Anatomical augmented reality with 3D commodity tracking and image-space alignment

Armelle Bauer, Debanga Raj Neog, Ali-Hamadi Dicko, Dinesh K. Pai, François Faure, Olivier Palombi, Jocelyne Troccaz

▶ **To cite this version:**

Armelle Bauer, Debanga Raj Neog, Ali-Hamadi Dicko, Dinesh K. Pai, François Faure, et al.. Anatomical augmented reality with 3D commodity tracking and image-space alignment. Computers and Graphics, 2017, 69, pp.140 - 153. 10.1016/j.cag.2017.10.008 . hal-01633632

HAL Id: hal-01633632
https://hal.science/hal-01633632v1

Submitted on 13 Nov 2017

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Computers & Graphics (2017)

Contents lists available at ScienceDirect

Computers & Graphics

journal homepage: www.elsevier.com/locate/results-in-physics

Anatomical Augmented Reality with 3D Commodity Tracking and Image-space Alignment

Armelle Bauer[a,b], Debanga Raj Neog[e], Ali-Hamadi Dicko[b,d], Dinesh K. Pai[e], François Faure[b,d], Olivier Palombi[b,c], Jocelyne Troccaz[a]

[a] *Univ. Grenoble Alpes, CNRS, Grenoble INP, TIMC-IMAG, F-38000 Grenoble, France*
[b] *Univ. Grenoble Alpes, CNRS, Grenoble INP, INRIA, LJK, F-38000 Grenoble, France*
[c] *Grenoble Alpes Hospital, LADAF, F-38000 Grenoble, France*
[d] *AnatoScope SA, F-34000 Montpellier, France*
[e] *Department of Computer Science, University of British Columbia, Vancouver, British Columbia V6T 1Z4, Canada*

ARTICLE INFO

Article history:
Received September 30, 2017

Keywords: User-specific Anatomy, Augmented Human, Real-Time, Motion Capture, Augmented Reality, Markerless Device, Image Warping, Handled Occlusion and Self-occlusion

ABSTRACT

This paper presents a mirror-like augmented reality (AR) system to display the internal anatomy of the current user. Using a single Microsoft V2.0 *Kinect* (later on referenced as the *Kinect*), we animate in real-time a user-specific model of internal anatomy according to the user's motion and we superimpose it onto the user's color map. Users can visualize their anatomy moving as if they where looking inside their own bodies in real-time.

A new calibration procedure to set up and attach a user-specific anatomy to the *Kinect* body tracking skeleton is introduced. At calibration time, the bone lengths are estimated using a set of poses. By using *Kinect* data as input, the practical limitation of skin correspondence in prior work is overcome. The generic 3D anatomical model is attached to the internal anatomy registration skeleton, and warped on the depth image using a novel elastic deformer subject to a closest-point registration force and anatomical constraints.

The noise in *Kinect* outputs precludes direct display of realistic human anatomy. Therefore, to enforce anatomical plausibility, a novel filter to reconstruct plausible motions based on fixed bones lengths as well as realistic angular degrees of freedom (DOFs) and limits are introduced. Anatomical constraints, applied to the *Kinect* body tracking skeleton joints, are used to maximize the physical plausibility of the anatomy motion while minimizing the distance to the raw data. At run-time, a simulation loop is used to attract the bones towards the raw data. Skinning shaders efficiently drag the resulting anatomy to the user's tracked motion.

Our user-specific internal anatomy model is validated by comparing the skeleton with segmented MRI images. A user study is established to evaluate the believability of the animated anatomy.

As an extension of Bauer et al. (2016), we also propose an image-based algorithm that corrects accumulated inaccuracy of the system steps: motion capture, anatomy transfer, image generation and animation. These inaccuracies show up as occlusion and self-occlusion misalignments of the anatomy regions when superimposed between them and on top of the color map. We also show that the proposed work can efficiently reduce these inaccuracies.

© 2017 Elsevier B. V. All rights reserved.

Introduction

The emergence of commodity depth cameras such as *Kinect* sensors motivates new educational, medical and healthcare applications. However, previous studies show that raw *Kinect* data cannot be easily employed in human motion tracking (see Pfister et al. (2014), and also Malinowski and Matsinos (2015)). In this paper, a new calibration and motion capture sufficiently accurate for AR applications are introduced and demonstrated by superimposing internal anatomy on the user's color map in real-time. At calibration time, the length and width of body segments are estimated based on specific body poses and silhouettes. A novel anatomically sound deformer is applied to fit a high-quality generic 3D biomechanical model in order to generate a user-specific anatomical model. At run-time, our model tracks bone motions based on the *Kinect* body tracking skeleton joints output, while enforcing anatomical plausibility rules such as constant lengths and joint limits. Our user study preliminary shows that the precision is sufficient to superimpose the user-specific anatomical model onto the color image, using linear blend skinning. We also propose an efficient image-based corrective registration method to correct any anatomy misalignments that can occur during 3D registration.

This paper is organized as follows: *Section 1* briefly survey related work. *Section 2* introduces a body size measurement procedure based on multiple poses, silhouette points and an anatomically sound deformer. *Section 3* describes how it is animated based on robust motion capture using anatomical constraints. *Section 4* goes through results and the validation process. *Section 5* explains how final adjustments can be made at the image level in order to improve the rendering and to solve occlusion issues. *Section 6* finally concludes by presenting possible applications of this work and under development features.

This paper is an extended version of the paper Bauer et al. (2016), presented at MIG'16. Superimposing the anatomy onto the users image allows us to create a real-time augmented reality experience. However, the AR paradigm impose us recurrent evolutions to improve immersion. As seen in Bauer et al. (2016) attached video (see https://youtu.be/Ip17-Vaqqos), in the previous version we had good but not optimal results. This paper presents an image-based correction system that is a major contribution: it allows perfect alignment of virtual data onto streamed video.

1. Related Work

Nowadays, human body modeling and tracking are widely studied for a variety of applications such as motion capture or morphometric studies.

Skin Registration is the most accessible approach to generate a wide range of human bodies. Most studies are based on skin statistical models generated by a shape and pose database (Helten et al. (2013)). Gilles et al. (2011) use frame-based skinning methods to deform a generic skin to fit at best the user data. Other approaches using point cloud (Li et al. (2013)) or multi-position silhouettes (Vlasic et al. (2008)) may also be used to reconstruct the body skin. Most often, raw data comes from acquisition of people wearing clothes and this may lead to non-realistic bodies. Bălan and Black (2008), as well as Zeng et al. (2015) intend to find ways to pass through these limitations. Since they rely only on skin models and do not include internal anatomy, those methods may result in unrealistic skin twisting.

Anatomy Registration The most accurate subject-specific anatomy registration methods come from the medical imaging fields (Sotiras et al. (2013)). However, 3D medical images are not easily used in a non medical context and are not adapted to real-time capture. Several other methods have been proposed. Quah et al. (2005) present a pose-dependent method to register a 3D anatomical model onto 2D images. Based on key points, they register skin and skeleton (no soft tissue). However this method gives static results. Using *Kinect* point cloud, Zhu et al. (2015) register user-specific skin and skeleton during motion. Ali-Hamadi et al. (2013) as Gilles et al. (2010), present a pose-dependent method to transfer a 3D anatomical model to a target skin. This method is static and time consuming. The method introduced by Saito et al. (2015) achieves near-interactive run time skin, skeleton and soft tissue 3D model editing. *Tab. 1* compares state of the art anatomy registration methods.

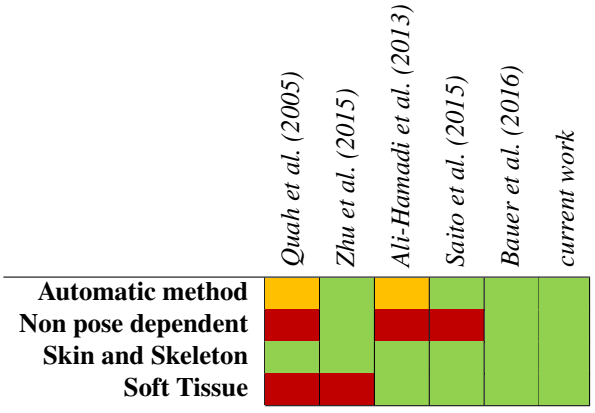

Table 1. Comparison between state of the art anatomy registration methods. Legend: green means that the characteristic is totally handled by the method, orange that it is partly, and red that it is not.

User Tracking In Pfister et al. (2014), the authors assess that the rough *Kinect* body tracking data are enough for basic motion measurements such as gait analysis, or joint angles during motion, but are far beyond VICON cameras in terms of software and hardware.

The tracking algorithm used in this paper is based on the *Kinect* body tracking skeleton which is really noisy. Whereas we add constraints to upgrade the tracking, Meng et al. (2013) ask the user to pinpoint anatomical key points to help positioning the data. Shen et al. (2012) use an example-based method to learn how to correct initially tracked poses. Because body

tracking is a critical step, other methods like those introduced by Zhou et al. (2014) or Wei et al. (2012) use the *Kinect* depth map and implement their own posture registration process using probabilities or pose estimations. Zhu et al. (2015) use multi-*Kinect* depth maps and anatomical knowledge to enhance realistic limb motions.

Nowadays, in the game industry, sports and fitness training applications based on depth map tracking devices are commonly used (e.g. Nike *Kinect*+, Get Fit With Mel B, Your Shape, *etc...*). To our knowledge, the best tracking games are based on the Microsoft V2.0 *Kinect* technology. All these games only show the user depth map or silhouette .

By presenting the anatomy superimposed onto the user's color map (AR), a precision and a realism constraint are added compared to this field state of the art. Associated Supplemental materials presents experiments we did to determine the overall *Kinect* body tracking precision and quality.

AR Systems In the last few years, the number of AR applications increased in the medical education field (see Kamphuis et al. (2014)).

The Magic Mirror, by Blum et al. (2012), superimposes statically CT scans of the abdomen onto the user's image with gesture-based interaction. A more recent version of the Magic Mirror (see also Ma et al. (2016)) also presents an "organ explosion effect" and the possibility to visualize precomputed muscle simulation of the user's arm in real-time. The Digital Mirror, by Maitre (2014), shows full body CT scans but does not superimpose them on the user image. In these two cases, data follow the user's motion but are not deformed with respect to these motions. The Anatomical Mirror, by Borner and Kirsch (2015), allows full-body motion by using the *Kinect* body tracking, but displays animated generic 3D models while we show a user-specific one.

Thanks to the use of rules coming from anatomical knowledge, we significantly improve AR realism and anatomy motion plausibility with respect to Bauer et al. (2014) and Bauer et al. (2015) in the Living Book of Anatomy project. *Tab. 2* summarizes comparisons between state-of-the-art demos and our work.

Data Validation Validation of anatomical data requires in-vivo measurements, the simplest way is to use as ground truth body measurements (see Dao et al. (2014)) or/and anatomical landmarks (see Espitia-Contreras et al. (2014)) taken directly onto the user's body. The study made by Malinowski and Matsinos (2015) gives limb bones length during motion and compare them with ground truth body measurements.

