Sensor Fusion for 3D Human Body Tracking with an Articulated 3D Body Model

Steffen Knoop, Stefan Vacek, and Rüdiger Dillmann
Industrial Applications of Informatics and Microsystems
Institute of Computer Science and Engineering
University of Karlsruhe, Germany
Email: {knoop,vacek,dillmann}@ira.uka.de

***Abstract*— This paper proposes a tracking system called *VooDoo* for 3d tracking of human body movements based on a 3d body model and the *Iterative Closest Point* (ICP) algorithm. The proposed approach is able to incorporate raw data from different input sensors, as well as results from feature trackers in 2d or 3d. All input data is processed within the same model fitting step by modeling all input measurements in 3d model space. The system has been implemented and runs in realtime at appr. 10-14 Hz. Experiments with complex human movements exhibit the characteristics and advantages of the proposed approach.**

I. Introduction

Robots that are meant to cooperate closely with humans, and especially with untrained persons which are not familiar with the domain of robotics, need a deep understanding of the intentions, activities, actions and movements of their interaction partner.

This is on the one hand due to the fact that the robot needs the ability to predict the global plan as well as single movements of the human in order to plan its own actions and movements in an efficient way with respect to the overall goal. Even if parts of the goal can be explicitly communicated between human and robot, there are in most cases several ways to reach a given goal, especially in a cooperation context. Thus, not only motion prediction, but also activity recognition is an indispensable feature for such a robot.

On the other hand, a shared workspace between robot and human puts up high safety demands. This includes not only collision detection, but also haptic interaction and shared object and tool manipulation. Therefore, observation and prediction of the human's movements is badly needed in a robot system that is designed to work together with humans.

Many tracking systems for humans have been proposed in literature, some of which are discussed in sec. II. Most of these are designed for one special input sensor, and all internal models are based on this assumption.

This paper introduces a 3d body model based tracking system called *VooDoo*, and especially proposes a new approach for fusion of different input sensors and cues for tracking. This approach is able to incorporate tracking information from 3d sensors like *Time-of-Flight*-cameras (ToF) or stereo reconstruction together with cues from 2d based trackers like a monocular camera. The system is designed to work only with sensors on-board the robot.

The system is able to track a person in realtime at about 10-14 Hz in 3d. Results are shown with different input sensors.

II. State of the Art

For observation and tracking of human movements, many different sensors and models have been used. This includes invasive sensors like magnetic field trackers (see [1], [2]) 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 (see [3]). Systems which rely on distributed sensors (see [4]) are not practicable in the given domain; the tracking system must be able to rely only on sensors mounted on the robot.

Tracking of humans and human body parts using vision is investigated by a lot of research groups and several surveys exist (see [5], [6], [7], [8]). Hence, there is a big variety of methods ranging from simple 2d approaches such as skin color segmentation (e.g. [9]) or background subtraction techniques (e.g. [10]) up to complex reconstructions of the human body pose. [11] shows how to learn the appearance of a human using texture and color.

Sidenbladh [12] 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. [13] use a *shape-from-silhouette* approach to estimate the human's pose. A similar particle filtering approach is used in [14]. The whole body is tracked based on edge detection, with only one camera. The input video stream is captured with 60 Hz, 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 [15]. The authors use cylinders to model each body part. In [16] 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 paper are based on the work of Demirdjian et al.

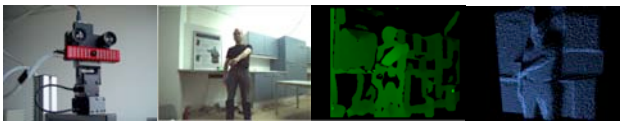

Fig. 1. Sensor head (left), 2d image (middle left), disparity image (middle right), 3d image (right)

III. Used Framework

This section describes the framework which is used for the presented work: Used sensors, the ICP algorithm which forms the basis, the articulated 3d human model and the joint model within the body model.

A. Sensor Data

In the described framework, two different sensors are used to demonstrate the capabilities of the algorithm: A time-of-flight (ToF) camera and a standard stereo camera head with depth data reconstruction generate 3d point clouds, and the color information of the camera is used to track face and hands with a simple skin color model in 2d.

