# 3D Human Model Reconstruction from Sparse Uncalibrated Views

Xiaoguang Han, Kwan-Yee K. Wong, Yizhou Yu

***Abstract*—This paper presents a novel two-stage algorithm for reconstructing 3D human models wearing regular clothes from sparse uncalibrated views. The first stage reconstructs a coarse model with the help of a template model for human figures. A non-rigid dense correspondence algorithm is applied to generate denser correspondences than traditional feature descriptors. We fit the template model to the point cloud reconstructed from dense correspondences while enclosing it with the visual hull. In the second stage, the coarse model from the first stage is refined with geometric details, such as wrinkles, reconstructed from shading information. To successfully extract shading information for a surface with nonuniform reflectance, a hierarchical density based clustering algorithm is adapted to obtain high-quality pixel clusters. Geometric details reconstructed using our new shading extraction method exhibit superior quality. Our algorithm has been validated with images from an existing dataset and images captured by a cell phone camera.**

***Index Terms*—3D human reconstruction, multi-view stereo, shape from shading**

## I. Introduction

With the recent advances and growing popularity in 3D data capturing techniques, it has now become more convenient to acquire realistic 3D models of human figures. Such 3D models can be used to produce 3D self-portraits using 3D printers, or used as custom-designed avatars in computer games and virtual reality.

Common techniques for acquiring 3D human models can be categorized into those based on depth cameras (e.g., Microsoft's Kinect) [1], [2] and those based on regular cameras [3]. Depth camera based methods often require complicated capturing steps or produce low-quality results. For example, Tong et al. [1] used three Kinect cameras to capture different parts of a person standing still on a rotating turntable. A major limitation of such methods is that depth cameras can only be used for short-range indoor scanning. Methods based on regular cameras commonly rely on multi-view stereo (MVS) algorithms. Most of these methods, such as the system in [3], require fully-calibrated cameras. With uncalibrated input images, structure-from-motion (SFM) [4] can be applied to estimate camera parameters from feature correspondences, and the resulting calibrated images can then be fed to MVS systems, such as PMVS [5]. Such an approach usually requires a large number of images with a sufficient number of feature correspondences to ensure the successes in camera calibration and dense reconstruction. For instance, Autodesk 123D catch[1] requires around 40 images for producing a human figure, and the images must have textured areas for feature point extraction.

This paper aims at developing a technique for reconstructing high quality 3D models of human wearing regular clothes from sparse uncalibrated cameras. This is a very challenge problem for the following reasons. First, traditional feature descriptors (e.g., Harris, DOG or SIFT) are incapable of finding a sufficient number of feature correspondences from a sparse image sequence for dense point cloud reconstruction. Second, camera parameters estimated using SFM are usually inaccurate and the generated point cloud are usually noisy and incomplete. Third, it is a common practice to enhance a coarse geometry resulting from MVS using shading information. However, existing intrinsic image decomposition algorithms cannot handle the intricate patterns often appear on the clothes, and fail to extract accurate shading information necessary for detailed geometric reconstruction.

In this paper, we propose a two-stage algorithm to tackle the aforementioned problems. In first stage of our algorithm, we first apply the non-rigid dense correspondences (NRDC) algorithm [6] to generate dense correspondences for camera calibration and point cloud reconstruction. Although NRDC works reasonably well in featureless regions, the reconstructed point cloud is still noisy with missing regions and the estimated camera parameters are of low accuracy which causes an imprecise visual hull. A watertight template model for human figures is used as a deformable prior to fit the incomplete point cloud as well as the inaccurate visual hull. This produces a coarse model of the human figure. In the second stage, we refine this coarse model with geometric details, such as wrinkles, reconstructed from shading information. To successfully extract shading information from a surface with a nonuniform reflectance pattern (e.g., clothes), we adopt a hierarchical density based clustering algorithm to produce high-quality pixel clusters with uniform reflectances. Geometric details reconstructed from shading information estimated using our new method exhibit superior quality.

We have validated our proposed algorithm using dataset from [3] as well as images captured by our own cameras, and compared our results against those of the previous methods. Note that our algorithm only requires a dozen of images to reconstruct a 3D human model.

In summary, our major contributions are as follows.

Xiaoguang Han, Kwan-Yee K. Wong and Yizhou Yu are with Department of Computer Science, The University of Hong Kong, Hong Kong, Email: xghan@cs.hku.hk, kykwong@cs.hku.hk, yzyu@cs.hku.hk

[1] Autodesk 123D catch http://www.123dapp.com/catch

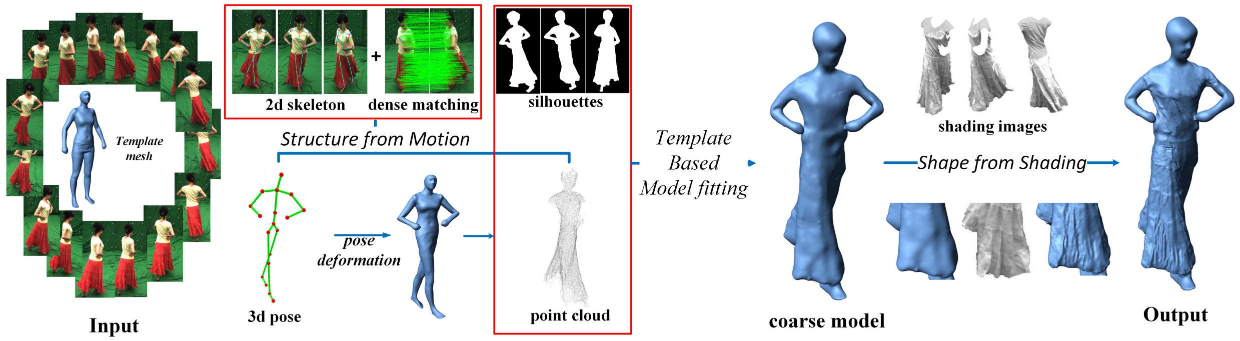


Fig. 1: The flowchart of our system.

