28

A Human Body Model for Articulated 3D Pose Tracking

Steffen Knoop, Stefan Vacek, Rüdiger Dillmann
University of Karlsruhe (TH)
Karlsruhe, Germany

1. Introduction

Within the last decade, robotic research has turned more and more towards flexible assistance and service applications. Especially when cooperating with untrained persons at small distances in the same workspace, it is essential for the robot to have a deep understanding and a reliable hypothesis of the intentions, activities and movements of the human interaction partner.

With growing computational capacities and new emerging sensor technologies, methods for tracking of articulated motion have become a hot topic of research. Tracking of the human body pose (often also referred to as Human Motion Capture) without invasive measurement techniques like attaching markers or accelerometers and gyroscopes demands (1) for algorithms that maximally exploit sensor data to resolve ambiguities that compulsorily arise in tracking of a high-degree-of-freedom system, and (2) for strong models of the tracked body that constrain the search space enough to enable fast and online tracking.

This chapter proposes a 3D model for tracking of the human body, along with an iterative tracking approach. The body model is composed of rigid geometric limb models, and joint models based on an elastic band approach. The joint model allows for different joint types with different numbers of degrees of freedom. Stiffness and adhesion can be controlled via joint parameters.

Effectiveness and efficiency of these models are demonstrated by applying them within an Iterative Closest Point (ICP) approach for tracking of the human body pose. Used sensors include a Time-of-Flight camera (depth camera), a mono colour camera as well as a laser range finder. Model and sensor information are integrated within the same tracking step for optimal pose estimation, and the resulting fusion process is explained, along with the used sensor model. The presented tracking system runs online at 20-25 frames per second on a standard PC.

We first describe related work and approaches, which partially form the basis for the presented models and methods. Then, a brief introduction into the ICP is given. The model for body limbs and joints is explained in detail, followed by a description of the full tracking algorithm. Experiments, examples and different evaluations are given. The chapter closes with a discussion of the achieved results and a conclusion.

Source: Humanoid Robots, New Developments, Book edited by: Armando Carlos de Pina Filho
ISBN 978-3-902613-02-8, pp.582, I-Tech, Vienna, Austria, June 2007

2. Related work

Tracking of human body motion is a highly active field in current research. Depending on the target application, many different sensors and models have been used. This includes invasive sensors like magnetic field trackers (Ehrenmann et al., 2003; Calinon & Billard, 2005) that are fixed to the human body. Within the context of human robot interaction in every-day life, this approach is not feasible; non-invasive tracking approaches must be applied. Most of these are based on vision systems, or on multi-sensor fusion (Fritsch et al., 2003). Systems which rely on distributed sensors (Deutscher et al., 2000) are not practicable in the given domain; the tracking system must be able to rely only on sensors mounted on the robot.

Several surveys exist on the area of tracking humans (Aggarwal & Cai, 1999; Gavrila, 1999; Moeslund & Granum, 2001; Wang et al., 2003). Possible applications range from the mentioned human-robot interaction to surveillance and security domains. Hence, there is a big variety of methods ranging from simple 2d approaches such as skin colour segmentation (Fritsch et al., 2002) or background subtraction techniques (Bobick & Davis, 2001) up to complex reconstructions of the human body pose. (Ramanan & Forsyth, 2003) shows how to learn the appearance of a human using texture and colour.

Sidenbladh (Sidenbladh, 2001) used a particle filter to estimate the 3d pose in monocular images. Each particle represents a specific configuration of the pose which is projected into the image and compared with the extracted features. (Cheung et al., 2003) use a *shape-from-silhouette* approach to estimate the human's pose.

A similar particle filtering approach is used in (Azad et al., 2004). The whole body is tracked based on edge detection, with only one camera. The input video stream is captured with 60Hz, which implies only small changes of the configuration between two consecutive frames. As it is a 2d approach, ambiguities of the 3d posture can hardly be resolved.

An ICP-based approached for pose estimation is shown in (Demirdjian & Darrell, 2002). The authors use cylinders to model each body part. In (Demirdjian, 2003) the same authors show how they model joint constraints for their tracking process. However, it the effect of the ICP is partially removed when the constraints are enforced. Nevertheless, parts of the work described in this chapter are based on the work of Demirdjian.

3. Sensors and framework

For tracking of the human in a human-robot interaction context, only the sensors onboard the robot can be used. In our setup, we use several different sensors as input for the tracking algorithm, which fuses all available information to obtain an optimal estimation of the current pose of the human.

3.1 Sensors

3D point clouds are acquired by a Time-of-Flight camera. This depth camera called Swissranger (CSEM, 2006) has a resolution of 160 x 124 pixels and a depth range of 0.5 to 7.5 meters. Fig. 1 shows the depth image of an example scene. Alternatively, point clouds from reconstruction of stereo images can be used.

A standard single FireWire camera is used to obtain colour images. These are processed by a standard skin-colour based algorithm to track head and hands in the image. All image regions which are candidates for skin regions are provided to the tracking algorithm.

A SICK laser range finder which is mounted for navigation purposes delivers 3D points in its measurement plane. These points are also provided to the tracking.