The *Swissranger* ToF camera uses a resolution of 160×124 pixels. The output consists of a dense depth image and an intensity image. The depth range is configured to $0.5\ m \leq$ *range* $\leq 7.5\ m$, the accuracy lies within a few centimeters. Intensity data is not used within the current context, as the intensity image has very low resolution and high noise due to the sensor concept.

The stereo camera (*mega-d* from *videre design*) is used at a resolution of 320×240. The disparity image is computed based on a calibration obtained offline.

The sensors and the raw data can be seen in fig. 1.

B. 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 $\mathbf{R}$ that transform the first coordinate system into the second. For person 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 [17], 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 $\vec{x_i}$.

Because the sensor data is always corrupted with noise, no exact solution exists. Instead, the problem is transformed into the minimization of a sum of squared distances:

$$f(\mathbf{R}, \vec{t}) = \frac{1}{N} \sum_{i=1}^{N} ||\mathbf{R}(\vec{x_i}) + \vec{t} - \vec{p_i}||^2 \qquad (1)$$

For a complete description of how to compute the optimal translation and rotation, see [18].

The sensor data consists of a list of data points, which has to be matched to a geometrical description of the body. To retrieve the ordered list of point pairs needed for the ICP, the correspondences between data and model have to be constructed.

This is done by calculating for each data point $\vec{p_i}$ the geometrically *closest point* on the model giving $\vec{x_i}$. In the second step the optimal translation and rotation can be estimated and applied to the model. This process is then repeated until the absolute value of the transformation is below some threshold. The *Iterative Closest Point* steps are:

1) For the given model and the data points calculate the closest points 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 new set of closest point with the new position of the model giving CP_i
5) Calculate the sum of squared distances between data points and model points giving $d_i(M, CP_i)$
6) If $d_{i-1}(M, CP_{i-1}) - d_i(M, CP_i) < \epsilon$ the iteration stops, otherwise go to step 3.

Note that computation of *closest point relations* is by far the most time consuming step in the ICP process, since it includes a set of geometric calculations for each data point in the point cloud.

C. Human 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 is described by an ellipse. The ellipses are not rotated to each other and the planes are parallel.

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 on the left of fig. 2. It should be mentioned that this body model is not necessarily restricted to humans, and also other bodies can be modeled easily.

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* (see [19]).

D. Joint Model

The joint model is based on the concept of introducing elastic bands into the body model. These elastic bands represent the joint constraints. For the ICP algorithm, these elastic bands can be modeled as artificial correspondences and will thus be considered automatically in each computation step (see sec. IV-B.6).

For each junction of model parts, a set of elastic bands is defined (see fig. 2). 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

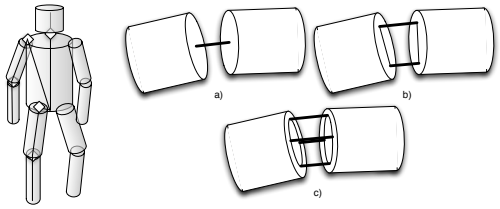

Fig. 2. Different joint type models. Universal Joint 3 DoF a), Hinge Joint 1 DoF + 2 restricted DoF b), Elliptic Joint 3 restricted DoF c)

configuration according to all sensor data input and to the defined constraints.

This approach allows for modeling of different joint types. Within the described tracking system, three types are used:

- **Universal Joints** have 3 full degrees of freedom (e.g. human shoulder).
  Universal joints are modeled by one point-to-point correspondence (one elastic band), see fig. 2 a).
- **Hinge Joints** have one real degree of freedom, the others being almost fixed (e.g. elbow or knee).
  Hinge joints are modeled by a set of correspondences which are distributed along a straight line, see fig. 2 b).
- **Elliptic Joints** have all degrees of freedom highly restricted. An example on the human body is the neck: Motion is possible in all 3 degrees of freedom, but very limited in range.
  Elliptic joints are modeled by a set of correspondences distributed along an ellipse, see fig. 2 c).

For details of the joint model, see also [20].

IV. Sensor Fusion Algorithm for Tracking

The goal of the *VooDoo* 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 section IV-A. The configuration values we have identified will be described in sec. IV-B along with the processing steps.

