Capture of Arm-Muscle Deformations using a Depth-Camera

Nadia Robertini
University of Saarland
Saarbruecken, Germany
nadia.robertini@gmail.com

Thomas Neumann
HTW
Dresden, Germany
tneumann@htw-dresden.de

Dr. Kiran Varanasi*
Max-Planck-Institut Informatik
Saarbruecken, Germany
varanasi@mpi-inf.mpg.de

Prof. Dr. Christian Theobalt
Max-Planck-Institut Informatik
Saarbruecken, Germany
theobalt@mpi-inf.mpg.de

ABSTRACT

Modeling realistic skin deformations due to underneath muscle bulging has a wide range of applications in medicine, entertainment and art. Current acquisition systems based on dense markers and multiple synchronized cameras are able to record and reproduce fine-scale skin deformations with sufficient quality. However, the complexity and the high cost of these systems severely limit their applicability. In this paper, we propose a method for reconstructing fine-scale arm muscle deformations using the Kinect depth camera. The captured data from the depth camera has no temporal contiguity and suffers from noise and sensory artifacts, and thus unsuitable by itself for potential applications in visual media production or biomechanics. We process noisy depth input to obtain spatio-temporally consistent 3D mesh reconstructions showing fine-scale muscle bulges over time. Our main contribution is the incorporation of statistical deformation priors into the spatio-temporal mesh registration progress. We obtain these priors from a previous dataset of a limited number of physiologically different actors captured using a high fidelity acquisition setup, and these priors help provide a better initialization for the ultimate non-rigid surface refinement that models deformations beyond the range of the previous dataset. Thus, our method is an easily scalable framework for bootstrapping the statistical muscle deformation model, by extending the set of subjects through a Kinect based acquisition process. We validate our spatio-temporal surface registration method on several arm movements performed by people of different body shapes.

1. INTRODUCTION

Reconstructing high-quality muscle deformations in a non-intrusive manner is a key problem in the areas of entertainment, human biomechanics and human-centered design. Depth cameras like Microsoft Kinect, that recently appeared on the consumer market, provide a relatively cheap and easy mechanism to capture 3D images. However, the captured depth images have significant artifacts due to sensor noise, occlusions and the lack of temporal contiguity in capture. As such, these are unusable for researchers in biomechanics or human-computer interfaces who want to build accurate user-specific models of muscle deformations. In the current paper, we propose a method for reconstructing high quality and temporally aligned 3D meshes from depth images captured by the Kinect camera, for the human shoulder-arm region. Thus our method bridges an important gap and enlarges the research scope for many areas concerned with modeling muscle deformations, making them capitalize on cheap consumer hardware. For example, realistic virtual humans and their muscle movements can be modeled for visual media production in a cost-effective manner, through the use of cheap acquisition systems and fewer hours of manual work by artists. Sports scientists and medical practitioners can observe the physiological action of muscles on a day-to-day basis and provide personalized advice to sportsmen and patients without the use of expensive and intrusive sensors.

At present, modeling realistic muscle deformations of virtual humans remains a highly labour intensive task. Commercial systems use specialized kinematic rigs for virtual characters, which have hundreds of control parameters to derive localized bulging effects on a fine scale by approximating them with a set of bones. As an alternative, bio-mechanically based simulation of human anatomy and physics-based muscle deformation can be performed. However, this remains computationally very expensive and such rigs are hard to control and adapt to new characters. Thus, data-driven simulation methods have been developed in order to overcome some of these limitations. Based on a training set of artist-given deformation examples or 3D scans acquired directly from the real world, an artistic interface can be developed that is simple to use, but which reproduces complex muscle deformation behavior as visible in the training set. These data-driven simulation methods bridge an important gap in the artistic production pipeline. However, acquiring

*Currently at Technicolor Research & Innovation, Rennes, France

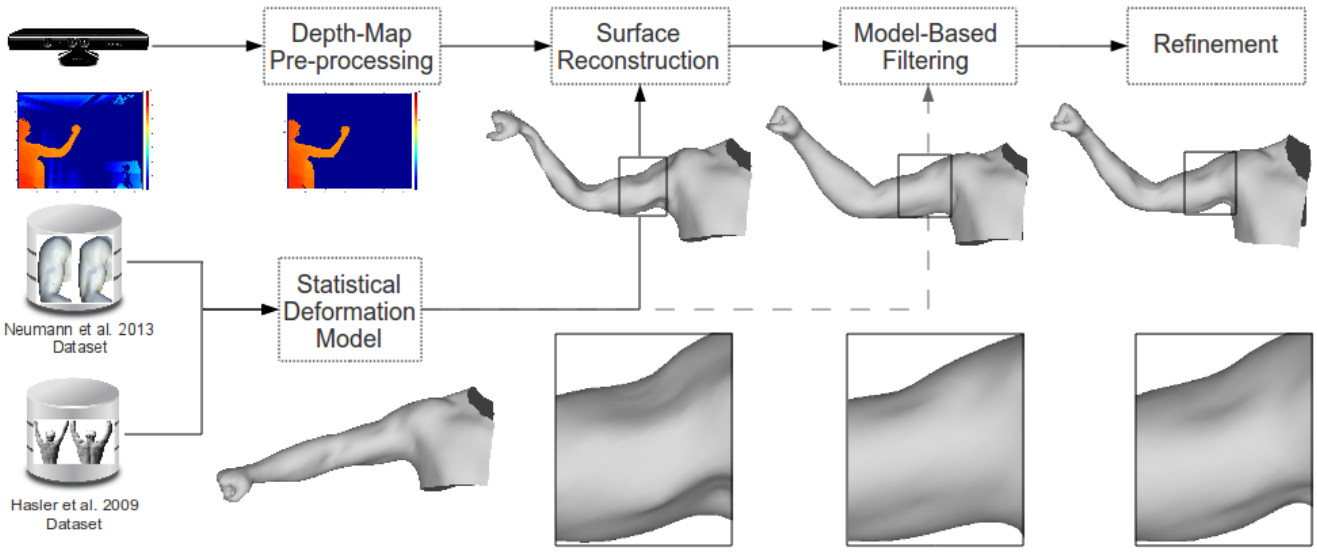


Figure 1: Overview of our capture pipeline: depth measurements are first filtered and then used for an initial surface reconstruction. A statistical deformation model based on two different datasets is built and then used to clean the initial reconstruction within the space of learned deformations. A last refinement step is used to capture fine scale details not captured by previous shapes in the database.

the training set of spatio-temporally aligned muscle deformation examples remains a challenging problem. In order to acquire fine-scale deformation, a lot of markers have to be placed on the human body and tracked using expensive imaging systems [9]. The complexity of this acquisition process places a high limiting barrier for novice artists and practitioners from taking advantage of the research advances in data-driven muscle simulation. Furthermore, people exhibit an enormous statistical variation in muscle deformations with respect to body pose. The data-driven simulation methods are, by their very design, restricted in their modeling ability to the limited set of human subjects captured in the training data. Unless the acquisition process becomes cheap and simple to use, it is difficult to capture a substantially large set of people and model the statistical variation in their muscle deformations. In this paper, we make a contribution in this regard by proposing a novel acquisition method based on the Kinect depth camera.