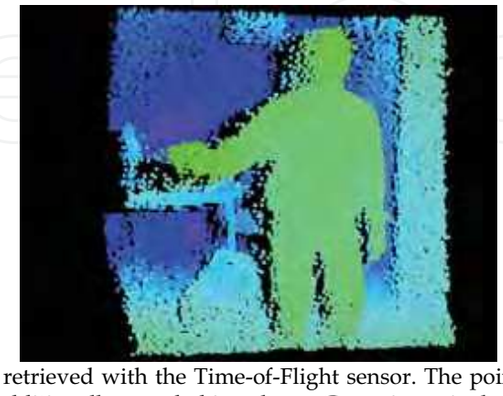

Fig. 1 Depth image, retrieved with the Time-of-Flight sensor. The point cloud is visualized in 3D, and depth is additionally encoded in colours. Green is equivalent to near, and blue to distant measurements.

3.2 Iterative Closest Point algorithm

This section gives a short introduction to the Iterative Closest Point (ICP) algorithm. The goal of the ICP is to match two indexed sets of the same points which are given in different coordinate systems and calculate the translation $\vec{t}$ and rotation R that transform the first coordinate system into the second.

For tracking, the first set corresponds to the data points of the sensor and the second set corresponds to points on the surface of a rigid body. Following (Besl & McKay, 1992), the first set is denoted $P = \{\vec{p}_i\}$, the second one $X = \{\vec{x}_i\}$. Both sets have the same size with $N_x = N_p = N$ and each point $\vec{p}_i$ corresponds to point $\vec{x}_i$.

With six degrees of freedom, at least three points are necessary to compute translation and rotation. Because sensor data is always corrupted with noise, no exact solution exists; instead, the problem is solved as a minimization problem for the sum of squared distances:

$$f(R,\vec{t}) = \frac{1}{N}\sum_{i=1}^{N} \left\| R(\vec{x}_i) + \vec{t} - \vec{p}_i \right\|^2 \tag{1}$$

With $\vec{\mu}_p$ and $\vec{\mu}_x$ as the mean value of P and N respectively, and with setting $\vec{p}_i' = \vec{p}_i - \vec{\mu}_p$, $\vec{x}_i' = \vec{x}_i - \vec{\mu}_x$ and $\vec{t}' = \vec{t} + R(\vec{\mu}_x) - \vec{\mu}_p$, equation (1) can be written as

$$f(R,\vec{t}) = \frac{1}{N}\sum_{i=1}^{N} \left\| R(\vec{x}'_i) - \vec{p}'_i + \vec{t}' \right\|^2 \tag{2}$$

This leads to

$$f(R,\vec{t}) = \frac{1}{N}\left(\sum_{i=1}^{N} \left\| R(\vec{x}'_i) - \vec{p}'_i \right\|^2 - 2\vec{t}\sum_{i=1}^{N} \left(R(\vec{x}_i') - \vec{p}_i' \right) + N\left\| \vec{t}' \right\|^2 \right) \tag{3}$$

In equation (3), the first part is independent from $\vec{t}'$, the second part reveals to zero. Therefore the function becomes minimal if $\vec{t}'=0$. Transformation yields

$$\vec{t} = \vec{\mu}_p - R(\vec{\mu}_x) \tag{4}$$

Having the optimal translation (giving $\vec{t}'=0$), equation (2) becomes:

$$f(R,\vec{t}) = \frac{1}{N}\sum_{i=1}^{N} \left\| R(\vec{x}'_i) - \vec{p}'_i \right\|^2 \tag{5}$$

Considering $\left\| R(\vec{x}_i') \right\| = \left\| \vec{x}_i' \right\|$, the equation can be rewritten as

$$f(R,\vec{t}) = \frac{1}{N}\left(\sum_{i=1}^{N} \left\| \vec{x}'_i \right\|^2 - 2\sum_{i=1}^{N} R(\vec{x}_i')\vec{p}_i' + \sum_{i=1}^{N} \left\| \vec{p}'_i \right\|^2 \right) \tag{6}$$

Maximizing

$$\sum_{i=1}^{N} R(\vec{x}'_i)\cdot \vec{p}'_i \tag{7}$$

Gives the optimal rotation. See (Horn, 1987) for details.

The described method can be used to compute the translation and rotation for two point sets P and N, where each $\vec{p}_i$ corresponds to $\vec{x}_i$. These point sets and their association has to be set up from the measurements and the surface model of the tracked body. The first set can be directly derived from the measured 3D point cloud. N is constructed by calculating for each data point $\vec{p}_i$ the *closest point on the model surface*, giving $\vec{x}_i$. This can be done with geometrical relations. Having these two indexed point sets, the optimal rotation and translation can be computed and applied to the model.

Association of model surface points to measurements can only be estimated. This estimation usually results in a non-optimal transformation, and the algorithm is iteratively applied until a minimum is reached. In summary, the *Iterative Closest Point* (ICP) algorithm works as follows:

1. For the given mode position and the given measurement M, calculate the closest points on the model giving CP_0.
2. Calculate the sum of squared distances between data points and model points, giving $d_0(M, CP_0)$.
3. Estimate rotation and translation and apply to the model.
4. Calculate the new set of closest points with the new model position, giving CP_i.
5. Calculate the sum of squared distances between data points and model points giving $d_i(M, CP_i)$.
6. If $d_{i-1}(M, CP_{i-1}) - d_i(M, CP_i) < \varepsilon$, stop iteration. Otherwise go to step 3.

Note that the computation of the closest points on the model is the most time consuming step in the ICP loop, since it includes geometrical calculations for each data point in the point cloud.

