# Single-view RGBD-based Reconstruction of Dynamic Human Geometry

Charles Malleson, Martin Klaudiny, Adrian Hilton and Jean-Yves Guillemaut
Centre for Vision, Speech and Signal Processing, University of Surrey, GU2 7XH, UK
{c.malleson, martin.klaudiny, a.hilton, j.guillemaut}@surrey.ac.uk

## Abstract

*We present a method for reconstructing the geometry and appearance of indoor scenes containing dynamic human subjects using a single (optionally moving) RGBD sensor. We introduce a framework for building a representation of the articulated scene geometry as a set of piecewise rigid parts which are tracked and accumulated over time using moving voxel grids containing a signed distance representation. Data association of noisy depth measurements with body parts is achieved by online training of a prior shape model for the specific subject. A novel frame-to-frame model registration is introduced which combines iterative closest-point with additional correspondences from optical flow and prior pose constraints from noisy skeletal tracking data. We quantitatively evaluate the reconstruction and tracking performance of the approach using a synthetic animated scene. We demonstrate that the approach is capable of reconstructing mid-resolution surface models of people from low-resolution noisy data acquired from a consumer RGBD camera.*

## 1. Introduction

Reconstruction of the dynamic geometry and appearance of scenes has several application areas including content creation, scene navigation, digital cartography and biometrics. Current approaches require multiple video cameras and/or depth sensors. We aim to reconstruct scenes using only a single low-cost commodity RGBD sensor (such as a Kinect). This work aims to extend previous work on depth-based tracking and reconstruction of rigid surface geometry to articulated structures with piecewise rigid surface geometry, in particular people. We focus on scenes containing static background geometry and a moving human subject.

The proposed method takes as input a sequence of RGB and depth maps captured from an RGBD sensor that may be either fixed or hand-held. A further input to the system is the approximate and noisy skeletal pose of the subject at each frame, as obtained from the depth maps by an off-the-shelf skeletal tracker. The output of the system is a set of textured meshes and their poses at each frame in the sequence, which together form a piecewise rigid representation of the dynamic scene.

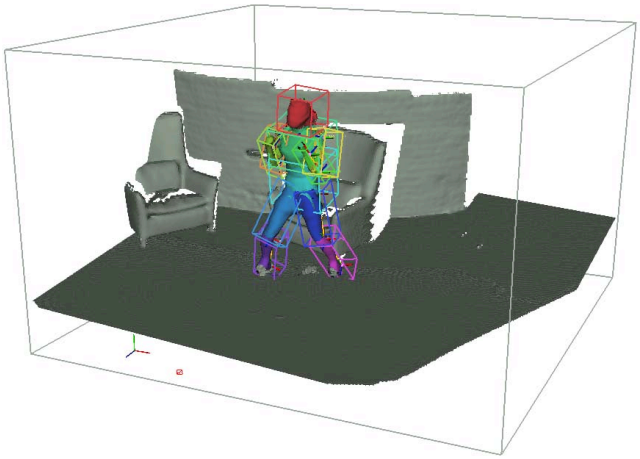


Figure 1: Reconstruction of dynamic human subject and static background scene, showing partitioning.

The proposed approach is summarised in Figure 2. We use a moving voxel grid for each rigid part to integrate surface measurements. The poses of these grids are tracked sequentially using the skeletal pose as initialization and refining the pose by performing an ICP-like registration between incoming frames and synthetic frames ray-cast from the integrated model. The registration also includes matches based on optical flow between successive RGB frames as well as terms based on the deviation of part pose from the input skeletal pose. We use rules (with parameters trained online) to assign depth measurements to single parts in cases where their voxel grids overlap, this helps prevent generation of spurious geometry.

The model surfaces are incrementally extended and refined as new depth measurements are integrated. The set of accumulated surfaces can be extracted at any frame using marching cubes [10] after which a per-vertex texture is applied to the resulting mesh by back projecting colours from the RGB images onto each vertex, subject to a visibility test

with all parts. These meshes together with the sequence of sensor and part poses allow the reconstructed dynamic sequence to be played back. The resulting 3D geometry is more complete and less noisy than the raw 2.5D geometry contained in each input depth map.

### 1.1. Previous work

The KinectFusion system [12] produces models of rigid scenes from a Kinect depth sensor using truncated signed distance function (TSDF) [7] measurement integration and point-to-plane ICP registration between incoming depth frames and the synthetic depth frames ray-cast from the TSDF model. GPU parallelization allows KinectFusion to run at video-rates. In this work we build on this approach to allow reconstruction of piece-wise rigid scenes using the additional input of a tracked skeletal pose.

In [4], three depth sensors are used to automatically derive articulation constraints and reconstruct motion and geometry, while [15] use three Kinects to perform articulated tracking with prior laser-scanned models of the subjects. In this work we use only a single sensor view, and use no prior surface scans.

Iterative closest point (ICP) algorithms have been widely used for the alignment of point clouds, typically using the sum of squared point-to-point [2] or point-to-plane [5] distances between matched points, where the point matches are re-estimated at each iteration. These registration algorithms tend to work best on geometry that has enough implicit features to constrain the transforms. In the registration of human parts, there are typically at least two degrees of freedom which are not well constrained by their geometry (consider an upper/lower limb which could rotate about its axis and translate along its axis without affecting the closest point error). This motivates the need for additional constraints in the registration cost function. Image assisted depth map registration has been proposed in [14], where optical flow on luminance images is used to obtain point correspondences. We combine optical flow correspondences and point-to-point error, with point-to-plane distances and skeletal pose constraints.

