Rapid Avatar Capture and Simulation using Commodity Depth Sensors

Ari Shapiro*[1], Andrew Feng†[1], Ruizhe Wang‡[2], Hao Li§[2], Mark Bolas¶[1], Gerard Medioni‖[2], and Evan Suma**[1]

[1]Institute for Creative Technologies, University of Southern California
[2]University of Southern California

Abstract

We demonstrate a method of acquiring a 3D model of a human using commodity scanning hardware and then controlling that 3D figure in a simulated environment in only a few minutes. The model acquisition requires 4 static poses taken at 90 degree angles relative to each other. The 3D model is then given a skeleton and smooth binding information necessary for control and simulation. The 3D models that are captured are suitable for use in applications where recognition and distinction among characters by shape, form or clothing is important, such as small group or crowd simulations, or other socially oriented applications. Due to the speed at which a human figure can be captured and the low hardware requirements, this method can be used to capture, track and model human figures as their appearances changes over time.

Keywords: image capture, animation, avatar

1 Introduction

Recent advances in low-cost scanning have enabled the capture and modeling of real-world objects into a virtual environment in 3D. For example, a table, a room, or work of art can be quickly scanned, modeled and displayed within a virtual world with a handheld, consumer scanner. There is great value to the ability to quickly and inexpensively capture real-world objects and create their 3D counterparts. While numerous generic 3D models are available for low- or no-cost for use in 3D environments and virtual worlds, it is unlikely that such acquired 3D model matches the real object to a reasonable extent without individually modeling the object. In addition, the ability to capture specific objects that vary from the generic counterparts is valuable for recognition, interaction and comprehension within a virtual world. For example, a real table could have a noticeable scratch, design, imperfection or size that differs greatly from a stock 3D model of a table. These individual markers can serve as landmarks for people interacting with the virtual scene.

The impact of recognizing living objects in a virtual environment can be very powerful, such as the effect of seeing a relative, partner or even yourself in a simulation. However, living objects present simulation challenges due to their dynamic nature. Organic creatures, such as plants, can be difficult to scan due to their size and shape, which requires high levels of details and stable scanning environments. Similarly, other living objects such as people or animals, can be scanned, but require much more complex

*shapiro@ict.usc.edu
†feng@ict.usc.edu
‡ruizewa@gmail.com
§hao@hao-li.com
¶bloas@ict.usc.edu
‖medioni@usc.edu
**suma@ict.usc.edu

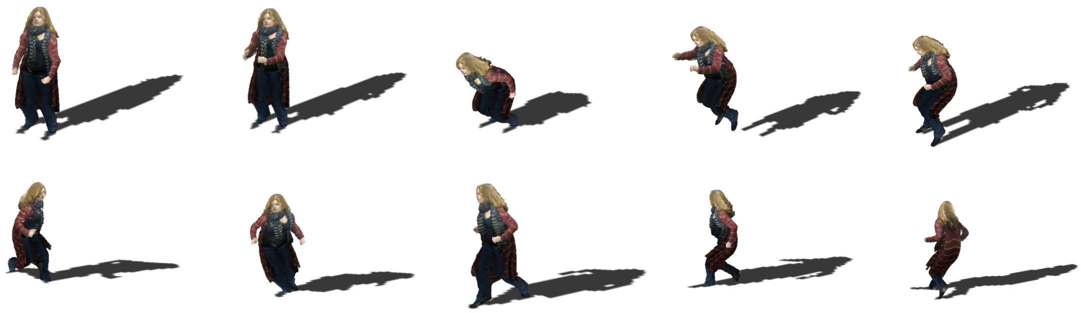


Figure 1: The 3D models captured in our system can be readily applied in real-time simulation to perform various behaviors such as jumping and running with the help of auto-rigging and animation retargeting.

models to model motion and behavior. In addition, the particular state of the living object can vary tremendously; an animal may grow, a plant can blossom flowers, and a person can wear different clothes, inhale or exhale, as well as gain or lose weight. Thus, capturing a moment in time of a living object is usually not sufficient for its representation in dynamic environments, where the 3D representation of that living object is expected to breath, move, grow as well as respond to interaction in non-trivial ways.

In this work, we demonstrate a process for capturing human subjects and generating digital characters from those models using commodity scanning hardware. Our process is capable of capturing a human subject using still four poses, constructing a 3D model, then registering it and controlling it within an animation system within minutes. The digital representation that our process is able to construct is suitable for use in simulations, games and other applications that use virtual characters. Our technique is able to model many dynamic aspects of human behavior (see Figure 1). As shown in Figure 2, our main contribution in this work is a near-fully automated, rapid, low-cost end-to-end system for capture, modeling and simulation of a human figure in a virtual environment that requires no expert intervention.