The ICP as mentioned above can only cope with one rigid body as model. For extension of the method to articulated models, the tracked body has to be modelled with a set of rigid bodies, which are connected to one articulated model. Two problems have to be solved in

this context: (i) data points must be associated with body parts, before they can be used for tracking, because each body part is transformed separately by the ICP. (ii) A mechanism must be established which introduces the joint constraints of the model in the tracking process, to avoid drift of the limbs due to separate tracking of each part.
The first issue is solved by checking each measurement point with all body part models, and assigning it to the geometrically closest body part. In addition, a threshold is used to discard measurements which have too high distance to the model.
To avoid that the body parts drift apart, we introduce a novel joint model. The complete body model together with the joint model will be explained in the next section.

4. Proposed body model

For the tracking system a 3d body model is used. Each body part is represented with a *degenerated cylinder*. The top and the bottom of each cylinder are described by an ellipse. The ellipses are not rotated to each other and the planes are parallel. In total such a body is described by five parameters: major and minor axis of each ellipse, plus the length of the cylinder.
The overall body model is built in a tree-like hierarchy starting with the torso as root body part. Each child is described with a degenerated cylinder and the corresponding transformation from its parent. Up to now the body model consists of ten body parts (torso, head, two for each arm and two for each leg) which is depicted in Fig. 2. It should be mentioned that this body model is not necessarily restricted to humans, and also other bodies can be modelled easily.

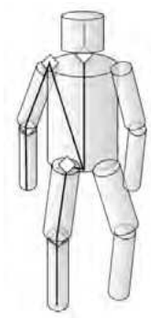

Fig. 2. Hierarchical body model, consisting of 10 cylinders and 9 joints. The torso describes the root limb.

If the fusion algorithm also incorporates data from feature trackers (like some vision based algorithms, or magnetic field trackers that are fixed on the human body), it is required to identify certain feature points on the human body. This is done following the H|Anim Specification (H|Anim, 2003).

4.1 Joint constraint model

The joint model we propose is based on the concept of introducing elastic bands into the body model. These elastic bands represent the joint constraints. For the ICP algorithm, the elastic bands can be modelled easily as artificial correspondences and will thus be considered automatically in each computation step.

For each junction of model parts, a set of elastic bands is defined (see Fig. 3). These relations set up corresponding points on both model parts. The corresponding points can then be used within the model fitting process to adjust the model configuration according to any sensor data input and to the defined constraints.

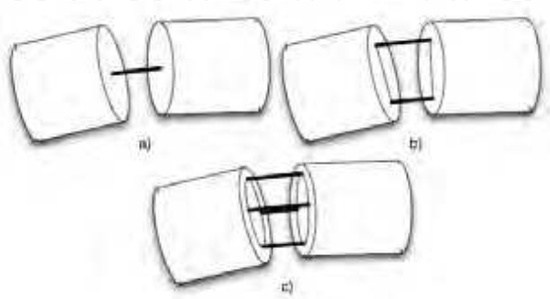

Fig. 3. Different joint types. Universal joint with 3 degrees of freedom (a), hinge joint with one full and 2 restricted degrees of freedom (b), and (c) elliptic joint with 3 restricted degrees of freedom.

4.2 Joint Constraint Types

With this approach, different types of joints can be modelled. Looking at a model for the human body, different kinds of joints with varying degrees of freedom are required:

- Universal Joints have 3 full degrees of freedom. This joint type can be found e.g. in the shoulder. The upper arm can rotate up/down, forward/backward and around its main axis. Universal joints are modelled by a point-to-point correspondence (one elastic band) between both body parts with one point on each, see Fig. 3 a).
- Hinge Joints have one real degree of freedom, the others being almost fixed. This can be found e.g. in the human knee or elbow (only 1 DoF), or in the hip (1 real DoF, the other two existing, but highly restricted in motion). Hinge joints are modelled by a set of correspondences which are distributed along a straight line on both body parts. The same restriction can be achieved with correspondences only at each end of the line (two elastic bands), see Fig. 3 b).
- Elliptic Joints have all degrees of freedom highly restricted. An example on the human body is the neck (or the wrist): Motion is possible in all 3 degrees of freedom: Left/right, forward/backward, and turning. Each direction is very limited in range. Elliptic joints are modelled by a set of correspondences distributed along an ellipse on both body parts. This restriction can be achieved with correspondences on each end of the main axes of the ellipse (four elastic bands), see Fig. 3 c).

Universal and hinge joints are special cases of the elliptic joint. For the hinge joint, one major axis of the ellipse is set to zero, resulting in a straight line. Setting both axes to zero produces a universal joint, because all correspondences are reduced to one point-to-point relation.

Following these definitions, each joint is modelled with a set of parameters describing the type of joint and its behaviour. This parameter set consists of the major axes of the ellipse, its position and orientation on both body parts, and the weight of the given correspondence. These parameters and the resulting behaviours are now described in detail.