The Kinect depth sensor has been deployed with great success for various tasks by researchers in robotics, computer vision and human-computer interaction. However, most of these tasks have been restricted to reconstructing a static world [6] or recovering the motion dynamics only at a coarse scale [15]. Modeling fine scale non-rigid surface deformation, with using a consumer-grade depth sensor like the Kinect, remains an immense challenge. The noise artifacts that occur in the depth image make the simultaneous recovery of accurate 3D geometry and motion severely under-constrained. The artifacts in the depth image can arise from limitations in the imaging process, the limited resolution of the capture sensor, and due to surface occlusions that naturally occur during motion. In this paper, we propose a method to process these noisy depth images and reconstruct high-quality spatio-temporally aligned 3D meshes. Our main observation is that a previously acquired dataset of muscle deformations (from a set of 10 subjects captured with a multi-camera acquisition system, kindly made available by [9]) provides useful priors for initializing the 3D registration, and ultimately to reconstructing fine-scale 3D surface deformations beyond the previous dataset. We make the following key-contributions to push the research agenda in this field.

1. We provide a method for filtering three-dimensional correspondence estimation in the noisy depth map input, by using geometric priors of deformation and statistical priors learned from a capture dataset.
2. We provide a framework for extending the generalization scope of a statistical model of deformation, by capturing more people than present in the initial dataset.

2. RELATED WORK

Modeling fine-scale muscle deformations has long been an active research topic in computer graphics. We refer the readers to the related work section in [9] for an elaborate review of muscle deformation models: skinning approaches, physiologically-based simulation models and data-driven simulation models. In the following, we review only certain important related works in data-driven modeling of muscle deformations.

In 2006, Park and Hodgins [10] developed and demonstrated an acquisition system for capturing fine-scale muscles deformations at high-speed motions on several actors. They used a very large set of reflective markers (350 against 40-60 previously used) placed on muscular and fleshy parts of the body. They first captured the rigid body motion of the markers and then used the found residual deformations to deform a hand-designed subject-specific model. However, the marker application time and acquisition complexity were extremely high. This inhibits the acquisition of a larger number of subjects, which can contribute with more muscle data and skin motion. Another limitation of their system is the impossibility to generalize the acquired dynamic captured motion for different body types. In their later work in 2008 [11], they presented a data-driven technique for synthesizing skin deformation from skeletal motion. Using the same input data they used in the previous work, they build up a database of deformation data separately parametrized by pose and acceleration. Afterward they learned respectively pose and acceleration specific deformation using Principal Component Analysis (PCA) and built a statistical model. Because of the complexity of the acquisition step, they filled the database with a huge amount of poses from a single subject, causing the statistical model again to be highly shape dependent. Although they introduced the possibility to generate novel motions of subjects with similar body shapes as the one contained in their database. Using similar acquisition system and pipeline, a later work presented by Hong et al. in 2010 [5] showed an improved skeleton configuration that, combined with standard skinning algorithms, generates a more visually pleasing and physically accurate skin deformation. Focusing on the shoulder complex, consisting of shoulder, elbow and wrist joint, they concluded that inserting one additional segment between the chest and the upper arm greatly improves the motion simulation of the shoulder. The main drawbacks are caused by the generality of their learning algorithm, highly dependent on the completeness of the captured poses. Furthermore, their model is subject-specific and suffers from the same limitations as the previous discussed work.

A recent work proposed by Neumann et al. [9], addresses some of the limitations in previous work, and builds a more generalized statistical model across multiple people with different body shapes. As Hong et al. [5] they focused on the shoulder complex, first capturing shape variations, using a novel acquisition and reconstruction approach, and secondly modeling the deformations as a function of body pose, shape and external forces. Even though using only a low number of parameters, their model is capable of reproducing fine-scale muscle deformations in novel poses and shapes, and for the first time, under the action of several external forces on the arm. Because of its efficiency, the model can be interactively used by artists to reproduce appealing complex skin deformations effects. However, the complexity of the acquisition system limited their acquisition to just a small number of 10 subjects. In this work, we use the dataset kindly made available to us by [9] to derive statistical priors for registering the template mesh showing the human arm to a noisy depth image captured from the Kinect camera. Thus, we propose an easily scalable framework for extending the statistical model by capturing more subjects in a much easier acquisition setup Please note that, unlike [9], we do not consider the problem of modeling the effect of external forces on surface deformation, owing to the capture limitations of the depth sensor. We limit ourselves to modeling arm muscle deformations due to body pose and body shape variations amongst people.

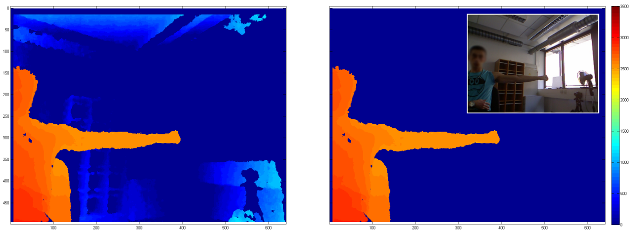
3. OVERVIEW

Our method aims for the reconstruction of high quality spatio-temporally aligned 3D meshes of the human arm from noisy depth images. We expect the human subjects to stand closely to the Kinect camera and perform arm movements through shoulder and elbow joints in a slow and natural manner. As argued by [9] (please also refer to [8]) these movements can be interpreted as *quasi-static* with respect to their underlying biomechanics, without the need to consider dynamics, such as jiggling of the flesh or skin in rapid motions.

As input to our method, we take depth image frames which are disconnected depth measurements affected by noise and quantization artifacts, that together with low resolution, poorly represent the original captured shape. Further, they lack measurements on the back side of the arm, which is invisible from the sensor point of view. From this input, we produce a sequence of temporally consistent 3D mesh reconstructions that show the arm motion at fine-scale detail. The overview of our acquisition and reconstruction pipeline is shown in Figure 1.

We start by performing depth data cleaning to improve its quality (section 4). After that, we fit a template arm mesh to the depth points, by registering the template against the measurements (section 5). This step allows to overcome limitations such as unknown overall shape, including the interpolation of measurements from unseen areas, like the back side. Furthermore, the use of the template allows to describe the motion by explaining the relation between depth measurements taken at different time steps. For the registration phase, we perform an improved version of the classical Non-Rigid Iterative Closest Point (ICP) algorithm. Specifically, we filter the correspondences through a statistical deformation model (section 6) which predicts local muscle bulging with respect to change of pose and shape as a prior for tracking. For each subject, we obtain an initial template mesh from the model that represents the subject's specific shape, and restricts the motion to the allowed pose deformations described by the model. This way we prevent unwanted distortions caused by the general Non-Rigid ICP approach and obtain a good looking arm which roughly aligns to the measurements through time. At this point, when the arm mesh and the measurements are close enough to each other, we perform a final refinement step (section 7), deforming the mesh towards the measurements, which are already filtered and temporally coherent, without any restriction imposed by the model. This way, we can capture new fine scale detail, even when it lies outside of the space represented by the current statistical arm model and we can fead that additional detail back into the mathematical model to extend it.