- A novel two-stage pipeline is proposed for reconstructing 3D models of human wearing regular clothes from sparse uncalibrated views.
- In the first stage (coarse reconstruction), we exploit a non-rigid dense correspondence algorithm for generating dense correspondences from a sparse image sequence for camera calibration and point cloud reconstruction. We also develop an algorithm to fit a deformable human model to both the incomplete and noisy point cloud, as well as the silhouettes in images to produce a coarse model.
- In the second stage (refinement), the coarse model is refined using shading information. A novel shading extraction algorithm is proposed which can handle surfaces with nonuniform reflectance patterns. This algorithm is built on a hierarchical density based clustering algorithm to produce high-quality pixel clusters with uniform reflectances.

## II. Related work

Researchers pay less attention to dense multi-view stereo from uncalibrated images in recent years. A quasi-dense reconstruction approach from uncalibrated images was proposed in [7], which generated more robust and accurate geometry estimation and required fewer images than sparse methods for modeling. Wu et al. [8] also presented a quasi-dense approach which exploited 3D tensors to provide a unified approach for the implementation of a match-propagate-filter pipeline. Both of these methods required initial confident correspondences. Our proposed algorithm based on NRDC requires fewer images than these methods, as NRDC does not require an initial sparse matching.

Our work is closely related to the topic of human model reconstruction from images. In [9], [10], the authors attempted to reconstruct human body from a single image. They constrained the 3D model by a parametric human body space. Another related area is template based marker-less performance capture. Most of the recent techniques (e.g. [11], [12], [13]) in this area used calibrated multi-view video and a well-reconstructed mesh with the same character wearing the same garment as input. The input mesh was then deformed to fit the silhouettes or the point cloud resulting from MVS in each frame. Different from these methods, our method begins with a template body mesh and aims at generating a detailed model of human with regular garments.

Geometry refinement by shape-from-shading (SfS) under general illumination is another area closely related to our paper. In [14], [15], the authors used a non-linear optimization solver to deform rough 3D models to match the shadings in images. Their initial 3D models were obtained using MVS or performance capture. Similar optimization approaches were used in [16], [17] to refine coarse depth data from Kinects. However, most of these methods were designed to work with surfaces with constant albedo. With the help of coarse geometry, shading extraction can be performed by a clustering-optimization strategy [15], [17], [16]. Based on color information, the input image was firstly segmented into different regions of approximately constant albedo. The albedos for each segment and the global lighting model were then solved as an optimization problem. Wu et al. [15] used graph based image segmentation for clustering and formulated a MAP problem for optimization. Mean shift was used in [17] for clustering, and the global lighting model and relative albedos were optimized in an alternating manner. A simple method is proposed in [16] which performed k-means for clustering and used the dominant cluster for global lighting estimation. The albedos of other groups were then determined by the estimated global lighting model. The output quality of these methods depended on the clustering results. However, all the previous image clustering methods usually produced inaccurate results especially when the shading varying too much. To address this issue, a novel density based pixel clustering algorithm is proposed in this paper.

## III. System Overview

The workflow of our system is shown in Fig 1. Our system takes as input $n$ images captured around a real person. These images are indexed using a circular order. Our approach begins with the silhouettes of the human in the images and labeled 2D skeletons in a subset of the images. We first apply the NRDC algorithm in [6] to obtain dense correspondences across the images, which are combined with the joint locations in the 2D skeletons as the input to SFM for simultaneous point cloud reconstruction and camera calibration.

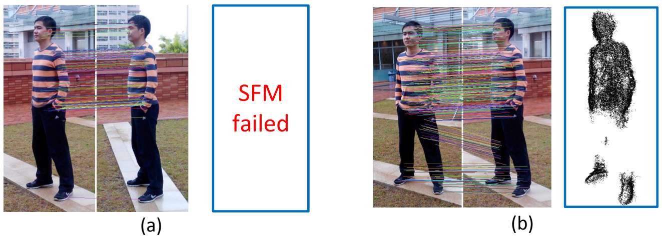


Fig. 2: Comparison of point correspondences obtained using ASIFT and NRDC. (a) ASIFT feature matching result (all correspondences are shown). SFM fails on such sparse correspondences. (b) NRDC matching result (only $1\%$ correspondences are shown). SFM can successfully reconstruct a dense point cloud from such dense correspondences.

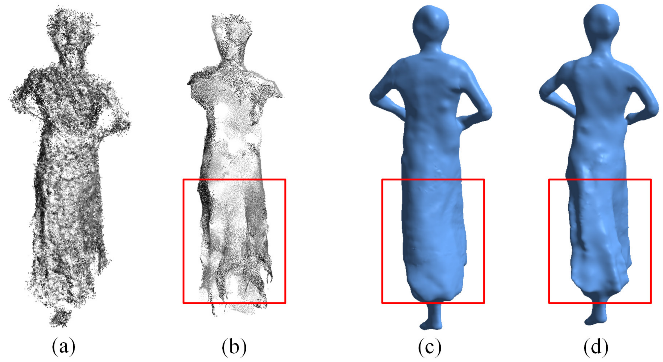


Fig. 3: (a) Original point cloud reconstructed by SFM from NRDC matching result. (b) Updated point cloud after filtering and resampling. (c) Coarse model generated using the visual hull only. (d) Coarse model generated using both the point cloud and the visual hull.

Given the roughly calibrated cameras and a point cloud, our system computes a watertight human model from a template model in two stages. In our system, the template model is used as a deformable prior. We fit this template to the point cloud as well as various cues in the images, including silhouettes and shading. In the first stage, the template model is deformed to fit the reconstructed point cloud while the contours of its projections are made as close as possible to the silhouettes in the images. In the second stage, geometric details are recovered using shape from shading and added onto the coarse mesh obtained in the first stage. In this part, a novel shading extraction method is proposed through an adapted density-based clustering algorithm.

## IV. Coarse Model Generation

Suppose the input images are $\{I_k\}$, the silhouettes in the images are $\{S_k\}$, and the template mesh is $T$. The template model is equipped with a skeleton. We use the same skeleton representation as [10]. Our system provides a simple drag-and-drop user interface as [18] for skeletons labeling in a subset (3 images are used in our experiments) of the input images.

### A. Point Cloud Reconstruction

To obtain correspondences between images, all of previous sparse feature matching methods failed for our sparse and textureless input images. This is shown in Fig 2, where Affine SIFT matching, one of the best sparse matching methods to date, are compared with NRDC matching. The dense point cloud reconstructed by SFM is shown as dark points.