Major axes: The model type (universal, hinge, elliptic) and the valid range of each degree of freedom control the choice for the major axes sizes and ratio. Universal joints are modelled with both ellipse axes set to zero. For hinge and elliptic joints, the axis direction defines the rotation axis, and the axis length defines the stiffness of the other two rotational degrees of freedom.
In Fig. 3 b), rotational flexibility around the z-axis (perpendicular to the image plane) and around the symmetric axis of the cylinders is very limited due to the modelled joint.
Position and orientation: Position and orientation of the point-to-point, hinge or elliptic joint model with respect to both body parts define the connection between both parts.
Weight: When the joint model is used within a bigger tracking framework the elastic bands can be used as correspondences which are included as tracking constraints. The use of measured correspondences together with artificial ones puts up the need for correct weighting strategies between input and model constraints.
To incorporate this, each joint model can be weighted with respect to measured input. This parameter is then used within the model-fitting algorithm to balance between measured input and joint constraints. The weight parameter is defined in relation to the number of 'natural' correspondences to keep the ratio between measurements and constraints.
To increase the weight of a joint model tightens the coupling between both model parts, by decreasing the coupling becomes looser. For hinge and elliptic joints, higher weight also increases stiffness of a kinematic chain. This makes sense especially for joints like the human neck or wrist which shows a very tight coupling and very limited angular range.
The proposed joint model provides a ''soft'' way to restrict the degrees of freedom for the model parts. It additionally provides means to control the ''degree of restrictiveness'' for each DoF in a joint. Applying elastic strips is tantamount to introducing a set of forces which hold the model parts together. The connected points and magnitude define the joint behaviour.
The soft joint model can e.g. be applied to the joint between human pelvis and thigh: While in forward/backward direction the movement is almost unrestricted, there is a high restriction for the left/right movement. But still a small movement is possible, and a 1-DoF model would not be sufficient.
Nevertheless, it is possible to model plain bending joints with a hinge joint model with one large axis.

4.3 Joint Model within the ICP

One of the main advantages of this joint model is that it can be very easily integrated in tracking algorithms. The joint model is added as a second data source, which adds correspondences between model and real world.
Introducing the joint model correspondences in the ICP framework (see sec. 3.2) is done by transforming the elastic band constraints into artificial input points according to the following rules:

- For each correspondence with the weight W, W artificial point pairs are generated.
- The artificial point pairs have to be added to the correspondence list after computation of the Closest Point Relations.
- Because each body part is processed separately, each joint model has to be added twice, once to each associated body part.

- The generated point pairs each represent one point on the model and the associated artificial data point. So each pair has to be added to one body part as Model - Data and to the other Data - Model relation to retrieve the desired forces (from the elastic bands) on both model parts. These forces then try to establish the modelled connections.
- The artificial correspondences are recalculated in each ICP step.
- The chosen weight of each joint depends on the desired stiffness of the model. To always achieve the same stiffness during tracking, the ratio between measured and artificial point relations has to be constant. This means that the number of generated artificial points for one body part in each step depends on the quantity of measurements for this part of the model. The generated relations are linearly scaled with the number of measurements.
- From our experience, the ratio r between measured and artificial points should be chosen as approximately $0.4 < r < 0.7$. This gives enough cohesion within the model without implying to hard and static relationships.

It is important to note that the introduction of multiple identical correspondences within the ICP does not increase computation time with the order of point counts (like a set of different measured points would). The only additional effort consists of one multiplication of the resulting point matrix (4x4) with the scalar weight W.

5. Data fusion for tracking

The goal of the tracking system is to track the posture of a human body in 3d by matching the internal 3d body model with the current input sensor data. Thus, the tracking system offers three interfaces: sensor data stream (input), parameter configuration (input), and current posture estimation (output). All sensor data formats that can be exploited are described in sec. 5.1. The configuration values we have identified will be described in sec. 5.2 along with the processing steps.

The current posture estimation output is given with respect to the hierarchical body model defined in sec. 4. In each time step, the whole body model is provided. This allows for changes not only in the body pose (joint angle space), but also for changes in the model itself (configuration and parameters of the body model). This may concern scaling of the model for different persons with varying body heights, or even addition and deletion of body parts in case of changing tracking targets or other effects. This can be useful e.g. if the tracked person is holding and handling a big object, which then can be added easily to the tracked configuration.

The fusion algorithm, which is implemented in a tracking system called *VooDoo*, is depicted in Fig. 4. The next section describes possible input data, while sec. 5.2 depicts the processing steps within the tracking loop.

5.1 Input data

The proposed tracking algorithm is able to include, process and fuse different kinds of sensor data (see also Fig. 4):

- *Free 3d points* from ToF-sensors or from pure stereo depth images. The system has to decide whether to use these points as measurements of the tracked model. For a point that is not discarded, the corresponding point on the model surface is computed.

- *3d points on the human body* that are e.g. generated by a stereo vision system that tracks a person in image space and generates the corresponding 3d points by stereo reconstruction.
- *3d points assigned to a single body part* may also be generated by a stereo vision system tracking special body parts like the face or the hands.
- *3d point-to-point relations* are 3d points that can be assigned to a given point on the tracked human body. Thus, tracking of special features or points (e.g. with markers, or magnetic field trackers attached to the human body) can be integrated.
- *2d point-to-line relations* can e.g. be derived from a 2d image space based tracker. The pixel in the image plane together with the focal point define a ray in 3d, which corresponds to the point on the human body that has been detected in the image.

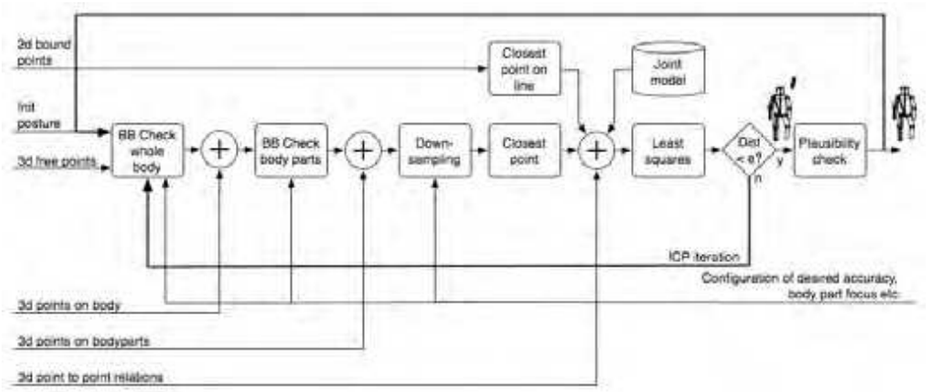

Fig. 4. The complete tracking algorithm, containing all input data streams. "BB" denotes Bounding Box.

