227, Feb. 2009.
[32] N.D.M. Paul and J. Besl, "A Method for Registration of 3-D Shapes," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 14, no. 2, pp. 239-256, Feb. 1992.
[33] D.M. Mount, and S. Arya, "ANN: A Library for Approximate Nearest Neighbor Searching," http://www.cs.umd.edu/mount/ANN/, 2014.
[34] T. Baltrusaitis, P. Robinson, and L.-P. Morency, "3D Constrained Local Model for Rigid and Non-Rigid Facial Tracking," *Proc. IEEE Conf. Computer Vision and Pattern Recognition (CVPR)*, June 2012.



Kangkan Wang received the BS degree in computer science from Northwestern Polytechnical University, China, in 2009. He is currently working toward the PhD degree in computer science at the State Key Laboratory of CAD&CG, Zhejiang University, China. His research interests include face modeling, object tracking, and 3D reconstruction.

Xianwang Wang received the BE and ME degrees from Sichuan University, China, in 1998 and 2001, respectively, and the PhD degree from the University of Kentucky in 2010, all in computer science. He was a postdoctoral researcher at Hewlett-Packard (HP) Labs from August 2010 to May 2012. He is currently a research scientist of HP Company in Palo Alto, California. His main research interests include face recognition, object detection, tracking, and recognition, large-scale image retrieval, motion modeling, and 3D reconstruction.

Zhigeng Pan received the bachelor's and master's degrees from the Computer Science Department at Nanjing University in 1987 and 1990, respectively, and the PhD degree in 1993 from Zhejiang University. He is the director of Digital Media and HCI Research Center at Hangzhou Normal University, and his research interests include HCI virtual reality, multimedia, and digital entertainment. He is the editor-in-chief of *Transactions on Edutainment*.



Kai Liu received the BS and MS degrees in computer science from Sichuan University, China, in 1996 and 2001, and the PhD degree in electrical engineering from the University of Kentucky in 2010, respectively. He is currently a professor in the School of Electrical Engineering and Information at Sichuan University, China. He had been a postdoctoral researcher in the Information Access Lab at the University of Delaware from September 2010 to July 2011. His main research interests include computer/machine vision, active/passive stereo vision and image processing. His works have been featured in *Optics Express*, *Journal of the Optical Society of America A*, *Optics Letter*, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *IEEE Transactions on Image Processing*, and *SPIE*. He is a senior member of IEEE.

▷ **For more information on this or any other computing topic, please visit our Digital Library at** www.computer.org/publications/dlib.