Using user body anatomical landmarks introduces measurement errors due to body position and skin curvature. We decided to use MRI data as ground truth to be able to obtain internal specific points (e.g. femoral head of bone) in addition to externally visible specific anatomical points.

2. User-Specific Anatomy

We present a novel approach using *Kinect* SDK outputs (color map, body tracking skeleton and point cloud) and a 3D reference model including skin surface and internal anatomy (skeleton, muscles, organs, etc) to generate user-specific anatomical data.

	Blum et al. (2012)	*Maitre (2014)*	*Borner and Kirsch (2015)*	*Bauer et al. (2014)*	*Bauer et al. (2015)*	*Bauer et al. (2016)*	*Ma et al. (2016)*	*current work*
Application								
Real-Time								
Anatomy Choice								
Advanced Interaction								
3D Model								
User-Specific								
Male/Female or Unisex	F	F	T	T	T	T	T	T
Whole Body	F	T	T	T	T	T	F	T
Skeleton System	T	F	T	T	T	T	T	T
Muscular System	F	T	T	T	T	T	T	T
Organs	F	T	T	T	T	T	T	T
Anatomy Motion								
User Position								
User Posture								
Precision								
Validation								
User Study	T	T	F	F	F	T	T	T
Augmented Reality								

Table 2. Comparison between anatomical mirror-like applications. Legend: green for good (or True), orange for average, red for bad (or False).

The method consists of four steps. First, the user-specific body segment lengths and widths are computed using the *Kinect* SDK outputs (see *Section 2.1*) to define a list of 3D key points. In the second step the generic skin is deformed based on key points and the partial user's point cloud (*Section 2.2*). The third step consists in transferring the reference skeleton inside the user-specific skin (*Section 2.3*). Finally soft tissue between the bones and the skin is determined using Laplacian interpolation in a way similar to Ali-Hamadi et al. (2013). These different steps are summarized in *Fig. 1*.

To ease the understanding of the rest of this *Section*, descriptions of each type of deformation skeleton used are provided below:

- ***Kinect* body tracking skeleton**: composed of 25 joints, this animation skeleton is given by the *Kinect* SDK.
- **skin registration skeleton**: composed of 22 control frames and 18 control points. Control frames (*Fig. 4, red dots*) are defined on the generic 3D skin and they corresponds to some of the *Kinect* body tracking joints. Control points are defined on the generic 3D skin contour (*Fig. 4, green dots*) and corresponds to the silhouette key points

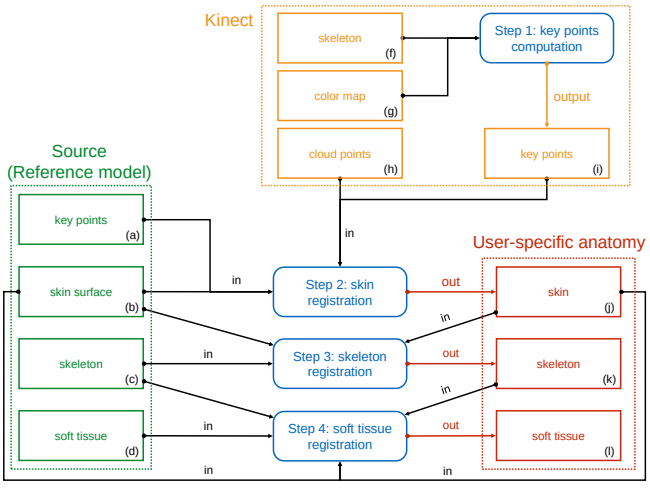

Fig. 1. Pipeline of the user-specific anatomy generation.

computed in *Section 2.2*. This system is used for skin registration.

- **Internal anatomy registration skeleton**: composed of 96 joint constraints between bones and 373 control frame positions (see *Fig. 7*). This system is used to keep anatomical consistency during internal skeleton registration and is defined based on anatomical rules presented in *Section 2.3*.

2.1. Body size

The *Kinect* SDK provides a simple body tracking skeleton, without temporal coherence: links may have different lengths at each frame. At calibration time: starting from a T-pose (*Fig. 2.a*), the user flexes his or her elbows (*Fig. 2.b*) and knees (*Fig. 2.c*). This allows us to estimate the length of upper and lower limb segments.

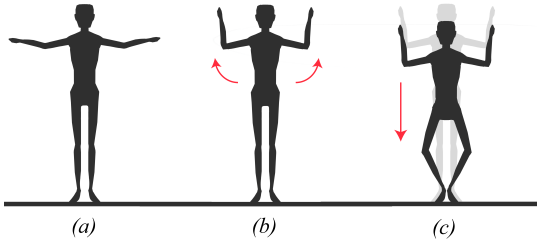

Fig. 2. ***Calibration process: (a)*** **to have global proportions (head and torso),** ***(b)*** **for real upper limb parts lengths,** ***(c)*** **for real lower limb parts lengths.**

The user silhouette and the body tracking skeleton given by *Kinect* are needed to compute body measurements (see *Fig. 3.b*) and define the 18 key points used for skin registration, as presented in *Section 2.2*. The *Kinect* body tracking skeleton is mapped from camera space to image space using *Kinect* SDK tools.

A key point corresponds to the intersection between the user's silhouette edge pixel and a perpendicular line computed using a Bresenham algorithm. For robustness, we have designed a silhouette detection criterion: an edge pixel is defined by a black pixel followed by three white pixels to avoid silhouette holes.

For each key point, the Bresenham algorithm is initialized using the middle of links as starting point and the perpendicular vector as the direction to follow. For instance using the point in-between the shoulder and elbow link gives us the upper arm width.

The 2D key points are mapped from image space to camera space using *Kinect* SDK tools, *Fig. 3.c* shows the key points we use.

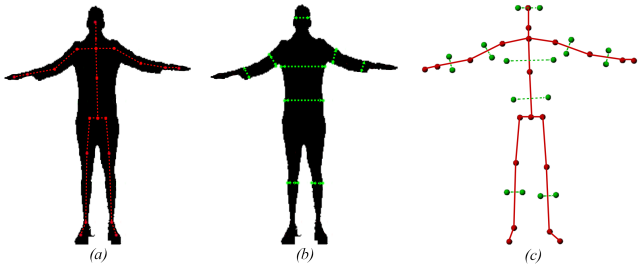

Fig. 3. ***(a)*****: skeleton key points.** ***(b)*****: body measurements key points.** ***(c)*****: 3D key points used in skin registration.**

Due to clothing and occlusion, some dimensions might be unreliable, especially thigh widths. Firstly, by assuming the human body symmetric along the sagittal plane, small errors in limb lengths are avoided. For each limb the average length value is used as real length in both sides. Other key point positions are inferred based on the user silhouette and basic anatomical knowledge. Based on an average human body, we defined ratios between body parts. For instance, knowing that the thigh measurement should be half of the hip measurement, the thigh width can be inferred. Some validations are shown in *Section 4*.

2.2. Skin registration

The skin registration method is based on the silhouette key points computed in *Section 2.1* and the *Kinect* point cloud. The main difficulties are the inaccuracy of the *Kinect* output data and the fact that people clothes are captured within the *Kinect* point cloud. To solve these issues, a new elastic deformer is introduced.

The skin is rigged using frame-based elastic deformers (Gilles et al. (2011)) corresponding to the *Kinect* body tracking skeleton joints (red dots in *Fig. 4*). Each skin vertex is controlled by several frames, using linear blend skinning. The skinning weights are computed using Voronoi shape functions as in Faure et al. (2011). The silhouette key points (green dots in *Fig. 4*) are mapped onto the skin to optimize the final result.

Instead of using global affine transformations (12DOFs) as in Ali-Hamadi et al. (2013); we use 9DOFs scalable rigids as frames, each bone matrix combines 3 translation, 3 rotation and 3 scale parameters. The advantage over affine control frames is obtaining a better non-uniform local scaling to avoid shearing artifacts.

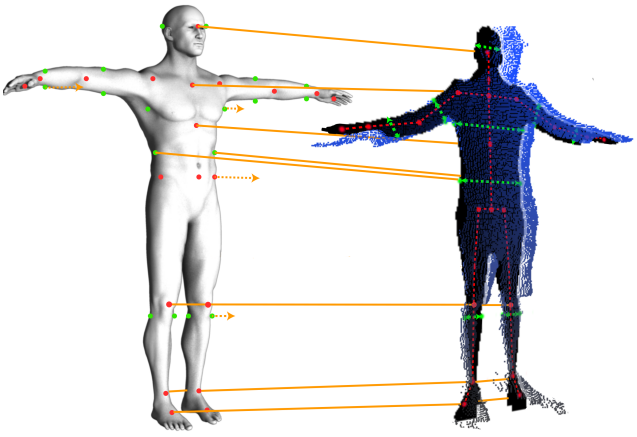

Fig. 4. Skin registration. Red dots: origins of control frames; green dots: silhouette key points; blue dots: *Kinect* point cloud.

The skin model is registered to the target by minimizing a weighted sum of three energies (see Gilles et al. (2011, 2013)) using an implicit solver.

The predominant energy $E_{skeleton} = \sum_{i=1}^{22} \frac{1}{2} K d_i^2$ (where K is the stiffness and d the distance to rest) defined by point to point zero length springs, attracts the control frames of the template to the bones of the user-specific model (red points in *Fig. 4*). Then the energy $E_{keypoint} = \sum_{i=1}^{18} \frac{1}{2} K d_i^2$, also defined by point to point zero length springs, attracts the silhouette points (green points in *Fig. 4*). Minimizing these first two energies scales the limbs, the torso, the neck and the head of the generic model according to the target body measurements as illustrated in *Fig. 5.a* and *b*. The energy $E_{cloudpoint}$ attracts the skin to the target point cloud using an ICP approach minimizing the distance between two sets of points. At each iteration, the point to point correspondence is reevaluated and the quadratic energy (as $E_{skeleton}$ and $E_{keypoint}$) attracts each skin point to the closest target point. The forces are propagated from the skin vertices to the skeleton control frames (see *Fig. 4* and *Fig. 5.c*). Thanks to the fact that a small set of control frames are used, awkward configurations are avoided and no smoothness or kinematic constraint terms are needed.

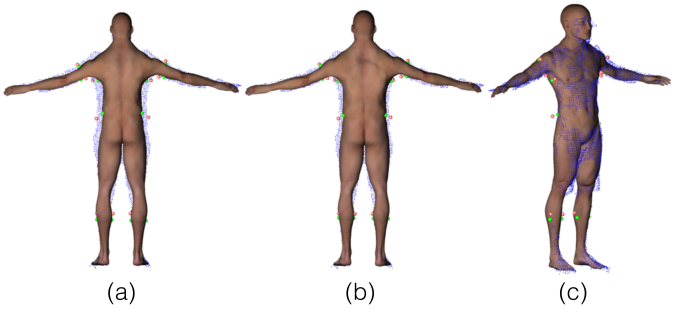

Fig. 5. Skin registration result at the end of each step of the optimization process. *(a)*: minimizing $E_{skeleton}$. *(b)*: minimizing $E_{skeleton}$ and $E_{keypoint}$. *(c)*: minimizing the three energies.