One simple way to prepare correspondences for SFM is applying NRDC to every pair of images. However, this results in few correspondences when two images only have a small overlap, and it is also time consuming. We use a *match-propagate-filter* pipeline to obtain dense correspondences for SFM.

***Match*** The NRDC algorithm [6] is first applied to every pair of adjacent images $I_i$ and $I_{i+1}$. Their dense correspondences are noted as $C_{i,i+1} : R^2 \rightarrow R^2$. For each pixel $p \in I_i$, $C_{i,i+1}(p)$ returns the corresponding pixel $q \in I_{i+1}$ if the confidence of this correspondence is over a threshold (0.5 in our experiments); otherwise it returns $null$. Meanwhile, we also compute $C_{i,i-1}$ from $I_i$ to $I_{i-1}$ for every image $I_i$.

***Propagate*** We further compute $C_{i,i+2}...C_{i,i+k}$ and $C_{i,i-2},...,C_{i,i-k}$ through propagation to obtain correspondences between more pairs of views. Such propagation is made possible by the higher density of the correspondences obtained from the NRDC algorithm. Here, $C_{i,i+k} = C_{i,i+1} \circ C_{i+1,i+2} \circ \ldots \circ C_{i+k-1,i+k}$ where $\circ$ is a compound operator. And $C_{i,i-k}$ is defined similarly.

Once the correspondences have been computed and propagated, they are combined with the joint correspondences from the labeled 2D skeletons, and used as the input to SFM. We use the bundle adjustment algorithm in [19] as the SFM solver. The output includes the camera projection matrices $\{Proj_k\}$ for all views, a 3D skeleton $S_{3d}$ and a rough point cloud $\widetilde{P}$.

***Filter*** Because of missing and inaccurate correspondences, the point cloud from SFM is noisy and incomplete. We clean it as follows. First, we enforce spatial consistency of pixel correspondences. Considering two adjacent images $I_i$ and $I_{i+1}$. Suppose $p'_1, p'_2 \in I_{i+1}$ are the corresponding pixels of two adjacent pixels $p_1, p_2 \in I_i$. We remove the two related points in $\widetilde{P}$ if the Euclidean distance $dist(p'_1, p'_2)$ is greater than a threshold (5 pixels is setting in our experiments). Then we apply the density constraint, PCA eigenvalue constraint and normal constraint in [3] for further filtering. After this two-steps cleaning, we run SFM and filtering again. The algorithm in [20] is applied to the clean point cloud afterwards. Noted that, in addition to cleaning, another objective of using edge-aware resampling here is preserving major cloth wrinkles in the point cloud (as shown by the highlight region in Fig. 3).

The final point cloud is denoted as $P$. Fig 3 shows point cloud before and after filtering and resampling. It also illustrates the importance of this point cloud during the coarse model generation.

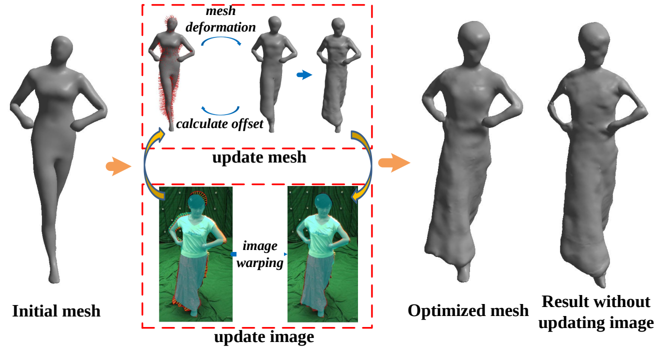


Fig. 4: The flowchart of our template based model fitting.

### *B. Template-Based Model Fitting*

The template mesh is viewed as a deformable model $M$ which aims to fit the given silhouettes $\{S_k\}$ and point cloud $P$. This can be formulated as an optimization problem. The energy is defined as

$$E = E_{dist}(M, \{S, P\}) + E_{smooth}(M). \tag{1}$$

$E_{dist}$ means the distance between $M$ and the point cloud $P$ and the visual hull defined by the silhouettes $\{S_k\}$. $E_{smooth}$ here is to ensure the smoothness of $M$ during its deforming procedure. Our optimization is performed in an iterative manner as follows:

***Initial value*** We first make the embedded 3D skeleton of the template mesh $T$ the same as the reconstructed 3D pose $S_{3d}$, and the template mesh is deformed together with its skeleton using the skinning scheme in [21]. The deformed mesh is set to $M$ as its initial value.

***Update Mesh*** To deform the mesh, the updated vertex positions are solved by a constrained Laplacian system, which minimizes

$$E = \sum_i \|v_i - t_i\|^2 + w_c \sum_i \left\| v_i - \sum_{j \in N(i)} c_{ij} v_j \right\|^2, \tag{2}$$

where $w_c$ is the smoothness weight, $N(i)$ means the neighborhood of vertex $i$, $c_{ij}$ is defined using the cotangent weight, and $t_i$ represents the target position of $v_i$ during deformation. Note that the first and second terms in (2) correspond to $E_{dist}$ and $E_{smooth}$ in (1), respectively.

The key task here is to determine the target $t_i$ using the silhouettes and the point cloud for each vertex. Inspired by [22], we define $t_i = v_i + \lambda_i n_i$, where $n_i$ is the normal at $v_i$, and the offset $\lambda_i$ is defined as

$$\lambda_i = \min(\min_k \lambda^S_{ik}, \lambda^P_i), \tag{3}$$

where $\lambda^S_{ik}$ is the offset of $v_i$ calculated from silhouette $S_k$ and $\lambda^P_i$ is the offset of $v_i$ calculated from point cloud $P$. They are computed as follows.

$$\lambda^S_{ik} = \text{argmin}_\lambda (dist_2(\bar{v}^k_i, \bar{t}^k_i) - dist_C(\bar{v}^k_i, S_k))^2,$$
$$\lambda^P_i = dist_3(v_i, P),$$

where $\bar{t}^k_i = Proj_k(v_i + \lambda n_i), \bar{v}^k_i = Proj_k(v_i)$. $dist_2$ stands for the distance between two pixels in an image, and $dist_C$ means the chamfer distance from a pixel to the silhouette, which can be pre-calculated using the fast marching method. $dist_3$ is the distance between a 3D point and its closest point in a point cloud. The $argmin$ here can be solved via a quadratic equation, and the sign of the root is determined by if projection of the vertex lies inside the corresponding silhouette. This step is performed iteratively (usually 3 iterations are sufficient based on our experiments) and the resolution of $M$ is also changed dynamically using subdivision during deformation procedure as [22].