The current posture estimation output is given with respect to the hierarchical body model defined in sec. III-C. 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 *VooDoo* tracking algorithm is depicted in fig. 3. The next section describes possible input data, while sec. IV-B depicts the processing steps within the tracking loop.

A. Input data

The proposed tracking algorithm is able to include, process and fuse different kinds of sensor data (see also fig. 3):

- *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.

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.

B. Processing

For the ICP matching algorithm, a list of corresponding point pairs has to be set up for each limb (see also sec. III-B). 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 tracking algorithm and the sensor fusion approach are now described step by step.

1) 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. 3, 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.

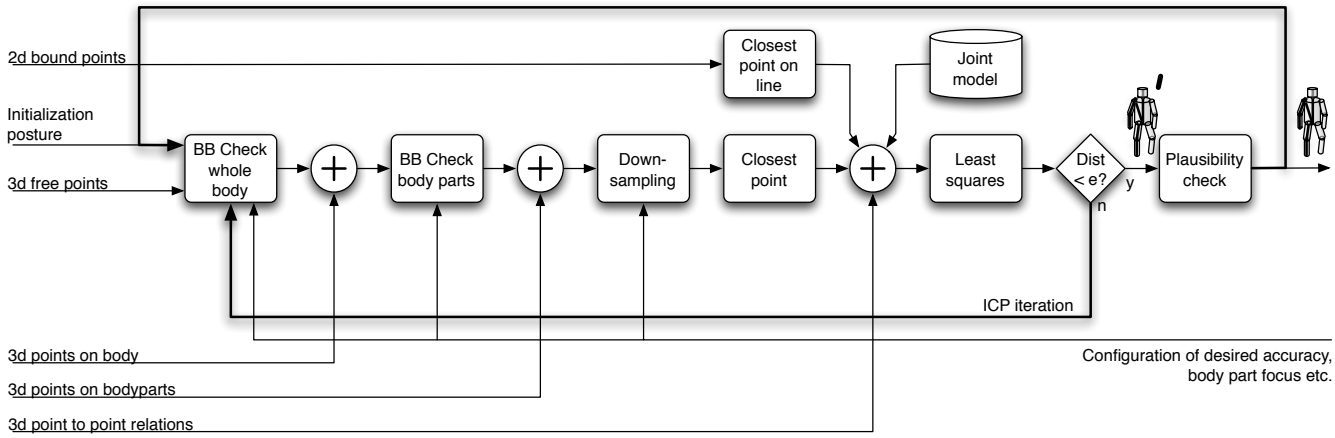

Fig. 3. The complete VooDoo algorithm ("BB" = Bounding Box)

The resulting point list is concatenated with any sensor data input that has already assigned its measured 3d points with the tracked body (see sec. IV-A). This results in a list of 3d points which are close to the body model and thus are candidates for measurements of the tracked body.

2) 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. 3, 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 behaviors 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 (see sec. IV-A). The resulting point list contains candidates for measurements of each limb.

3) 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. 3, 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.

4) 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. 3, 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 (see sec. IV-A) can now be added to the resulting list, which holds now corresponding point pairs between data set and model.

5) 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. 3, 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.

6) 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. 3, step *Joint model*). According to sec. III-D, 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.

7) 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. III-B (fig. 3, step *Least squares*). The transformation is computed seperately for each limb.

When all transformations have been computed, they can be applied to the model. The quality measure defined in sec. III-B is used for the fitting. Steps IV-B.1 to IV-B.7 are repeated until the quality measure is below a given threshold or a maximum number of steps have been performed.

C. 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 ToF camera.

It is important to note that an increased weight for a single point does not affect the time needed for the computation.

V. Experiments and Results

The described tracking procedure has been implemented and tested with the sensors described in sec. III-A. The tracking runs online at a framerate of appr. 10-14 Hz on a Pentium 4 with 3.2 GHz.

Different test series have been performed to evaluate the *VooDoo* system: First, the same data sequences have been processed using different input sensor configurations to test the fusion, and second, a set of 100 sequences has been recorded and processed. The tracking result has then been evaluated manually for consistency with the recorded body movements to evaluate the overall system performance.