Fig. 6 presents the skin results after registration with the corresponding *Kinect* point cloud. By using $E_{cloudpoint}$, the torso skin is slightly deformed to refine the model in the same way, the user being a woman or a man.

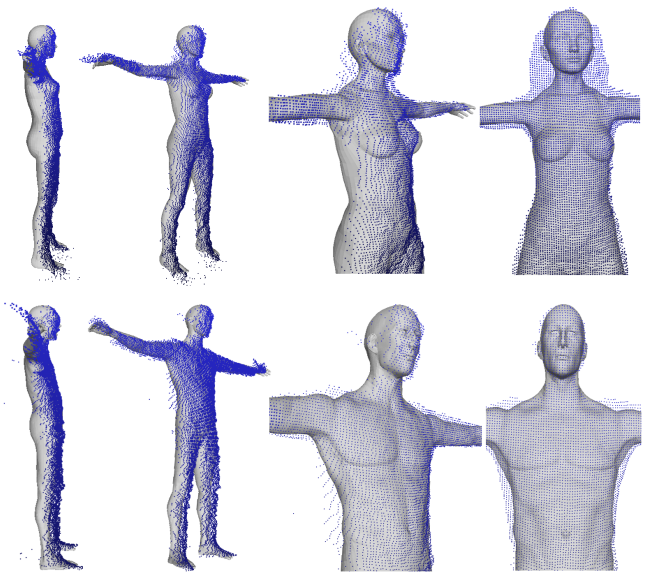

Fig. 6. *Kinect* point cloud and corresponding registered skin. Top: 1.55m female. Bottom: 1.85m male.

2.3. Internal Anatomy Registration

User-specific anatomy reconstruction is divided in two subparts: anatomical skeleton registration and soft tissue registration. Tissues are deformed as described in Ali-Hamadi et al. (2013); here the only focus is on internal skeleton registration. Inputs are the 3D reference of the skin and skeleton models and the estimate of the user skin registered obtained in *Section 2.2*.

First, our method uses a volumetric interpolation to estimate the user anatomical skeleton. As in Ali-Hamadi et al. (2013), the use of Laplacian interpolation (*Fig. 8.a*) with as boundary condition the transformations between the two skins ensures that all the internal anatomy is bounded inside the user's skin after transfer.

A major limitation of the Anatomy Transfer (Ali-Hamadi et al. (2013)) is the fact that the joint structure of the generic model is not maintained. Nothing prevents a bone from passing through another one (*Fig. 8.b*) or from being disconnected from a bone to which it should be connected (*for instance ribs and thoracic vertebra, or ulna and humerus around the elbow joint*, see *Fig. 8.c*). To keep correct joint structures and avoid these shortcomings, joint constraints between the elements of our elastic bone model are added. The joint location, kinematics and limits are set according to Nordin and Frankel (2001) (see *Fig. 7*).

Thus, the internal anatomy registration skeleton is defined using frame based elastic deformations (defined in Gilles et al. (2010)) with weights computed using a Voronoi shape function as in Faure et al. (2011) to smoothly propagate along the bone each control frame transformation. 9DOFs scalable rigids for the control are used to keep head bone consistency as it is in the generic model. This guarantees that the bone heads can only

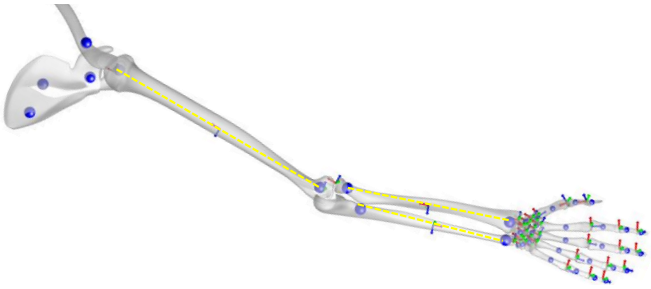

Fig. 7. Right arm internal anatomy registration skeleton. Blue dots for control frame positions; yellow lines and middle bone frames for alignment constraints; other frames for joint constraints.

translate, rotate and scale, and thus they keep a similar type of shape as in the generic bone model.

The list of anatomical rules used to define the internal anatomy registration skeleton follows:

- *R01*: Keep long bones straightness (no bending or twisting)
- *R02*: Keep 3D model consistency: the complete set of entities is transferred to avoid holes
- *R03*: Keep bone head consistency
- *R04*: Keep consistency of rib cage and limbs: symmetry with respect to the sagittal plane
- *R05*: Keep body joints consistency: type of joint and movement amplitude

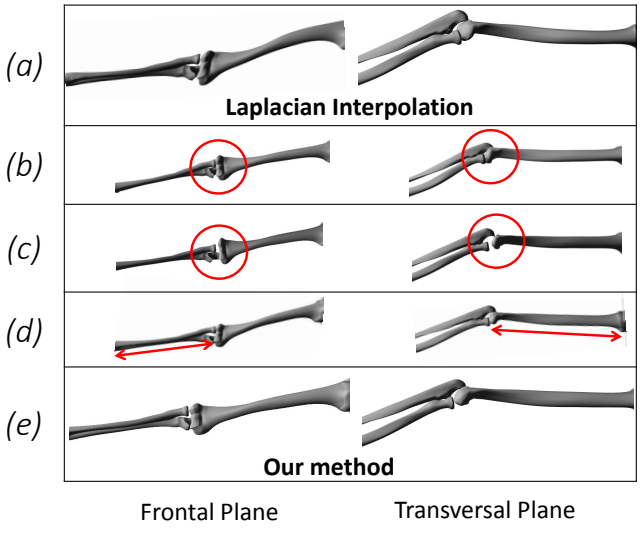

Fig. 8. *(a)*: Laplacian Interpolation, *(b)*: registration without joint constraint (overlaps between bones), *(c)*: registration without joint constraint (bone heads disconnection), *(d)*: registration without alignment constraint (bent and twisted bones), *(e)*: our method.

To avoid bending bones (*Fig. 8.d*), an alignment constraint is added between the two bone heads. This constraint restrains the possible displacements between the control frames in only one direction defined by the line between them (see yellow lines in *Fig. 7*). Thereby, the control frames can translate in one direction, but can still scale in all three directions. This alignment constraint is applied to long bones only.

It has been shown in Zhu et al. (2015) and similar approaches has been explored in Saito et al. (2015) that non-uniform scaling can be used to get more plausible bone deformations. This is why we introduced more control frames per anatomical bone. The number of frames varies according to bone type, the goal being to give enough deformability to each (*for the registration process*) while keeping good computation times (see blue dots in *Fig. 7*). For the short bones such as carpal bones, one frame per bone is used. For the long bones such as the femur two frames per bone are needed: one at the center of each bone head. For the flat bones such as the ribs three frames per bone are defined to keep ribs close to the skin in terms of curvature: two on bone heads (*e.g. close to the joints rib-vertebra and rib-sternum*), and one between the two others (*middle of the rib*). For bones with more complex shape such as vertebrae three frames per bone allows enough deformability to register the model while avoiding overlaps (*e.g. overlaps between facet joints, and spinous process of two different vertebrae*).

3. User Tracking

A single *Kinect* is used to perform body tracking. To reduce tracking noise, we record *Kinect* data in daylight, *Kinect* gives better results with background and ground matte materials. We observed that if the user's ground reflection is too visible, the *Kinect* includes it as part of the user silhouette which leads to lower limb length errors. The *Kinect* position is 60*cm* off ground for good lower-limb tracking results as determined in Pfister et al. (2014).

Because *Kinect* segments the depth map to compute body tracking joints at each frame, link distances change from frame to frame. This may lead to a disconnected anatomical skeleton (on the limbs) or elongated meshes (on the torso zone). We present the pipeline of our enhanced body tracking system in *Fig. 9*.

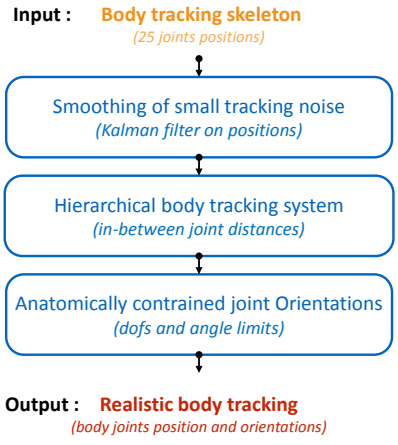

Fig. 9. Enhanced body tracking pipeline.

Firstly we define a hierarchical body tracking system by constraining the limb lengths and by recomputing joint orientations (see *Section 3.1* for more details).

To smooth out small tracking noise, we then apply a *Kalman* filter onto the joint positions. Joint orientations are recomputed from the filtered joint positions. Finally, we anatomically constrain the joint orientations: more details are given in *Section 3.2*.

3.1. Hierarchical body tracking system

Our hierarchical body tracking system is composed of 25 joints according to the *Kinect* SDK body tracking system.
To define each joint f, the position and the orientation of its parent p is required. To overcome this, we begin by computing the joints from the root (spine base joint) to the leaves (e.g. hand tips, foot joints and head joint). The root joint is defined by keeping the filtered *Kinect* position and orientation.

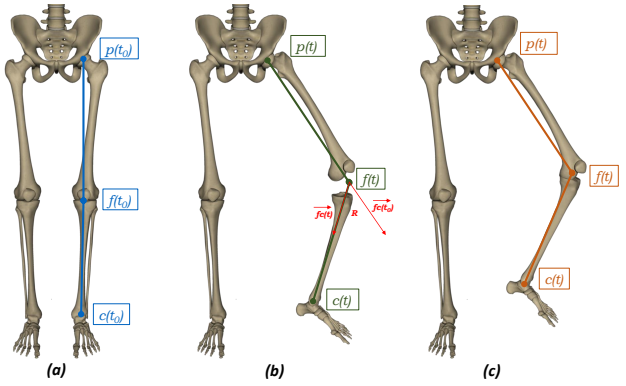

Fig. 10. ***(a)*: our hierarchical body tracking skeleton at** (t_0)**.** ***(b)*: *Kinect* body tracking skeleton at** (t)**.** ***(c)*: our result.**

At initialization time t_0 (see *Fig. 10.a*), each joint position is defined by our generic anatomical model and link distances computed after calibration (see *Section 2*); and each joint orientation is defined by the initial *Kinect* orientation determined in *Kinect* SDK.

The advantage of using a hierarchical skeleton is to obtain the body pose at each time t using only the joint rotations. We use the current *Kinect* body tracking skeleton to retrieve these rotations.

Most often, orientations given by *Kinect* are incorrect so we decided to recompute them using link directions by finding the smallest rotation R between initial direction ($fc(t_0)$) and current direction ($fc(t)$), see *Fig. 10.b*. *Fig. 10.c* shows our hierarchical body skeleton system at step t.

