KinectAvatar: Fully Automatic Body Capture Using a Single Kinect

Yan Cui[†], Will Chang[‡], Tobias Nöll[†], Didier Stricker[†]

[†]Augmented Vision, DFKI

Abstract. We present a novel scanning system for capturing a full 3D human body model using just a single depth camera and no auxiliary equipment. We claim that data captured from a single Kinect is sufficient to produce a good quality full 3D human model. In this setting, the challenges we face are the sensor's low resolution with random noise and the subject's non-rigid movement when capturing the data. To overcome these challenges, we develop an improved super-resolution algorithm that takes color constraints into account. We then align the super-resolved scans using a combination of automatic rigid and non-rigid registration. As the system is of low price and obtains impressive results in several minutes, full 3D human body scanning technology can now become more accessible to everyday users at home.

1 Introduction

Three-dimensional geometric models of real world objects, especially human models, are essential for many applications such as design, virtual prototyping, quality assurance, games, and special effects. However, highly trained artists are mainly responsible for performing this modeling task, often using specialized modeling software. While 3D scanning technology has been available as an alternative for some time, it is still not used very widely for capturing models. This is because scanning devices are still expensive and often require expert knowledge for operation. And many scanning systems are limited for only rigid objects.

In this work, we present a novel approach to full 3D human body scanning which employs a single Microsoft Kinect [1] sensor. The Kinect has a variety of advantages over existing 3D scanning devices. It can capture depth and image data at video rates without a significant dependence on specific lighting and texture conditions. Also the Kinect is compact, low-price, and as easy to use as a normal video camera.

The challenge in using a single Kinect for scanning is that it provides relatively low-resolution data (320×240) with a high noise level. With these data characteristics, it is difficult to produce high-quality 3D models. Devices such as the Kinect were designed for use in object detection and natural user interfaces, not for high-quality 3D scanning. In addition, since we observe our subject only from a single, fixed location, movement of the subject is unavoidable during scanning. Therefore, we must estimate and compensate for this motion in order to combine scans from different viewpoints and produce a complete model.

[‡]Author has no affiliation at this time. Contact address: william.y.chang@gmail.com

We claim that data captured from a single Kinect is sufficient to produce a good quality full 3D human model, without any additional equipment or shape priors (e.g. a turntable, additional Kinects, or a database of human body models). A key benefit of our algorithm is that it only relies on the input data and does not require an explicit shape model of the scanned subject. This allows us to reproduce personalized detail geometry such as faces and clothing. Our pipeline leverages prior work in depth super-resolution and articulated non-rigid registration. The contributions are

- a pipeline for automatically scanning full human body using a single Kinect,
- an improved depth super-resolution algorithm, taking color constraints into account,
- and a unified formulation of rigid and non-rigid registration under a probabilistic model, improving upon prior work [2] [3] for registration quality in high noise scenarios and for solving the loop closure problem.

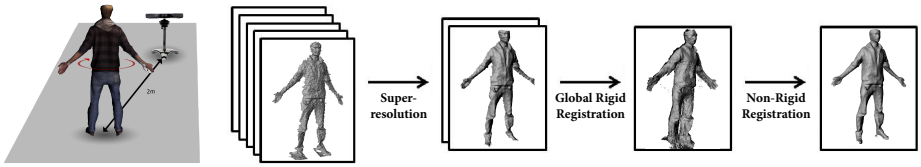
In the following section, we discuss the relationship of this paper to previous work. Afterwards, we give technical details of our scanning system and verify our main claim by demonstrating results on several real-world datasets.

2 Related Work

Recent developments in low cost real-time depth cameras have opened up a new field for 3D content acquisition. In particular, several publications about 3D shape scanning with the Kinect have appeared. Henry et al. [4] build dense 3D maps of indoor environments, and Newcombe et al. [5] present a real-time 3D scanning system for arbitrary indoor scenes. These projects focus mainly on scanning static scenes of indoor environments.

Specifically concerning the problem of human body scanning, Cui et al. [6] try to scan a human body with one Kinect. However, they do not use the color information, and fundamental proplem is that they can not handle non-rigid movement, so the reconstructed results in the arm and leg parts are not of high quality. The work by Weiss et al. [7] estimates the body shape by fitting the parameters of a SCAPE model [8] to depth data and image silhouettes from a single Kinect. In contrast, our work does not require a prior shape model and relies mainly on registration. Thus, our method reproduces personalized details such as faces or dresses from the scans. Most recently, Tong et al. [9] present a system to scan a body using three Kinects and a turntable. They also utilize a global non-rigid registration algorithm which uses a rough template constructed from the first depth frame. While we share some similarities, our system setup is much simpler (using only a single Kinect), and our global registration works directly with the data without requiring a rough template.