The ICP framework has been extended to articulated bodies [13], [8]. Such approaches are unsuitable for this application where we assume noisy skeletal input where joint positions may differ from the true joint positions by several cm, and bone lengths are not maintained. We therefore use the skeleton tracking as an additional data term in per-part registration rather than enforcing articulation as a hard constraint.

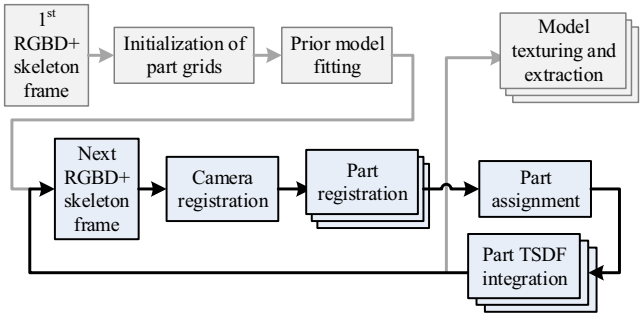



Figure 2: Overview of the proposed reconstruction system. The core registration/integration cycle is in bold.

## 2. Registration and model integration

### 2.1. Problem statement

A single moving RGBD sensor is used to capture a scene containing static background geometry and a moving human subject. Using the captured depth maps, RGB images and approximate skeletal tracking as input, we aim to track the sensor and simultaneously build a piecewise rigid model of the dynamic human subject without using any prescanned surface models.

### 2.2. Summary of approach

We treat the background scene as a static object fixed in the global reference frame, and the human body as a piecewise rigid set of surface parts associated with a hierarchical articulated skeleton. We use the TSDF to integrate depth measurements into models. The TSDF is an intermediate volumetric representation which allows incremental building and de-noising of surface models as new measurements are added. TSDFs are well suited to integration of surface measurements of general rigid scenes where shape and topology are unknown. We extend the TSDF representation to integrate observations of both the static background and piecewise rigid parts of the foreground.

The moving RGBD camera pose is estimated using point-to-plane ICP between input depth maps and background TSDF model. This strategy is shown in [12] to be less susceptible to drift than raw frame-to-frame ICP. To increase the robustness of this camera pose estimation approach we also use point-to-point terms obtained via optical flow on the RGB images.

Because the subject is moving during the capture, we also need to track the parts over time. One of the main difficulties with using ICP for this is that the parts typically occupy a relatively small portion of the frame (compared to a static background) and therefore provide far fewer measurements. This, along with symmetries in the parts, makes standard ICP inadequate for part tracking, even when colour images are used as well. We propose a method for includ-

ing noisy joint pose data from the skeletal track in the part registration in order to initialize part pose at the start of registration and also as an extra constraint in the optimization.

In practice, the bounds of two or more part volumes often partially overlap with one another. To mitigate the generation of spurious geometry in the reconstructed surfaces, a mechanism is therefore required for the assignment of surface measurements to individual parts. This part data association approach is based on the measurement's relative proximity to simple prior models of each part, trained for the subject online using the first depth frame.

### 2.3. Definitions

All subscripted matrices $\mathbf{T}$ denote $4\times4$ rigid body transforms. For example the (known and fixed) pose matrix of the colour sensor relative to the depth sensor is $\mathbf{T}_{cd}$ (and $\mathbf{R}_{cd}$ and $\mathbf{t}_{cd}$ are its rotation matrix and translation vector components, respectively). The depth and RGB sensors have known and fixed $3\times3$ camera matrices $\mathbf{K}_d$ and $\mathbf{K}_c$, respectively. The depth sensor pose at the first frame, $\mathbf{T}_0^d = \mathbf{I}$ is defined as the global coordinate system. The sensor may move from frame having pose $\mathbf{T}_k^d$ at frame $k$.

Let a dot above a 3-vector $\mathbf{u}$ denote its homogeneous form $\dot{\mathbf{u}} := [\mathbf{u}^T \quad 1]^T$ and $\dot{\mathbf{K}}$ denote the $4\times4$ homogeneous form of $\mathbf{K}$. In a similar vein, let $\dot{\mathbf{T}}$ denote the $3\times4$ matrix formed by discarding the last row of a transform $\mathbf{T}$. The operator $\rho$ denotes conversion from homogeneous to image pixel coordinates: $\rho([x \quad y \quad z \quad 1]^T) := (\lfloor x/z \rceil, \lfloor y/z \rceil)$.

The input to the system is a sequence of frames $F_k := \{D_k, C_k, S_k\}$ where $D_k$ is a depth map, $C_k$ is an RGB image and $S_k$ is the skeletal pose estimate at frame index $k$.

The depth map $D_k := \{d_k(u,v) : 0 \le u < w_d, 0 \le v < h_d\}$ where $d_k(u,v)$ is the measured depth in metric units at pixel coordinates $(u, v)$ and $h_d$ and $w_d$ are the image dimensions. For pixels where no measurement is available $d_k = 0$. We define a point map as $V_k := \{\mathbf{v}_k(u,v) : 0 \le u < w_d, 0 \le v < h_d\}$ where

$$\mathbf{v}_k(u,v) = d_k(u,v)\mathbf{K}_d^{-1}[u \quad v \quad 1]^T \qquad (1)$$

is the re-projected depth point in depth sensor coordinates, from which a normal map $N_k := \{\mathbf{n}_k(u,v) : 0 \le u < w_d, 0 \le v < h_d\}$ is estimated using nearest neighbours [12].