3.2. Anatomically constrained joint orientations

To correct non-anatomically plausible behaviors due to tracking errors, each *Kinect* hierarchical body tracking joint orientation is constrained by limiting the number of possible rotations based on anatomical motion knowledge (e.g. knee joint can be approximated as a 1DOF joint, whereas the hip joint is a 3DOFs joint). This is done by constraining a given quaternion using Euler-angle constraints to find the closest rotation matrix defined only with valid axis within the joint limits. Computation is made using the *Geometric Tools* library by Eberly (2008). *Fig. 11.a* illustrates in red a raw *Kinect* tracking and in gray the result after applying this constraint. To add even more anatomical plausibility to the result, joint limits are added to each rotation axis. *Fig. 11.b* highlights this constraint by showing *Kinect* raw data in red and realistic angular limits obtained in gray.

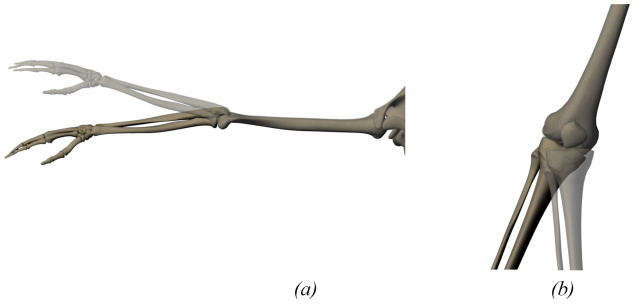

Fig. 11. ***Kinect* data in red and corrected in gray.** ***(a)*: off angular limits rotation.** ***(b)*: rotation axis error (DOFs).**

4. Results and Validation

To our knowledge, dealing with realistic anatomy visualization and motion is one of the most complex AR system ever because superimposing 3D anatomical data onto the user's color map reveals all the user measurement and tracking errors.

Our calibration method is a little time consuming (1-2*sec* for skin registration, 15-30*sec* for skeleton registration and 30-60*sec* for soft tissue registration) but allows us to obtain a 3D model with accurate user measurements; moreover the motion capture pipeline, even with the introduction of delay during quick motions, leads to realistic and stable user tracking.

Thanks to these two features, the presented method allows a realistic experience for understanding anatomy. The described method is implemented in C++ and runs on a commodity laptop (Intel CoreI7 processor at 3 GHz, Nvidia Quadro K2100M and 8GB of RAM). The real-time AR visualization runs between 35 to 62 fps depending on the 3D feedback: full-body musculoskeletal system (49211 vertices, 95189 faces) will run at 35 fps whereas internal organs (20144 vertices, 39491 faces) will run at 62 fps.

The computational bottleneck of our system is the quality of the 3D model (number of faces and vertices) alongside the quality of the user color map (*Kinect* gives a high definition color map, which is reloaded at each frame).

We provide the visual feedback on a commodity laptop screen and onto a 1.50m/2.0m screen for a demo display (see right side of *Fig. 12*).

Fig. 12 presents snapshots of the provided visualization. In a first set of experiments, the motion sequences were acquired for 4 men with an average height of 1.70m, and 3 women with an average height of 1.60m. To get uniform results we work with *Kinect* sequences made in similar environment conditions (daylight, background material reflections, *Kinect* position, *etc*...).

Fig. 13 presents two tracking data of the same user wearing different clothing and with different hair styles. It can be

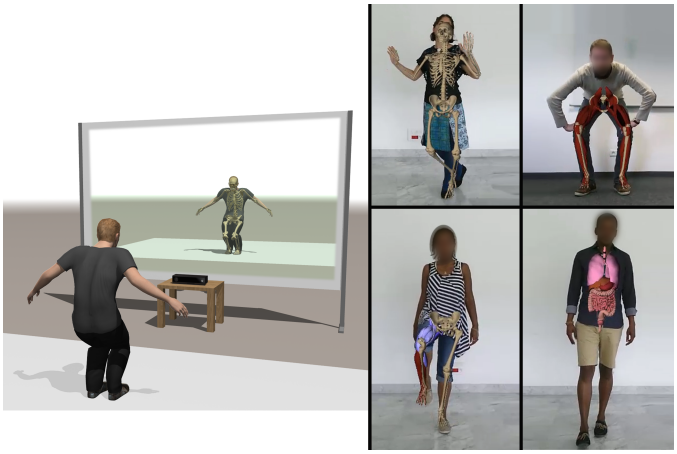

Fig. 12. Left: system set-up. Right: snapshots of results.

seen on the right side that the registered skeleton for these two datasets are almost identical; the red one is slightly bigger (1.2% for the limbs lengths and 2.5% for torso widths) than the other one (green). This comparison allows the validation of our skin registration process (see *Section 2.2*).

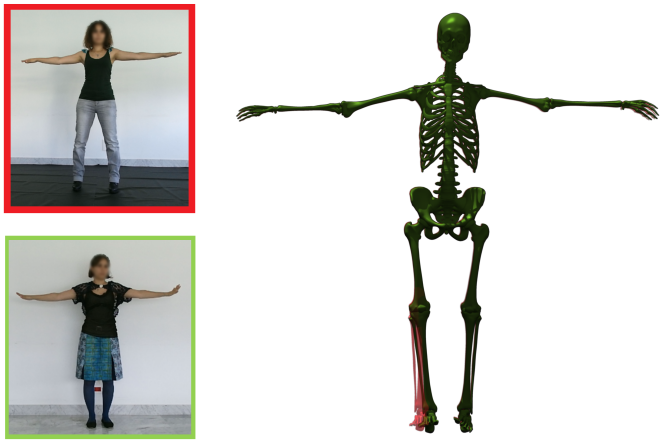

Fig. 13. For the same user with different clothing and hair style (Left), we obtain almost identical results (Right).

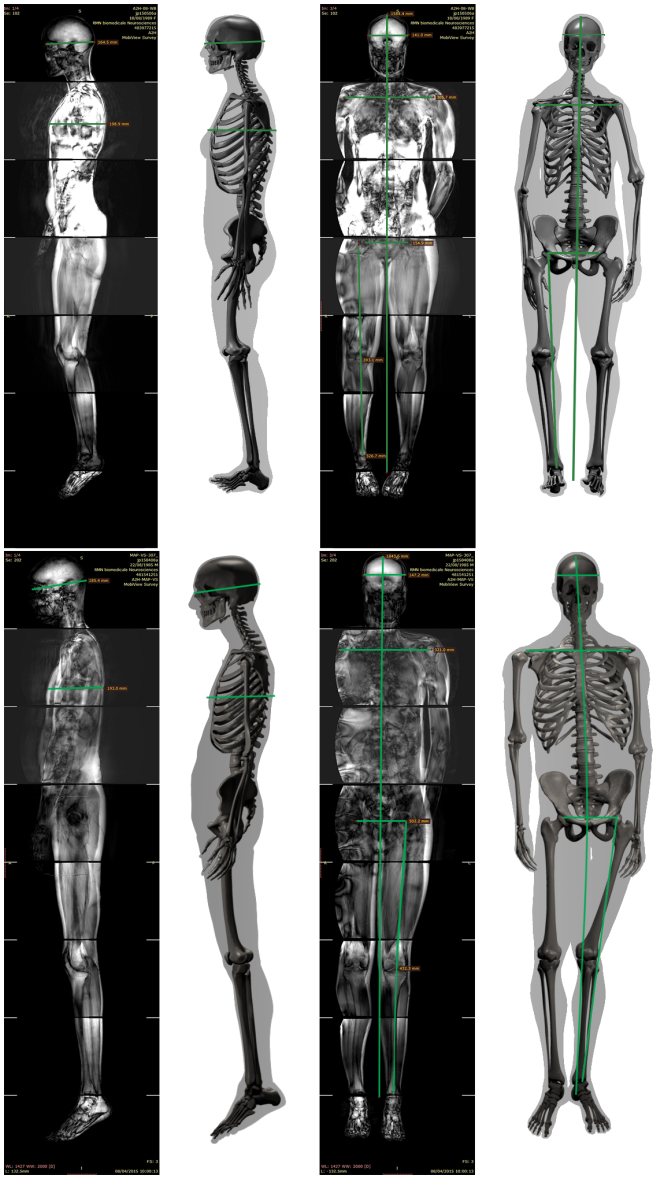

Fig. 14. Morphometric measurements (green lines) used to compare our results with ground truth MRI data. Top: 1.55m female. Bottom: 1.85m male.

4.1. Validation with MRI

The major contribution of our work, and also the most critical point is the closeness between the user-specific anatomy generated and the user's own. As explained in *Section 1*, using MRI data as ground truth allows us to obtain external as well as internal specific anatomical points for validation purpose.

Fig. 14 presents MRI data of two users (1.55m female and 1.85m male) in front and lateral views side to side with the corresponding 3D user-specific registered anatomies. The internal anatomy registration skeleton introduced in *Section 2.3* is used to set the 3D model in a pose similar to MRI acquisitions. After comparing body height, we found an average error of 1.5% between 3D and MRI data, 3D data being always smaller than MRI data. This quite small error is mainly due to limited skull deformations.

In the *Kinect* body tracking skeleton data, we observe a lot of change in limbs lengths. Thus, we pinpoint anatomical specific points (long bones protuberances) onto the MRI and onto the user-specific associated 3D model. With these specific points, we compare ulna and tibia lengths in real and 3D data. We suffer an average 5.2% error in limb lengths : most often the user-specific 3D model lacks a few centimeters. This percentage seems quite acceptable taking into account *Kinect* raw data noisiness.

To evaluate the torso body part realism, we propose to compare the user-specific 3D model and full-body MRI data by comparing the distance between left and right humerus bone heads. The average error between 3D and MRI data is rather small: 1.5%.

Fig. 6 shows that the point cloud and skin are fairly close; our generic skin being registered for a woman or a man, but

what about internal anatomy? We know that women hips are in average larger than men to allow birth. The 3.1% error between MRI and 3D data (the 3D data being always bigger than the MRI data) demonstrates that the distances between left and right femoral bones head difference between women and men is well transcribed in internal anatomy.

Using the lateral view, we pinpointed specific points to find rib cage depth. The user-specific rib cage is always bigger than the MRI data (around 20% bigger). It may be due to the difference in posture during acquisition : for MRI data, the user is lying whereas for *Kinect* data the user is standing. It may also come from the use of a partial point cloud instead of a complete one. Due to front view capture, we observe depth errors in the skull as well: the skull is about 12% bigger in depth in 3D than in MRI data.

4.2. Validation with User Study

In a second set of experiments 20 different subjects with no motor problems and working everyday on tools involving medicine or medical imaging were involved. For each subject, we captured a range of full body motions involving upper and lower limb motion as well as torso motion. *Fig. 15* presents snapshots obtained during the user study.

Fig. 15. snapshots of our system obtained during the user study.

The group is composed of 13 men between 24 and 54 years old (average height: 181*cm*, average weight: 82.6*kg*), and 7 women between 22 and 44 years old (average height: 164*cm*, average weight: 61.7*kg*). This user study was designed to evaluate the believability of our system. The 4-page user study is composed of yes/no questions (with text space for elaboration if no is the answer), augmented reality screenshot comparison and correctness evaluation, rating-based questions (1 to 5), and opened questions (for global feedback). Below we present eight criteria extracted from the user study results, which evaluation is given in *Tab.* 3.