This data can originate from any sensor that gives data in the described format. Obviously, all input data has to be transformed into the tracker coordinate system before it is used within the system. In our setup, we use the 3d point clouds from laser range finder and depth camera as free 3d points. Extracted features (hands, head) from the colour image are projected onto 2d point-to-line relations.

5.2 Processing

For the ICP matching algorithm, a list of corresponding point pairs has to be set up for each limb. Therefore, all "free" 3d points have to be analyzed in order to decide whether they correspond to points on the tracked model. Otherwise, they are discarded. Additionally, all given correspondences from other tracking procedures and the background knowledge on joint constraints have to be added to the correspondences list. Then, the optimal resulting model configuration has to be computed. These steps are performed iteratively until an optimum of the configuration is reached.

Before the input data of one time step is processed, it is possible to adjust internal model parameters. This can be e.g. the model scale factor, or particular cylinder sizes. Even limbs can be added to or removed from the model.

The processing steps are now described in detail.

Prefiltering free 3d points: The whole point cloud of free 3d points from used depth sensors is processed in order to remove all points that are not contained within the bounding box of the body model (see Fig. 4, step *BB Check whole body*). This is done on the assumption that the body configuration changes only locally between two time frames. A parameter defines an additional enlargement of the bounding box prior to this filtering step. The resulting point list is concatenated with any sensor data input that has already assigned its measured 3d points with the tracked. It results in a list of 3d points which are close to the body model and thus are candidates for measurements of the tracked body.

Assigning points to limb models: The point list is now processed in order to assign measured points to dedicated limb models based on the bounding box of each limb model (see Fig. 4, step *BB Check body parts*). Again, the bounding boxes can be enlarged by a parameter to take the maximum possible displacement into account. Points that do not fall in any bounding box are again removed. Several behaviours can be selected for points that belong to more than one bounding box (overlap): These points are either shared between limb models, exclusively assigned to one limb or shared only in case of adjacent limbs. This last method avoids collisions between limbs that are not directly connected. The resulting point list can be joined with any sensor data input that has already assigned its measured 3d points with dedicated limbs of the tracked body. The resulting point list contains candidates for measurements of each limb.

Point Number reduction: The resulting point list can be downsampled before the calculation of the closest points to reduce the overall number of points (see Fig. 4, step *Downsampling*). This step is controlled by three parameters: the sampling factor, and minimum and maximum number of points per limb. Thus, it is possible to reduce the number of points for limbs with many measurements, but maintain all points for limbs which have been measured with only a few points.

Closest point computation: The closest point calculation is the most time-consuming step in the whole loop. For each remaining data point, the corresponding model point on the assigned limb model has to be computed for the ICP matching step (see Fig. 4, step *Closest Point*). This involves several geometric operations. Depending on the resulting distance between data and model point, all points within a given maximum distance are kept and the correspondence pair is stored in the output list. All other points are deleted.

3d point-to-point relations from input data can now be added to the resulting list, which holds now corresponding point pairs between data set and model.

Addition of 2d measurements: Each 2d measure (e.g. tracked features in 2d image plane of a camera) of a feature on the human body defines a ray in 3d which contains the tracked feature. This fact is used to add the 2d tracking information to the 3d point correspondences (see Fig. 4, step *Closest point on line*): For each reference point on the body model, the closest point on the straight line is computed and added to the list.

Joint model integration: The joint model for each junction is added as artificial point correspondences for each limb, depending on the limb type (see Fig. 4, step *Joint model*). According to sec. 4.1, the correspondences can be interpreted as elastic bands which apply dedicated forces to the limbs to maintain the model constraints. Thus, artificial correspondences will keep up the joint constraints in the fitting step.

Model fitting: When the complete list of corresponding point pairs has been set up, the optimal transformation between model and data point set can be computed according to sec. 3.2 (Fig. 4, step *Least squares*). The transformation is computed separately for each limb.

When all transformations have been computed, they can be applied to the model. The quality measure defined in sec. 3.2 is used for the fitting. All are repeated until the quality measure is below a given threshold or a maximum number of steps has been performed.

6. Sensor model

Each used data source has its own stochastic parameters which have to be taken into account. The described approach offers a very simple method for this: each input date is weighted with a measure that describes its accuracy. The ICP algorithm then incorporates these weights in the model-fitting step. Thus it is possible to weight a 2d face tracker much higher than a single 3d point from a Time-of-Flight camera, or to weight 3d points from a Time-of-Flight-camera slightly higher than points from the stereo reconstruction due to the measuring principle and the sensor accuracy.
It is important to note that an increased weight for a single point does not affect the time needed for the computation. This is very important and is due to the fact that in the presented approach, each measurement is projected into model space. This is different to e.g. particle filtering approaches, where each particle is projected into each sensor's measurement space to compute the likelihood. In consequence, adding a sensor source to the tracking framework increases computation time only with the number of different measurements from the sensor.
An example configuration can be seen in Fig. 5. The model consists of two cylinders, connected by a linear joint. The measurements contain a 3d point cloud, and a 3d measurement of one end point. This configuration can e.g. result from a stereo depth image of a human arm and a colour based hand tracker.