2 Related Work

2.1 3D Shape Reconstruction

3D shape reconstruction has been extensively explored, among which the 3D shape reconstruction of human subjects is of specific interest to computer vision and computer graphics, with its potential applications in recognition, animation and apparel design. With the availability of low-cost 3D cameras (e.g., Kinect and Primesense), many inexpensive solutions for 3D human shape acquisition have been proposed. The work by [1] employs three Kinect devices and a turntable. As the turntable rotates, multiple shots are taken with the three precalibrated Kinect sensors to cover the entire body. All frames are registered in a pairwise non-rigid manner using the Embedded Deformation Model [2] and loop-closure is explicitly addressed at the final stage. The work done in [3] utilizes two Kinect sensors in front of the self-turning subject. The subject stops at several key poses and the captured frame is used to update the online model. Again the dynamic nature of the turning subject is considered under the same non-rigid registration framework [2] and the loop is implicitly closed.

More recently, solutions which utilize only a single 3D sensor have been proposed, and this allows for home-based scanning and applications. The work in [4] asks the subject to turn in front of a fixed 3D sensor and 4 key poses are uniformly sampled to perform shape reconstruction. The 4 key poses are registered in a top-bottom-top fashion, assuming an articulated tree structure of human body. Their reconstructed model, however, suffers from a low-resolution issue at a distance. To overcome the resolution issue, KinectAvatar [5] considers color constraints among consecutive frames for super-resolution. They register all super-resolution

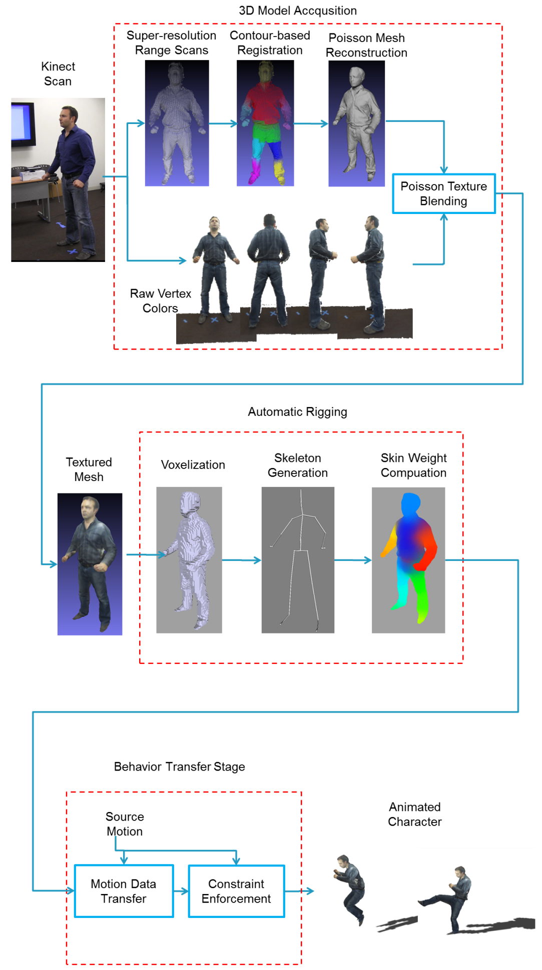


Figure 2: The overall work flow of our fast avatar capture system.

frames under a probabilistic framework. More recently, the work in [6] asks the subject to come closer and obtain a super-resolution scan at each of 8 key poses. The 8 key poses are then aligned in a multi-view non-rigid manner to generate the final model. Inspired by their work, we follow the same idea of asking the subject to get closer, but employ a different super-resolution scheme. Unlike [6] where they merge all range scans using the Iterative Closest Point (ICP) algortihm [7] along with the Poisson Surface Reconstruction algorithm [8], we use the KinectFusion algorithm [9] which incrementally updates an online volumetric model.

All these works capture the static geometry of human subjects, and additional efforts are necessary to convert the static geometry into an animated virtual character. The research works [10, 11] focus on capturing the dynamic shapes of an actor's full body performance. The capturing sessions usually require a dedicated setup with multiple cameras and are more expensive than capturing only the static geometry. The resulting dynamic geometries can be played back to produce the animations of the scanned actor. The work in [12] combines dynamic shapes from multiple actors to form a shape space. The novel body deformations are driven by motion capture markers and can be synthesized based on an actor's new performance.

Other research has created a database of people that show the diversity of shape, size and posture in a small population of shape, size and posture [13]. The data set has be employed for human body modeling by fitting the model to input range scans of subject of interest [14]. This data set has also been used to manipulate a scanned human model by modifying the models proportions according to the data [15].