Body position range (criterion *C01*) corresponds to motions while standing, crouching or sitting. In most cases, the results are well received. For other cases, limitations are directly connected to *Kinect* occlusion limitations (for more information cf. Associated supplemental materials).

Body orientation range (criterion *C02*) corresponds to body orientation from *Kinect* point of view: e.g. facing, profile, 3/4, back. When *Kinect* raw data are occluded or self-occluded, our system returns incorrect motion poses: the more occlusion in *Kinect* raw data, the more errors we will have (for more information cf. Associated supplemental materials). A major topic is to be able to handle important occlusion zones, this motivates the work presented in *Section 5*.

Motion range (criterion *C03*) defines simple motions like flexion/extension of the knee, as well as complex motions in the extremities like finger motion or supination/pronation of the arm. We obtain high motion quality for simple motions; for complex motions we are limited by *Kinect*: this criterion suggest further improvements. The *Kinect* SDK outputs a small number of joints which limits the body motion possibilities (e.g. spine bending).

For **Motion fluidity and delay** (criterion *C04*) and **Motion consistency** (criterion *C05*), the goal is reached. Motion consistency refers to the absence of outliers during motion. We should state the fact that part of the visual latency that might occur comes from the low frame rate of the color map display.

Motion plausibility (criterion *C06*) corresponds to joint DOFs and angular limits. For this criterion we obtain different results depending on the body segment studied. For instance, it is easier to implement constraint for 1DOF joints than for 3DOFs joints such as spine or shoulders joints due to motion range. Work presented in *Section 5* allows us to obtain better results on this criterion.

Anatomy realism (criterion *C07*) gives a feedback on the registration method by focusing on limb length and torso width. For this criterion, people with professional knowledge in anatomy were the only ones to rate the user-specific anatomy as average.

For almost everyone, the **Augmented reality** (criterion *C08*) results were of good level. The overall quality can even be increased with mesh texturing, or by adding a transition effect between virtual and real data (e.g. 3D anatomy and the user's color map).

	C01	*C02*	*C03*	*C04*	*C05*	*C06*	*C07*	*C08*
−−	00%	15%	05%	00%	00%	10%	00%	00%
+−	20%	50%	30%	10%	10%	25%	15%	05%
++	80%	35%	65%	90%	90%	65%	85%	95%

Table 3. User study compiled results according to quality criteria for a mirror-like augmented reality system. For each criterion: the percentage of bad/average/good reviews.

5. Towards Image-based Corrective Registration

In *Section 2* and *3* we proposed a system that can efficiently register user-specific anatomy and provide interactive visualization of it in motion based on data captured by *Kinect*. In

this section, we will refer to this system as the "*3D registration system*".

The feedback obtained during the user study (see *Section 4.2*) suggests that a supplementary process for better quality of the overlay is needed. Anatomy misalignments are in particular visible when presenting anatomy superimposed onto the user's color map (AR). For example, anatomical limbs can sometimes be out of the user's silhouette as shown in *Fig. 16* (e.g. arms and hands). We avoid performing these corrections in our 3D registration system as they make it overly constrained.

Image-based Correction In this section, we describe a method to solve this problem efficiently in image domain. Our image-based corrective registration allows to reduce the errors that accumulate during previous system steps such as motion capture, anatomy transfer, image generation and animation.

Occlusion Handling We perform these image-based corrections for separate regions of the registered 3D anatomy (see *Fig. 16*). To generate the final overlay image we have to determine in which order these corrected images should be layered. Therefore, we propose an image-based occlusion handling technique that automatically decides how to overlay these images relatively to each other in the final AR rendering.

5.1. Input Data From 3D Registration

By projecting and rendering registered 3D anatomy on the 2D color map we generate *anatomy images* (*Fig. 16* top row) corresponding to a set of predefined 3D *anatomy regions*. Each anatomy region is composed of a specific set of bones, organs and muscles of the registered 3D anatomical model. For example, in *Fig. 16*, we show the anatomy images corresponding to the five anatomy regions we use in our examples. Unfortunately, after superimposing them onto the color map we can clearly observe that they are misaligned (see *Fig. 16* bottom-left). Using our image-based correction, we deform or warp these anatomy images to correct the misalignments. In *Section 3* we defined our enhanced body tracking system composed of 25 joints. By mapping it to image space, we generate *Anatomy landmarks*. The same is done with the *Kinect* SDK body tracking skeleton to generate *Kinect landmarks*. Our image-based correction method uses these two sets of landmarks and the *Kinect* depth map to correct the anatomy misalignments.

As mentioned earlier, to solve the problem of anatomy misalignment due to accumulation of errors in the 3D registration system, we propose an image-based correction method. Correcting these errors in a reduced dimension (in image space) causes small projective errors to occur, but these corrections can be performed very efficiently. Furthermore, the result of the 3D registration system is a 2D image that mixes the real world captured by the *Kinect* with a simulated image of the animated model; therefore making the correction at the very end of the process makes it possible to correct errors of all the stages including the final visualization in 2D. The two main stages of our algorithm are: (a) image-based correction and (b) occlusion estimation and layering. The overview of this pipeline can be seen in *Fig. 17*.

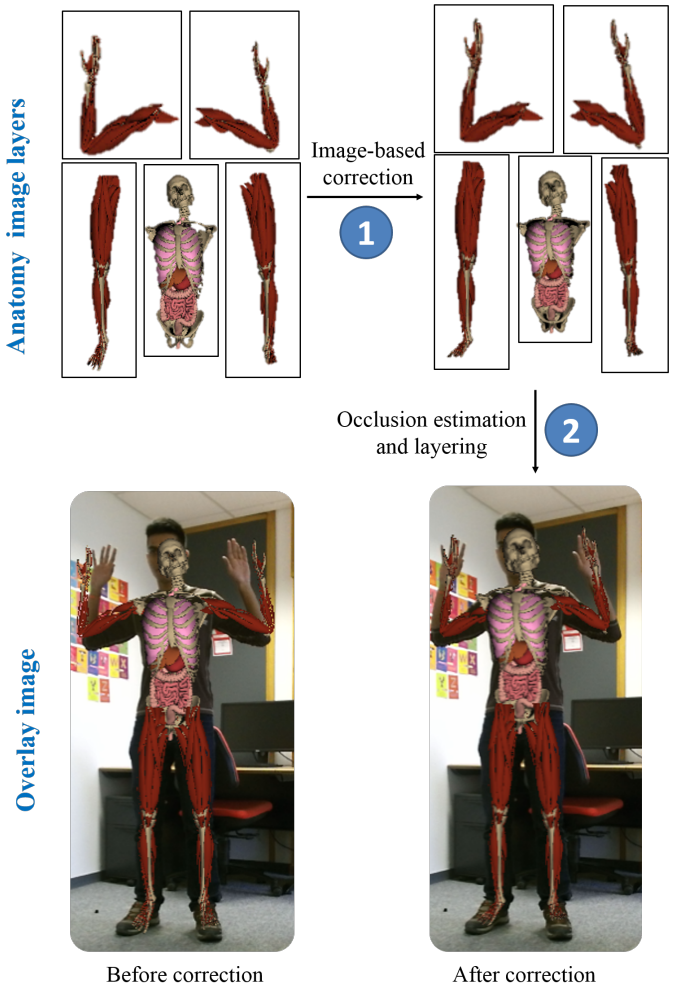

Fig. 16. We propose an image-based correction (step 1), and an occlusion estimation and layering technique (step 2). In the first step, we correct anatomy regions separately. In the second step, we combine them in correct order to generate the final overlay image.

5.2. Image-based Correction

The image-based correction stage can be further divided into three steps: (a) feature estimation, (b) landmark correction and (c) updated anatomy image generation.

5.2.1. Feature Estimation

We estimate two types of image features: first, we find a set of *anatomy features* $\mathcal{S}$ in the anatomy images, and second, a set of *depth features* $\mathcal{D}$ in the *Kinect* depth map. Let's consider that we have N (=5 in our examples) anatomy regions. Subsets of the *Kinect* and anatomy landmarks are assigned to each of the anatomy regions based on the contribution of the 3D joints corresponding to these landmarks in producing soft tissue movements in that region. We also describe below, how we can estimate N depth contours corresponding to the anatomy regions and estimate depth features from these regions.

We estimate depth contour points by first detecting edges corresponding to the depth discontinuities in the *Kinect* depth map using Canny edge detector (Canny (1986)) and then computing the external contour using contour detection algorithm

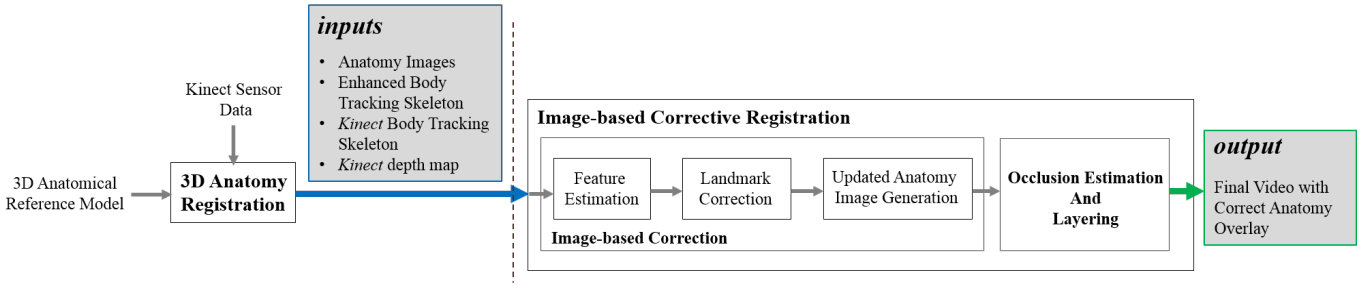

Fig. 17. Overview of our corrective registration system.

proposed in Suzuki et al. (1985). Contour segmentation is a well researched topic and more recent work, such as Hickson et al. (2014); Abramov et al. (2012); Hernandez-Lopez et al. (2012) can also be used. For each contour point we find the closest *Kinect* landmark. Conversely, after that, for each *Kinect* landmark we obtain a set of depth contour points. For an anatomy region, we estimate *depth contour* by taking union of all depth contour points corresponding to the *Kinect* landmarks of that region. Similarly, we estimate *anatomy contour* of an anatomy region by estimating the contour around the rendered anatomy in the corresponding anatomy image.

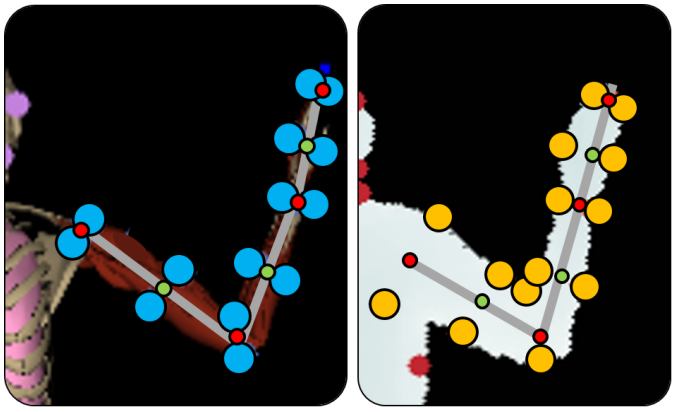

Fig. 18. Feature estimation for left arm. In left: anatomy features ($\mathcal{S}$) (blue) are estimated using anatomy landmarks (red) and sub-landmarks (green). In right: depth features ($\mathcal{D}$) are estimated using *Kinect* depth landmarks (red) and sub-landmarks(green). Here we are showing one sub-division of landmarks.