The RGB image $C_k := \{\mathbf{c}_k(u,v) : 0 \le u < w_c, 0 \le v < h_c\}$ where $\mathbf{c}_k(u,v)$ is the measured RGB vector at pixel coordinates $(u, v)$ and $h_c$ and $w_c$ are the image dimensions.

The input skeletal pose estimate $S_k := \{J_k^j : 0 \le j < n\}$ where $n$ is the number of joints. A joint $J_k^j := \{\mathbf{T}_k^j, r_k^j, t_k^j\}$, consists of a pose $\mathbf{T}_k^j$ (w.r.t. the depth sensor coordinate system), and pose estimation confidences $r_k^j \in [0,1]$ and $t_k^j \in [0,1]$ for the joint position and orientation, respectively.

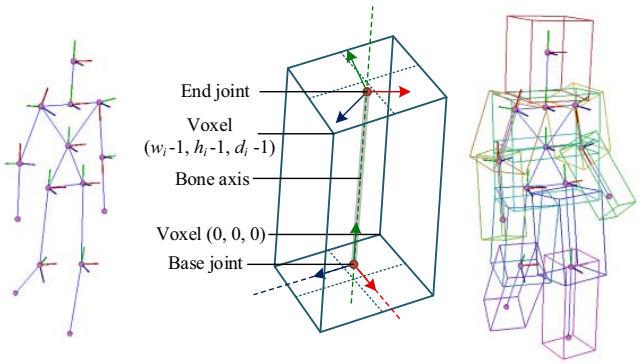



Figure 3: Left: skeletal pose. Centre: defining part pose and size in terms of joint poses. Right: initialized part sized and positions.

We define a part as $P_i := \{G_i, \mathbf{s}_i, \mathbf{t}_i^o, \mathbf{T}_k^i\}$ where $G_i$ is its voxel grid, $\mathbf{s}_i$ is the voxel size (in metric units), $\mathbf{t}_i^o$ is the metric offset between the grid coordinate origin and the origin of the local coordinate system of the part, and $\mathbf{T}_k^i$ is the global pose of the part at frame $k$. We define a voxel grid as $G_i := \{\{f_i(u,v,w), g_i(u,v,w)\} : 0 \le u < w_i, 0 \le v < h_i, 0 \le w < d_i\}$ where $f_i(u,v,w)$ and $g_i(u,v,w)$ are, respectively, the signed distance value and weight at voxel position $(u,v,w)$, and $w_i$, $h_i$ and $d_i$ are, the width, height and depth in voxels, respectively.

### 2.4. Skeletal model and initialization of parts

Our representation consists of two volumetric models which are simultaneously built from the RGBD observations: a global rigid background scene model $P_{bg}$ and a generic piecewise rigid model consisting of $m$ human body parts $P_h := \{P_0, \ldots, P_{(m-1)}\}$. The parts in $P_h$ have their grid dimensions and coordinate system defined using the skeletal data from the first frame $S_0$ (refer to Figure 3). The known skeleton joint connectivity defines a base and end joint for each part. Let $\lambda(i)$ denote index of the base joint of part $P_i \in P_h$. The base joint pose of the part defines the part coordinate system with the part bone along one of its axes. Thus given this origin and orientation, it remains to set the six sides of the grid. The sides orthogonal to the bone axis are set using the base and end joint positions, the sides parallel to the bone axis are set based on expected anthropometric ratios for limb width to length dimensions giving part volumes which are sufficient to enclose the part surface.

### 2.5. Camera and part registration

In a reconstruction cycle at time $k$, we refer to the current frame $k$ as the *source* and the previous frame $k-1$ as the *target*. We define $F_{k-1}^i := \{D_{k-1}^i, C_{k-1}, S_{k-1}\}$, where $D_{k-1}^i$ is a synthetic version of $D_{k-1}$ obtained by ray-casting into the current estimated model $G_i$ from the perspective of the

depth camera at frame $k-1$ [12].

The depth camera pose $\mathbf{T}_k^d$ is estimated by registering $F_k$ against $F_{k-1}^{bg}$ using $\mathbf{T}_{k-1}^d$ as initialization. Then for each human part $P_i \in P_h$ we estimate pose $\mathbf{T}_k^i$ by registering $F_k$ against $F_{k-1}^i$ using the estimated skeletal pose from $S_k$ as initialization. Note that in the case of background part registration the camera moves and the part is fixed in global coordinates (moving source), but in the case of human part registration, the camera is fixed and the part moves (moving target). However, we find it more convenient to have a unified formulation for both camera and part registration. Therefore for each $P_i \in P_h$ we re-register the camera using $P_i$ to give an 'apparent camera pose' $\mathbf{T}_k^{d,i}$ and then obtain the actual part pose $\mathbf{T}_k^i$ using the inverse of apparent camera pose change:

$$\mathbf{T}_k^i = (\mathbf{T}_k^{d,i}(\mathbf{T}_k^d)^{-1})^{-1}\mathbf{T}_{k-1}^i. \tag{2}$$