Fig. 5. Different weights for measurements from different sensors, projected into 3D model space. The depicted point sizes correspond to the sensor data weight, the lines indicate the closest-point relations. These pictures motivated the system name *VooDoo*.

7. Experiments, evaluation and results

The described tracking procedure has been implemented and tested with a Time-of-Flight camera and a stereo camera. The tracking runs online at a frame-rate of appr. 20-25Hz on a Pentium4 with 3.2GHz with a model of a human body, consisting of 10 cylinders with 9 joints. For the experiments, the same data sequences have been processed using different input sensor configurations to test the fusion.
Fig. 6 shows example images from a sequence of 15 seconds containing a "bow" and a "wave" movement. The first row shows the scene image, which has been also used for segmentation of face and hands. The second and third row contain the tracking result

with 3d data only (row 2) and 2d data only (row 3), where the 3d data has been acquired with the Time-of-Flight camera and the 2d data is derived from skin colour segmentation in one image of the stereo camera. The rays in 3d defined by the skin colour features can be seen here. Row 4 shows the tracking result with both inputs used.
For the shown results, the following weights for the input data have been used: 3d data points w = 1.0, face tracker w = 30.0, hand tracker w = 20.0.

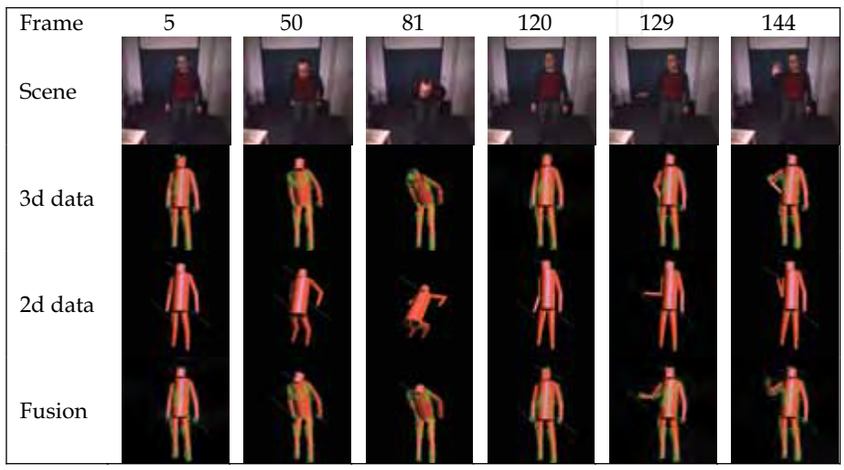

Fig. 6. Experiments with different sensor inputs, taken from a sequence containing a “bow” and a “wave” movement. The frame number is displayed on the top. The used 2d and 3d correspondences have been added to the resulting model images.

Different conclusions can be drawn from the results:

- Huge movements are easily detected by the 3d data based tracking: The ``bow'' movement is tracked quite well. On the other hand, fast movements with the extremities may cause failures when only 3d data is used, as with the ``wave'' movement.
- Tracking only with a 2d feature tracker works quite well for the tracked body parts. Nevertheless, the body configuration cannot be determined only from 2d features (see frame 81). To do this, a lot more background information on the human body would be needed.
- Fusion of both input sensors in 3d shows very good results: Huge body movements as well as fine and fast movements of the extremities can be recognized, and the algorithm is able to reliably track the body configuration.

As already stated in sec. 6, the computational effort and thus the frame-rate depends on the true number of different measurements, independent of the particular weights. To evaluate the computational performance and frame-rate of the presented method, several analyses have been carried out. The model corresponds again to the human body model used for the analysis in Fig. 6.

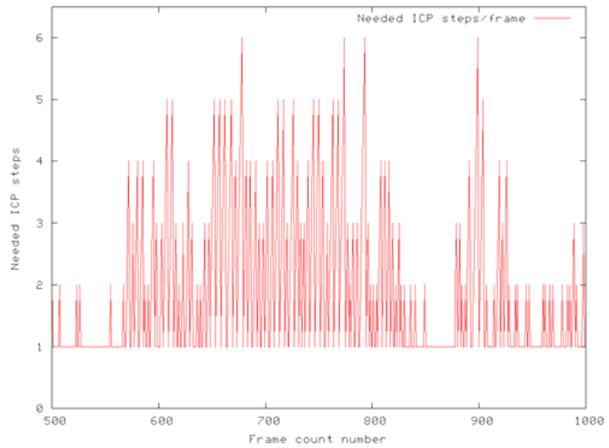

Fig. 7. Number of ICP steps required for a typical tracking sequence.

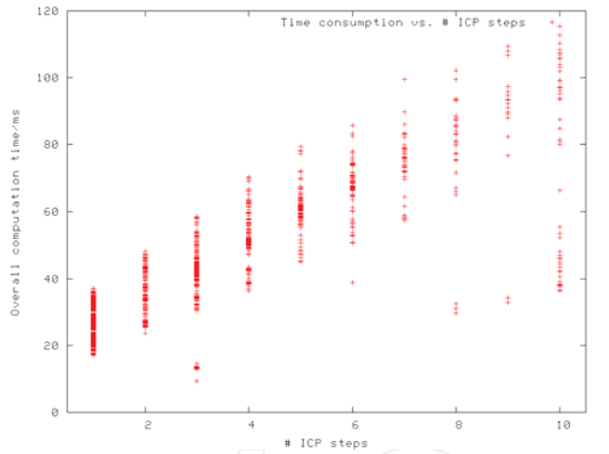

Fig. 8. Time consumption per ICP step vs. number of ICP steps.