Anatomy Feature Estimation We denote the anatomy features for the ith anatomy region as $\mathcal{S}_i$, which are estimated in the corresponding anatomy image. For each anatomy landmark of the anatomy region we estimate a normal vector with a direction which is average of the normals to the lines connecting the landmark to the adjacent anatomy landmarks. This vector intersects the anatomy contour at two points and we add these points to $\mathcal{S}_i$. If desired, the lines can be further sub-divided to generate *sub-landmarks* increasing the number of features, in our implementation we subdivided the lines 6 times. Adding more features increases the robustness by reducing the contribution of outliers, see *Fig.18*.

Depth Feature Estimation We denote depth features of the ith anatomy region as $\mathcal{D}_i$. Similar to the estimation of anatomy features, we can estimate depth features by finding intersections of the normal vectors from *Kinect* landmarks with the depth contour of the anatomy region (see *Fig. 19*). Depth contours are mostly fragmented and not closed because the transition between anatomy regions generates depth discontinuities. Since we have missing points in the contour, sometimes normal vectors do not intersect with depth contours. In that case, we do not add any depth features to the depth landmark. At the same time, we drop anatomy features of the corresponding anatomy landmark to ensure one-to-one correspondences between anatomy and depth features. Depth maps are often noisy and causes erroneous depth feature estimation due to noise in the *Kinect* depth sensor raw data. We apply a *Kalman* filter onto the depth feature locations to remove the noise effect.

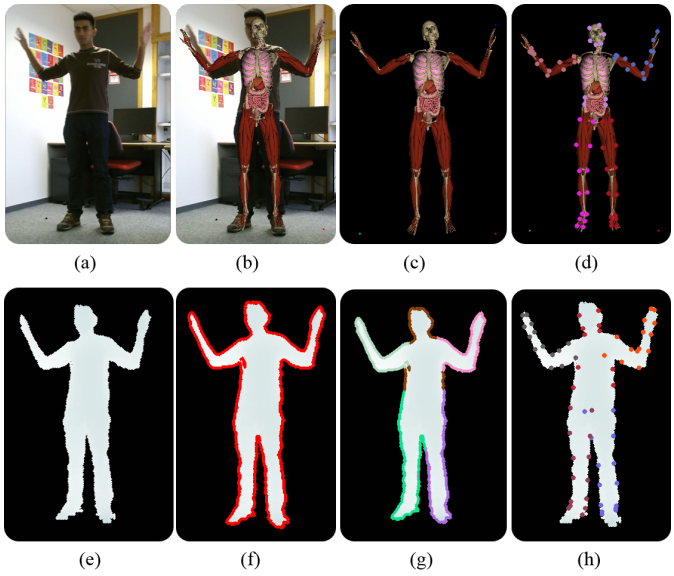

Fig. 19. *Feature estimation:* Preliminary 3D registration when superimposed on input image (a) we can observe anatomy misalignments (b). We estimate anatomy features (d) from the intermediate anatomy images (c). In the bottom row, we show the estimation (f) and segmentation (g) of depth contours from the depth map (e), and estimated depth features (h).

5.2.2. Landmark Correction

Kinect landmarks provide a reasonable estimate of the skeleton in the 2D image space, we will call it *Kinect 2D skeleton*. Similarly, we will call the corresponding skeleton formed by

the anatomy landmarks as *anatomy 2D skeleton*. But if we warp our anatomy regions using a map learned from anatomy landmarks to *Kinect* landmarks, it does not ensure that the mapped regions will entirely reside within the depth contours. To maintain smoothness in shape at boundary of the warped region, we look into a reliable warp map, we call $\mathcal{T}$.

We use a thin plate spline-based interpolation (see Bookstein (1989)) represented with radial basis functions (RBF) to approximate the mapping $\mathcal{T}$ from anatomy features to depth features. This mapping is composed of an affine transformation and a non-affine deformation. We use a regularizer parameter λ to control the influence of the non-affine deformation part, which refers to a physical analogy of bending. More recent techniques, such as Sorkine and Alexa (2007) can also be used to compute the image warp map.

For an anatomy region with M feature points. The parameters of the RBF function are estimated by solving the following *Eq. 1*:

$$\mathcal{D}_i = A\ \mathcal{S}_i + \lambda \sum_{i=1:M} \sum_{j=1:M} (w_j\ R(\|s_i - d_j\|)) \tag{1}$$

Where, $\mathcal{D}_i$ and $\mathcal{S}_i$ are Mx2 matrices containing respectively locations of depth and anatomy features. s_i and d_i are locations of ith anatomy and depth features, A is a MxM affine transformation matrix. w represents weight of non-affine bending transformation, and $R(r) = r^2\ log(r)$ is a RBF function. Additionally, we include *Kinect* landmarks and anatomy landmarks respectively in the matrices $\mathcal{D}$ and $\mathcal{S}$.

We can rewrite *Eq. 1* in matrix form as:

$$\mathcal{D}_i = \begin{bmatrix} K(R) & P^T(\mathcal{S}_i) \\ P(\mathcal{S}_i) & 0 \end{bmatrix} \begin{bmatrix} W \\ A \end{bmatrix}, \tag{2}$$

Where, P contains homogenized $\mathcal{S}_i$, K contains values of RBF functions and W is a vector with non-affine weights. We can further simplify *Eq. 2* as $\mathcal{D}_i = M_i(\mathcal{S}_i)\ X_i$, where M_i and X_i represent the first and second matrix of the right hand side of *Eq. 2*. Now, by combining equations of all the body parts in one global equation, we can write:

$$\begin{bmatrix} \mathcal{D}_1 \\ \vdots \\ \mathcal{D}_N \end{bmatrix} = \begin{bmatrix} M_1(\mathcal{S}_1) & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & M_N(\mathcal{S}_N) \end{bmatrix} \begin{bmatrix} X_1 \\ \vdots \\ X_N \end{bmatrix} \tag{3}$$

We can rewrite *Eq. 3* as:

$$\mathcal{D} = \tilde{M}(\mathcal{S})\ \tilde{X} \tag{4}$$

In our current implementation N=5, size of $\tilde{M}$ and $\mathcal{D}$ are 240×240 and 240×2 respectively. Finally, we can also write *Eq. 3* as $\mathcal{D} = \mathcal{T}(\tilde{M}(\mathcal{S}), \tilde{X}$. where, $\mathcal{T}$ maps anatomy features in $\mathcal{S}$ to the depth features in $\mathcal{D}$, and $\tilde{X}$ includes the parameters of the mapping.

We solve *Eq. 4* to estimate mapping parameters $\tilde{X}$ of $\mathcal{T}$ by formulating it as a linear least squares system for given $\mathcal{S}$ and $\mathcal{D}$ which includes anatomy and depth features that have one-to-one correspondences. In our regularized optimization framework, we constrain the anatomy landmarks with soft constraints to map *Kinect* landmarks. $\mathcal{T}$ can be used to warp anatomy regions such that they remain enclosed within the depth contour while maintaining a smooth boundary shape. Note that $\mathcal{T}$ is composed of N separate mappings corresponding to the anatomy regions. By remapping the anatomy landmarks using $\mathcal{T}$, we also obtain a better estimate of the original *Kinect* 2D skeleton formed by *Kinect* landmarks, which we call *updated 2D skeleton*. Furthermore, to ensure the connectivity of landmarks across different anatomy regions, we set the location of shared landmarks to the average of their estimates for different anatomy regions. *Fig. 20* shows the resulting landmark corrections.

(a) (b) (c)

Fig. 20. *Landmark correction*: Our skeleton correction algorithm corrects initial *Kinect* 2D skeleton in image space (b) to produce more consistent configurations (c).

Note that depth contours are noisy if the user wears loose clothes, which in turn makes the depth features noisy. Therefore, we prefer to maintain a smooth shape of the mapped anatomy region instead of mapping anatomy features exactly to the depth features. By picking a suitable λ in *Eq. 1* we can control the smoothness. In our current implementation we chose $\lambda = 0.01$.

5.2.3. *Updated Anatomy Image Generation*

As explained previously, to each anatomy region corresponds an anatomy image, therefore, to warp an anatomy region, we simply need to warp the corresponding anatomy image. We decided to separately warp each of these images based on the transformation from anatomy 2D skeleton to updated 2D skeleton. To obtain the final anatomy corrected rendered image, we combine these warped anatomy images to a single composite image. For each pixel of that composite image, we should render the anatomy region closest to the camera (e.g. smallest depth value). To estimate the closest anatomy region, we propose a novel occlusion estimation and layering algorithm.

Image Warping We generate bounding boxes, which we call *anatomy cages*, around links of the updated 2D skeleton. Now, our goal is to deform these anatomy cages based on the deformation field. *Fig. 21* shows warping of the right upper limb with cages.

Using $\mathcal{T}$ as a deformation field is not a good choice for two reasons: first, $\mathcal{T}$ does not include the additional deformations

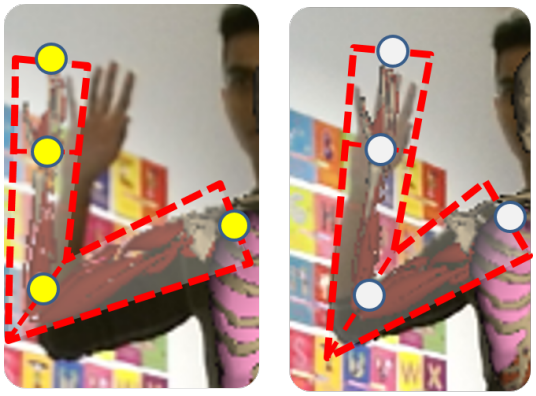

Fig. 21. ***Cage generation and warping of the right upper limb. Yellow points are anatomy landmarks, White ones are updated landmarks. The red zones present anatomy cages.***

we introduced in the post-processing step (where we modify the locations of mapped anatomy landmarks). Second, $\mathcal{T}$ is reliable within the regions enclosed by the anatomy landmarks, but if we use these landmarks to generate the cages, we will end up having a piecewise linear approximation of the boundary of the anatomy in the image space. Therefore, we chose to use wider cages to get a smoother anatomy boundary after warping. This puts the cage points outside the regions enclosed by the landmarks and so we cannot use $\mathcal{T}$ reliably. Therefore, instead of $\mathcal{T}$, we use dual-quaternion skinning (Kavan et al. (2008)) driven by the skeleton to perform image warping from anatomy to updated 2D skeleton landmarks. Using this, we estimate *deformed anatomy cages* corresponding to each of the anatomy cages. To warp the anatomy images, we triangulate the anatomy cages, and then we estimate affine transformation of the individual element of the cages from original anatomy cage to the deformed anatomy cage configuration. Using bilinear interpolation we then warp the anatomy image pixels inside the triangles. We call the new warped images as *warped anatomy images*. We define one for each anatomy region. *Fig. 22* shows warping results for the complete set of anatomy images.