4. DEPTH-MAP ACQUISITION AND PRE-PROCESSING



(a) (b)

Figure 2: The capturing depth maps (a) are first segmented which removes unwanted background objects (b). The small box on the top-right side shows the corresponding RGB image from the image sensor (which we do not use for reconstruction). Distance is color coded using the scale on the right. The dark blue pixels are missing measurements.

Experiments have shown that Kinect's measurements suffer mainly from two limitations: random uncontrolled noise, that increases with increasing distance to the sensor, and missing data. There are multiple reasons that cause the sensor to miss depth measurements on certain scene areas. Some of them are connected with the sensor range and some other with the so-called "shadows" [1, 2, 7], which are due to the disparity between the camera and projector of the Kinect. In order to improve the initial depth-map quality, we first perform segmentation. Our goal here is to retain measurements that are placed on the subject's arm, which are our main and only interest (see Figure 2). Our segmentation algorithm uses simple thresholding based on the distance from the sensor.

Next, we improve the segmented depth-data quality by running

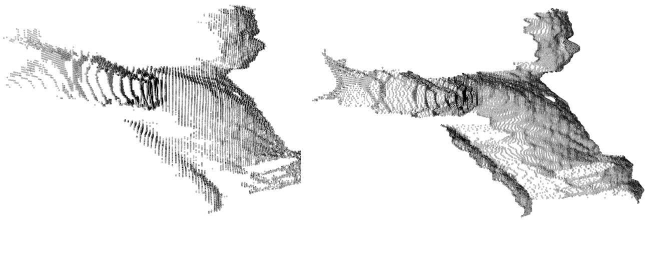

(a) (b)

Figure 3: Depth map smoothing: (a) input depth map, (b) depth map after the smoothing. Notice how in (a) structural/quantization artifacts are visible, whereas the result in (b) shows a smoother surface that is better suited for surface reconstruction.

well-studied filtering algorithms: a median and a Gaussian filter. The median filter removes outliers arising from random noise, and the Gaussian smoothing reduces structural noise caused by the quantization of the data (see Figure 3). We get rid of flying pixels around the borders, by morphological thinning.

5. SURFACE REGISTRATION

In order to reconstruct the original surface from disconnected measurements obtained from the Kinect, we use a surface fitting based approach. Such approach guarantees consistent surfaces with fixed topology, which are optimal for tracking fine-scale skin deformations over time. Our algorithm is similar to the one proposed by Stoll et al. [14]. In particular, after performing an initial alignment, which rigidly aligns the template mesh to the sampled points, we start a non-rigid ICP method, that iteratively deforms the template mesh towards the measurements. The deformation process is guided by local point correspondences.

5.1 Rigid Alignment

In this step, we fix the space within which the human subject moves the arm relative to the location of the template mesh, such that the imaged point cloud is close to the template mesh. This step is particularly important in the registration process, where we are going to register the template against the measurements. Accurate initial surface approximation (in terms of vicinity) of the sample points, obtained from the Kinect, considerably increases the probability of succeeding in generating high-quality reconstructions. By constraining the subject's position and orientation with respect to the sensor (e.g. facing the sensor roughly in a fronto-parallel orientation) to be fixed throughout the entire arm motion sequence, it is possible to find a global rigid transformation (scaling + rotation + translation) relative to the template mesh, and we apply this to all the point clouds in the sequence. We ask the subject to orient the arm and shoulder joints at the first frame such that the imaged point cloud is already near to the template mesh, which can later be tracked over the sequence.

5.2 Finding and Filtering Correspondences

The problem of finding correspondences between a source and a target representation set has been intensively studied, particularly in surface matching and registration. If correct correspondences are known it is possible to find the correct transformation that aligns the sets. Since we have already quite close surface and measurements, coming from the previous step, we can proceed by computing the closest point correspondences. For each mesh vertex we find the nearest point sample and set it as possible good correspondence. The next step is filtering the correspondences [12]. We propose the following filtering strategies to suit to our specific problem setting.

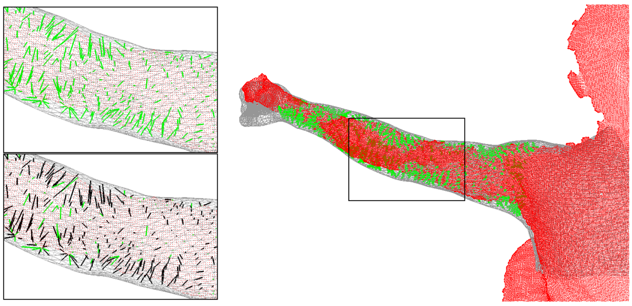


Figure 4: Finding and filtering correspondences example. (Gray) The template mesh, (Red) Point cloud measurements, (Green) Nearest correspondences, (Black) Filtered correspondences.

Arm vertices only: Our mesh model is composed of the arm and part of the chest encompassing biceps, triceps, deltoid and pectoralis muscles. A hand is attached to the arm for aesthetic visualization but not explicitly considered for tracking or deformation modeling. As we would like to focus on deforming the arm shapes, we need to restrict the considered pairs to the ones placed on the arm.

Front side only: We remove all the correspondence pairs coming from the back side of the arm model. In fact, from the sensor's point of view only half of the arm is visible (the front-side), and possible pairs should lie on the same model side. To this end, we compare vertex normals $\mathbf{n}_v$ directions with the known sensor's view direction $\mathbf{s}$, and reject all inconsistent pairs, that do not satisfy the condition:

$$\arccos\left(\mathbf{n}_v \cdot (-\mathbf{s})\right) < \frac{\pi}{2} \tag{1}$$

Normals check: We compute the normals at both end points of each correspondence (from surface vertices with normal $\mathbf{n}_v$ and their correspondent sample points with normal $\mathbf{n}_p$) and check their angular distance. We retain a pair only if the angular difference does not exceed a given threshold of maximum angle T_N,

$$\arccos(\mathbf{n}_p \cdot \mathbf{n}_v) < T_N\ . \tag{2}$$

We estimate the normal at a point by fitting a plane in the neighborhood around the point and then take the normal among the two possible (one pointing upward and one downward), which is consistent with the known sensor's view direction $\mathbf{s}$.