The computational effort for one frame depends first of all on the number of ICP steps needed. The number of iterations again depends on the body displacement between two consecutive frames. Fig. 7 shows the number of required ICP steps during a typical tracking sequence for a human body model. During phases without large movements, one iteration is enough to approximate the body pose (frame 500 to 570). Extensive movements are compensated by more ICP iteration steps per frame (650 to 800).

The required time per frame obviously increases with the number of needed ICP steps. This relation is shown in Fig. 8. A maximum number of 6 ICP steps has turned out to be a good trade-off between time consumption per frame and tracking accuracy. This leads to a frame period of 20 - 70ms, which corresponds to a frame-rate of 14.2 to 50Hz. The maximum frame-rate in our framework is only constrained by the camera frame-rate, which is 30Hz.

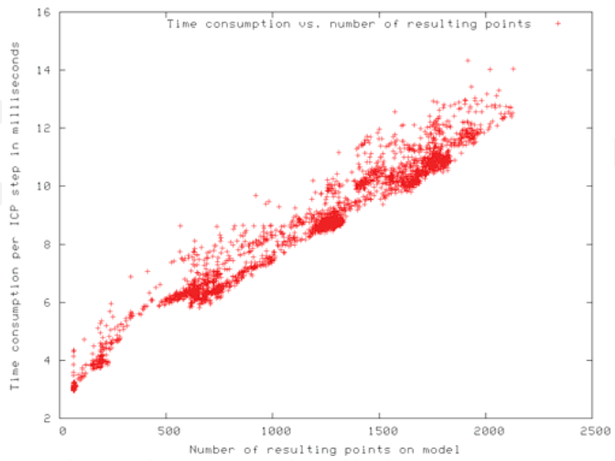

Fig. 9. Time consumption per frame vs. number of body measurements.

The relation between the number of body measurements and the computational effort for one ICP step is depicted in Fig. 9. For each measurement of the target, several computations have to be carried out. This leads to the dependency in Fig. 9. As expected, the time scales linearly with the number of measurements.

These results show that the presented tracking approach is able to incorporate several thousand measurements with reasonable computational effort. One disadvantage of the depicted iterative process is the negative dependency between target displacement and computational effort: The faster the target moves, the longer the tracking needs for one frame, which again leads to larger displacements due to the low frame-rate.

To overcome this, one has to find a good trade-off between accuracy and frame-rate. This compromise depends on the tracking target characteristics, as well as on the application which utilizes the Human Motion Capture data. It is also possible to switch between different behaviours, taking into account the requirements the applications which depend on the Motion Capture data: in case the data is used for physical interaction (e.g. handing over objects), the required accuracy is high, along with usually low dynamics. On the other hand, if the target is only to observe a human in the robot's environment, the required accuracy is low, but the person moves with high velocity.

8. Discussion and conclusion

This paper has proposed a geometric human body model, a joint model and a way for fusion of different input cues for tracking of an articulated body. The proposed algorithm is able to process 3d as well as 2d input data from different sensors like ToF-cameras, stereo or monocular images. It is based on a 3d body model which consists of a set of degenerated cylinders, which are connected by an elastic bands joint model. The proposed approach runs in real-time. It has been demonstrated with a human body model for pose tracking.

The main novelty and contribution of the presented approach lies in the articulated body model based on elastic bands with soft stiffness constraints, and in the notion of point correspondences as a general measurement and model format. Different joint behaviours

can be modelled easily by distributing the elastic bands along two axes in the joint. The joint constraints are incorporated in the ICP as artificial measurements, so measurements and model knowledge are processed identically. The model can also be refined by adding cylindrical primitives for hands, fingers and feet. This is reasonable if the accuracy and resolution of the available sensors are high enough to resolve e.g. the hand posture, which is not the case in our approach due to the large distance between human and robot and the low measurement resolution.

The idea of introducing artificial correspondences into the fitting step can even be exploited further. Current works include further restriction of the joints in angular space by adding angular limits to certain degrees of freedom, which are maintained valid by artificial point correspondences. These will be generated and weighted depending on the current body configuration.

Our implementation of the described tracking framework has been released under the GPL license, and is available online at wwwiaim.ira.uka.de/users/knoop/VooDoo/doc/html/, along with sample sequences of raw sensor data and resulting model sequences.

9. References

Aggarwal, J. K.; Cai, Q. (1999) *Human motion analysis: A review*, Computer Vision and Image Understanding: CVIU, vol. 73, no. 3, pp. 428–440.

Azad, P.; Ude, A.; Dillmann, R.; Cheng, G. (2004) *A full body human motion capture system using particle filtering and on-the-fly edge detection*, in Proceedings of the IEEE-RAS/RSJ International Conference on Humanoid Robots. Santa Monica, USA.

Besl, P. J.; McKay, N. D. (1992) *A method for registration of 3-d shapes*, IEEE Transactions on pattern analysis and machine intelligence, vol. 14, no. 2, pp. 239–256, February.

Bobick, A. F.; Davis, J. W. (2001) *The recognition of human movement using temporal templates*, IEEE Transactions on Pattern Analysis and Machine Intelligence, no. 3, pp. 257–267.

Calinon, S.; Billard, A. (2005), *Recognition and reproduction of gestures using a probabilistic framework combining PCA, ICA and HMM*, in Proceedings of the International Conference on Machine Learning (ICML), Bonn, Germany

Cheung, G. K. M.; Baker, S.; Kanade, T. (2003) *Shape-from-silhouette of articulated objects and its use for human body kinematics estimation and motion capture*, in Computer Vision and Pattern Recognition.