***Update Image*** Due to imprecise projection matrices estimation from SFM, the projection of the mesh cannot toward the silhouettes very well in the above step as the conflicts of $\lambda^S_{ik}$ between different views. Instead of re-calibrating the cameras which is very challenging, our system fixes $\{Proj_k\}$ and tries to deform the silhouettes in images toward the projections of the mesh generated by the above steps as much as possible. To do this, we firstly apply the 2D-to-3D matching method in [23] to obtain correspondences between $M$ and the silhouettes in those views. We define the distance of one correspondence as the Euclidean distance of the projected point pairs on image. Thus, the correspondences whose distance larger than a threshold (5 pixels is set in our experiments) are considered as control points to move the silhouettes toward the projected $M$ by image warping. We use as rigid as possible method [24] in our paper to preserve the shape of contour in the image, for example, thickness of the arm in Fig 4 should not be changed too much during image warping.

The mesh and images are updated iteratively to achieve the best mesh $M$ with its well-aligned silhouettes. Fig 4 shows the whole optimization procedure. The result generated without image updating is also shown, which is shrunk too much caused by the inaccurate projection matrices. It is notice that our image updating strategy usually does not change the image too much and thus produces few affections on the quality of our reconstruction, which is validated by the results shown later.

At last, we perform a final laplacian deformation to move $M$ towards $P$ as closely as possible, where the target position $t_i$ are determined by ICP, and we reduce the smoothness weight in this step. Note that, self-intersections usually occur in the model after mesh updating, for instance, the two legs of the girl in Fig 4 intersect during the mesh evolving procedure. We remove the self-intersections using the technique in [25] to only keep the outermost surface of the model before ICP registration.

## V. Model Refinement

We develop a novel shading extraction technique and integrate it with a state-of-the-art shape-from-shading algorithm to recover geometric details in this section. Since the clothes worn by a human character may have spatially varying color patterns, which imply spatially varying albedo. To extract

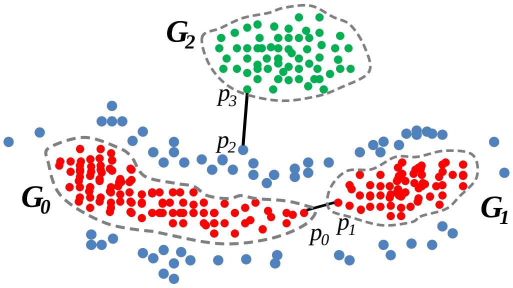


Fig. 5: Our adapted density based clustering.

shading information from pixel colors for such a surface with spatially varying albedo, it is necessary to group pixels into clusters such that pixels in the same cluster share the same albedo. This is nontrivial because pixelwise albedo is unknown at the time of clustering.

### A. Density Based Pixel Clustering

Even when one region of an image has a constant albedo, its color can still have much variation due to shading. This property usually makes traditional K-means or mean shift algorithms produce inaccurate clustering results. However, it can be observed that, over a region with constant albedo, the pixel color is smoothly varying on the image plane and the pixel cloud in the color space is also connected. Density based clustering algorithms are capable of handling such scenarios, where the data points are noisy and their distribution has an irregular shape. A recent hierarchical density based clustering algorithm [26] is modified to extract pixel clusters with varying densities. This clustering approach only requires two input parameters and can determine the number of clusters automatically.

We first introduce notations similar to their original definitions in [26].

**Core Distance:** The core distance of a pixel $p$, $d_{core}(p)$, is the Euclidean distance from $p$ to its $m_{pts}$-th nearest neighbor in the $Lab$ color space.

**Density:** The density of a pixel $p$ is defined as $density(p) = 1/d_{core}(p)$.

**Mutual Reachability Distance:** The mutual reachability distance between two pixels $p$ and $q$ is defined as $d_{mreach}(p,q) = \max\{d_{core}(p), d_{core}(q), d(p,q)\}$, where $d(p,q)$ means the Euclidean distance between $p$ and $q$ in $Lab$ space.

**$\varepsilon$-Mutual Reachability Graph:** It is a graph, $G_{m_{pts},\varepsilon}$, where the nodes represent the pixels in the image. For each pixel $p$, only its neighboring pixels with a distance to $p$ below $\varepsilon$ are used to form edges with $p$. The weight of each edge is set to the mutual reachability distance.

Clustering begins with a $\varepsilon$-mutual reachability graph $G_{m_{pts},\varepsilon}$ ($\varepsilon$ is a given parameter), which may contain a few connected subgraphs, and performs the following steps.

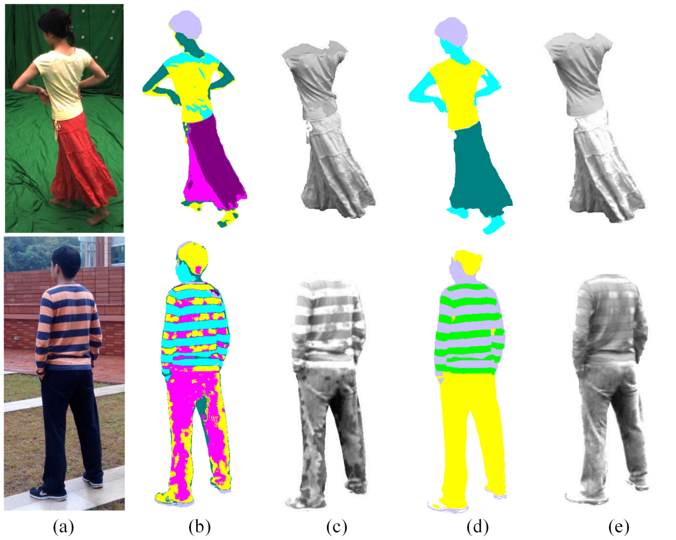


Fig. 6: A comparison between K-means and density based clustering. (a) Input image. (b) K-means clustering result (k=5). (c) Shading extraction based on K-means clustering. (d) Density based clustering result ($\varepsilon = 0.8$, $m_{pts} = 8$). (e) Shading extraction based on our density based clustering (skin and hair clusters are excluded).