As the raw Kinect depth data is of low resolution with high noise levels, a smoothing algorithm should be applied as a pre-processing step. Newcombe et al. [5] apply a bilateral filter [10] to the raw Kinect depth map to obtain a discontinuity preserved depth map with reduced noise. Schuon et al. [11] develop a super-resolution algorithm (LidarBoost) to improve the depth resolution and data quality of a ToF range scan, and Cui et al. [2] further develop this method. In this paper, we compare these existing



(a) Working setup (b) Processing pipeline

Fig. 1. Outline of our working setup and processing pipeline.

methods and provide a new super-resolution algorithm for the Kinect depth and color data. This improves the resolution, reduces the noise, and preserves shape detail.

The global rigid registration problem is an important topic in object scanning. Iterative Closest Points (ICP) and its variants [12] can solve the local rigid alignment problem. Global rigid alignment techniques [13][14][15] can be used to register the scans against each other and solve the well-known loop closure problem. However, the input data used in these algorithms are acquired using structured light or laser scanning, which has high resolution and little noise. Cui et al. [2] present a probabilistic scan alignment approach that explicitly takes the sensor's noise characteristics into account. However, this method can only solve a local alignment task. We improve upon their approach and develop a probabilistic global alignment algorithm.

Even after the global rigid alignment, the scans do not align well because of the human bodies' non-rigid deformation during scanning (especially in the arms and legs). Aligning these scans is a challenging task that often requires high quality scan data and small changes of the pose in each scan [3][16][17][18][19][20][21]. While most of these approaches mainly deal with high-resolution scan data, we propose a robust non-rigid global alignment that can work well even with the Kinect's resolution and noise levels. Since the human body is largely articulated, we build on the global articulated registration work by Chang and Zwicker [3] and significantly improve the algorithm using a probabilistic scan alignment approach robust to the noise of the scans.

3 Overview

First, we give an overview of our scanning system, easily built as shown in Fig. 1a. The user stands before a Kinect in a range of about 2 meters, such that the full body falls in the Kinect's range of view. Then, the user simply turns around 360 degrees for about 20 to 30 seconds while maintaining an approximate "T" pose.

While scanning, we capture $s = 10$ frames as one view "chunk" with depth maps $\mathcal{D}_\ell = (D_1, \ldots, D_s)$ and color maps $\mathcal{C}_\ell = (C_1, \ldots, C_s)$. We then stop 0.5 seconds, capture another chunk, stop, capture, and so on until the human body has turned completely. The capture process yields about 10 raw data frames for each view, with the body turning approximately 30 degrees between the views. This ensures that there are enough overlapping areas for registration while providing full 360 degree coverage. The depth maps and color maps are calibrated and aligned (calibration off-line). In total, we get K chunks, where $K = 30 \sim 40$. Since the Kinect captures at about 30 FPS, the displacement is relatively small within a single chunk.

The resulting range data is then processed by our reconstruction algorithm, comprised of the following steps (Fig. 1b). First, a super-resolution step takes each chunk as input, and produces a single super-resolved depth scan with both greater depth accuracy and resolution (Sec. 4). Second, global rigid and non-rigid alignment steps combine the super-resolved scans into a final model (Sec. 5). Here, by "global" we mean that all of the frames in the sequence are aligned together. After the alignment is complete, we reconstruct a single mesh using Poisson mesh reconstruction [22] as a post processing step. Finally, textures are created with the Kinect's raw color data (Sec. 6). Our results show that this system can capture impressive 3D human shapes with personalized geometric details. In subequent sections, we describe each step of the system in more detail.

4 Super-resolution

We apply our super-resolution algorithm to each chunk of depth images, yielding K new super-resolved depth maps H_ℓ with much higher X/Y resolution and less noise. Our algorithm is largely based on Cui et al [23].

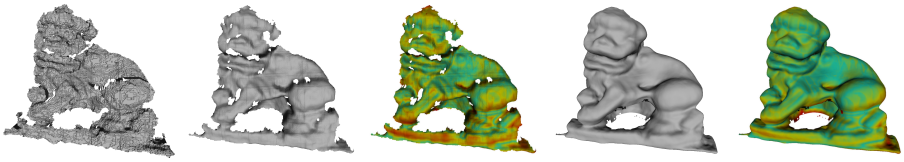
First, we align all depth and color maps of a chunk to its middle frame using optical flow. This is sufficiently accurate since the maximum viewpoint displacements throughout the entire chunk are typically one to three pixels. This corresponds to a turning speed of aobut 30 seconds per revolution by the user. Second, we extract a high-resolution denoised depth map H_ℓ from the aligned low resolution depth and color maps by optimizing the following objective function:

$$\min_{H_\ell} \mathcal{E}_{\text{data}}(D_1, \ldots, D_s, C_1, \ldots, C_s, H_\ell) + \gamma \mathcal{E}_{\text{reg}}(H_\ell) \ , \tag{1}$$

$$\begin{aligned} \mathcal{E}_{\text{data}} &= \sum_{k=1}^{s} \left\| W_k \circ \left(H_\ell - f\left(D_k\right) \right) \right\|^2 \\ W_k &= \frac{1}{C_k - \frac{1}{s} \sum_{k=1}^{s} C_k}. \end{aligned} \tag{2}$$

Here, we upsample the depth data by a factor $\beta = 2$ in both X and Y dimensions. After the optimization, the depth map H_ℓ is converted into a 3D point cloud $Y_\ell = \{y_j \mid j = 1 \ldots \beta^2 N_X N_Y\}$ using the Kinect's intrinsic parameters after optimization (N_X, N_Y are the resolution of the raw depth data).