Neighbors check: We filter out geometrically incompatible correspondences in a neighborhood that cause twisting or other undesirable artifacts. Specifically, we compare each correspondence vector $\mathbf{c}_i$, which starts at a vertex and ends at a sample point, to all the closer ones in a neighborhood. We retain a pair only if the median angle deviation α of the correspondence vectors in the neighborhood does not exceed a given threshold of maximum angle T_{Ng},

$$\alpha = \text{median}\left(\left\{\arccos\left(\frac{\mathbf{c}_i \cdot \mathbf{c}_j}{\|\mathbf{c}_i\|\|\mathbf{c}_j\|}\right), j \in N(i)\right\}\right) < T_{Ng}\ . \tag{3}$$

Length limitation: We require the correspondences to not stretch the neighborhood around the point beyond a given threshold. We impose this constraint by requiring the correspondence vector $\mathbf{c}_i$ between mesh vertex and the target point to not deviate in length beyond a threshold T_L over the median length ℓ in the neighborhood.

$$\ell = \text{median}\left(\{\|\mathbf{c}_i\|, \|\mathbf{c}_j\|, j \in N(i))\}\right) < T_L \tag{4}$$

Duplicates elimination: In the last step, we deal with duplicates. Duplicated correspondences are most frequently close to the depth map borders, and are caused by the quantization of the measurements. We retain the point for which the normals of the mesh vertex and that of the point normal agree best. We check for each potential match point, the angular distance $\arccos(\mathbf{n}_v \cdot \mathbf{n}_p)$ between the correspondent normals (vertex normal $\mathbf{n}_v$ and correspondent sample point normal $\mathbf{n}_p$), and take the pair with smaller angular distance.

5.3 Non-Rigid Registration using Mesh Deformation

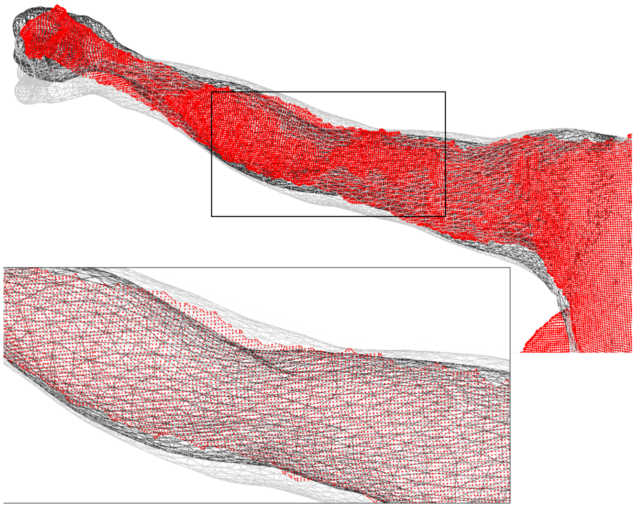


Figure 5: Non-Rigid Registration using As-Rigid-As-Possible deformation for regularization. (Gray) The initial template mesh, (Black) The final mesh obtained after registration. (Red) Point cloud measurements.

Using the correspondences found in previous section, we now deform the template arm model into the target point cloud. To this end, we apply the algorithm proposed by Sorkine and Alexa in 2007 [13]. This method proposes a geometric deformation scheme that preserves local surface detail as encoded by the mesh curvature. Physically, this is similar to deforming an elastic membrane with a thin-plate spline bending energy with certain preset material properties. From now on, in the paper we refer to [13] by the term ARAP ("As-rigid-as-possible") deformation. Although the arm muscles do not deform exactly as an elastic membrane, this simple geometric prior approximates the deformation well (see Figure 5). We use more elaborate statistical priors for arm muscle deformation in the later section.

ARAP deformation is robust under noise as it tends to create smooth surfaces rather than introduce details. This limits the ability of the algorithm from generating fine scale skin deformations. Furthermore, unless all correspondences are correct, ARAP deformations might explain the arm motion incorrectly, and introduce distortions. This happens because the volume is not preserved and there is no additional constraint that forces sections of the arm to deform rigidly, for instance points between the shoulder and the elbow. In the next section, we are going to discuss a way to overcome these limitations by using a template mesh generated from the statistical model as prior for tracking.

6. MODEL-BASED FILTERING

ARAP deformations alone are not able track muscle bulging. ARAP techniques focus on minimizing localized surface deformations, making the whole mesh deform elastically, and thus lose sharp features. Especially in the case of fast motion, where correspondences may be inaccurate, ARAP deformations introduce several irreversible unwanted artifacts, such as arm distortions, which compromise the entire remaining sequence (see an example in Figure 6(a)). In order to avoid such distortions, we project the obtained ARAP mesh estimate to the closest mesh in the plausible space of deformations, which we represent by a statistical model learnt from a previous dataset [9]. This requires the computation of the physiological and body pose parameters of the statistical model such that the corresponding mesh model best fits to the mesh template registered to the point cloud using the previous step. The mesh predicted by the statistical deformation model is artifact-free, because it stays within the plausible space of deformations in the training set. Depending on the model accuracy and the precision by which parameters are computed, the resulting mesh is already close to the measurements. Furthermore, in presence of strong motion, the model does not introduce elastic deformations, as ARAP does, and therefore sharp features are preserved.

6.1 Learning

In order to build a statistical arm deformation model that suits our needs, we use an existing dataset comprising a variety of performance-captured arm motion sequences, which was made available by Neumann et al. [9]. The dataset constitutes a discrete amount of various upper limb movements (30-40), recorded from 10 physically different subjects using a dense make-up of markers on the skin captured under a multi-camera acquisition system . Each movement is stored as a sequence of detailed geometrical mesh representations plus underneath skeleton and blend skinning weights, recorded and formatted by means of an advanced high-quality acquisition system. Based on this dataset, we build a regression model based on two biologically motivated input parameters (shape θ and pose ρ). We learn a linear mapping Ψ between the chosen parameters, in order to simulate sufficiently good meshes M for tracking purposes:

$$M = \Psi(\theta, \rho) \tag{5}$$

Shape parameters are studied in terms of BMI, muscle proportion (or muscularity), height and arm length, while pose parameters in terms of joint angles.

We represent a shape (*i.e,* vertex coordinates of our template mesh) as residual vertex displacements from an initial given base mesh. The base mesh is obtained from the arm section of the average mesh of a set of human full body scans, originally generated by Hasler et al. [4]. This base mesh is particularly suitable for our needs, since it has the same geometrical and skeletal representation as the meshes collected in Neumann's dataset. It is composed of the arm and part of the chest encompassing biceps, triceps, deltoid and pectoralis muscles (see Fig. 9). A hand is attached to the arm for aesthetic visualization but not explicitly considered for tracking or deformation modeling. We select a restricted amount of meshes (2-3) from each subject in the dataset in the same pose as the base mesh pose (*base pose*). We check for this by comparing joint angles. Afterwards, for each extracted mesh we compute vertex displacements d_i from the base mesh and learn a linear regression model that connects this data to physiological body shape parameters. We aim to find weights w_1, w_2 and w_3 relative to BMI β, muscularity μ and height η respectively (plus the intercept weight w_0), such that each vertex displacement d_i can be obtained by computing:

$$d_i = w_0 + w_1 \cdot \beta + w_2 \cdot \mu + w_3 \cdot \eta \ , \qquad (6)$$

(for simplicity we write d_i to denote vector component x, y, and z). Specifically, for each vertex i separately, we minimize the sum of the square residuals:

$$\underset{w_0,w_1,w_2,w_3}{\text{minimize}} \sum_j \left(d_i - (w_0 + w_1 \cdot \beta_j + w_2 \cdot \mu_j + w_3 \cdot \eta_j)\right)^2 . \qquad (7)$$

Because of the insufficient amount of data in the dataset (only 10 people), we cannot rely on Neumann's dataset for estimating the required arm length to initialize the arm to a new subject in the base pose. We instead make use of Hasler's dataset [4], which constitutes far more subjects (114 people in static poses) across shape and age. We find the wanted vertex displacements, by solving for weights w_j a simple linear system of equations, similar to equation (6), given by:

$$d_i = \sum_j w_j \cdot e_j \qquad (8)$$

where e_j are eigenvectors obtained learning PCA on Hasler's dataset. The found approximate solution results from a least squares minimization of the original problem, with a regularization term included (Tikhonov-Miller regularization):

$$\underset{w_j}{\text{minimize}} \sum_j \left(d_i - \left(w_j \cdot e_j\right)\right)^2 + \left(q \cdot w_j\right)^2 \qquad (9)$$

Here q is the regularization weight that allows us to control the strength of the regularization.