2.2 *Automatic Rigging and Retargeting*

While it is relatively easy to obtain static 3D character models, either from the internet or through 3D scanning, it requires much more efforts to create an animated virtual character. A 3D model needs to be rigged with a skeleton hierarchy and appropriate skinning weights. Traditionally, this process needs to be done manually and is time consuming even for an experienced animator. An automatic skinning method is proposed in [16] to reduce the manual efforts of rigging a 3D model. The method produces reasonable results but requires a connected and watertight mesh to work. The method proposed by [17] complements the previous work by automatically skinning a multi-component mesh. It works by detecting the boundaries between disconnected components to find potential joints. Thus the method is suitable for rigging the mechanical characters that usually consist of many components. Other rigging algorithms can include manual annotation to identify important structures such as wrists, knees and neck [18].

Recent work has shown the capability of capturing a human figure and placing that character into a simulation using 48 cameras with processing time on the order of two hours [19]. Our method differs in that we use a single commodity camera and scanner and our processing time takes a few minutes. While this introduces a tradeoff in visual quality, the minimal technical infrastructure required makes our approach substantially more accessible to a widespread audience. In addition, our method requires no expert intervention during the rigging and animation phases.

3 3D Model Reconstruction

We propose a convenient and fast way to acquire accurate static 3D human models of different shapes by the use of a single commodity hardware, e.g., Kinect. The subject turns in front of the Kinect sensor in a natural motion, while staying static at 4 key poses, namely front, back and two profiles, for approximately 10 seconds each. For each key pose, a super-resolution range scan is generated as the Kinect device, controlled by a built-in motor, moves up and down (Sec 3.1). The 4 super-resolution range scans are then aligned in a multi-view piecewise rigid manner, assuming small articulations between them. Traditional registration algorithms (e.g., Iterative Closest Point [7]), which are based on the *shape coherence*, fail in this scenario because the overlap between consecutive frames is very small. Instead, we employ *contour coherence* (Sec 3.2) and develop a contour-based registration method [20], which iteratively minimizes the distance between the closest points on the predicted and observed contours (Sec 3.3). For more details on using *contour coherence* for multi-view registration of range scans, please refer to [20]. In this paper, we summarize their method and give a brief introduction. At the final stage, the 4 aligned key poses are processed to generate a water-tight mesh model using the Poisson Surface Reconstruction algorithm [8]. The corresponding texture information of the 4 super-resolution range scans are inferred using the Poisson Texture Blending algorithm [21] (Sec 3.4).

3.1 Super-resolution Range Scan

Given the field of view of the Kinect sensor, the subject must stand 2 meters away in order to cover the full body while turning in front of the device. The data is heavily quantized at that distance (Fig 3(b)), thus produces a poor quality scan, which results in a coarse model after integration. Here, instead, we ask the subject to come closer and stay as rigid as possible at the 4 key poses, while the Kinect device scans up and down to generate a super-resolution range scan. Each pose takes 10 seconds and approximately 200 frames are merged using the KinectFusion algorithm [9] (Fig 3(a))). This process greatly improves the quality of the input and allows us to capture more details, such as wrinkles of clothes and face as shown in Fig 3. It is worth mentioning that the ground is removed by using the RANSAC algorithm [22], assuming that the subject of interest is the only thing in the sensor's predefined capture range.

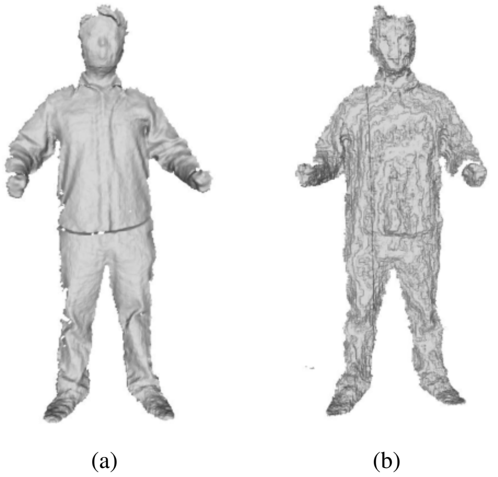
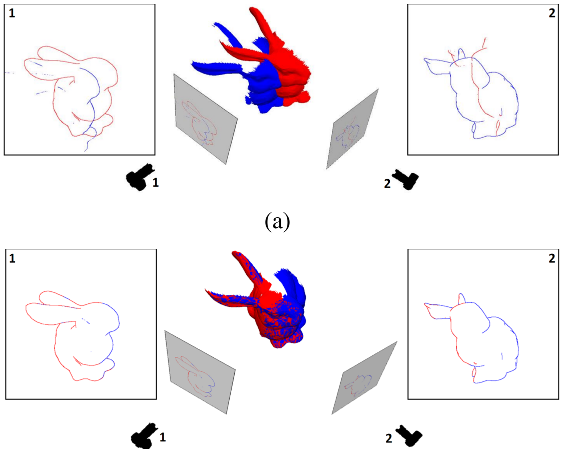


Figure 3: (a) Super-resolution range scans after integrating approximately 200 frames using the KinectFusion algorithm (b) Low-resolution single range scan at the distance of 2 meters

3.2 Contour Coherence as a Clue

The amount of overlap between two consecutive super-resolution range scans is limited as they are 90^o apart (i.e.wide baseline). As such, traditional *shape coherence* based methods (e.g., ICP and its variants [23]) fail, as it is hard to establish the point-to-point correspondences on two surfaces with small overlap.