The aim of registration is to find the rigid body transform that brings source points $V_k$ into alignment with target points $V_{k-1}^i$ according to an alignment cost function. The proposed registration system is based on ICP with point-to-point and point-to-plane distances as well as an additional constraint based on skeletal pose. At each iteration a cost function is minimized w.r.t. an incremental pose $\tilde{\mathbf{T}}$. The per-iteration registration cost function for a part $P_i$ at frame $k$ is

$$E_{i,k}(\tilde{\mathbf{T}}) = E_{i,k}^p(\tilde{\mathbf{T}}) + w_o E_{i,k}^o(\tilde{\mathbf{T}}) + w_s E_{i,k}^s(\tilde{\mathbf{T}}) \tag{3}$$

where $E_{i,k}^p(\tilde{\mathbf{T}})$ is the point-to-plane term, $E_{i,k}^o(\tilde{\mathbf{T}})$ is the point-to-point term and $E_{i,k}^s(\tilde{\mathbf{T}})$ is the skeletal pose term as described in the following subsections. Weights $w_o$ and $w_s$ control the relative contribution of each error term. We use equal weighting for projective and and optical flow matches ($w_o = 1$) and set $w_s = 3000$. (The relatively large weighting for $w_s$ compensates for the relatively small number of skeletal constraints compared to depth data-points.)

An important part of the registration is the data association process. Formally, we define a data association as the assignment of each source depth point $\mathbf{v}_k(u_s, v_s)$ to a target depth point $\mathbf{v}_{k-1}(u_t, v_t)$ via a function $\Omega : (u_s, v_s) \rightarrow (u_t, v_t)$. We use the projective data association algorithm [3] defined as

$$\Omega_p(u_s, v_s) = \rho\big(\dot{\mathbf{K}}_d(\mathbf{T}_{k-1}^d)^{-1}\mathbf{T}_k^d\dot{\mathbf{v}}_k(u_s, v_s)\big) \tag{4}$$

and also an optical flow-based data association $\Omega_o(u_s, v_s)$ similar to [14], where the 3D depth point correspondences are inferred via the 2D correspondences from optical-flow on the images.

The point-to-plane distance is generally preferred over point-to-point as it tends to converge faster. However it does not constrain the transform when the geometry is highly uniform, even when given matches obtained using another modality (*e.g.* optical flow on images). We therefore opt to use both types of distance in the registration.

The point-to-plane error term $E_{i,k}^p(\tilde{\mathbf{T}})$ which uses the projective matches is defined as

$$E_{i,k}^p = \sum_{\Omega_p(u,v)\neq\text{null}} \big(\dot{\tilde{\mathbf{T}}}\mathbf{T}_k^d\dot{\mathbf{v}}_k(u,v) - \dot{\mathbf{T}}_{k-1}^d\dot{\mathbf{v}}_{k-1}^i(\Omega_p(u,v))\big)\cdot\mathbf{n}_k(u,v). \tag{5}$$

and the point-to-point term $E_{i,k}^o(\tilde{\mathbf{T}})$ which uses the optical flow matches is defined as

$$E_{i,k}^o = \sum_{\Omega_o(u,v)\neq\text{null}} ||\dot{\tilde{\mathbf{T}}}\mathbf{T}_k^d\dot{\mathbf{v}}_k(u,v) - \dot{\mathbf{T}}_{k-1}^d\dot{\mathbf{v}}_{k-1}^i(\Omega_o(u,v))||^2. \tag{6}$$

### 2.5.1 Skeletal pose constraints

The registration of each human part $P_i \in P_h$ employs additional constraints which serve to minimize the difference between its pose $\mathbf{T}_k^i$ and the pose of its corresponding joint $J_k^{\lambda(i)} \in S_k$, $\mathbf{T}_k^{j_i}$. We minimise the squared distance between the transform origins

$$t^2 = ||\mathbf{t}_k^i - \mathbf{t}_k^{j_n}||^2 \tag{7}$$

and the squared angles between coordinate axes,

$$\begin{aligned}\theta_x^2 &\approx ||\mathbf{R}_k^i\hat{\mathbf{i}} - \mathbf{R}_k^{j_n}\hat{\mathbf{i}}||^2\\ \theta_y^2 &\approx ||\mathbf{R}_k^i\hat{\mathbf{j}} - \mathbf{R}_k^{j_n}\hat{\mathbf{j}}||^2\\ \theta_z^2 &\approx ||\mathbf{R}_k^i\hat{\mathbf{k}} - \mathbf{R}_k^{j_n}\hat{\mathbf{k}}||^2\end{aligned} \tag{8}$$

where $\hat{\mathbf{i}}$, $\hat{\mathbf{j}}$ and $\hat{\mathbf{k}}$ are the unit basis vectors and a small angle assumption has been used to replace angles with straight line distances between the basis vectors, as depicted in Figure 4. The prior skeletal pose constraint term is defined as

$$\begin{aligned}E_{i,k}^s =& t_k^{\lambda(i)}||\mathbf{t}_k^i + \tilde{\mathbf{t}} - \mathbf{t}_k^{\lambda(i)}||^2\\ &+ (1/3)r_k^{\lambda(i)}\big(||\tilde{\mathbf{R}}\mathbf{R}_k^i\hat{\mathbf{i}} - \mathbf{R}_k^{j_n}\hat{\mathbf{i}}||^2\\ &+ ||\tilde{\mathbf{R}}\mathbf{R}_k^i\hat{\mathbf{j}} - \mathbf{R}_k^{j_n}\hat{\mathbf{j}}||^2\\ &+ ||\tilde{\mathbf{R}}\mathbf{R}_k^i\hat{\mathbf{k}} - \mathbf{R}_k^{j_n}\hat{\mathbf{k}}||^2\big)\end{aligned} \tag{9}$$

where the position and orientation constraints have been weighted by the confidences of the input skeletal track.