In order to learn deformations with respect to pose, we proceed similarly. For each subject in Neumann's dataset, we first bring each mesh in the base pose by setting the base pose joint angle parameters and using dual quaternion skinning, and then compute vertex displacements from the base mesh. Skinning might introduce so-called skinning artifacts, causing the surface to present severe distortions. Therefore, in our situation, we learn both vertex displacements arising from pose specific deformations and those arising from skinning artifacts indistinctly. Since we are going to use skinning during the reverse process as well, *i.e,* bring the base mesh to the wanted pose, skinning artifacts are going to cancel out. We learn the resulting displacements using linear regression, and find pose weights w_i relative to each 3 joint angles (shoulder J_S, elbow J_E and wrist J_W) coordinate (x,y,z) respectively (plus the intercept weight w_0), such that each vertex displacement d_i can be obtained by computing:

$$\begin{aligned} d_i = w_0 &+ w_1 \cdot J_{S_x} + w_2 \cdot J_{S_y} + w_3 \cdot J_{S_z} \\ &+ w_4 \cdot J_{E_x} + w_5 \cdot J_{E_y} + w_6 \cdot J_{E_z} \\ &+ w_7 \cdot J_{W_x} + w_8 \cdot J_{W_y} + w_9 \cdot J_{W_z} \end{aligned} \qquad (10)$$

6.2 Mesh Projection

Shape parameters are constants that depend on the particular subject's physiology, and can be measured once at the beginning of the captured motion sequence. Pose parameters instead need to be updated along with the arm movements. We estimate the pose parameters (joint angles) of the resulting ARAP deformed mesh in the current step, which is already close to the actual pose. The approach we use here was discussed by Neumann et al. [9] during the motion estimation step. We proceed finding for each body part separately (torso, upper arm, forearm and hand) the rigid transformation that maps the vertices of the base mesh $p = \{p_1, \ldots, p_n\}$ to their deformed positions in the ARAP deformed mesh $p' = \{p'_1, \ldots, p'_n\}$, and then solve for joint position and orientation. The problem of finding an orthogonal mapping between p and p', is known as the orthogonal Procrustes problem. It can be solved by minimizing the following three-dimensional Euclidean distance error, with respect to the unknown rotation R and translation t:

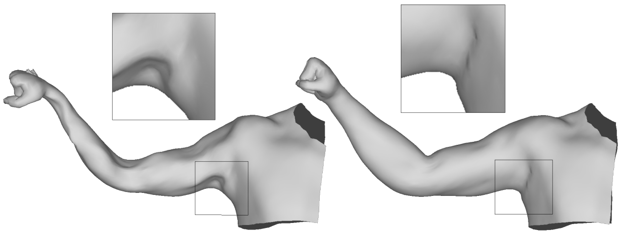


Figure 6: (a) Registered mesh using Non-rigid ICP. (b) Mesh projected using the statistical deformation model. The mesh after projection into the learned model space has less distortions.

$$\underset{R,t}{\text{minimize}} \sum_{i=1}^{n} w_i \left((Rp_i + t) - p'_i\right) \qquad (11)$$

where w_i are blending weights that associate vertex index i to the considered body part. We refine the obtained transformation (rotation + translation) iteratively by using the theory of rigid-body motion and the twist representation [3]. This approach is more flexible than skeleton based inverse kinematics (IK). In fact, ARAP regularization adapts better to fine-scale registrations, as compared against pure mechanical skeleton rotations. Additionally, the resulting mesh in the found pose is closer to the original measurements. This is very important for our pipeline.

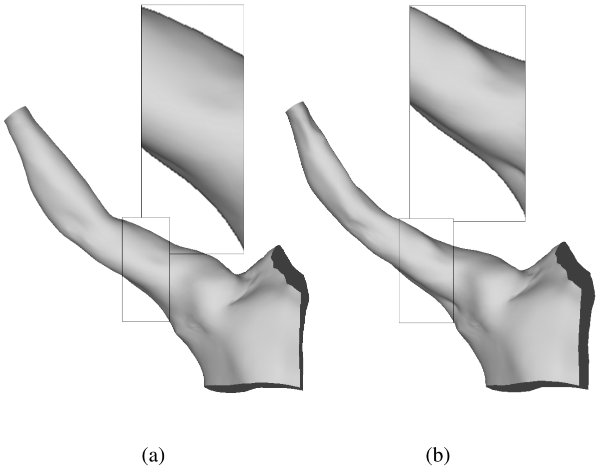
Having shape and pose parameters, all we need to do is to input their values to our linear regression model, which gives as output vertex displacements that we add to Hasler's base mesh. Finally, we perform linear blend skinning in order to bring the resulting displaced mesh in the wanted pose. In summary, the final vertex coordinates v are given by:

$$v = \text{skinning}\left(v_{base} + d_l + d_\theta + d_\rho\right) \qquad (12)$$

where v_{base} are the vertex coordinates of the base mesh, d_l, d_θ and d_ρ represent respectively length, shape and pose displacements. See a projection result in Figure 6

7. SURFACE REFINEMENT

This final step of our pipeline finalize the entire tracking process, by introducing new surface details that go slightly beyond the deformation space described by the previous dataset. To this end, we perform a last ARAP deformation driven by dense point correspondences that are temporally consistent across frames. See a refined surface result in Figure 7.



(a) (b)

Figure 7: (a) Mesh obtained after projecting the initial surface using the statistical deformation model, (b) this mesh is further refined using the depth map to obtain fine-scale folds outside the space of admissible body shapes and muscle bulges.