An example of two wide baseline range scans of the Stanford bunny with approximately 35%



(a)

(b)

Figure 4: (a) Two roughly aligned wide baseline 2.5D range scans of the Stanford Bunny with the observed and predicted apparent contours extracted. The two meshed points cloud are generated from the two 2.5D range scans respectively (b) Registration result after maximizing the **contour coherence**

overlap is given in Fig 4(a). Traditional methods fail, as most closest-distance correspondences are incorrect.

While the traditional notion of *shape coherence* fail, we propose the concept of *contour coherence* for wide baseline range scan registration. *Contour coherence* is defined as the agreement between the observed apparent contour and the predicted apparent contour. As shown in Fig 4(a), the observed contours extracted from the original 2.5D range scans, i.e.red lines in image 1 and blue lines in image 2, do not match the corresponding predicted contours extracted from the projected 2.5D range scans, i.e.blue lines in image 1 and red lines in image 2. We maximize *contour coherence* by iteratively finding closest correspondences among apparent contours and minimizing their distances. The registration result is shown in Fig 4(b) with the *contour coherence* maximized and two wide baseline range scans well aligned. The *contour coherence* is robust in the presence of wide baseline in the sense that, no matter the amount of overlap between two range scans, only the shape area close to the predicted contour generator is considered when building correspondences on the contour, thus avoiding the search for correspondences over the entire shape.

3.3 Contour Coherence based Registration Method

We apply the notion of *contour coherence* to solve the registration problem of 4 super-resolution range scans with small articulations. For simplicity, we start the discussion with the contour-based rigid registration of 2 range scans. As shown in Fig 4(a), the observed contour and the predicted contour do not match. In order to maximize the *contour coherence*, we iteratively find the closest pairs of points on two contours and minimize their distances. Assume point $\mathbf{u} \in \mathbb{R}^2$ is on predicted contour in image 1 of Fig 4(a) (i.e.blue line) and point $\mathbf{v} \in \mathbb{R}^2$ is its corresponding closest point on the observed contour in image 1 (i.e.red line), we minimize their distance as

$$\|\mathbf{v} - \mathcal{P}_1(T_1^{-1} T_2 \mathcal{V}_2(\tilde{\mathbf{u}}))\|, \tag{1}$$

where $\tilde{u}$ is the corresponding pixel location in image 2 of $\mathbf{u}$, $\mathcal{V}_2$ maps the pixel location $\tilde{u}$ to its 3D location in the coordinate system of camera 2, T_1 and T_2 are the camera to world transformation matrices of camera 1 and 2 respectively, and $\mathcal{P}_1$ is the projection matrix of camera 1. Assuming known $\mathcal{P}_1$ and $\mathcal{P}_2$, we iterate between finding all closest contour points on image 1 and 2 and minimizing the sum of their distances (Eq 1) to update the camera poses T_1 and T_2 until convergence. We use quaternion to represent the rotation part of T and Levenberg-Marquardt algorithm to solve for the minimization as it is nonlinear in parameters. It is worth mentioning that minimizing Eq 1 updates T_1 and T_2 at the same time, and this enables us to perform multi-view rigid registration in the case of 3 or more frames.

The extension from rigid registration to piecewise rigid registration is quite straightforward. Each segment (i.e., segmented body part) is considered rigid, and all the rigid segments are linked by a hierarchical tree structure in the case of body modeling. We again iteratively find the closest pairs on contours between all corresponding body segments and minimize the sum of their distances.

A complete pipeline of our registration method is given in Fig 5. First, the 4 super-resolution range scans are initialized by assuming a 90^o rotation between consecutive frames (Fig 6(a)). Second, they are further aligned by the multi-view rigid registration method considering the whole body as rigid (Fig 6(b)). While the translation part of the camera pose is not well estimated by the initialization procedure, it is corrected by the multi-view rigid registration step. As indicated by the red boxes, however, the small articulations between frames still remain unresolved under the rigid assumption. Third, the front pose is roughly segmented into 9 body parts in a heuristic way (Fig 6(c)). Fourth, we iteratively propagate the segmentation to other frames, find closest pairs on contours between corresponding rigid body parts, and minimize their distances to update the camera poses, as well as the human poses of each frame (Fig 6(d)).