To explain the optimization further, $\mathcal{E}_{\text{data}}$ measures the agreement of H_ℓ with the low resolution depth maps. Here, f is a function, which upsamples the low resolution $\mathcal{D}_\ell$ to higher resolution of $\beta^2 N_X N_Y$ and align to the center depth frame. We performed experiments to determine the best resampling strategy. It turned out that a nearest neighbor sampling from the low resolution images is preferable over any type of interpolated sampling. Interpolation implicitly introduces unwanted blurring that leads to a less accurate reconstruction of high-frequency shape details in the superresolved result. W_k is a per-pixel weight that measures the quality of the optical flow alignment, $\circ$ denotes element-wise multiplication, and $\|\|^2$ denotes sum of the square normal for each pixel. The $\frac{1}{\cdot}$ notation for W_k means that we invert the value per pixel. Since the human is moving in 3D space but optical flow just considers the movement in 2D image space,



(a) Raw data (b) Off color (c) Off color (AEE = 4.84) (d) With color (e) With color (AEE = 2.98)

Fig. 2. Super-resolved mesh and error distribution results, produced without the color constraint (b,c) and with the color constraint (d,e). AEE denotes average Euclidean error, computed by averaging the error values over all points in the mesh.

some points are not aligned properly in each chunk. The weight term W_k quantifies this by producing a larger value if the color in each raw frame is similar to the average color; otherwise, the point is not correctly aligned and the weight is smaller. Finally, the $\mathcal{E}_{\text{reg}}$ term is a feature-preserving smoothing regularizer tailored to the Kinect depth data, with γ as the weighting coefficient. We use the same definition as [23], with their four different versions of regularizers that are inspired by image processing counterparts [24]: linear, square nonlinear, isotropic nonlinear and anisotropic nonlinear. Compared to previous work [23], the new contribution is the weight term W_k which allows the optimization to the color into account as an additional constraint.

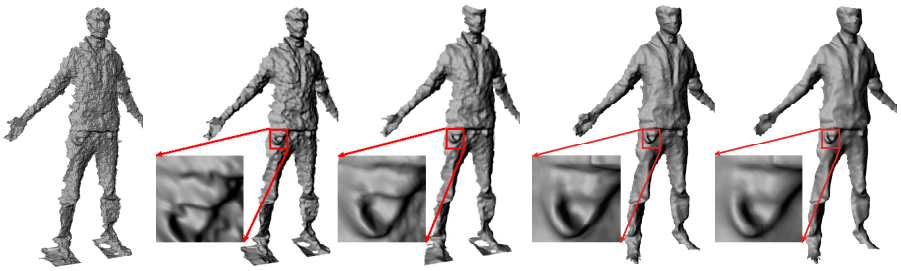
Our implementation uses the Euler-Lagrange equation to transform the optimization problem into a linear system of equations, which we then solve using Gauss-Seidel [23]. The runtime is shown in Tab. 1 with a C++ implementation.

Fig. 2 compares the difference between super-resolution without the color constraint (Fig. 2b) and with the color constraint (Fig. 2d). Incorporating the color constraint gives a smoother surface while preserving important shape detail. It also produces a result that compares more favorably to ground truth Fig. 5c. Here, the errors are shown in the color-coded error plots (Fig. 2c and Fig. 2e, error range in Fig. 5e). The average Euclidean error (AEE) with the color constraint is smaller as a result.

We also demonstrate the effect of the color constraint for a human scan. Fig. 3b shows results using the LidarBoost [11] filter (square regularization), and Fig. 3c shows results using the anisotropic nonlinear filter [23]. In general, anisotropic performs best because it employes a diffusion tensor instead of a simple square regularization. Notice that for both filters, the result that uses the color constraint gives a smoother appearance while preserving important shape detail. We thus use the anisotropic nonlinear operator with the color constraint to super-resolve the point clouds.

5 Global Registration

We consider the human body as articulated, with rigid structures connected by joints. Aligning such point clouds is challenging, especially considering that the Kinect depth data has much noise even after super-resolution. We solve this special global registration problem inspired by two approaches. First, we incorporate the maximum-likelihood formulation described by Myronenko et al. [25] and Cui et al. [2] which explicitly



(a) Raw data (b) LidarBoost off & with color (c) Anisotropic off & with color

Fig. 3. Super-resolved meshes using the LidarBoost and Anisotropic filters. The images compare results with and without the color constraint.

takes into account the sensor's noise characteristics. Second, we employ the articulated model described in Chang and Zwicker [3] to describe the non-rigid motion. Here, the surface motion is expressed in terms of an articulated model. We extend their ideas to our setting and develop an approach for global rigid and non-rigid registration for noisy Kinect data.

5.1 Problem Setup

The input to registration are 3D point clouds $Y_f = \{y_{f,n} \mid n = 1, \ldots, N_f\}$ from each super-resolved frame $f \in H_\ell$. Here, N_f is number of points in frame f.