***Step 1.*** Extract the minimum spanning tree ($MST$) of each subgraph in $G_{m_{pts},\varepsilon}$.

***Step 2.*** Label each pixel as an individual cluster.

***Step 3.*** Visit the edges in $MST$ in an increasing order of the weight. When an edge $(p,q)$ is visited ($p$ in cluster $G_p$ and $q$ in cluster $G_q$), perform one of three operations: pseudo-merge, absorb and merge.

i) *Pseudo-merge:* make $G_p$ and $G_q$ two sub-clusters of a new cluster $\widetilde{G_p}$.

ii) *Absorb:* assign $G_q$ to the noise set in $G_p$.

iii) *Merge:* really merge all sub-clusters and noise sets in $G_p$ and $G_q$ into one cluster $\widetilde{G_p}$.

The core of the algorithm is to determine which operation to perform. *'Absorb'* is performed if the size of $G_p$ or $G_q$ is smaller than a pre-defined parameter $m_{cltsize}$ (it is set to 20 in all experiments).

As illustrated in Fig 5, before deciding whether $G_0$ and $G_1$ need to be really merged (when the edge $(p_0, p_1)$ is visited), *'Pseudo-merge'* is performed to form a new cluster $\widetilde{G_0}$. Edge traversal then continues, while nearby noise pixels (blue points) are being absorbed, until absorbing is stopped by an edge across a gap (when the edge $(p_2, p_3)$ is visited). All of the absorbed pixels (blue points) in this process are assigned to the temporal noise set of $\widetilde{G_0}$. The stability of $\widetilde{G_0}$ is defined as follows.

$$S(\widetilde{G_0}) = \sum_{p \in \widetilde{G_0}} (\lambda_{max}(p, \widetilde{G_0}) - \lambda_{min}(\widetilde{G_0})). \tag{4}$$

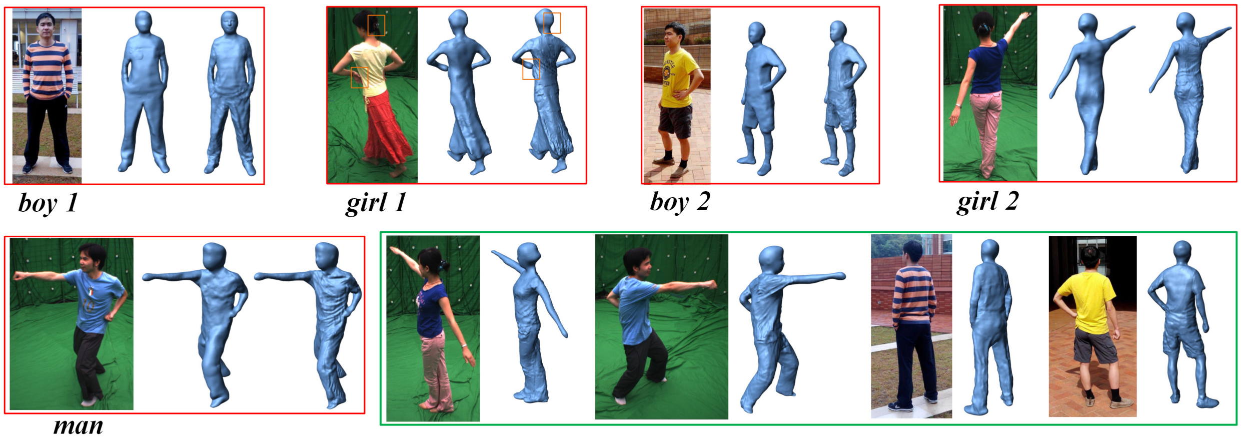


Fig. 8: The 5 examples used in our experiments are shown in the red box, listed as input image, coarse model and final model. Some results of other views are shown in green box.

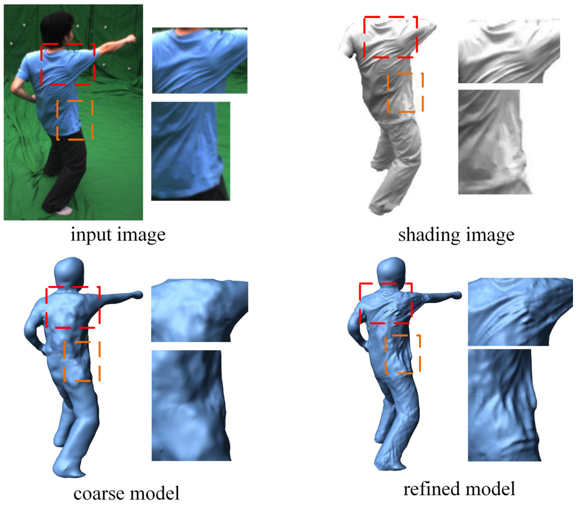


Fig. 7: Geometry refinement using extracted shading images.

where

$$\begin{aligned}\lambda_{max}(p,\widetilde{G_0}) &= \min(density(p), 1/d_{mreach}(p_0,p_1)),\\ \lambda_{min}(\widetilde{G_0}) &= 1/d_{mreach}(p_0,p_1).\end{aligned} \qquad (5)$$

A detailed explanation why (4) and (5) are used to define the stability is given in [26]. Intuitively, $\widetilde{G_0}$ is more stable if it is denser and farther away from $G_2$. If $S(\widetilde{G_0}) > S(G_0)+S(G_1)$, really merge $G_0$ and $G_1$, and the absorbed pixels are still kept in the noise set of $\widetilde{G_0}$. Once all edges have been visited, there is only one root cluster at the top of the hierarchy. The sub-clusters of this root cluster are taken as the real clusters we need, and its noise set contains all the noise pixels.

This clustering algorithm only has two primary parameters $\varepsilon$ and $m_{pts}$, which are set to 0.8 and 8 in all of our experiments. As shown in Fig 6, a comparison against the traditional K-means algorithm indicates that density based clustering generates much cleaner results even when pixelwise shading has much variation, which facilitates more accurate shading extraction.