7.1 Time Consistent Correspondences

Because of imperfect alignment between the template mesh and the measurements, and and noise in the captured depth data, the direction and length of the found correspondences may vary strongly over time, producing visually unpleasant flickering skin deformations. A way to reduce this effect in a physically correct manner, is averaging the correspondences through time. In fact, at each frame, skin deformations happen smoothly with respect to previous and following frames. Therefore, a weighted average over frames in temporal vicinity helps fix visual flickering and better simulate consistent muscle bulging. Specifically, we compute a weighted average between all n correspondences in the current frame $C_i = \{c_1, c_2, ..., c_n\}$ and the relative ones (based on the same vertex) in the previous C_{i-1} and following frames C_{i+1}:

$$C_i = \frac{w_{i-1} \cdot C_{i-1} + w_i \cdot C_i + w_{i+1} \cdot C_{i+1}}{w_{i-1} + w_i + w_{i+1}} \qquad (13)$$

where the weights w_i, w_{i-1} and w_{i+1} are computed using a Gaussian based on the temporal inter-frame distance δ from the current frame i:

$$w_{i+\delta} = e^{\frac{-(\delta)^2}{2\cdot\sigma^2}} \qquad (14)$$

7.2 Weighting Correspondences

Once all the correspondences are computed by averaging them with respect to the neighbor frames, we proceed by assigning weights to the pairs. The weight specifies the reliability of the connection and is extremely important for the next deformation step. We have observed that the reliability of correspondences cannot depend solely on the accuracy of the normals at both ends, since this value is often unreliable. Further, considering the consistency in a limited neighborhood alone does not give an accurate measure of reliability. Instead, we impose a regularity in the vertex normal orientations *i.e,* we favor those correspondences that are oriented similarly to the relative vertex normals. This is a valid assumption in our case, where the template and sample points are approximately aligned and the relative displacements are small. Visually, such point correspondences prefer bulges or dimples along the normal direction to uncontrolled shifting and shearing, that are favored less. Applying this expedient, we set the weights according to how the correspondence vector of a vertex aligns with the respective normal direction. Considering that surface regions around the bone joints deform more dramatically, we also scale the weights by a linear factor, which specifies the closeness of a vertex to a joint position. This lets vertices around the joint to deform more freely.

8. RESULTS

8.1 Experimental Setup

Subject	BMI	Muscularity	Height	Weight
	(Kg/m^2)	(%)	(m)	(Kg)
P_1^f	18.1	35.0	1.65	49
P_2^f	20.3	37.0	1.72	60
P_3^f	21.1	35.0	1.60	54
P_4^m	21.1	50.0	1.78	67
P_5^m	21.2	50.0	1.92	78
P_6^f	30.8	38.0	1.58	77
P_7^m	34.2	40.0	1.76	106

Table 1: Physiological parameters for each recorded subject (ordered by BMI parameter). The superscript f or m in the person ID indicates female and male gender.

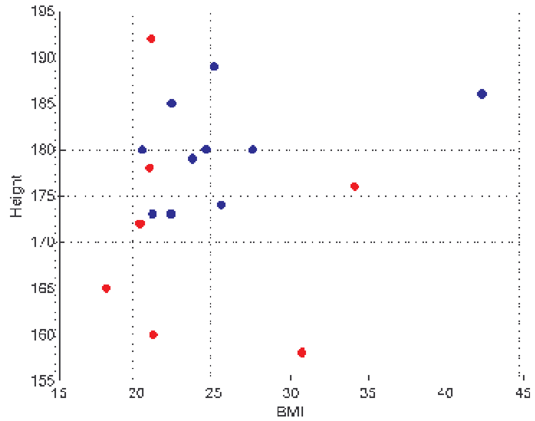


Figure 8: BMI and Height (in cm) of the subjects from the dataset by [9] used to build the statistical body shape model (blue) together with the new captured subjects (red).

Our acquisition setup requires a single depth sensor, in our case we use the Kinect. During data capture, test subjects were sitting on a chair while performing various arm motions. The sensor was placed at 1 meter distance at the same height as the subject's chest, giving a full frontal view of the shoulder and arm section. Prior to the acquisition, we asked the actors to undress the arm up to the shoulder and adopt an initial arm position resembling the pose of the template mesh (base pose). This way, we simplify the initial alignment step and ensure accurate tracking. To avoid self occlusions and facilitate tracking, we asked the people to keep the arm parallel to the sensor and perform slow movements. To concentrate on muscle bulges in the elbow and shoulder area, subjects were asked to make a fist and avoid wrist movement during the course of the acquisition.

We captured arm movements from 7 people of different physical conditions (Table 1). On purpose, we chose some subjects outside the range of body shapes represented in the mathematical model,

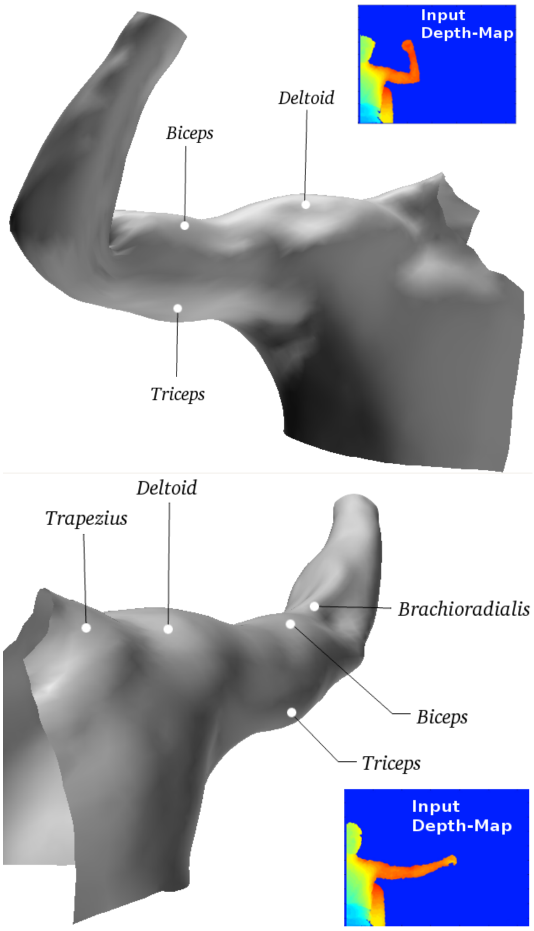


Figure 9: Resulting captured arm-muscle for person P_4^m. Thanks to the statistical deformation model, we can reconstruct the invisible back side of the arm. On the front side, minute muscle bulges can be captured from the low-resolution and noisy input depth-map.

Figure 8. This allowed evaluating wether our method is able extend the space of body shapes. On top of that, the previous dataset of Neumann [9] comprised only male subjects, while we included female subjects as well (with f attribute in table 1). We captured two overweight (P_6^f and P_7^m), one underweight (P_1^f), two normal weight (P_2^f and P_3^f) and two athletic subjects who reported to exercise regularly (P_4^m and P_5^m). For each subject we recorded around 200 frames where they perform elbow flexion and extension movements, accompanied with shoulder abduction and adduction, without external weight.

8.2 Evaluation of Captured Sequences

We show results from three physiologically different subjects in detail and evaluate the muscle bulges that our method is able to reconstruct in those sequences as well as the algorithm parameters used for those sequences. We also discuss the generalization capability of our method to capture previously unseen body shapes.