Fig. 4 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 ToF camera and the 2d data is derived from skin color segmentation in one image of the stereo camera. The rays in 3d defined by the skin color 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 $w = 20.0$.

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 can not 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.

The second evaluation step consisted in recording a set of 100 sequences which contained ten different movements from several persons: e.g. *point somewhere*, *walk*, *wave*, *shake hands* with somebody, *bow* or *clap*. The tracking result has then been evaluated and classified manually into one of three classes: (0) *Tracking lost* somewhere within the sequence, (1) *acceptable deviations* like a temporally lost (but recovered) forearm within a walking sequence, and (2) *good congruence* between original and resulting model movements. The evaluation result is depicted in tab. I, the average result is $\oslash = 1.58$.

TABLE I
Evaluation result with 100 sequences

Tracking result	0	1	2
# of sequences	5	32	63

VI. Discussion

The proposed tracking approach does not include any background knowledge apart from kinematic constraints, i.e. no assumptions like "the torso stands always upright" are made. This implies on the one hand that all possible configurations can be recognized; on the other hand, the tracking can only succeed if the input data contains all necessary information to determine the human posture, and no tracking hypothesis can be generated for temporarily invisible body parts or ambiguous configurations.

The current framerate is appr. 10-14 Hz. The computation time depends on several factors: It scales linearly with the number of measured 3d points on the model; background points are removed in an early stage and do not distinctly influence framerate. It also depends on the number of ICP steps performed in each frame, which is appr. 3-15, depending on the desired accuracy and the speed of the movement.

Sec. V has shown that tracking based only on the measurements of the ToF camera is not sufficient. Especially movements along the main axis of the body (e.g. sitting down) can hardly be detected, which substantiates again the use of different data inputs for a fusion algorithm.

VII. Conclusion

This paper has proposed a new way for fusion of different input cues for tracking of a human 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 realtime and is able to track complex movements like walking or bowing. It even recognizes postures with the arms outstretched directly towards the sensor.

The described way of adding 2d measurements to a 3d matching process is one of the main innovations. The idea of adding artificial point correspondences from non-3d sensors or background knowledge to the 3d matching process can even be exploited further: Future works will investigate methods to include valid ranges for joints via addition of artificial correspondences. Other unsolved issues are the initialization process, or the computation of an optimal scale factor for the model to incorporate the ability to track persons of different height without manually resizing the model.