We need to solve for the motion of each frame so that all frames align to each other. To parameterize the motion, we define a set of rigid transformations $M_f^0, M_f^1, M_f^2, \ldots$ for each frame f. This set will describe the motion of the frame from its original location. Since there are multiple transformations per frame, we will need to define which transformation associates with each point. We define this association as a label $i(n)$: the index of the transformation assigned to $y_{f,n}$. Therefore, $M_f^{i(n)}$ will be the transformation for the point $y_{f,n}$. This parameterization of the motion is essentially an articulated model, since the points divide into rigid parts according to their label [3].

When we set up the problem in this way, the alignment task then reduces to solving for the transformations and labels that give the best alignment possible. We model a likelihood function describing the quality of the alignment, and we maximize this likelihood for all frames simultaneously to produce the result.

Conformal Geometric Algebra. We use an exponential map based on conformal geometric algebra to express the transformations. The Euclidean transformations of a point X into X' in the conformal space caused by a motor M and a reverse motor $\tilde{M}$ is approximated as:

$$X' = MX\tilde{M} = E + e_\infty(x - l \cdot x - m), \tag{3}$$

where E is the identity matrix, e_∞ is the point at infinity, and $l, m \in \mathbb{R}^3$ corresponds to the rotation and traslation in the conformal geometric algebra, respectively. This formulation has the advantage that we only have 6 degrees of freedom (DoF). A formulation with a $3 \times 3 \in \mathbb{R}$ matrix and a $3 \times 1 \in \mathbb{R}$ translation vector would have

12 variables instead. A further advantage is that we can gain linear equations with respect to minimizing an energy function using the motor. For notational convenience, we use the expression $\mathcal{T}_f^{i(n)}(X)$ to denote the transformation of point X using $M_f^{i(n)}$ ($\mathcal{T}_f^{i(n)}(X) = M_f^{i(n)} X \tilde{M}_f^{i(n)}$). For more background information for Eqn. 3 and its optimization, please refer to the references [26][27][28].

Deformation model. To aid the optimization, we find neighbor relations between the rigid parts and constrain such neighbors to be joined together at a common location. These locations we call "joints" and infer them during the optimization. Specifically, we use ball joint relations (3 DoF) between rigid parts, expressed by a single point denoted $y_{f,ab} \in \mathbb{R}^3$ relating two rigid transformations M_f^a and M_f^b. The joints constrain neighboring transformations to agree on this common ball joint location. This ensures that the rigid parts stay connected and do not drift away. We estimate the joint locations automatically using the same technique as in previous work [3].

Difference to Previous Work. Our problem setup is similar to that of Chang and Zwicker [3]. However, there are notable differences in our work. First, we do not subsample the frames but instead define a label for all points in all frames. Therefore, we use the mesh connectivity in each frame to define neighborhood relationships between the points.

Second, we do not define a reference frame or employ a dynamic sample graph. The transformations operate directly on the scanned points so that every scan aligns to all others. In addition, since we associate the labels directly to the scan points, we obtain a separate, independent segmentation per frame (based on the motion occuring in the frame). This means that, unlike prior work, all frames are moving independently. Also, we have a separate set of joint locations per frame, whereas the joint locations were defined on the reference frame previously. Changing the problem setup in this manner allows us to optimize the alignment of all frames in a single optimization step.

Probabilistic model. The key ingredient for making the registration robust to noise and outliers is the probabilistic modeling of the point clouds. We consider each Y_f to be generated from a Gaussian Mixture Model (GMM), with density as follows [2][25]:

$$p(x) = \sum_{n=1}^{N_f} \frac{1}{N_f} p\left(x \mid n\right) \;\text{ with }\; p\left(x \mid n\right) \propto N(y_{f,n}, \sigma^2 I)\ . \tag{4}$$

Simply speaking, $p(x)$ gives the probability that $x \in \mathbb{R}^3$ is generated by Y_f. As the equation shows, we model the GMM using the original point set Y_f and center each multi-variate Gaussian $N(y_{f,n}, \sigma^2 I)$ at the scanned points $y_{f,n}$. All Gaussians share the same isotropic covariance matrix $\sigma^2 I$, with I as a 3×3 identity matrix and σ^2 as the variance in all directions.

5.2 Energy Function

We define a measure of how well the point sets align using the following function:

$$\arg\min_{\mathcal{M},\mathcal{L}} E_{\text{data}}(\mathcal{M}, \mathcal{L}) + \lambda E_{\text{reg}}(\mathcal{M}, \mathcal{L}), \tag{5}$$

where $\mathcal{M}$ is the entire transformation set and $\mathcal{L}$ are the labels for all the points of the K frames. E_{data} measures the alignment distance between points in each frame, E_{reg} constrains the labels for a smooth segmentation and also neighboring transformations to agree on a common joint location. λ is weighting coefficient.