### *B. Shading Extraction*

We extract shading information from multiple input images in three main steps:

**Clustering:** The above density based clustering algorithm is first performed on each input image. The resulting clusters in different views are then merged according to two cues, correspondence and color similarity. A cluster $G_1$ in view $I_i$ is merged with another cluster $G_2$ in view $I_j$ if some pixels from $G_1$ correspond to pixels from $G_2$ and the difference between the mean colors of $G_1$ and $G_2$ falls below a predefined threshold.

**Global Lighting Estimation:** After merging clusters across different views, we choose the largest cluster and set its albedo to be the mean color of its pixels. Pointwise shading (image color divided by the albedo) in this cluster is then used to estimate the global lighting and visibility function using the method in [14]. The estimated global lighting model is represented using spherical harmonics.

**Shading Calculation:** The predicted shading at every vertex on the coarse model are calculated using the normal vector at the vertex and the global lighting, and the albedo at the vertex is simply the ratio between the predicted shading and the image color corresponding to the vertex. By assuming the vertices in the same cluster share the same albedo, we set the albedo of a cluster to be the mean albedo of these vertices. Finally, a shading image is calculated as the ratio between the original image and the mean albedo of the cluster a pixel belongs to.

To fulfill the Lambertian assumption, we skip skin and hair regions during shading extraction. The skin region is identified with a skin color detection algorithm and the hair region is identified as the spatially separated cluster at the top of the segmented human figure. Some results on shading extraction are shown in Fig. 6.

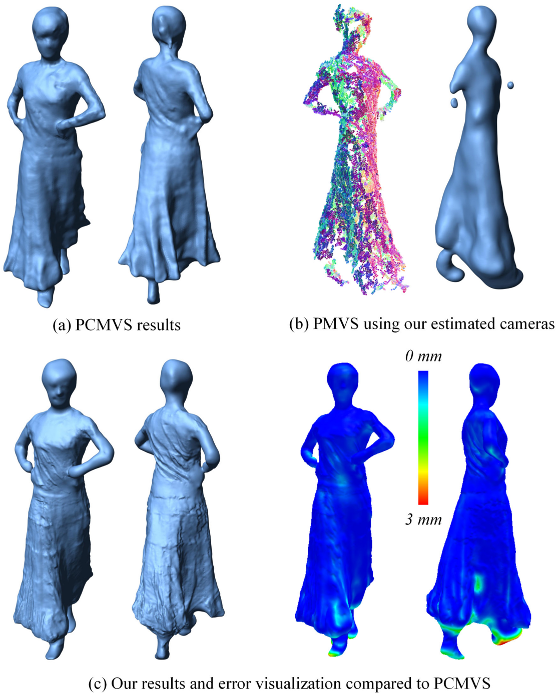
### C. Geometry Refinement

To reconstruct geometric details in each view from the extracted shading information, we apply the algorithm in [14] to minimize the following cost function

$$E(\{\lambda_i\}) = \sum_i (I_i^s - S_i)^2 + w_f \sum_i \left\| v_i - \sum_{j \in N(i)} c_{ij} v_j \right\|^2 , \quad (6)$$

where $I_i^s$ represents the shading value at the pixel location $Proj(v_i)$ which is the projection of $v_i$, $S_i$ is the predicted shading value based on the normal vector $n_i$ at $v_i$ and the estimated global lighting. We also constrain every vertex to move along its normal from the coarse model during the minimization and $\lambda_i$ means the offset at $v_i$. The second part of this cost function ensures the smoothness of the mesh, $w_f$ is the smoothness weight and $c_{ij}$ is the cotangent weight. This nonlinear optimization is solved using $L-BFGS$.

In surface regions visible to multiple views, the computed offsets at the same vertex but from different views are usually inconsistent due to inaccurate camera projection matrices. To overcome this issue, we find an optimal seam on the mesh surface to stitch together the refined geometry from every pair of adjacent views. The graph cut algorithm is used here to find the optimal seam. Fig 7 gives an example to show the results of geometry refinement.



(a) PCMVS results

(b) PMVS using our estimated cameras

(c) Our results and error visualization compared to PCMVS

Fig. 9: Comparison of final 3D reconstruction with PMVS and PCMVS. (a) Result from PCMVS using 20 calibrated images. (b) Reconstructed point cloud and mesh from PMVS using our roughly estimated camera parameters. (c) Our result from 16 uncalibrated images.

The face part of our coarse model is enhanced with a simple template based deformation scheme. Facial feature points are first detected in a frontal view, and they are backprojected onto the coarse model using the estimated camera projection matrix to find their corresponding vertices there. A pre-labeled face template is first rigidly registered with the backprojected 3D facial feature points. Displacements parallel to the frontal view are then added to the pre-labeled feature points on the template such that their projections on the frontal view match the detected facial features. Finally, we deform the facial region of our coarse model towards the displaced face template using the ICP strategy.

## VI. Experimental Results and Discussion

We have successfully evaluated our proposed algorithm on multiple examples, including the ones shown in Fig. 8. Among these examples, *"man"*, *"girl 1"* and *"girl 2"* are taken from the indoor dataset in [3]. Each of them has 20 images from different views. In our experiments, we use 15 images of *"man"*, 16 images of *"girl 1"* and 13 images of *"girl 2"* for 3D reconstruction. The *"boy 1"* and *"boy 2"* examples are captured with a cell phone camera in an outdoor environment with 15 images and 16 images respectively. The entire image acquisition process for each example takes around 3 minutes. These examples exhibit quite a few challenges: complicated pose (*"man"*), complicated garment (*"girl 1"*) and textureless clothes (*"boy 1"*). Some sample views of the reconstructed results are also shown in Fig 8.

### A. Comparison with PMVS/PCMVS

PMVS [5] is one of the best multi-view stereo algorithms. However, it requires camera parameters and cannot handle our sparse uncalibrated views directly. We use our estimated camera parameters as input and perform the reconstruction by PMVS. Fig 9 $(b)$ shows that this method can only reconstruct an incomplete point cloud which is insufficient for surface reconstruction. Our result is also compared against PCMVS [3] on the "*girl 1*" example in Fig 9. The result of PCMVS is obtained from 20 views with well-calibrated camera parameters, while our result is reconstructed from 16 views without making use of their existing camera parameters. To validate the quality of our generated mesh $M$, we use the result of PCMVS as the ground truth $G$ and take the nearest neighbor distance for each vertex of $M$ to $G$ as the error. The max distance error is 3.8 millimeters (where the girl is of 170 cm height) and the distribution is visualized in Fig 9 $(c)$ . It is also shown that the biggest errors happen at the bottom of skirt and feet and are due to the lacking of point cloud and the low accuracy of visual hull at those parts.