5.3. Occlusion Estimation and Layering

As mentioned before, the main challenge in generating final *composite image* of the warped anatomy images is to figure out which anatomy region is closest to the camera for a given view. If images are naively combined as layers, occlusions such as in *Fig. 23, (b)* can occur. In this case, the anatomy region corresponding to the torso is occluding the hands, which is not what we expect. Our method described below tackles this problem.

We first generate synthetic depth images for the anatomy regions based on the *Kinect* 2D skeleton. For each anatomy region, we know 3D configuration of the corresponding *Kinect* joints in the *Kinect* 2D skeleton. We model cylindrical primitives around the bones. The radius is set equal to the maximum cross-section radius of the corresponding anatomy region. Using the projection matrix of the camera, for each anatomy region, we render a depth map. We call this image *anatomy depth image*.

(a) (b)

Fig. 22. ***Image-based corrective registration*: Misalignments in the anatomy regions observed in (a) are corrected by our image-based corrective algorithm to produce (b).**

The size of the warped anatomy images and the composite image are the same. For each warped anatomy image, we categorize the pixels into two types: *valid* when pixels belong to anatomy, and *invalid* when they do not. In the composite image domain we loop through all the pixels: for each pixel, we check if at that location any of the warped anatomy images contains a valid pixel. If not, we set that pixel to black. If yes, we check which of the warped anatomy images contain valid pixels. Out of all those warped anatomy images we pick the one that is closest to the camera. The distance from the camera is determined based on anatomy depth images. We then update the pixel of the composite image with the color of that closest warped anatomy image. In *Fig. 23* (c) we can see how our algorithm corrected the problem of occlusion (b).

(a) (b) (c)

Fig. 23. ***Occlusion handling*: Our image-based corrective algorithm corrects misalignments in the rendering (a) of initial 3D registration by warping anatomy regions in image space and in separate layers. Rendering them without knowing their relative distances from camera create occlusions (b). Our occlusion handling algorithm can recover these relative distances and render these regions in correct order (c).**

5.4. Evaluation

We have shown qualitative results of our landmark correction in *Fig. 20*, where we used non-linear thin plate spline-based interpolation to model the deformation. *Fig. 22* shows how we improve anatomy registration by applying dual-quaternion skinning based on deformations produced by landmark correction. The results of our occlusion handling algorithm is shown in *Fig. 23*. We quantitatively analyze the results of our image-based corrective registration using an *anatomy intersection coefficient* η. If n_f is the total of anatomy pixels in the final composite image and n_k is the total of anatomy pixels that also belong to the user according to the *Kinect* depth map. We can define η as: $\eta = \frac{n_k}{n_f}$. In *Tab. 5.4*, results are shown for two video sequences and they clearly indicate the improvement after applying our image-based corrective registration algorithm. Our unoptimized routines take around 75ms per frame to perform this correction.

Video	Before	After
Squat motion	0.939 ± 0.184	0.994 ± 0.026
Hand crossing	0.961 ± 0.159	0.988 ± 0.089

Table 4. Image-based corrective registration results: anatomy intersection coefficient before and after corrections

Fig. 24 shows the temporal profile of η for individual anatomy regions in the squat motion video. Anatomy regions are represented with different colors. As we can see, η consistently remains close to 1 after our corrections. Thus, image-based correction drastically reduced the alignment errors.

Furthermore, we can make our corrective registration faster by generating warped anatomy images only when the η value of an anatomy region is below a certain threshold (which means they are not well aligned). For example, in the 81 frames of the squat motion video we originally estimate 405 warped anatomy images (e.g. 81 (number of video frames) × 5 (number of anatomy regions)). If we set the threshold of η to be 0.9, we reduce this number to 137: this is a 66.2% reduction.

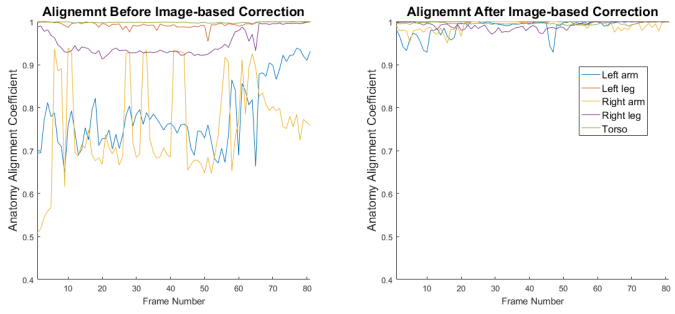

Fig. 24. The anatomy alignment coefficients for anatomy regions are shown before and after image-based correction for the squat sequence.

In the 3D registration system, the errors in orientation of the anatomy regions produce wrong color maps of the anatomy regions. Since we use these color maps as color or texture of the anatomy regions, we cannot correct orientation errors in the image-based correction step. Currently, integration of the proposed image-based corrective registration step within the 3D registration system is not real-time. The image-based correction uses color maps rendered by the 3D registration system. In our current implementation, we save these color maps to the disk and read them later for image-based corrections. These computationally expensive file operations prevent real-time image-based misalignment correction. Furthermore, with the current latency, the combined system does not satisfy "motion fluidity and delay"(criterion *C04* as mentioned in *Section* 4.2).

6. Conclusion

We present the first live system of personalized anatomy in motion. Superimposing the anatomy onto the user's image allows us to create a real-time augmented reality experience. The first paper version Bauer et al. (2016) attached video (see *https://youtu.be/Ip17-Vaqqos*) illustrates the application pipeline and shows AR results of our system (before image-based corrective registration presented in *Section 5*).

We also proposed an image-based corrective registration to correct the errors that build up during system steps: motion capture, anatomy transfer, image generation and animation. Currently, the combined pipeline is not real-time due to expensive file read and write operations. Using unoptimized code for image-based corrective registration we currently achieve a frame rate of 12fps on average for the combined pipeline. In future, we plan to read color maps from memory instead, and build a combined real-time system. Another limitation of the current image-based correction is that we cannot correct the errors in orientation of the anatomy regions relative to the bones. In future, to solve this problem, we can use an image-based hybrid solution, such as Zhou et al. (2010); Jain et al. (2010); Richter et al. (2012) that use a 3D morphable model to fit to some features in 2D images. In our case, we can model our 3D anatomical reference model as a morphable model and then fit it based on 2D joint locations of the updated *Kinect* 2D skeleton. Then, we can re-render the anatomy regions from camera view to generate updated anatomy images. This should be able to recover the color of anatomy regions that get occluded due to orientation error in 3D registration system. After that we can follow our usual skinning and occlusion handling routines to generate final results. We believe that the basic *Kinect* body tracking enhanced with our method is sufficiently accurate for our needs.

The system could be extended or improved in different ways. Posture reconstruction proposed by the *Kinect* SDK could be replaced by more sophisticated approaches such as presented in Liu et al. (2016). This could make the system not only more robust but also more independent of the selected sensor. Another solution could be the use of physical priors such as introduced in Andrews et al. (2016). It would certainly enable suppressing some outliers resulting of *Kinect* data. On the other hand, the addition of biomechanical simulations could allow to get more realistic deformations of soft tissue and organs but this could be at the cost of interactivity.

To show full body muscular activity for every possible body

motion, inverse dynamics (Murai et al. (2010)) will also be developed.

An improvement in the skin registration can be done by reducing the 9DOFs controllers to 6DOFs (3 rotations and 3 scales). This can be done by exploiting appropriately the hierarchical structure and would allow more robustness and skin consistency around body joints.

Our work is designed to be used as a tool for anatomy learning for medical and sports students. This is why in the future it is planned to display anatomical educational content (text, images, videos, *etc*) in addition to the AR visualization.

This system could also be used as a way to communicate between medical practitioners and their patients, about surgery, rehabilitation or any other health issue. Novel artistic content might also be produced using our technology, as well as interactive advertising.

Our system has been featured as a live demo during two conferences and at the *Consumer Electronic Show*. More than 400 people have been able to test it out. Most of them enjoyed the experience and a lot of them recommended it and came back with others.

Acknowledgments

Many thanks to Estelle Charleroy our Graphist Artist, and to Olivier Carre for engineering advices. We also would like to thanks all people involved in the user study, and all people helped testing the application. MRI data used for validation are from the A2H study promoted by Grenoble University Hospital.

This work was partially supported by the ANR "Investissements d'Avenir" program, LabEx PERSYVAL-LAB (ANR-11-LABX-0025). It was also supported in part by grants from NSERC, Canada Foundation for Innovation, MITACS, and the Canada Research Chairs Program.

References

Bauer, A, Dicko, AH, Faure, F, Palombi, O, Troccaz, J. Anatomical Mirroring: Real-time User-specific Anatomy in Motion Using a Commodity Depth Camera. In: Proceedings of the 9th International Conference on Motion in Games. MIG '16; New York, NY, USA: ACM. ISBN 978-1-4503-4592-7; 2016, p. 113–122. URL: http://doi.acm.org/10.1145/2994258.2994259. doi:10.1145/2994258.2994259.

Pfister, A, West, AM, Bronner, S, Noah, JA. Comparative abilities of Microsoft Kinect and Vicon 3D motion capture for gait analysis. Journal of medical engineering & technology 2014;38(5):274–280.

Malinowski, MJ, Matsinos, E. On using the Microsoft Kinect sensors to determine the lengths of the arm and leg bones of a human subject in motion. Medical Physics 2015;.

Helten, T, Baak, A, Bharaj, G, Muller, M, Seidel, HP, Theobalt, C. Personalization and Evaluation of a Real-Time Depth-Based Full Body Tracker. In: Proceedings of the 2013 International Conference on 3D Vision. 3DV '13; Washington, DC, USA: IEEE Computer Society. ISBN 978-0-7695-5067-1; 2013, p. 279–286. URL: http://dx.doi.org/10.1109/3DV.2013.44. doi:10.1109/3DV.2013.44.

Gilles, B, Bousquet, G, Faure, F, Pai, DK. Frame-based Elastic Models. ACM Trans Graph 2011;30(2):15:1–15:12. URL: http://doi.acm.org/10.1145/1944846.1944855. doi:10.1145/1944846.1944855.

Li, H, Vouga, E, Gudym, A, Luo, L, Barron, JT, Gusev, G. 3D Self-portraits. ACM Trans Graph 2013;32(6):187:1–187:9. URL: http://doi.acm.org/10.1145/2508363.2508407. doi:10.1145/2508363.2508407.

Vlasic, D, Baran, I, Matusik, W, Popović, J. Articulated Mesh Animation from Multi-view Silhouettes. In: ACM SIGGRAPH 2008 Papers. SIGGRAPH '08; New York, NY, USA: ACM. ISBN 978-1-4503-0112-1; 2008, p. 97:1–97:9. URL: http://doi.acm.org/10.1145/1399504.1360696. doi:10.1145/1399504.1360696.