Data Term E_{data}. To achieve a closed model, all frames should be aligned globally with minimal distance. For each pair of frames f and g, the alignment task is performed by minimizing the negative log-likelihood based on the probabilistic model:

$$E_{\text{data}}(\mathcal{M},\mathcal{L}) = -\sum_{f,g}\sum_{n=1}^{N_f}\log\sum_{m=1}^{N_g}\exp\left(\frac{\left\|\mathcal{T}_f^{i(n)}(y_{f,n})-\mathcal{T}_g^{j(m)}(y_{g,m})\right\|^2}{-2\sigma_{f,g}^2}\right), \tag{6}$$

where $i(n)$ is the index of the transformation assigned to the point $y_{f,n}$, and $j(m)$ the same for the point $y_{g,m}$ but based on the segmentation of frame g. The variance $\sigma_{f,g}^2$ of mixture components is estimated separately for each point using

$$\sigma_{f,g}^2 = \frac{1}{N_f N_{\text{near}}}\sum_{n=1}^{N_f}\sum_{m\in\text{Near}(y_{f,n})}^{N_{\text{near}}}\left\|\mathcal{T}_f^{i(n)}(y_{f,n}) - \mathcal{T}_g^{j(m)}(y_{g,m})\right\|^2, \tag{7}$$

where $m \in \text{Near}(y_{f,n})$ denotes the indices of the nearest N_{near} points in frame g for point $y_{f,n}$. In our experiments, we use a value of $N_{\text{near}} = 20$.

Regularization term E_{reg}. There are two parts for the regularization term E_{reg} [3]. The first part is a smoothness term for the labels, which constrains neighboring points n, m to have a similar label to ensure a smooth segmentation result. If the label is not the same ($i(n) \neq i(m)$), we apply a penalty $I(\cdot) = 1$ which is added to E_{reg}.

The second part is the joint constraint which ensures that the rigid parts do not drift away from each other. Each ball joint specifies that its point $y_{f,ab}$ should move to the same location when applying M_f^a and M_f^b. With α providing a relative weighting of the two constraints, the resulting energy function is

$$E_{\text{reg}}(\mathcal{M},\mathcal{L}) = \sum_f\left(\underbrace{\sum_{(n,m)\in f} I\left(i(n)\neq i(m)\right)}_{\text{Label Constraint}} + \alpha\underbrace{\sum_{\text{Joints }(a,b)}\left\|\mathcal{T}_f^a\left(y_{f,ab}\right) - \mathcal{T}_f^b\left(y_{f,ab}\right)\right\|^2}_{\text{Joint Constraint}}\right). \tag{8}$$

5.3 Expectation-Maximization

Therefore, for each of the K frames, we minimize the above energy to get a the set of transformations per frame. We use an iterative Expectation Maximization (EM) like procedure to find a maximum likelihood solution of Eqn. 5.

During the E-step, the best alignment parameters from the previous iteration are used to compute an estimate of the posterior $P_{f,g}^{\text{old}}(m \mid y_{f,n})$ of mixture components using Bayes theorem [2]. This posterior is a matrix of dimension $N_g \times N_f$, where each matrix entry p_{mn} gives a conditional probability for a pair of points $(y_{f,n}, y_{g,m})$ from

frame f and g. Computing this matrix is intensive and would spend about 10 hours for two frames. It turns out that most entries are zero, and a relatively small number of pairs are actually close enough to yield a non-zero probability. Therefore, we only consider the N_{near} closest points in frame g for each point in frame f. As mentioned earlier, we use $N_{\text{near}} = 20$ which we empirically determined to be sufficient for our experiments. In addition, we compute the posterior matrix values only for four frames before and four frames after Y_f. These eight neighbor frames were enough for the registration, since the subject turns continuously in our experiments. Using these approximations, we can simultaneously optimize the alignment of all frames in a reasonable computation time.

During the M-step, the new alignment parameter values are found by minimizing the negative log-likelihood function, or more specifically, its upper bound Q which evaluates to:

$$Q\left(\mathcal{M},\mathcal{L}\right) = \sum_{f,g}\left(\sum_{n=1}^{N_f}\sum_{m=1}^{N_g} P_{f,g}^{\text{old}}\left(m \mid y_{f,n}\right) \frac{\left\|\mathcal{T}_f^{i(n)}(y_{f,n}) - \mathcal{T}_g^{j(m)}(y_{g,m})\right\|^2}{2{\sigma_{f,g}}^2}\right). \quad (9)$$

Here, the variances $\sigma_{f,g}^2$ in Eqn. 7 are continuously recomputed which is similar to an annealing procedure, in which the support of the Gaussians is reduced as the point sets get closer. Since the deformation parameters change after each M-step, the variances are recomputed after the M-step update.

The EM procedure converges to a local minimum of the negative log-likelihood function. To improve convergence, we optimize in two phases. In the first phase, we perform rigid registration by setting all labels $i(n),\ n = 1,\ldots,N_f$ to be the same within each frame Y_f. Thus, each frame is exactly one rigid part. We iterate the E and M steps until the transformations converge, yielding a rigid registration of all frames.

After completion of the first phase, we move to the second phase. This time, we run the EM procedure once again, but relaxing the restriction on the labels. Thus, while the E-step remains the same, we solve for both labels and transformations in the M-step. Iterating the E and M steps in this fashion completes the non-rigid registration.