### B. Number of Input Images and Skeleton-labeled Images

One important question is how many images are required for 3D reconstruction with our algorithm. Generally speaking, the more input images the more accurate the result. As shown in Fig 10, we run our algorithm on the "*girl 1*" example

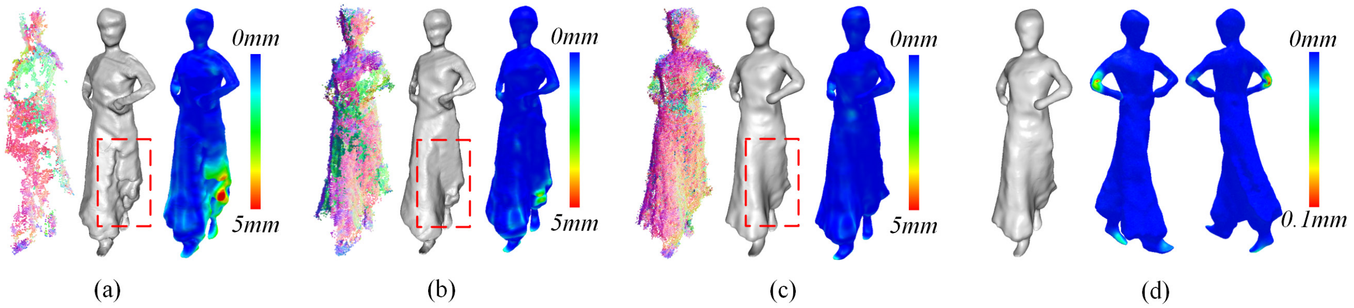


Fig. 10: Our result from different numbers of input images and skeleton-labeled images. The point clouds and coarse models are reconstructed from (a) 8 images, (b) 12 images, and (c) 16 images, respectively. And, (c) is generated by labeling skeletons of 3 images and (d) uses 6 labeled images. The color maps of (a), (b) and (c) show the differences of their models with the model generated by PCMVS using the nearest neighbor distance, while the color map of (d) shows the difference between its model with model in (c).

using different numbers of input images. 8, 12 and 16 images are respectively chosen from the original 20 images as the input. As seen in the results, both 8 and 12 views cannot generate a sufficiently dense point cloud to capture the shape of the highlighted region on the dress. According to our experiments, around 15 uniformly sampled views around a person are recommended. We also compared the generated results with different number of skeleton-labeled images as shown in Fig  10, where the mesh of $(c)$ is generated from 3 labeled images and $(d)$ used 6 labeled views. The nearest neighbor distance between these two meshes is also shown in $(d)$. One can notice that there only exists some tiny differences at the parts of arms and feet. This shows that our algorithm is insensitive to the number of skeleton-labeled images. It is because the labeled skeletons in our system only served for the 3d skeleton reconstruction and initial mesh generation, and the final result is mostly determined by the point cloud and visual hull which are reconstructed primarily from the pixel correspondences. Based on our experiments, 3 skeleton-labeled images in different views are sufficient and all of our results in this paper are generated by 3 labeled images.

### C. Runtime Performance

Our algorithm is implemented using C++ and runs on a standard PC with a 3.4GHz Intel quad core processor. Take “*girl 1*” as an example, it is generated in 33 minutes in total(the input images have a 300x600 resolution). NRDC matching and propagation takes 643 seconds in total (where NRDC matching takes on average 20 seconds for each pair of adjacent images), SFM takes 46 seconds, coarse model generation takes 110 seconds (where the template mesh has 6,449 vertices), clustering and shading extraction take 54 seconds, and final geometry refinement with nonlinear optimization takes 1,132 seconds for a coarse model with 253,212 vertices. The other four models have similar runtime performance and all of them are produced less than 35 minutes.

### D. Limitations and Future work

Requiring a small number of skeleton-labeled images is the primary limitation of our algorithm which prevents the system from performing automatically. Eliminating the user interaction during the reconstruction procedure is left as one of our future works. Another one important limitation of our system is that it usually generates low-quality results for some detailed parts such as hand and hair as shown by the highlighted orange box in “*girl 1*” example of Fig  8. This is due to the low-quality of visual hull, reconstructed point cloud and extracted shading at these parts, and it is also very challenging to evolve the mesh into these small areas while keeping the smoothness. Additionally, reconstruction of high-quality hair and face from the input images is another direction of the future work.

## VII. Conclusions

We have presented a novel two-stage algorithm for reconstructing 3D human models from sparse uncalibrated views. The first stage reconstructs a coarse model with the help of a template model for human figures. We fit the template model to the point cloud reconstructed from dense correspondences while enclosing it with the visual hull. In the second stage, the coarse model from the first stage is refined with geometric details reconstructed from shading information. A novel shading extraction algorithm has been proposed for surfaces with nonuniform reflectance. This algorithm builds on an adapted density based clustering algorithm. Our algorithm has been validated with images from an existing dataset as well as images captured by a cell phone camera.

## VIII. Acknowledgements

We wish to thank the anonymous reviewers for their valuable comments. This work was partially supported by Hong Kong Research Grants Council under General Research Funds (HKU718712).

## References

[1] J. Tong, J. Zhou, L. Liu, Z. Pan, and H. Yan, "Scanning 3d full human bodies using kinects," *IEEE Transactions on Visualization and Computer Graphics (Proc. IEEE Virtual Reality)*, vol. 18, no. 4, pp. 643–650, 2012.

[2] H. Li, E. Vouga, A. Gudym, L. Luo, J. T. Barron, and G. Gusev, "3d self-portraits," *ACM Transactions on Graphics (Proceedings SIGGRAPH Asia 2013)*, vol. 32, no. 6, 2013.

[3] Y. Liu, Q. Dai, and W. Xu, "A point-cloud-based multiview stereo algorithm for free-viewpoint video," *IEEE Transactions on Visualization and Computer Graphics*, vol. 16, no. 3, pp. 407–418, 2010.

[4] R. I. Hartley and A. Zisserman, *Multiple View Geometry in Computer Vision*, 2nd ed. Cambridge University Press, 2004.