### 2.5.2 Optimization

Assuming small incremental rotations and linearising the rotation matrices, the cost function (3) can easily be written as a $6\times 6$ symmetric linear system of the form $\sum \mathbf{A}^T\mathbf{A}\mathbf{x} =$

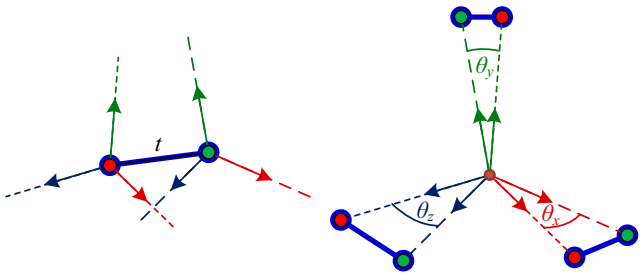



Figure 4: Position and orientation constraints based on relative pose between part and its joint from the skeleton. Point-to-point distances approximate angles in the rotation constraints.

$\sum \mathbf{A}^T\mathbf{b}$ and solved for the 6D incremental transform vector $\mathbf{x}$ using Cholesky decomposition (similar to [12]). The incremental transform $\mathbf{x}$ is used to generate $\tilde{\mathbf{T}}$ which is composed onto $\mathbf{T}_k^{d,i}$ at each iteration.

## 2.6. Measurement integration

After the camera pose and all part poses for new frame $F_k$ have been estimated, each of the raw measured surface points $\mathbf{v}_k(u,v) \in V_k$ is integrated into the background model $P_{bg}$ and/or part models $P_h$ as appropriate. When $\mathbf{v}_k(u,v)$ occupies the voxel grid of more than one part, there is a risk of incorrectly updating the surface models such that surfaces measurements from one part (*e.g.* the upper arm) may be integrated into the surface model of another (*e.g.* the thorax). Furthermore it is often the case that the background volume $P_{bg}$ overlaps completely with the parts in $P_h$. We therefore assign each observation $\mathbf{v}_k$ to a single part.

### 2.6.1 Prior on part surface geometry

The assignment decision is ambiguous when the depth point occupies two or more body parts. In these cases we make use of a simple prior surface model representing the approximate size and shape of each part. The prior model $C_i$ for each part $P_i \in P_h$ is a cylinder or elliptic cylinder aligned with and centred on $P_i$'s bone axis (requiring only 1 or two parameters to be estimated). For the head and limb parts we use a cylinder, for the trunk parts we use elliptic cylinders (Figure 5). While fitting algorithms such as RANSAC could be used, we find it sufficient to fit by exhaustive sampling of the permitted range of radii (in increments of 15 mm) and choosing the radii which lead to the highest number points in the first frame depth point map $V_0$ falling within a tolerance distance of its candidate (elliptic) cylinder. As shown in Figure 2, this fitting process is performed once (prior to the commencement of the reconstruction cycle).

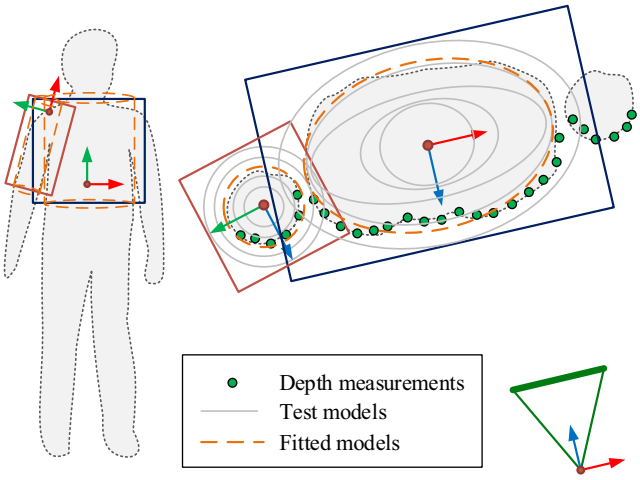



Figure 5: Using measured depth points to fit cylinders/elliptic cylinder prior surface models depth map surface measurements. The rectangles are the grid boundaries and dotted grey lines are the true surface.

### 2.6.2 Assignment of depth measurements to parts

Let the function $c(P_i, \mathbf{v}_k(u,v))$ denote the distance between $\mathbf{v}_k(u,v)$ and the fitted elliptic cylinder for $P_i$. Let the operator $\Psi(\mathbf{v}_k(u,v), P)$ denote the number of parts in a set of parts $P$ that are occupied by $\mathbf{v}_k(u,v)$. Algorithm 1 defines the part assignment rules, which assign depth points to the closest prior surface model in cases of part overlap. Figure 6 illustrates the approach.