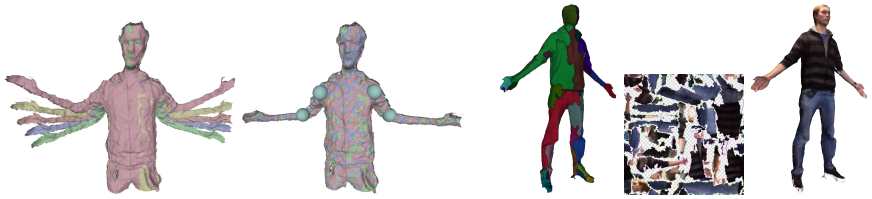
5.4 Non-Rigid M-step

Since the rigid registration is essentially the same as previous work, we refer to the references for further information [2][3][28]. Instead, we describe the non-rigid M-step optimization in a little more detail.

For the non-rigid registration with joint constraints, we need to minimize the whole term given in Eqn. 9, including both the labels and the transformations. Using the rigid transformation results from the first phase as the initial input of the non-rigid energy, we perform the M-step in two sub-steps iteratively until convergence.

Sub-Step 1. Fix labels $\mathcal{L}$ and solve for transformations $\mathcal{M}$. For E_{data}, the labels $i(n)$ and $j(m)$ are fixed, and the variables are the transformations M. For the regularization E_{reg}, only the joint constraint remains, since the labels are fixed. We use the same optimization method as the rigid registration, except that the joint constraints are added as additional terms, and we solve for more transformations simultaneously.

Sub-Step 2. Fix transformations $\mathcal{M}$ and solve for labels $\mathcal{L}$. The labels $i(n)$ are the variables that we are solving for, and this affects the location of the points because it



(a) Registration of upper body (b) Charts and textures on mesh

Fig. 4. (a) Challenging task for the non-rigid registration. Here, the joints are shown as blue balls. (b) Texture mapping.

changes which transformation is being applied. Therefore, the goal is to re-segment the points to yield a better registration. In E_{reg}, the joint constraint can be ingnored, and only the label constraint is left to ensure that the number of segmented parts in each frame is not too high. We solve the resulting discrete optimization problem using the α-expansion algorithm [29].

6 Texture Mapping

After the registration is complete, we reconstruct a closed mesh and apply textures to reproduce the person's appearance. In order to assemble a texture, we first compute a 2D parameterization for the reconstructed mesh. Since 3D surfaces cannot be mapped to 2D without any form of distortion, we segment the mesh in regions homeomorphic to discs that can be unfolded to the 2D domain without exceeding a certain threshold of distortion. These regions are called *charts* (Fig. 4b). We use the method of Sander et al. [30] that automatically segments the mesh into charts. Then we compute the 2D parameterization of each chart independently. To obtain a global non-overlapping parameterization of the whole mesh we pack the 2D parameterizations of each chart into a common *texture atlas* [31] (Fig. 4b).

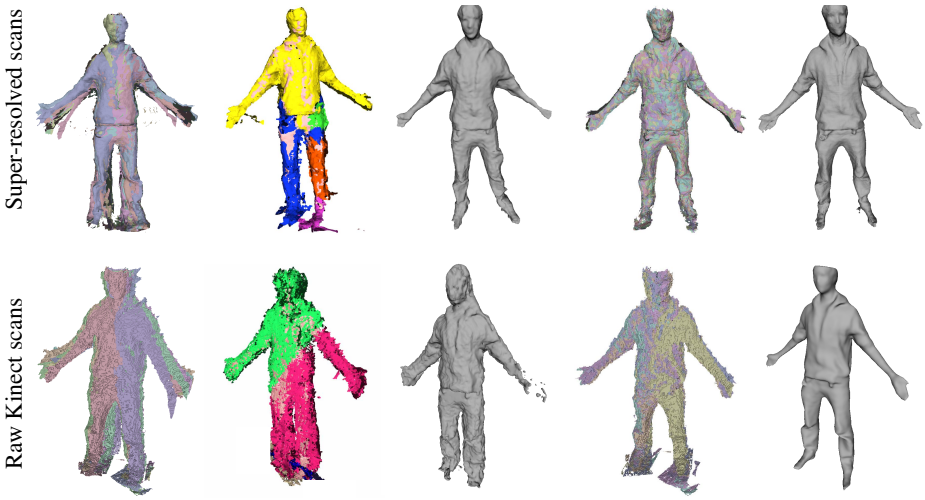
Based on the non-rigid global alignment, we build a textured depth map for each view. This map provides the combined depth and texture information (after non-rigid registration) for each pixel of the depth camera. For each pixel p in the texture map we compute a corresponding 3D point P on the reconstructed surface using the parameterization. We project P into each textured depth map and use the difference in depth to decide whether P is visible in the respective view. The final color of p is assembled as the weighted average of the texture information from each view where P is visible. We use the scalar product of the surface normal and the viewing direction as weight. Fig. 4b depicts our texture mapping result.