Bălan, AO, Black, MJ. The naked truth: Estimating body shape under clothing. In: Computer Vision–ECCV 2008. Springer; 2008, p. 15–29.

Zeng, M, Cao, L, Dong, H, Lin, K, Wang, M, Tong, J. Estimation of human body shape and cloth field in front of a kinect. Neurocomputing 2015;151:626–631. URL: http://www.sciencedirect.com/science/article/pii/S0925231214013824. doi:http://dx.doi.org/10.1016/j.neucom.2014.06.087.

Sotiras, A, Davatzikos, C, Paragios, N. Deformable Medical Image Registration: A Survey. Medical Imaging, IEEE Transactions on 2013;32(7):1153–1190. URL: http://ieeexplore.ieee.org/xpl/articleDetails.jsp?arnumber=6522524. doi:10.1109/TMI.2013.2265603.

Quah, CK, Gagalowicz, A, Roussel, R, Seah, HS. 3D Modeling of Humans with Skeletons from Uncalibrated Wide Baseline Views. In: Proceedings of the 11th International Conference on Computer Analysis of Images and Patterns. CAIP'05; Berlin, Heidelberg: Springer-Verlag. ISBN 3-540-28969-0, 978-3-540-28969-2; 2005, p. 379–389. URL: http://dx.doi.org/10.1007/11556121_47. doi:10.1007/11556121_47.

Zhu, L, Hu, X, Kavan, L. Adaptable Anatomical Models for Realistic Bone Motion Reconstruction. Comput Graph Forum 2015;34(2):459–471. URL: http://dx.doi.org/10.1111/cgf.12575. doi:10.1111/cgf.12575.

Ali-Hamadi, D, Liu, T, Gilles, B, Kavan, L, Faure, F, Palombi, O, et al. Anatomy Transfer. ACM Trans Graph 2013;32(6):188:1–188:8. URL: http://doi.acm.org/10.1145/2508363.2508415. doi:10.1145/2508363.2508415.

Gilles, B, Reveret, L, Pai, D. Creating and animating subject-specific anatomical models. Computer Graphics Forum 2010;:http://onlinelibrary.wiley.com/doi/10.1111/j.1467–8659.2010.01718.x/abstractdoi:10.1111/j.1467-8659.2010.01718.x.

Saito, S, Zhou, ZY, Kavan, L. Computational Bodybuilding: Anatomically-based Modeling of Human Bodies. ACM Trans Graph 2015;34(4):41:1–41:12. URL: http://doi.acm.org/10.1145/2766957. doi:10.1145/2766957.

Meng, M, Fallavollita, P, Blum, T, Eck, U, Sandor, C, Weidert, S, et al. Kinect for interactive AR anatomy learning. In: Mixed and Augmented Reality (ISMAR), 2013 IEEE International Symposium on. IEEE; 2013, p. 277–278.

Shen, W, Deng, K, Bai, X, Leyvand, T, Guo, B, Tu, Z. Exemplar-based human action pose correction and tagging. In: Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on. IEEE; 2012, p. 1784–1791.

Zhou, L, Liu, Z, Leung, H, Shum, HPH. Posture Reconstruction Using Kinect with a Probabilistic Model. In: Proceedings of the 20th ACM Symposium on Virtual Reality Software and Technology. VRST '14; New York, NY, USA: ACM. ISBN 978-1-4503-3253-8; 2014, p. 117–125. URL: http://doi.acm.org/10.1145/2671015.2671021. doi:10.1145/2671015.2671021.

Wei, X, Zhang, P, Chai, J. Accurate Realtime Full-body Motion Capture Using a Single Depth Camera. ACM Trans Graph 2012;31(6):188:1–188:12. URL: http://doi.acm.org/10.1145/2366145.2366207. doi:10.1145/2366145.2366207.

Kamphuis, C, Barsom, E, Schijven, M, Christoph, N. Augmented reality in medical education? Perspectives on medical education 2014;3(4):300–311.

Blum, T, Kleeberger, V, Bichlmeier, C, Navab, N. mirracle: An augmented reality magic mirror system for anatomy education. In: Virtual Reality Short Papers and Posters (VRW), 2012 IEEE. IEEE; 2012, p. 115–116.

Ma, M, Fallavollita, P, Seelbach, I, Von Der Heide, AM, Euler, E, Waschke, J, et al. Personalized augmented reality for anatomy education. Clinical Anatomy 2016;29(4):446–453. URL: http://dx.doi.org/10.1002/ca.22675. doi:10.1002/ca.22675.

Maitre, X. NewScientist : Digital mirror reveals what lies under your skin. 2014. URL: https://www.newscientist.com/article/mg22229653-800-digital-mirror-reveals-what-lies-under-your-skin/.

Borner, A, Kirsch, S. Anatomie Spiegel. 2015. URL: http://www.qwertzus.com/anatomie-spiegel/.

Bauer, A, Paclet, F, Cahouet, V, Dicko, AH, Palombi, O, Faure, F, et al. Interactive Visualization of Muscle Activity During Limb Movements: Towards Enhanced Anatomy Learning. In: Proceedings of the 4th Eurograph-

ics Workshop on Visual Computing for Biology and Medicine. VCBM '14; Aire-la-Ville, Switzerland, Switzerland: Eurographics Association. ISBN 978-3-905674-62-0; 2014, p. 191–198. URL: http://dx.doi.org/10.2312/vcbm.20141191. doi:10.2312/vcbm.20141191.

Bauer, A, Dicko, AH, Palombi, O, Faure, F, Troccaz, J. Living Book of Anatomy (LBA) Project: See Your Insides in Motion! In: SIGGRAPH Asia 2015 Emerging Technologies. SA '15; New York, NY, USA: ACM. ISBN 978-1-4503-3925-4; 2015, p. 16:1–16:3. URL: http://doi.acm.org/10.1145/2818466.2818470. doi:10.1145/2818466.2818470.

Dao, NL, Deng, T, Cai, J. Fast and automatic body circular measurement based on a single kinect. In: Asia-Pacific Signal and Information Processing Association, 2014 Annual Summit and Conference (APSIPA). IEEE; 2014, p. 1–4.

Espitia-Contreras, A, Sanchez-Caiman, P, Uribe-Quevedo, A. Development of a Kinect-based anthropometric measurement application. In: Virtual Reality (VR), 2014 iEEE. IEEE; 2014, p. 71–72.

Faure, F, Gilles, B, Bousquet, G, Pai, DK. Sparse Meshless Models of Complex Deformable Solids. In: ACM SIGGRAPH 2011 Papers. SIGGRAPH '11; New York, NY, USA: ACM. ISBN 978-1-4503-0943-1; 2011, p. 73:1–73:10. URL: http://doi.acm.org/10.1145/1964921.1964968. doi:10.1145/1964921.1964968.

Gilles, B, Faure, F, Bousquet, G, Pai, DK. Frame-based interactive simulation of complex deformable objects. In: Hidalgo González, M, Mir Torres, A, Varona Gómez, J, editors. Deformation Models; vol. 7 of *Lecture Notes in Computational Vision and Biomechanics*. Springer; 2013, p. 145–166. doi:10.1007/978-94-007-5446-1_6.

Nordin, M, Frankel, VHVH. Basic biomechanics of the musculoskeletal system. Philadelphia (Pa.): Lippincott Williams & Wilkins; 2001. ISBN 0-683-30247-7. URL: http://opac.inria.fr/record=b1133407.

Eberly, D. Constrained Quaternions Using Euler Angles. 2008. URL: http://www.geometrictools.com/Documentation/ConstrainedQuaternions.pdf.

Canny, J. A Computational Approach to Edge Detection. IEEE Trans Pattern Anal Mach Intell 1986;8(6):679–698. URL: http://dx.doi.org/10.1109/TPAMI.1986.4767851. doi:10.1109/TPAMI.1986.4767851.

Suzuki, S, et al. Topological structural analysis of digitized binary images by border following. Computer vision, graphics, and image processing 1985;30(1):32–46.

Hickson, S, Birchfield, S, Essa, I, Christensen, H. Efficient hierarchical graph-based segmentation of RGBD videos. Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition 2014;:344–351.

Abramov, A, Pauwels, K, Papon, J, Wörgötter, F, Dellen, B. Depth-supported real-time video segmentation with the Kinect. Applications of Computer Vision (WACV), 2012 IEEE Workshop on 2012;:457–464.

Hernandez-Lopez, JJ, Quintanilla-Olvera, AL, López-Ramírez, JL, Rangel-Butanda, FJ, Ibarra-Manzano, MA, Almanza-Ojeda, DL. Detecting objects using color and depth segmentation with Kinect sensor. Procedia Technology 2012;3:196–204.

Bookstein, FL. Principal Warps: Thin-Plate Splines and the Decomposition of Deformations. IEEE Trans Pattern Anal Mach Intell 1989;11(6):567–585. URL: http://dx.doi.org/10.1109/34.24792. doi:10.1109/34.24792.

Sorkine, O, Alexa, M. As-rigid-as-possible surface modeling. Symposium on Geometry processing 2007;4.

Kavan, L, Collins, S, Žára, J, O'Sullivan, C. Geometric Skinning with Approximate Dual Quaternion Blending. ACM Trans Graph 2008;27(4):105:1–105:23. URL: http://doi.acm.org/10.1145/1409625.1409627. doi:10.1145/1409625.1409627.

Zhou, S, Fu, H, Liu, L, Cohen-Or, D, Han, X. Parametric Reshaping of Human Bodies in Images. In: ACM SIGGRAPH 2010 Papers. SIGGRAPH '10; New York, NY, USA: ACM. ISBN 978-1-4503-0210-4; 2010, p. 126:1–126:10. URL: http://doi.acm.org/10.1145/1833349.1778863. doi:10.1145/1833349.1778863.

Jain, A, Thormählen, T, Seidel, HP, Theobalt, C. MovieReshape: Tracking and Reshaping of Humans in Videos. ACM Trans Graph 2010;29(6):148:1–148:10. URL: http://doi.acm.org/10.1145/1882261.1866174. doi:10.1145/1882261.1866174.

Richter, M, Varanasi, K, Hasler, N, Theobalt, C. Real-time reshaping of humans. In: 3D Imaging, Modeling, Processing, Visualization and Transmission (3DIMPVT), 2012 Second International Conference on. IEEE; 2012, p. 340–347.

Liu, Z, Zhou, L, Leung, H, Shum, HP. Kinect posture reconstruction based on a local mixture of gaussian process models. IEEE transactions on visualization and computer graphics 2016;22(11):2437–2450.

Andrews, S, Casado, IH, Komura, T, Sigal, L, Mitchell, K. Real-time physics-based motion capture with sparse sensors. In: CVMP. 2016, p. 5–1.

Murai, A, Kurosaki, K, Yamane, K, Nakamura, Y. Musculoskeletal-see-through mirror: Computational modeling and algorithm for whole-body muscle activity visualization in real time. Progress in Biophysics and Molecular Biology 2010;103(2-3):310–317.