Sensor fusion for model based 3D tracking

Steffen Knoop, Stefan Vacek, Klaus Steinbach, Rüdiger Dillmann
Institute of Computer Science and Engineering (CSE)
University of Karlsruhe
Germany
Email: {knoop,vacek,ksteinba,dillmann}@ira.uka.de

Abstract— **In this paper, we present a new approach for fusion of different measurements and sensors for 3D model based tracking. The underlying model of the tracked body is defined geometrically with generalized cylinders, which can hierarchically be connected by different kinds of joints. This results in an articulated body model with constrained kinematic degrees of freedom. The fusion approach incorporates this model knowledge together with the measurements, and tracks the target body iteratively with an extended *Iterative Closest Point* approach.**

The resulting tracking system named *VooDoo* is used to track humans in a Human-Robot Interaction (HRI) context. We only rely on sensors on board the robot, i.e. a color camera, a 3d time-of-flight camera and a laser range finder. The system runs in realtime ($\sim$ 20Hz) and is able to robustly track a human in the vicinity of the robot. The pose and trajectory of the human interaction partner can then be used for haptic interaction like hand-overs, and for activity and gesture recognition.

I. Introduction

With growing computational capacities and new emerging sensor technologies, tracking of articulated motion has become popular during the last decade. Especially when the field of application does not allow invasive measurement methods like marker based tracking, attachment of magnetic field trackers or active light or sound source devices, it has become evident that fusion of different complementary sensors is the only way to obtain stable and robust tracking results.

Human Motion Capture (HMC) is one important application for model based tracking methods. Especially the field of robotics puts up high demands for HMC systems. Robots that are meant to cooperate closely with humans, and especially with untrained persons which are not familiar with the domain of robotics, need accurate knowledge about the current human pose. 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. 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.

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. It is able to track a person in realtime at about 20-25 Hz in 3d. Results are shown with different input sensors.

Sec. II gives an overview of related work, sec. III and IV describe in detail the tracking target model and the proposed algorithm. The implied sensor model is described in sec. V, and sec. VI gives and evaluates the results.

II. Related work

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 (Iterative Closest Point) 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, 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.

III. Tracking target model

The articulated tracking target model consists of *limbs* or body parts and *joints* which define the model structure through establishing connections between the body parts.

A. 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. Each body part is described with a degenerated cylinder and the corresponding transformation from its parent. The model for a human body e.g. consists of ten body parts (torso, head, two for each arm and two for each leg) which is depicted on the left of fig. 1. However, the model definition is not restricted to humans.

If the fusion algorithm also incorporates data from feature trackers (like some vision based algorithms, or magnetic field trackers that are fixed to the body), it is required to identify certain feature points on the body. For humans, this is done following the *H—Anim Specification* (see [18]).

B. Joint Model

The joint model is based on the concept of introducing elastic bands between two connected model parts. 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).

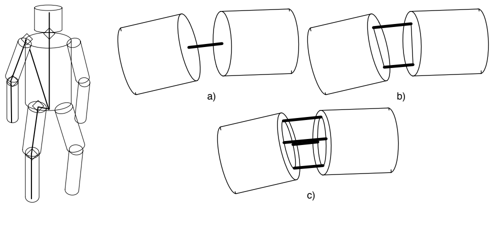

Fig. 1. Different joint type models. Universal Joint 3 DoF a), Hinge Joint 1 DoF + 2 restricted DoF b), Elliptic Joint 3 restricted DoF c)

For each junction of model parts, a set of elastic bands is defined (see fig. 1). 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 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. 1 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. 1 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. 1 c).

For details of the joint model, see also [19].

IV. Sensor Fusion Algorithm for Tracking

The goal of the *VooDoo* tracking system is to track the posture of a modeled body in 3d by matching the internal 3d body model with the current input sensor data. A full description can also be found in [17].

The current posture estimation is given with respect to the hierarchical body model defined in sec. III-A. 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 targets 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. 2. 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. 2):

- *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 with reference to the target* 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 human face or the hands.
- *3d points with reference points on the tracking target* for tracking of special features or points (e.g. with markers,

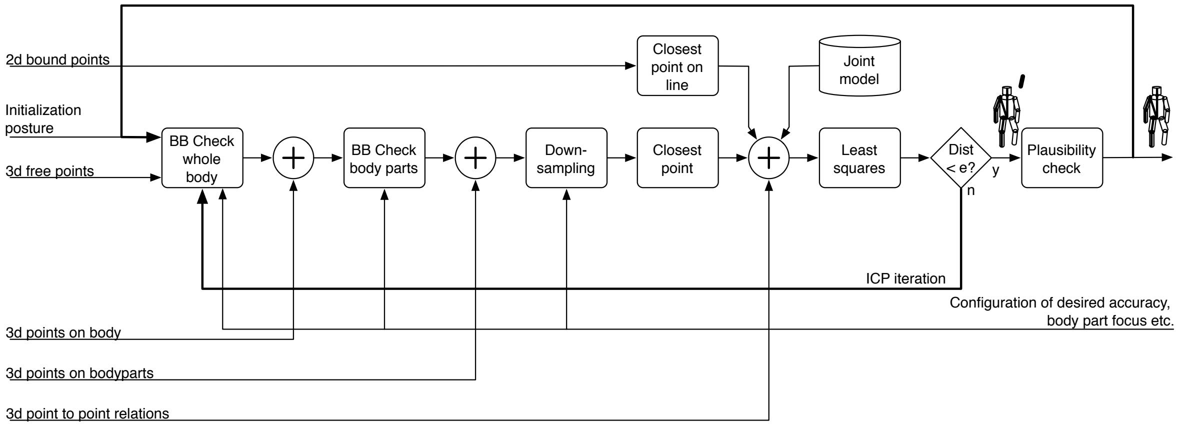

Fig. 2. The complete VooDoo algorithm ("BB" denoting "Bounding Box")

or magnetic field trackers attached to the body) can be integrated.