**Algorithm 1** Assignment of depth points to parts (the indices of $\mathbf{v}_k(u,v)$ have been omitted for brevity)

**if** $\Psi(\mathbf{v}_k, P_{bg}) = 1$ **and** $\Psi(\mathbf{v}_k, P_h) = 0$ **then**
  $\mathbf{v}_k$ is assigned to $P_{bg}$
**else if** $\exists P_i \in P_h : \Psi(\mathbf{v}_k, P_i) = 1$ **and** $\Psi(\mathbf{v}_k, P_h) = 1$ **then**
  $\mathbf{v}_k$ is assigned to $P_i$
**else if** $\Psi(\mathbf{v}_k, P_h) > 1$ **then**
  $\mathbf{v}_k$ is assigned to $\underset{P_i \in P_h : \Psi(\mathbf{v}_k, P_i) = 1}{\operatorname{argmin}} c(P_i, \mathbf{v}_k)$
**else**
  $\mathbf{v}_k$ is assigned to null
**end if**

While it is important not to introduce *surfaces* into the wrong part, it is also important that observed *free space* is integrated into all parts which lie between that surface and the depth camera, otherwise any spurious geometry in free space between the measured point and the depth sensor camera would be allowed persist.

We define a function $\boldsymbol{\phi}_i(u,v,w) := \text{diag}(\mathbf{s}_i)[u \quad v \quad w]^T + \mathbf{t}_i^0$ which transforms from voxel

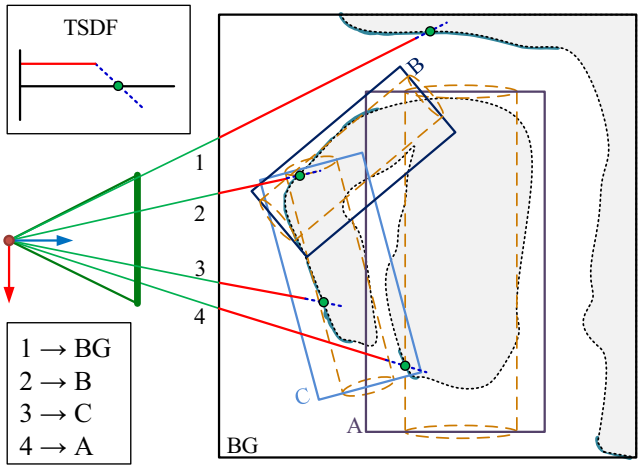



Figure 6: Integration of measurements with multiple rigid parts. Note that in the variable SDF region (dashed blue) a depth point only contributes to the part to which it was assigned, but in the free-space SDF region (red) it may contribute to any parts along its line of sight.

index coordinates to local part coordinates. The projective signed distance $\hat{f}_i^m(u,v,w)$ between a voxel $(u,v,w)$ and its associated depth point $\mathbf{v}_k$ is:

$$\hat{f}_i^m(u,v,w) = \pm \,||\mathbf{v}_k\big(\rho(\dot{\mathbf{K}}_d(\mathbf{T}_k^d)^{-1}\mathbf{T}_k^i\dot{\boldsymbol{\phi}}_i(u,v,w))\big) - \mathbf{I}_{3\times4}(\mathbf{T}_k^d)^{-1}\mathbf{T}_k^i\dot{\boldsymbol{\phi}}_i(u,v,w)|| \tag{10}$$

where the sign of $\hat{f}_i^m$ is the sign of the $z$-value of the argument of the norm. The update rule is given by Algorithm 2, where $g_i^m(u,v,w)$ is the weighting of the measurement (constant in our case), $\mu$ is the truncation distance and $f_i^m(u,v,w)$ is the truncated version of $\hat{f}_i^m(u,v,w)$.

**Algorithm 2** Multi-part voxel grid update rules for part $P_i \in \{P_{bg} \cup P_h\}$ at frame $k$

**for** $\forall\{f_i(u,v,w), g_i(u,v,w)\} \in G_i$ **do**
  **if** ( $|\hat{f}_i^m(u,v,w)| < \mu$ **and** pixel is assigned to $P_i$ ) **or** ( $|\hat{f}_i^m(u,v,w)| \geq \mu$ ) **then**
    $f_i(u,v,w) \leftarrow (f_i \cdot g_i + f_i^m \cdot g_i^m)/(g_i + g_i^m)$
    $g_i(u,v,w) \leftarrow g_i + g_i^m$
  **else**
    do not update voxel $(u,v,w)$
  **end if**
**end for**

# 3. Experimental results

We tested our system both qualitatively on real data from a consumer RGBD camera and quantitatively on synthetic data with known ground truth articulated motion and surface shape. We use a skeletal model containing 15 joints from which we derive 11 parts (see Figures 3 and 1).

## 3.1. Real and virtual sensor

We used an Xtion Pro Live, which is based on the same PrimeSense sensor as the ubiquitous Kinect, but allows synchronization of the 30 fps $640 \times 480$ RGB and depth stream and also allows locking of the RGB exposure and white balance (which helps make the final model texture more consistent). It was calibrated using a chart-based calibration tool to obtain $\mathbf{K}_d$, $\mathbf{K}_c$ and $\mathbf{T}_{cd}$.

We make use of an off-the-shelf depth map-based skeletal tracker from OpenNI's NiTE middle-ware to obtain the initial pose estimate $S_k$ for each frame. The NiTE skeleton representation provides 15 joint position and orientation in the coordinate system of the depth sensor (Figure 3). The tracker exhibits significant amounts of jitter in the joint positions and bone length is not maintained. A significant limitation of the NiTE tracker is that it is unable to function in the presence of sensor motion[1], therefore our real-world experiments are restricted to a fixed sensor.