CSEM (2006) Swissranger website. http://www.swissranger.ch

Demirdjian, D.; Darrell, T. (2002) *3-d articulated pose tracking to untethered deictic references*, in Multimodel Interfaces, pp. 267–272.

Demirdjian, D. (2003) *Enforcing constraints for human body tracking*, in Conference on Computer Vision and Pattern Recognition, Workshop Vol. 9, Madison, Wisconsin, USA, pp. 102–109.

Deutscher, J.; Blake, A.; Reid, I. (2000), *Articulated body motion capture by annealed particle filtering*, in Computer Vision and Pattern Recognition (CVPR), Hilton Head, USA, pp. 2126–2133.

Ehrenmann, M.; Zöllner, R.; Rogalla, O.; Vacek, S.; Dillmann, R. (2003). *Observation in programming by demonstration: Training and execution environment*, in Proceedings of Third IEEE International Conference on Humanoid Robots, Karlsruhe and Munich, Germany.

Fritsch, J.; Lang, S.; Kleinehagenbrock, M.; Fink, G. A.; Sagerer, G. (2002) *Improving adaptive skin color segmentation by incorporating results from face detection*, in Proc. IEEE Int. Workshop on Robot and Human Interactive Communication (ROMAN). Berlin, Germany

Fritsch, J.; Kleinehagenbrock, M.; Lang, S.; Plötz, T.; Fink, G.A.; Sagerer, G. (2003), *Multi-modal anchoring for human-robot-interaction*, Robotics and Autonomous Systems, Special issue on Anchoring Symbols to Sensor Data in Single and Multiple Robot Systems, vol. 43, no. 2–3, pp. 133–147.

Gavrila, D. M. (1999) *The visual analysis of human movement: A survey*, Computer Vision and Image Understanding, vol. 73, no. 1, pp. 82–98.

H|Anim (2003), *Information technology – Computer graphics and image processing – Humanoid animation (H-Anim), Annex B*, ISO/IEC FCD 19774, Humanoid Animation Working Group, Specification.

Horn, B. K. P. (1987) *Closed-form solution of absolute orientation using unit quaternions*, Optical Society of America Journal A, vol. 4, pp. 629–642, Apr. 1987.

Knoop, S.; Vacek, S. & Dillmann, R. (2005). *Modelling Joint Constraints for an Articulated 3D Human Body Model with Artificial Correspondences in ICP*, Proceedings of the International Conference on Humanoid Robots (Humanoids 2005), Tsukuba, Japan, December 2005, IEEE-RAS

Knoop, S.; Vacek, S. & Dillmann, R. (2006). *Sensor Fusion for 3D Human Body Tracking with an Articulated 3D Body Model*. Proceedings of the IEEE International Conference on Robotics and Automation (ICRA), Orlando, Florida, May 2006

Knoop, S.; Vacek, S. & Dillmann, R. (2006). *Sensor fusion for model based 3D tracking*. Proceedings of the IEEE International Conference on Multisensor Fusion and Integration for Intelligent Systems (MFI), Heidelberg, Germany, September 2006

Moeslund, T. B.; Granum, E. (2001) *A survey of computer vision-based human motion capture*, Computer Vision and Image Understanding, vol. 81, no. 3, pp. 231–268.

Ramanan, D.; Forsyth, D. A. (2003) *Finding and tracking people from the bottom up*, in Computer Vision and Pattern Recognition, vol. 2, 18-20 June, pp. II–467–II–474.

Sidenbladh, H. (2001) *Probabilistic tracking and reconstruction of 3d human motion in monocular video sequences*, Ph.D. dissertation, KTH, Stockholm, Sweden.

Wang, L.; Hu, W.; Tan, T. (2004) *Recent developments in human motion analysis*, Pattern Recognition, vol. 36, no. 3, pp. 585–601, 2003.and Electronics Engineers.

Humanoid Robots: New Developments

Edited by Armando Carlos de Pina Filho

ISBN 978-3-902613-00-4

Hard cover, 582 pages

Publisher I-Tech Education and Publishing

Published online 01, June, 2007

Published in print edition June, 2007

For many years, the human being has been trying, in all ways, to recreate the complex mechanisms that form the human body. Such task is extremely complicated and the results are not totally satisfactory. However, with increasing technological advances based on theoretical and experimental researches, man gets, in a way, to copy or to imitate some systems of the human body. These researches not only intended to create humanoid robots, great part of them constituting autonomous systems, but also, in some way, to offer a higher knowledge of the systems that form the human body, objectifying possible applications in the technology of rehabilitation of human beings, gathering in a whole studies related not only to Robotics, but also to Biomechanics, Biomimmetics, Cybernetics, among other areas. This book presents a series of researches inspired by this ideal, carried through by various researchers worldwide, looking for to analyze and to discuss diverse subjects related to humanoid robots. The presented contributions explore aspects about robotic hands, learning, language, vision and locomotion.

InTech Europe

University Campus STeP Ri
Slavka Krautzeka 83/A
51000 Rijeka, Croatia
Phone: +385 (51) 770 447
Fax: +385 (51) 686 166
www.intechopen.com

InTech China

Unit 405, Office Block, Hotel Equatorial Shanghai
No.65, Yan An Road (West), Shanghai, 200040, China
中国上海市延安西路65号上海国际贵都大饭店办公楼405单元
Phone: +86-21-62489820
Fax: +86-21-62489821

IntechOpen