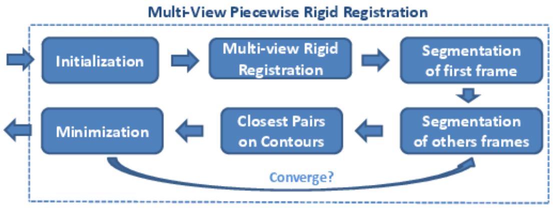

Multi-View Piecewise Rigid Registration
Initialization
Multi-view Rigid Registration
Segmentation of first frame
Segmentation of others frames
Closest Pairs on Contours
Minimization
Converge?

Figure 5: General pipeline of our registration method

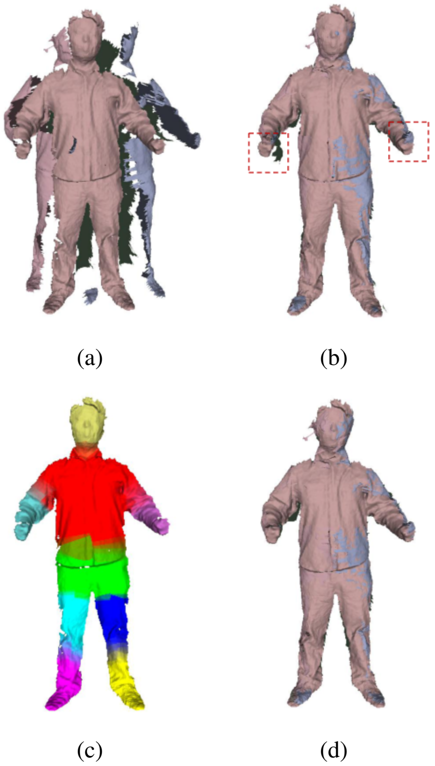


Figure 6: (a) 4 super-resolution range scans after initialization (b) 4 super-resolution range scans after multi-view rigid registration, with red boxes indicating unresolved small articulations under the rigid assumption (c) Rough segmentation of the front pose (d) 4 super-resolution range scans after multi-view piecewise rigid registration

3.4 *Water-tight Mesh Model with Texture*

At this point, we have aligned all four super scans to produce a point cloud with normal vectors. Poisson mesh reconstruction [8] is used to obtain a watertight mesh from the point clouds. The Kinect camera also captures the color information from the scanned person when generating the superscans at each pose. For each superscan, we also store a color image corresponding to the range scan and combine the color images to produce the texture for the watertight mesh. We follow a similar procedure as in [6] to corrode the color images and remove unreliable pixels. The corroded color images are then transferred onto the superscans as vertex colors to produce color meshes before going through the registration process. Finally, these aligned color meshes are used to texture the watertight mesh generated from Poisson reconstruction. We apply the Poisson texture blending algorithm in [21] to filling out the gaps and holes in the texture and produce the final color mesh.

4 Resolution Independent Automatic Rigging

Animating a 3D character model usually requires a skeletal structure to control the movements. Our system automatically builds and adapts a skeleton to the 3D scanned character. Thus, it can later apply the rich sets of behavior on the character through motion retargeting.

The auto-rigging method in our system is similar to the one proposed in [16]. The method builds a distance field from the mesh and uses the approximate medial surface to extract the skeletal graph. The extracted skeleton is then matched and refined based on the template skeleton. The method is automatic and mostly robust, but it requires a watertight and single component mesh to work correctly. This poses a big restriction on the type of 3D models the method can be applied to. For example, the production meshes usually come with many props and thus have multiple components. On the other hand, the mesh produced from range scans tend to contain holes, non-manifold geometry, or other topological artifacts that require additional clean-up. Moreover, the resulting mesh produced through the super-resolution scans usually consists of hundreds of thousands of vertices. Such high resolution meshes would cause the auto-rigging method fail during optimization process to build the skeleton. To alleviate this limit, we proposed a modified method that works both for generic production models and large meshes.

Our key idea is that the mesh could be approximated by a set of voxels and the distance field could be computed using the voxels. The voxels are naturally free from any topological artifacts and are easy to processed. It is done by first converting the mesh into voxels using depth buffer carving in all positive and negative x,y, and z directions. This results in 6 depth images that can be used to generate the voxelization of the original mesh. Although most small holes in the original mesh are usually removed in the resulting voxels due to discretization, some holes could still remain after the voxelization. To removing the remaining holes, we perform the image hole filling operation in the depth images to fill up the small empty pixels. After voxelization, we select the largest connected component and use that as the voxel representation for the mesh. The resulting voxels are watertight and connected and can be converted into distance field to construct the skeleton. Figure 2 demonstrates the process of converting the original mesh into voxel representation to produce the skeleton hierarchy and skinning weights.