7 Results

First, we demonstrate the accuracy of our rigid global registration with a static Lion model (Fig. 5). A local ICP algorithm optimizes the transformations frame by frame,

(a) ICP result (b) Global rigid result (c) Ground truth (d) Our result with error plot (e) Error range

Fig. 5. Global rigid registration for a Lion model (height 18cm). The rightmost two figures show a comparison with ground truth.



(a) Rigid align (b) Using previous work [3] (c) Our result

Fig. 6. Comparing global non-rigid registration for super-resolution data (top row) and raw Kinect data (bottom row). We show the registration and the reconstructed mesh for both cases. The total energy (Eqn. 5) for (b)-top: 66.82, (b)-bottom: 128.93, (c)-top: 4.18, (c)-bottom: 23.74.

so the registration drifts and loops are not closed properly (shown in Fig. 5a from a top view). However, our result (Fig. 5b) computes the transformation based on a global energy function and properly solves the loop closure problem. We also quantitatively measure the reconstruction quality with a laser-scanned ground truth model. Fig. 5d shows the final reconstruction result and the accuracy of the reconstruction compares favorably to the ground truth Fig. 5c, as seen in the color-coded error plots (Fig. 5d, 5e). While the ICP registration yields an error above 1cm for 50% of the points, our result yields an error below 3mm for 90% of the points.

Next, we compare the non-rigid alignment algorithm to previous work. We perform two types of experiments: one testing the alignment of super-resolved scans, and one testing in extreme noise conditions using the raw Kinect scans (Fig. 6). By raw data, we mean the raw output from the Kinect depth sensor without the super-resolution step. In both cases, our method gives more accurate alignment results, especially in the arm

Fig. 7. Results of full 3D body scanning with a single Kinect.

and hand regions that have much movement. Therefore, our algorithm is more robust to noise in a high-noise data scenario.

The most important reason our method is successful is the probabilistic distance model. This works very well in the presence of noise. In addition, unlike previous work [3] which uses a sliding window to solve for the transformations, our method always takes all frames into account and can perform loop closure more effectively.

In most of our experiments, the real non-rigid displacement for each frame is not large. However, we demonstrate the results of a more challenging task in Fig. 4a. Here, the Kinect has captured five frames of a upper body human model with waving arms. The displacement between each frame is about 20 pixels, and the width of the arm is only about 10 pixels. Even in this case, the non-rigid algorithm can still find the correct registration and joint positions.

Finally, Fig. 7 shows our scanning results on five users. Our result reproduces the whole human structure well (especially the arms and legs), and can reconstruct detailed geometry such as the face structure and the wrinkles of the clothes. To evaluate the accuracy of the reconstruction, we compare the biometric measurements of the reconstructed human models with data from the actual people in Tab. 1. The values are the average absolute distance among eight people.

Frames	**S. R.**	**Rigid**	**Non-Rigid**	**Poisson**	**All**
36	28 sec	110 sec	620 sec	68 sec	826 sec (13.8 min)
Neck-Hip	**Shoulder W.**	**Arm L.**	**Leg L.**	**Waist Girth**	**Hip Girth**
2.1cm	1.0cm	2.3cm	3.1cm	3.2cm	2.6cm

Table 1. Runtime for each processing part and biometric error measurements. Timings and measurements are averaged among eight people.

We also show average runtime statistics in Tab. 1. The whole processing time for each model is about 14 minutes on average, using an Intel(R) Xeon 2.67GHz CPU with 12GB of memory. Note that 90% of the time in our method is used for computing closest points. Previous work on human body reconstruction [7] can only capture nearly naked human bodies and spends nearly one hour of computation time, and prior work on articulated registration [3] computes the registration frame by frame in K minimization steps, taking nearly two hours to compute.

8 Conclusion and Future Work

In this paper, we demonstrate that a full 3D human body model can be scanned with a single Kinect, which at first glance seems completely inappropriate for the task. Currently, the system requires the user to maintain a relatively awkward "T" pose while turning. Too much motion in the arms and legs can throw the registration off. And there are three main cases caused possilbe failure: 1) Not enough overlapping area for corresponding views. 2) The non-rigid movement segmentations are similar shape. 3) The segmentations change a lot in different views. To make this process more comfortable, we plan to investigate more sophisticated noise and deformation models to be able to handle larger user movements. In addition, we would like improve our algorithm to run in real-time. This would provide real-time feedback to the user about inaccurate or unfilled areas during the scanning process.