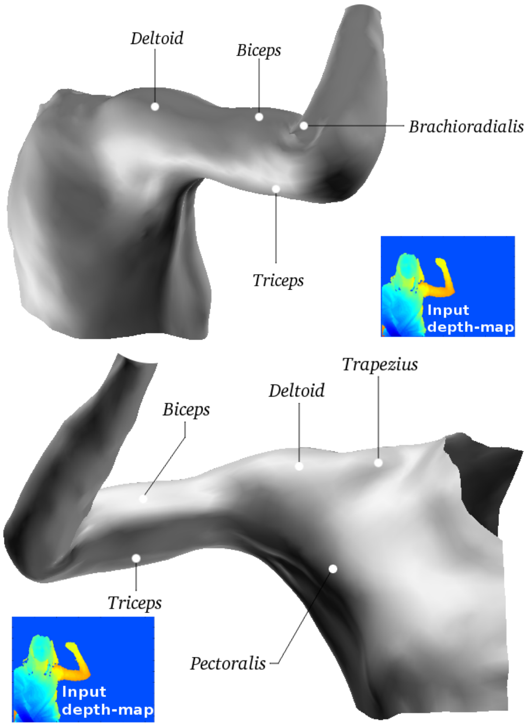


Figure 10: Captured arm-muscle deformation for person P_1^f.

Muscular Arm We show some selected frames of a 220 frame sequence of person P_4^m performing arm abduction and adduction as well as elbow flexion and extension. Together with P_5^m this is the most muscular arm we captured. His physical characteristics (i.e. BMI, height and muscularity) are within the boundaries of the previous dataset we used for learning our mathematical model. Figure 9 shows that our method faithfully captures bulges of the biggest muscles of the arm, triceps and biceps, as well as the most important shoulder muscle, the deltoid.

Skinny Arm We show the most salient frames from the sequence resulting from a skinny underweight person, P_1^f. This data is completely outside the boundaries of the original statistical arm model (see Figure 8). It is therefore challenging to reconstruct, because the mathematical model may not be able to correctly predict the body shape and bulges. The acquired sequence consists of about 150 frames, where the actor performed simple elbow flexion and extension movements only. The captured meshes of one frame are shown in Figure 10.

Flabby Arm This data shows arm motion of a flabby subject, P_6^f. It also falls completely outside of the initial dataset as well, because of the subject's height (particularly small) and BMI (particularly high). The sequence includes around 180 frames of slow elbow movements. Some representative frames are shown in Figure 11.

Muscle Bulges During arm abduction and adduction movements, the main force is sustained by muscles originating from the shoulder, like the pectoralis and trapezius muscles [8]. The mathematical model convincingly reconstructs and interpolates those areas as well as those which are invisible from the sensor's point view (posterior side of the arm), and for which we do not collect any data (the chest). During elbow flexion, the biceps and barchioradialis muscles are activated, while elbow extension is mostly exerted by the triceps muscle. Both kind of muscle contractions are well visible in the muscular arm sequence. In the skinny and flabby arm sequence, almost no muscle contractions are reconstructed, mainly because they are simply not visible in the recorded setting due to

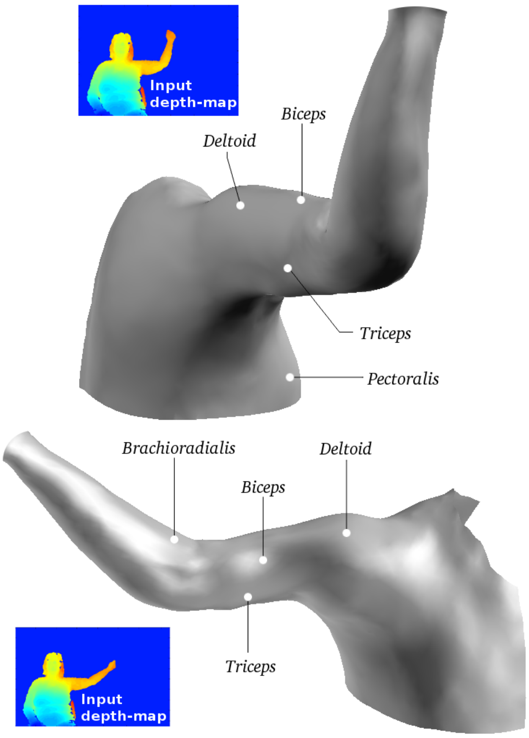


Figure 11: Captured arm-muscle deformation for person P_6^f.

the particular physiological characteristic of the subject.

Algorithm Parameters Our pipeline requires the selection of a number of parameters, which determine the accuracy and stability of the arm reconstruction. The parameters that are most dependent on the specific arm motion sequence are the normal threshold (T_N) and the neighbor threshold (T_{Ng}). The chosen parameter values for the muscular, skinny, and flabby arm sequence are collected in Table 2.

Arm	T_N	T_{Ng}
Muscular	30	60
Skinny	10	40
Flabby	12	60

Table 2: Chosen parameters for the skinny, muscular and flabby arms. The values for T_N and T_{Ng}, respectively the normals' and neighbors thresholds, are expressed in degrees.

Generalization to new Body Shapes Our method works well for arms with physiological characteristics within the boundaries of the initial dataset we use for learning the deformation model. Fig. 12 shows the 3D meshes generated by our deformation model, and also indicates the limits where our model starts to fail. BMI values smaller than 15 or greater than 45 are very rare and usually symptoms of poor health conditions. Our deformation model can benefit by enlarging the dataset to include small healthy people (e.g, children), which we leave to future work.

In order to reduce surface distortion due to a body shape or unusual body pose not within the range of the mathematical model, we constrain the displacement norms of vertices. In particular, we allow shape displacements smaller than a given value, and cut-off the rest. The sequence of meshes in Figure 13 shows model generated arms for a previously unseen pose, i.e. a pose which is not included in the learning dataset. We found a displacements norm limit of 90% (of the maximum displacement norm for the particular pose) to alleviate typical skinning artifacts (see the underarm and elbow area), without introducing large surface distortions produced by the model.

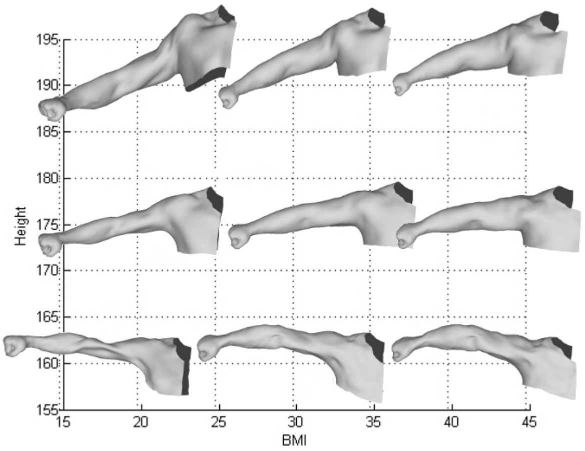


Figure 12: Plot showing different body shapes generated by the statistical deformation model for different BMI and height (cm) parameters. Muscularity is fixed to 50%. The limited training data permits generation of artifact-free meshes outside the capture range, our method can be used to extend the space of body shapes using a simple capture setup to improve results in those cases.