We aim to make the synthetic data resemble Kinect data so that the evaluation gives some insight into expected real-world performance. Thus for the virtual camera we use the same resolution and calibration as the Xtion. We add Gaussian noise and quantize the depth maps following the Kinect noise model in [9], where the standard deviation of the depth map random noise and the quantization steps both increase quadratically with distance, reaching 4 cm and 7 cm respectively at 5 m. We also add a moderate amount of Gaussian noise to the RGB images (resulting in a PSNR of 34 dB).

## 3.2. Evaluation on synthetic sequences

We generated a synthetic scene containing a background 'lobby' set and an animated dynamic character (Figure 7a). The character was animated using skeletal motion from the CMU Motion Capture Database [6] (Subject 14, Trial 01 - 'boxing'), simplified to the NiTE skeleton representation and re-targeted to the character. Subsequently a mesh sequence of the moving character was created using Linear Blend Skinning [11] with skinning weights automatically calculated according to [1]. The final sequence of textured 3D models along with the skeletal motion sequence used to drive the character provide the ground-truth data for quantitative evaluation.

We generated two RGBD videos of the 'boxing' sequence - one with the static virtual sensor (689 frames) and the other with shaky hand-held motion (510 frames).

[1] We believe this may be due to static background subtraction being used internally in the NiTE tracker.

We tested on both the noise-free and noise-corrupted versions of the data. For the noise corrupted sequences we also added Gaussian noise (with standard deviation 8.3 mm) to all joint positions in order to simulate the jitter of the real depth-map based skeletal tracker. Figure 7 shows selected results from this data. Videos of these results are provided in the supplementary material.

Figure 8 shows quantitative results for the synthetic sequences under different registration modes (obtained by disabling the appropriate terms in (3)). Figure 8a illustrates how the model becomes more complete as new depth frames are integrated. The extensive motion of the character throughout the sequence results in the inclusion of most of the scene surface after about 200 frames, after which few new areas become visible to the camera (the back of the character is never fully visible to the camera).

We also determine whether or not our registration approach improves the tracking of the parts compared to the noisy skeletal tracking input. For this we compute statistics on the relative pose between each part's ground truth pose and its registered pose throughout the sequence. Figure 8b shows the RMS error in each component of the relative pose of the upper left arm over the noisy fixed camera sequence. The proposed surface registration reduces both the translation and orientation error compared to the noisy skeletal pose. Including the optical flow point-to-point constraints improves the orientation error compared to point-to-plane alone.

Figure 8c shows the camera tracking error (for the noisy sequence) for both standard ICP and for the proposed image assisted registration. Note that the optical flow assisted registration term maintains tracking to within a few cm throughout, while the standard ICP diverges.

To evaluate the reconstructed surface quality we compute the RMS distance between every vertex in the reconstructed surface and its closest point the ground-truth surface (at the first frame). A visualization of this is shown in Figure 7d. The RMS surface error for the fixed camera sequence was 10 mm for the clean data and 14 mm for the noisy input data. For the moving camera sequence it was 12 mm and 22 mm for clean and noisy input, respectively. The clean data results in more accurate surface reconstruction. This is because the depth data is less noisy and also because skeletal pose error does not contribute. In the case of the moving camera, the camera pose estimation step also further contributes to the error.

### 3.3. Evaluation on real sequences

Figure 9 shows the result of running the system on a real 'turning' sequence. The subject is roughly 2 m from the sensor, resulting in very noisy depth map input (left). If the noisy NiTE skeleton tracking alone is used as the pose, the resulting surface reconstruction is inaccurate and lacks detail (centre). However, when the proposed registration system is used, a more accurate and detailed model is produced (right). The complete sequence is given as a video in the supplementary material (along with a further 'star jump' sequence).

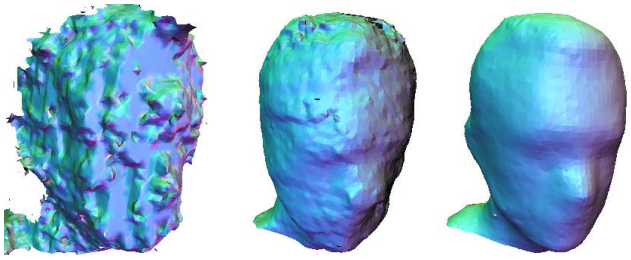


Figure 9: Normal colour-mapped visualization of the head in 'turning' sequence. From left to right: raw input frame, 3D reconstruction using noisy skeletal pose only, reconstruction using proposed registration.

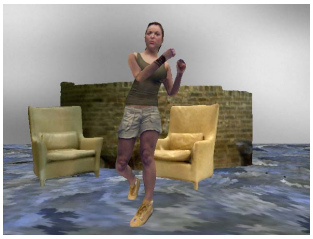
## 4. Conclusion and future work

We demonstrate a feasible approach to modelling of dynamic human geometry using a single RGBD sensor, producing a high quality piece-wise rigid model of a subject performing in a scene. The approach integrates noisy surface observations over time to reconstruct a complete surface with mid-resolution detail (creases, facial features) which are not visible/resolved in the individual depth images.

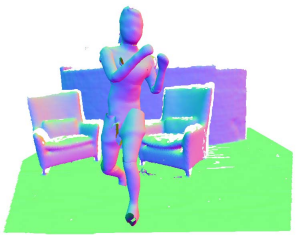
The dynamic scene is represented as a static background volume model and piece-wise rigid articulated volume structure. A novel data-association approach is introduced to robustly assign observations to the body parts in the presence of inter-part occlusion and overlap/close proximity. The novel representation is demonstrated to allow fusion of dynamic articulated surface observations over time to reconstruct a complete surface and integrate out sensor noise to resolve surface detail.