References

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

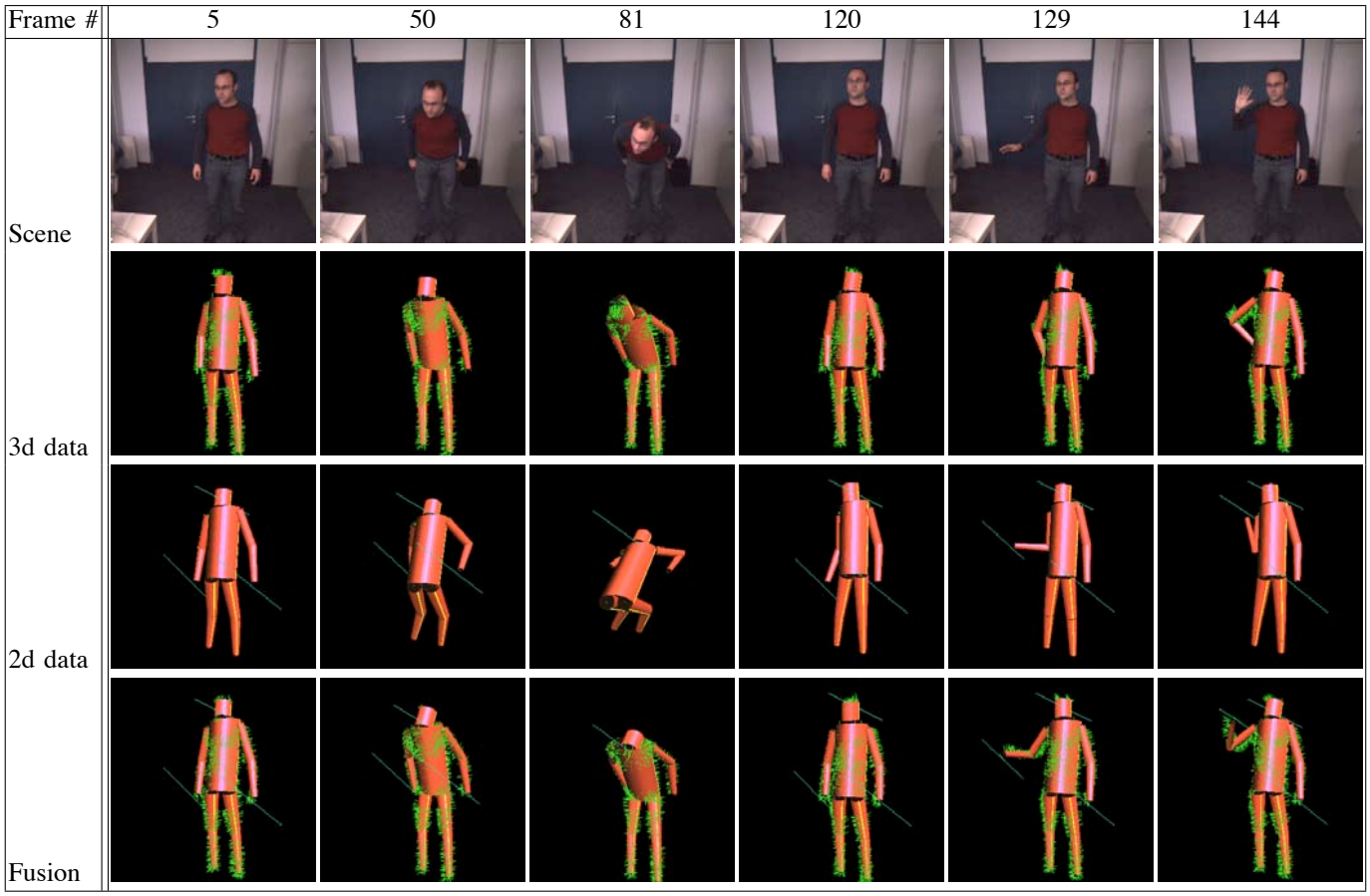

Fig. 4. Experiments with different sensor inputs, taken from a sequence containing a "bow" and a "wave" movement. The frame number is displayed on the top. The used 2d and 3d correspondences have been added to the resulting model images.

[2] S. Calinon and A. Billard, "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, 2005.

[3] J. Fritsch, M. Kleinehagenbrock, S. Lang, T. Plötz, G. A. Fink, and G. Sagerer, "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, 2003.

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

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

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

[7] T. B. Moeslund and E. Granum, "A survey of computer vision-based human motion capture," *Comput. Vis. Image Underst.*, vol. 81, no. 3, pp. 231–268, 2001.

[8] L. Wang, W. Hu, and T. Tan, "Recent developments in human motion analysis," *Pattern Recognition*, vol. 36, no. 3, pp. 585–601, 2003.

[9] J. Fritsch, S. Lang, M. Kleinehagenbrock, G. A. Fink, and G. Sagerer, "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: IEEE, September 2002, pp. 337–343.

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

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

[12] H. Sidenbladh, "Probabilistic tracking and reconstruction of 3d human motion in monocular video sequences," Ph.D. dissertation, KTH, Stockholm, Sweden, 2001.

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

[14] P. Azad, A. Ude, R. Dillmann, and G. Cheng, "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: IEEE Institute of Electrical and Electronics Engineers, 2004.

[15] D. Demirdjian and T. Darrell, "3-d articulated pose tracking to untethered diectic references," in *Multimodel Interfaces*, 2002, pp. 267–272.

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

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

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

[19] Humanoid Animation Working Group, "Information technology — Computer graphics and image processing — Humanoid animation (H-Anim), Annex B," ISO/IEC FCD 19774 - Humanoid Animation," Specification, 2003.

[20] S. Knoop, S. Vacek and R. Dillmann, "Modeling Joint Constraints for an Articulated 3D Human Body Model with Artificial Correspondences in ICP," in *Proceedings of the International Conference on Humanoid Robots (Humanoids 2005)*. Tsukuba, Japan: IEEE-RAS, 2005.