The voxel representation is only an approximation of the original mesh. Therefore the resulting distance field and consequently the skeleton could be different from the one generated with the original mesh. In our experiments, we found the resulting skeletons tend to be very similar as shown in Figure 7 and do not impact the overall animation quality in the retargeting stage. Once we obtain the skeleton, the skinning weights can be computed using the original mesh instead of the voxels since the weight computation in [16] does not rely on the distance field. Alternatively, the skinning weights can be computed using the techniques in [24], which uses voxels to approximate the geodesic distance for computing bone influence weights. Thus we can naturally apply their algorithm using our resulting voxels and skeleton to produce higher quality smooth bindings.

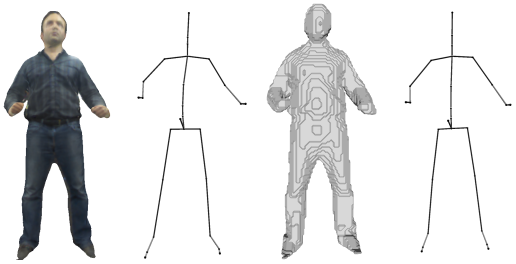


Figure 7: The voxelization produces the skeleton similar to the one extracted from original mesh. Left : original mesh and its skeleton. Right : voxel representation of original mesh and its corresponding skeleton.

5 Behavior Transfer

The behavior transfer stage works by retargeting an example motion set from our canonical skeleton to the custom skeleton generated from automatic rigging. Here we use the method from [25] to perform motion retargeting. The retargeting process can be separated into two stages. The first stage is to convert the joint angles encoded in a motion from our canonical skeleton to the custom skeleton. This is done by first recursively rotating each bone segment in target skeleton to match the global direction of that segment in source skeleton at default pose so that the target skeleton is adjusted to have the same default pose as the source skeleton. Once the default pose is matched, we address the discrepancy between their local frames by adding suitable pre-rotation and post-rotation at each joint in target skeleton. These pre-rotation and post-rotation are then used to convert the joint angles from source canonical skeleton to the target skeleton.

The second stage is using inverse kinematics to enforce various positional constraints such as foot positions to remove motion artifacts such as foot sliding. The inverse kinematic method we use is based on damped Jacobian pseudo-inverse [26]. We apply this IK method at each motion frame in the locomotion sequences to ensure the foot joint is in the same position during the foot plant stage. After the retargeting stage, the acquired 3D skinned character can be incorporated into the animation simulation system to execute a wide range of common human-like behaviors such as walking, gesturing, and etc.

6 Applications

6.1 3D Capture for Use in Games and Simulation

We demonstrate our method by showing the capture and processing, registration and subsequent simulation of a human figure in our accompanying video and in Figure 8 below. The construction of a 3D model take approximately 4 minutes, and the automatic rigging, skinning and registration of a deformable skeleton takes approximately 90 seconds. Models typically contain between 200k and 400k vertices, and 400k to 800k faces. Simulation and control of the character is performed in real time using various animations and procedurally-based controllers for gazing and head movement. The 3D models captured in this way are suitable for use in games where characters need to be recognizable from a distance, but do not require face-to-face or close interactions.

6.2 Temporal Avatar Capture

Since our method enables the capture of a 3D character without expert assistance and uses commodity hardware, it is economically feasible to perform 3D captures of the same subject over a protracted period of time. For example, a 3D model could be taken every day of the same subject, which would reflect their differences in appearance over time. Such captures would reflect changes in appearance such as hair style or hair color, clothing, or accessories worn. In addition, such temporal captures could reflect personal changes such as growth of facial hair, scars, weight changes and so on. Such temporal information could be analyzed to determine clothing preferences or variations in appearance.

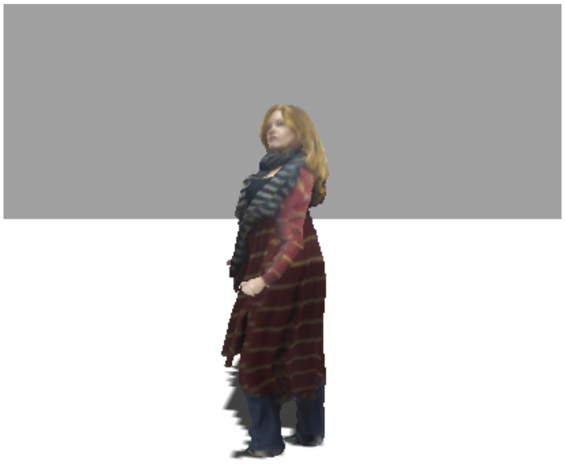


Figure 8: A representative captured character from scan containing 306k vertices and 613k faces. Note that distinguishing characteristics are preserved in the capture and simulation, such as hair color, clothing style, height, skin tone and so forth.