The proposed reconstruction system is 'online' in the sense that the required computational resources are independent of sequence length and it processes the frames sequentially. However, our implementation, while making use of the GPU, is not highly optimized and currently runs at ~2 fps on our hardware (GeForce GTX 560 Ti GPU, 3.4 GHz Intel Core i7 CPU).

Because of the piecewise rigid approach used, there is no concept of continuity at joints, therefore the extracted part surfaces exhibit seams between parts, which leads to visual artefacts. Future work will investigate extending the system for merging of the reconstructed geometry at the joints seems or using non-rigid representations for continuous integration between parts.

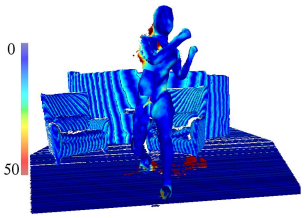


(a) RGB frame of synthetic sequence.



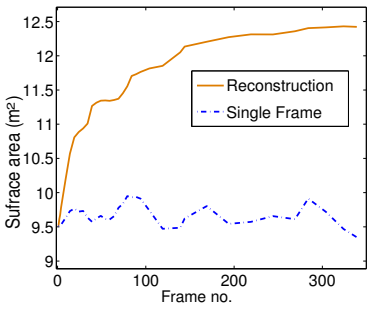
(b) Normal colour-mapped reconstructed geometry.

(c) Textured reconstructed geometry.

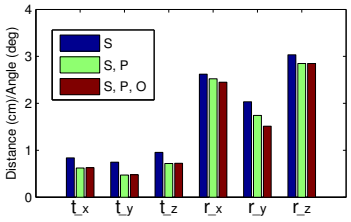



(d) Reconstructed surface error (blue - 0, red 5 cm).

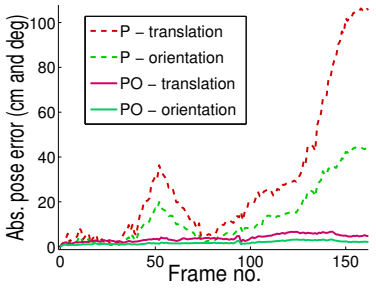
Figure 7: Results on synthetic 'boxing' sequence with noisy fixed sensor (FN).




(a) Surface area of individual depth frames vs. the integrated model over time (FN).




(b) RMS error in tracked pose for the upper left arm over all frames (FN).




(c) Absolute camera tracking (displacement and orientation) error for P and PO registration (MN).

Figure 8: Quantitative results for the synthetic sequences. Abbreviations are as follows: F - fixed sensor, M - moving sensor, C - clean data, N - noisy data. S - skeleton, P - projective data association/point-to-plane ICP, O - optical-flow based registration.


### Acknowledgement

This work was supported by the European Union FP7 project SCENE - www.3d-scene.eu.


## References


[1] I. Baran and J. Popović. Automatic rigging and animation of 3D characters. *ACM TOG*, 26, 2007. 6

[2] P. Besl and N. McKay. A method for registration of 3-D shapes. *TPAMI*, 14(2):239–256, 1992. 2

[3] G. Blais and M. Levine. Registering multiview range data to create 3D computer objects. *TPAMI*, 17(8):820–824, 1995. 4

[4] W. Chang and M. Zwicker. Global registration of dynamic range scans for articulated model reconstruction. *ACM TOG*, 30:15–26, 2011. 2

[5] Y. Chen and G. Medioni. Object modeling by registration of multiple range images. In *ICRA*, 1991. 2

[6] CMU Graphics Lab. Motion Capture Database. mocap.cs.cmu.edu. 6

[7] B. Curless and M. Levoy. A volumetric method for building complex models from range images. In *SIGGRAPH*, pages 303–312. ACM, 1996. 2

[8] P. Fechteler and P. Eisert. Recovering Articulated Pose of 3D Point Clouds. In *CVMP*, 2011. 2

[9] K. Khoshelham and S. O. Elberink. Accuracy and Resolution of Kinect Depth Data for Indoor Mapping Applications. *Sensors*, 12:1437–1454, 2012. 6

[10] W. Lorensen and H. Cline. Marching cubes: A high resolution 3D surface construction algorithm. In *SIGGRAPH*, 1987. 1

[11] N. Magnenat-Thalmann, R. Laperrière, and D. Thalmann. Joint-dependent local deformations for hand animation and object grasping. In *Graphics interface*, 1988. 6

[12] R. Newcombe, S. Izadi, O. Hilliges, D. Molyneaux, D. Kim, A. Davison, P. Kohli, J. Shotton, S. Hodges, and A. Fitzgibbon. KinectFusion: Real-time dense surface mapping and tracking. In *ISMAR*, 2011. 2, 3, 4, 5

[13] S. Pellegrini, K. Schindler, and D. Nardi. A generalisation of the ICP algorithm for articulated bodies. In *BMVC*, 2008. 2

[14] S. Weik. Registration of 3-D partial surface models using luminance and depth information. In *3DIM*, 1997. 2, 4

[15] G. Ye, Y. Liu, N. Hasler, X. Ji, Q. Dai, C. Theobalt, D. Wu, K. Li, Y. Deng, W. Deng, and C. Wu. Performance Capture of Interacting Characters with Handheld Kinects. In *ECCV*, 2012. 2