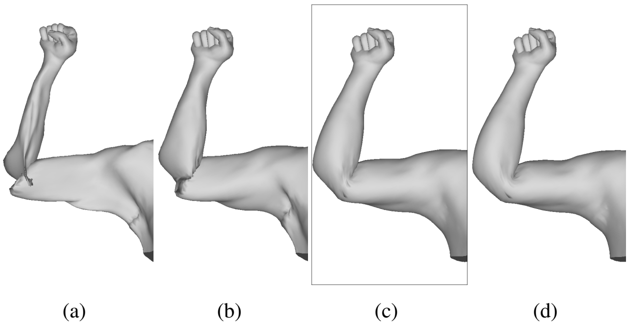


Figure 13: Meshes generated from the statistical deformation model for an unseen pose far off the captured pose range. Each mesh shows different displacement norm limitation used. The percentage are computed out of the maximum displacement norm for the particular pose. (a) 0% norm limitation = unlimited, (b) 50% norm limitation, (c) 90% norm limitation, (d) 100% norm limitation = simple skinning. Notice how this limiting process effectively restricts artifacts. We use the choice (c) in our method.

8.3 Discussion and Limitations

Highly detailed reconstruction of time-varying skin surface just from a noisy and low-resolution depth sensor is an extremely challenging problem. The use of a mathematical model as a prior in surface reconstruction, as done by our method, helps in such a setting to significantly constrain distortions caused by possibly very noisy input data. Using this idea, we can handle fast movements and still capture shape and volume preserving meshes. However, the underlying model may fail to generate accurate surfaces free of artifacts for body shapes or poses that are outside the boundaries of the original dataset. In case of strong flexion, some areas are not satisfactorily modeled (for example, the back-side of the arm, elbow and shoulder areas), with important fine-scale details missing and skinning artefacts not completely removed. These limitations are possibly due to the inability of our simple linear regression model to handle areas under occlusions.

Our pipeline handles a wide range of arm movements, however it fails to reconstruct forearm pronation or supination. Such rotations are not possible to capture only from a depth map, since such motions are only visible when tracking the skin and do not result in any changes of the depth map. A way to overcome this limitation is to simultaneously include knowledge from an RGB camera, and use tracking algorithms or optical flow to estimate the arm rotation. However, skin usually presents a homogeneous pattern that is not suitable for feature tracking. Application of colored markers is a typical solution for this problem, though this further increases acquisition setup time and may require calibration. Alternatively, a motion prior or hand tracking can be used in the future to resolve this ambiguity.

Another limitation of our pipeline is the handling of arm occlusion and self-occlusion. Occlusions cause a permanent or (in the best case) a temporary loss of measurements on a big arm area (more than 50% occluded, e.g. forearm and hand occluded). To improve reconstruction quality for such cases in the future, we would like to use spatio-temporal priors in our tracking framework.

9. CONCLUSIONS

In this paper we have shown that using a statistical deformation model as prior for tracking improves classical surface fitting methods based on Non-Rigid Iterative Closest Point (ICP) algorithms. Typical non-rigid ICP algorithms require very accurate correspondence pairs in order to correctly deform the original surface, and focus on minimizing local deformations rather than preserving the overall shape and volume. Such approaches are clearly too generalized to allow for distortion-free deformations. We proposed a reinforcement of such methods by constraining the allowed deformations to a subset allowed by the model. This reinforcement nearly nullifies distortions and provides a good initialization from which we can reconstruct coherent and visually appealing skin deformations at a fine scale. We have evaluated each step of the pipeline. Modeling the human arm is useful for many applications such as the developing virtual 3D characters, pointing interfaces in HCI, or detecting muscle fatigue in sports. But the contributions of this paper are also useful for modeling other body parts. Anatomically based deformation models that can be acquired easily from real people will have many applications in entertainment and medicine. Among the main contributions of this paper, we introduced the possibility of using a simple and cheap acquisition system for fine-scale reconstructions (the Kinect sensor). Such systems have the advantage to greatly accelerate the data acquisition process, compared to large and complex systems commonly used for this purpose. In the future, we would like to explore the possibility of coupling information from different sensors, such as force or motion sensors, together with RGB images and 3D point cloud data, in order to improve the overall surface registration.

10. REFERENCES

[1] M. Andersen, T. Jensen, P. Lisouski, A. Mortensen, M. Hanse, T. Gregersen, and P. Ahrendt. *Kinect Depth Sensor Evaluation for Computer Vision Applications*, 2012.

[2] J. Ballester and C. Pheatt. Using the xbox kinect sensor for positional data acquisition. In *Volume 81, Issue 1, pp. 71.* American Journal of Physics, 2013.

[3] C. Bregler and J. Malik. Tracking people with twists and exponential maps. In *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, CVPR '98, pages 8–, Washington, DC, USA, 1998. IEEE Computer Society.

[4] N. Hasler, C. Stoll, M. Sunkel, B. Rosenhahn, and H. Seidel. A statistical model of human pose and body shape. *Comput. Graph. Forum*, 28(2):337–346, 2009.

[5] Q. Y. Hong, S. I. Park, and J. K. Hodgins. A data-driven segmentation for the shoulder complex. *Comput. Graph. Forum*, 29(2):537–544, 2010.

[6] S. Izadi, D. Kim, O. Hilliges, D. Molyneaux, R. Newcombe, P. Kohli, J. Shotton, S. Hodges, D. Freeman, A. Davison, and A. Fitzgibbon. Kinectfusion: real-time 3d reconstruction and interaction using a moving depth camera. In *Proceedings of the 24th annual ACM symposium on User interface software and technology*, UIST '11, pages 559–568, New York, NY, USA, 2011. ACM.

[7] K. Khoshelham and S. Oude Elberink. Accuracy and resolution of kinect depth data for indoor mapping applications. *Sensors 2012*, 12:1437–1454, 2012.

[8] D. A. Neumann. *Kinesiology of the musculoskeletal system: foundations for physical rehabilitation*, chapter 5,6,7. Mosby, St. Louis, 2002.

[9] T. Neumann, K. Varanasi, N. Hasler, M. Wacker, M. Magnor, and C. Theobalt. Capture and statistical modeling of arm-muscle deformations. In *Proceedings Eurographics 2013*, 2013.

[10] S. I. Park and J. K. Hodgins. Capturing and animating skin deformation in human motion. *ACM Trans. Graph.*, 25(3):881–889, July 2006.

[11] S. I. Park and J. K. Hodgins. Data-driven modeling of skin and muscle deformation. *ACM Trans. Graph.*, 27(3):96:1–96:6, Aug. 2008.

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

[13] O. Sorkine and M. Alexa. As-rigid-as-possible surface modeling. In *Proceedings of EUROGRAPHICS/ACM SIGGRAPH Symposium on Geometry Processing*, pages 109–116, 2007.

[14] C. Stoll, Z. Karni, C. Rössl, H. Yamauchi, and H. Seidel. Template deformation for point cloud fitting. In M. Botsch, B. Chen, M. Pauly, and M. Zwicker, editors, *SPBG*, pages 27–35. Eurographics Association, 2006.

[15] T. Weise, S. Bouaziz, H. Li, and M. Pauly. Realtime performance-based facial animation. *ACM Transactions on Graphics (Proceedings SIGGRAPH 2011)*, 30(4), July 2011.