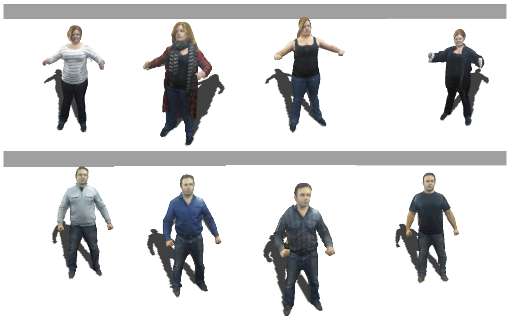


Figure 9: Models generated from captures over a period of 4 days. Note changes and commonality in clothing, hair styles, and other elements of appearance.

Note that our method will generate a skeleton for each 3D model. Thus avatars of the same subject will share the same topology, but have differing bone lengths.

6.3 Crowds

Many applications that use virtual crowds require tens, hundreds or thousands of characters to populate the virtual space. Research has experimented with saliency to show the needed variation in traditionally modeled characters to model a crowd [27] as well as the number of variations needed [28]. By reducing the cost of constructions of 3D characters, crowd members can be generated from a population of capture subjects rather than through traditional 3D means.

7 Discussion

We have demonstrated a technique that allows the capture and simulation of a human figure into a real time simulation without expert intervention in a matter of a few minutes.

7.1 Limitations

The characters generated are suitable for applications where recognizability of and distinction among the virtual characters is important. In the course of our experiments, we have found the virtual characters to be recognizable to those familiar with the subjects. The characters are not suitable for close viewing or in simulations where face details are needed, such as conversational agent or talking head applications. Higher levels of detail are needed for areas such as the face and hands before other models of synthetic motion, such as emotional expression, lip syncing or gesturing could be used. Additionally, our method makes no distinction between the body of the capture subject and their clothing. Thus, bulky clothing or accessories could change the skeletal structure of the virtual character. Also, the behaviors associated with the characters are retargeted from sets of motion data and control algorithms, but are not generated from movements or motion gleaned from the subject itself. Thus, motion transferred to all captured subjects shares the same characteristics, differing only by the online retargeting algorithm which accommodates differently sized characters. This homogeneity can be partially circumvented by including variations in the set of motion data, such as differing locomotion or gesturing sets for male and female characters. For future work, we plan on extracting movement models from the capture subjects in order to further personalize their virtual representation.

References

[1] Jing Tong, Jin Zhou, Ligang Liu, Zhigeng Pan, and Hao Yan. Scanning 3d full human bodies using kinects. *Visualization and Computer Graphics, IEEE Transactions on*, 18(4):643–650, 2012.

[2] Robert W Sumner, Johannes Schmid, and Mark Pauly. Embedded deformation for shape manipulation. In *ACM Transactions on Graphics (TOG)*, volume 26, page 80. ACM, 2007.

[3] Ming Zeng, Jiaxiang Zheng, Xuan Cheng, and Xinguo Liu. Templateless quasi-rigid shape modeling with implicit loop-closure. In *Computer Vision and Pattern Recognition (CVPR), 2013 IEEE Conference on*, pages 145–152. IEEE, 2013.

[4] Ruizhe Wang, Jongmoo Choi, and Gérard Medioni. Accurate full body scanning from a single fixed 3d camera. In *3D Imaging, Modeling, Processing, Visualization and Transmission (3DIMPVT), 2012 Second International Conference on*, pages 432–439. IEEE, 2012.

[5] Yan Cui, Will Chang, Tobias Nöll, and Didier Stricker. Kinectavatar: fully automatic body capture using a single kinect. In *Computer Vision-ACCV 2012 Workshops*, pages 133–147. Springer, 2013.

[6] Hao Li, Etienne Vouga, Anton Gudym, Linjie Luo, Jonathan T. Barron, and Gleb Gusev. 3d self-portraits. *ACM Transactions on Graphics (Proceedings SIGGRAPH Asia 2013)*, 32(6), November 2013.

[7] Yang Chen and Gérard Medioni. Object modelling by registration of multiple range images. *Image and vision computing*, 10(3):145–155, 1992.

[8] Michael Kazhdan, Matthew Bolitho, and Hugues Hoppe. Poisson surface reconstruction. In *Proceedings of the fourth Eurographics symposium on Geometry processing*, 2006.