[5] Y. Furukawa and J. Ponce, "Accurate, dense, and robust multi-view stereopsis," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 32, no. 8, pp. 1362–1376, 2010.

[6] Y. HaCohen, E. Shechtman, D. B. Goldman, and D. Lischinski, "Non-rigid dense correspondence with applications for image enhancement," ser. SIGGRAPH '11, 2011, pp. 70:1–70:10.

[7] L. Maxime and Q. Long, "A quasi-dense approach to surface reconstruction from uncalibrated images," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 27, no. 3, pp. 418–433, 2005.

[8] T. Wu, S. Yeung, J. Jia, and C. Tang, "Quasi-dense 3d reconstruction using tensor-based multiview stereo," in *CVPR*, 2010.

[9] P. Guan, A. Weiss, A. Balan, and M. J. Black, "Estimating human shape and pose from a single image," in *ICCV*, 2009, pp. 1381–1388.

[10] S. Zhou, H. Fu, L. Liu, D. Cohen-Or, and X. Han, "Parametric reshaping of human bodies in images," *ACM Trans. Graph.*, vol. 29, no. 4, pp. 126:1–126:10, 2010.

[11] C. Stoll, J. Gall, E. de Aguiar, S. Thrun, and C. Theobalt, "Video-based reconstruction of animatable human characters," *ACM Trans. Graph.*, vol. 29, no. 6, 2010.

[12] J. Gall, C. Stoll, E. D. Aguiar, C. Theobalt, B. Rosenhahn, and H. peter Seidel, "Motion capture using joint skeleton tracking and surface estimation," in *CVPR*, 2009.

[13] E. de Aguiar, C. Stoll, C. Theobalt, N. Ahmed, H.-P. Seidel, and S. Thrun, "Performance capture from sparse multi-view video," *ACM Trans. Graph.*, vol. 27, no. 3, 2008.

[14] W. Chenglei, B. Wilburn, Y. Matsushita, and C. Theobalt, "High-quality shape from multi-view stereo and shading under general illumination," in *CVPR*, 2011, pp. 969 –976.

[15] W. Chenglei, V. Kiran, L. Yebin, S. Hans-Peter, and T. Christian, "Shading-based dynamic shape refinement from multi-view video under general illumination," in *ICCV*, 2011, pp. 1108–1115.

[16] H. Yudeog, L. Joon-Young, and S. K. In, "High quality shape from a single rgb-d image under uncalibrated natural illumination," in *ICCV*, 2013.

[17] L.-F. Yu, S.-K. Yeung, Y.-W. Tai, and S. Lin, "Shading-based shape refinement of rgb-d images," in *CVPR*. IEEE, 2013, pp. 1415–1422.

[18] H. Fu, X. Han, and Q. H. Phan, "Data-driven suggestions for portrait posing," in *ACM SIGGRAPH Asia 2013, Technical Briefs*, 2013.

[19] N. Snavely, S. M. Seitz, and R. Szeliski, "Photo tourism: Exploring photo collections in 3d," *ACM Trans. Graph.*, vol. 25, no. 3, pp. 835–846, Jul. 2006.

[20] H. Huang, S. Wu, M. Gong, D. Cohen-Or, U. Ascher, and H. Zhang, "Edge-aware point set resampling," *ACM Transactions on Graphics*, vol. 32, pp. 9:1–9:12, 2013.

[21] I. Baran and J. Popović, "Automatic rigging and animation of 3d characters," *ACM Trans. Graph.*, vol. 26, no. 3, Jul. 2007.

[22] A. Sharf, T. Lewiner, A. Shamir, L. Kobbelt, and D. Cohenor, "Competing fronts for coarse-to-fine surface reconstruction," in *Computer Graphics Forum*, 2006, pp. 389–398.

[23] V. Kraevoy, A. Sheffer, and M. van de Panne, "Modeling from contour drawings," in *Proceedings of the 6th Eurographics Symposium on Sketch-Based Interfaces and Modeling*, ser. SBIM '09, 2009, pp. 37–44.

[24] T. Igarashi, T. Moscovich, and J. F. Hughes, "As-rigid-as-possible shape manipulation," *ACM Trans. Graph.*, vol. 24, no. 3, pp. 1134–1141, Jul. 2005.

[25] A. Zaharescu, E. Boyer, and R. Horaud, "Transformesh: A topology-adaptive mesh-based approach to surface evolution," in *ACCV*, 2007, pp. 166–175.

[26] R. J. G. B. Campello, D. Moulavi, and J. Sander, "Density-based clustering based on hierarchical density estimates," *PAKDD*, no. 2, pp. 160–172, 2013.

**Xiaoguang Han** is currently a Ph.D. student with the Department of Computer Science at the University of Hong Kong since 2013. He received his M.Sc. in Applied Mathematics (2011) from Zhejiang University, and his B.S. in Information and Computer Science (2009) from Nanjing University of Aeronautics and Astronautics, China. He was also a Research Associate of School of Creative Media at City University of Hong Kong during 2011 to 2013. His research interests include computer graphics and computer vision.

**Kwan-Yee K. Wong** received the BEng degree, with first class honors, in computer engineering from the Chinese University of Hong Kong in 1998, and the MPhil and PhD degrees in computer vision (information engineering) from the University of Cambridge in 2000 and 2001, respectively. Since 2001, he has been with the Department of Computer Science at the University of Hong Kong, where he is now an associate professor. His research interests are in computer vision and image processing, including camera calibration, motion tracking, model reconstruction and representation, and motion estimation from image sequences.

**Yizhou Yu** received the Ph.D. degree from University of California at Berkeley in 2000. He is currently a professor at the University of Hong Kong and an adjunct professor at University of Illinois, Urbana-Champaign. He is a recipient of the 2002 US National Science Foundation CAREER Award, 1998 Microsoft Graduate Fellowship and the Best Paper Awards at 2005 and 2011 ACM SIGGRAPH/EG Symposium on Computer Animation. He has served as an associate editor of IEEE Transactions on Visualization and Computer Graphics, Computer Graphics Forum and the Visual Computer, and is on the editorial board of International Journal of Software and Informatics. His current research interests include computer graphics, computer vision, digital geometry processing, video analytics and biomedical data analysis. He is a senior member of IEEE.