References

1. Microsoft. (http://www.xbox.com/en-US/kinect)
2. Cui, Y., Schuon, S., Chan, D., Thrun, S., Theobalt, C.: 3D shape scanning with a time-of-flight camera. In: IEEE Proc. CVPR. (2010)
3. Chang, W., Zwicker, M.: Global registration of dynamic range scans for articulated model reconstruction. ACM Trans. Graph. **30** (2011)
4. Henry, P., Krainin, M., Herbst, E., Ren, X., Fox, D.: RGB-D mapping: Using depth cameras for dense 3D modeling of indoor environments. In: In RGB-D: Advanced Reasoning with Depth Cameras Workshop in conjunction with RSS. (2010)
5. Newcombe, R.A., Kim, D., Kohli, P.: Kinectfusion: Real-time dense surface mapping and tracking. ISMAR (2011)
6. Cui, Y., Stricker, D.: 3D shape scanning with a kinect. ACM SIGGRAPH Posters (2011)
7. Weiss, A., Hirshberg, D., Black, M.J.: Home 3D body scans from noisy image and range data. Proc. IEEE Intl, Conf. Computer Vision (CVPR) (2011)
8. Anguelov, D., Srinivasan, P., Koller, D., Thrun, S., Rodgers, J., Davis, J.: SCAPE: shape completion and animation of people. ACM Trans. Graph. (SIGGRAPH) **24** (2005) 408–416

9. Tong, J., Zhou, J., Liu, L., Pan, Z., Yan, H.: Scanning 3d full human bodies using kinects. IEEE Transactions on Visualization and Computer Graphics (Proc. IEEE Virtual Reality), to appear. (2012)
10. Tomasi, C., Manduchi, R.: Bilateral filtering for gray and color images. In: Proceedings of the Sixth International Conference on Computer Vision. (1998)
11. Schuon, S., Theobalt, C., Davis, J., Thrun, S.: Lidarboost: Depth superresolution for ToF 3D shape scanning. Proc. CVPR (2009)
12. Besl, P.J., McKay, N.D.: A method for registration of 3-D shapes. IEEE PAMI. **14** (1992) 239–256
13. Benjemaa, R., Schmitt, F.: A solution for the registration of multiple 3D point sets using unit quaternions. In: Proc. ECCV. (1998) 34–50
14. Bergevin, R., Soucy, M., Gagnon, H., Laurendeau, D.: Towards a general multi-view registration technique. IEEE PAMI **18** (1996) 540–547
15. Weise, T., Wismer, T., Leibe, B., Gool, L.V.: Online loop closure for real-time interactive 3D scanning. Computer Vision and Image Understanding **115** (2011)
16. Mitra, N.J., Flory, S., Ovsjanikov, M., Gelfand, N., Guibas, L., Pottmann, H.: Dynamic geometry registration. In: Symposium on Geometry Processing. (2007) 173–182
17. Huang, Q., Adams, B., Wicke, M., Guibas, L.: Non-rigid registration under isometric deformations. Computer Graphics Forum **27** (2008) 1449–1457
18. Wand, M., Adams, B., Ovsjanikov, M., Berner, A., Bokeloh, M., Jenke, P., Guibas, L., Seidel, H.P., Schilling, A.: Efficient reconstruction of non-rigid shape and motion from real-time 3D scanner data. ACM Trans. Graph. (2009)
19. Popa, T., South-Dickinson, I., Bradley, D., Sheffer, A., Heidrich, W.: Globally consistent space-time reconstruction. Computer Graphics Forum (Proceedings of SGP) **29** (2010)
20. Li, H., Luo, L., Vlasic, D., Peers, P., Popović, J., Pauly, M., Rusinkiewicz, S.: Temporally coherent completion of dynamic shapes. ACM Trans. Graph. **31** (2012)
21. Tevs, A., Berner, A., Wand, M., Ihrke, I., Bokeloh, M., Kerber, J., Seidel, H.P.: Animation cartography - intrinsic reconstruction of shape and motion. (ACM Trans. Graph. (to appear))
22. Kazhdan, M., Bolitho, M., Hoppe, H.: Poisson surface reconstruction. In: Proceedings of the Fourth Eurographics Symposium on Geometry Processing. SGP '06 (2006) 61–70
23. Cui, Y., Schuon, S., Thrun, S., Stricke, D., Theobalt, C.: Algorithms for 3D shape scanning with a depth camera. IEEE T-PAMI (2012)
24. Weickert, J., Hagen, H.E.: Visualization and Processing of Tensor Fields. Springer, Berlin (2006)
25. Myronenko, A., Song, X., Carreira-Perpinan, M.: Non-rigid point set registration: Coherent Point Drift. NIPS **19** (2007) 1009
26. Murray, R.M., Li, Z., Sastry, S.S.: A Mathematical Introduction to Robotic Manipulation. CRC (1994)
27. Rosenhahn, B.: Pose estimation revisited. PhD thesis, Universität Kiel (2003)
28. Cui, Y., Hildenbrand, D.: Pose estimation based on geometric algebra. In: GraVisMa. (2009)
29. Boykov, Y., Kolmogorov, V.: An experimental comparison of min-cut/max-flow algorithms for energy minimization in vision. IEEE PAMI **26** (2004) 1124–1137
30. Sander, P.V., Wood, Z.J., Gortler, S.J., Snyder, J., Hoppe, H.: Multi-chart geometry images. In: Proceedings of the Eurographics/ACM SIGGRAPH Symposium on Geometry Processing. SGP (2003) 146–155
31. Nöll, T., Stricker, D.: Efficient packing of arbitrary shaped charts for automatic texture atlas generation. Computer Graphics Forum **30** (2011) 1309–1317