[9] Richard A Newcombe, Andrew J Davison, Shahram Izadi, Pushmeet Kohli, Otmar Hilliges, Jamie Shotton, David Molyneaux, Steve Hodges, David Kim, and Andrew Fitzgibbon. Kinectfusion: Real-time dense surface mapping and tracking. In *Mixed and augmented reality (ISMAR), 2011 10th IEEE international symposium on*, pages 127–136. IEEE, 2011.

[10] Chenglei Wu, Carsten Stoll, Levi Valgaerts, and Christian Theobalt. On-set performance capture of multiple actors with a stereo camera. In *ACM Transactions on Graphics (Proceedings of SIGGRAPH Asia 2013)*, volume 32, November 2013.

[11] Daniel Vlasic, Pieter Peers, Ilya Baran, Paul Debevec, Jovan Popovic, Szymon Rusinkiewicz, and Wojciech Matusik. Dynamic shape capture using multi-view photometric stereo. In *In ACM Transactions on Graphics*, 2009.

[12] Dragomir Anguelov, Praveen Srinivasan, Daphne Koller, Sebastian Thrun, Jim Rodgers, and James Davis. Scape: Shape completion and animation of people. In *ACM SIGGRAPH 2005 Papers*, SIGGRAPH '05, pages 408–416, New York, NY, USA, 2005. ACM.

[13] Dragomir Anguelov, Praveen Srinivasan, Daphne Koller, Sebastian Thrun, Jim Rodgers, and James Davis. Scape: shape completion and animation of people. In *ACM Transactions on Graphics (TOG)*, volume 24, pages 408–416. ACM, 2005.

[14] Alexander Weiss, David Hirshberg, and Michael J Black. Home 3d body scans from noisy image and range data. In *Computer Vision (ICCV), 2011 IEEE International Conference on*, pages 1951–1958. IEEE, 2011.

[15] Arjun Jain, Thorsten Thormählen, Hans-Peter Seidel, and Christian Theobalt. Moviereshape: Tracking and reshaping of humans in videos. *ACM Trans. Graph. (Proc. SIGGRAPH Asia 2010)*, 29(5), 2010.

[16] Ilya Baran and Jovan Popović. Automatic rigging and animation of 3d characters. *ACM Trans. Graph.*, 26(3), July 2007.

[17] Gaurav Bharaj, Thorsten Thormählen, Hans-Peter Seidel, and Christian Theobalt. Automatically rigging multi-component characters. *Comp. Graph. Forum (Proc. Eurographics 2012)*, 30(2), 2011.

[18] Mixamo auto-rigger, 2013. http://www.mixamo.com/c/auto-rigger.

[19] xxarray demo at ces, 2014. http://gizmodo.com/nikon-just-put-me-in-a-video-game-and-it-was-totally-in-1497441443.

[20] Ruizhe Wang, Jongmoo Choi, and Gérard Medioni. 3d modeling from wide baseline range scans using contour coherence. In *Computer Vision and Pattern Recognition (CVPR), 2014 IEEE Conference on*. IEEE, 2014.

[21] Ming Chuang, Linjie Luo, Benedict J Brown, Szymon Rusinkiewicz, and Michael Kazhdan. Estimating the laplace-beltrami operator by restricting 3d functions. In *Computer Graphics Forum*, volume 28, pages 1475–1484. Wiley Online Library, 2009.

[22] Martin A Fischler and Robert C Bolles. Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. *Communications of the ACM*, 24(6):381–395, 1981.

[23] Szymon Rusinkiewicz and Marc Levoy. Efficient variants of the icp algorithm. In *3-D Digital Imaging and Modeling, 2001. Proceedings. Third International Conference on*, pages 145–152. IEEE, 2001.

[24] Olivier Dionne and Martin de Lasa. Geodesic voxel binding for production character meshes. In *Proceedings of the 12th ACM SIGGRAPH/Eurographics Symposium on Computer Animation*, SCA '13, pages 173–180, New York, NY, USA, 2013. ACM.

[25] Andrew Feng, Yazhou Huang, Yuyu Xu, and Ari Shapiro. Fast, automatic character animation pipelines. *Computer Animation and Virtual Worlds*, pages n/a–n/a, 2013.

[26] Samuel R. Buss. Introduction to inverse kinematics with jacobian transpose, pseudoinverse and damped least squares methods. Technical report, IEEE Journal of Robotics and Automation, 2004.

[27] Rachel McDonnell, Michéal Larkin, Benjamín Hernández, Isaac Rudomin, and Carol O'Sullivan. Eye-catching crowds: Saliency based selective variation. *ACM Trans. Graph.*, 28(3):55:1–55:10, July 2009.

[28] Rachel McDonnell, Michéal Larkin, Simon Dobbyn, Steven Collins, and Carol O'Sullivan. Clone attack! perception of crowd variety. *ACM Trans. Graph.*, 27(3):26:1–26:8, August 2008.