- *2d features with reference points to the target* 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 reference point on the target.

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 (for a complete ICP overview, please refer to [20] and [21]). 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, along the process depicted in fig. 2.

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. 2, 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 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. 2, 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. 2, 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 are described by only a few measurements.

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. 2, 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. 2, 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. 2, step *Joint model*). According to sec. III-B, 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 (fig. 2, step *Least squares*). The transformation is computed seperately for each limb.

When all transformations have been computed, they can be applied to the model. Steps IV-B.1 to IV-B.7 are repeated until the displacement is below a given threshold or a maximum number of steps have been performed.

V. 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, or to weight 3d points from a ToF-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. 3. 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 color based hand tracker.

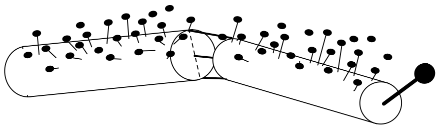

Fig. 3. Different weights for measurements from different sensors, projected into 3D model space. The depicted point sizes correspond to the sensor data weight, the lines indicate the closest-point relations. These pictures motivated the system name *VooDoo*.

VI. Results and evaluation

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 framerate of appr. 20-25 Hz on a Pentium 4 with 3.2 GHz 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. 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.

As already stated in sec. V, the computational effort and thus the framerate depend on the true number of different measurements, independent of the particular weights. To evaluate the computational performance and framerate of the presented method, several analyses have been carried out. The model corresponds again to the human body model depicted in fig. 4.

The computational effort for one frame depends first of all on the number of ICP steps needed. The number of

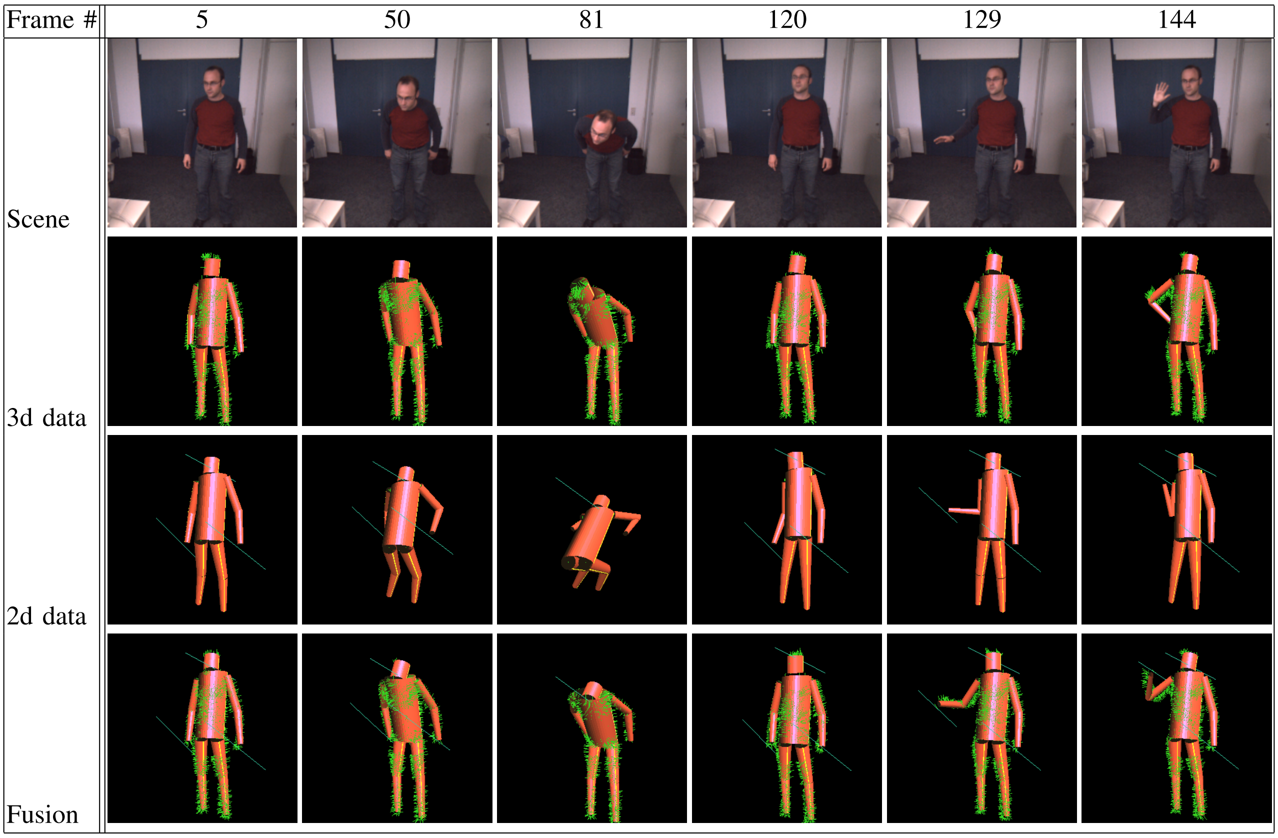

Fig. 4. Experiments with different sensor inputs, taken from a sequence containing a “bow” and a “wave” movement. The frame number is displayed on the top. The used 2d and 3d correspondences have been added to the resulting model images.

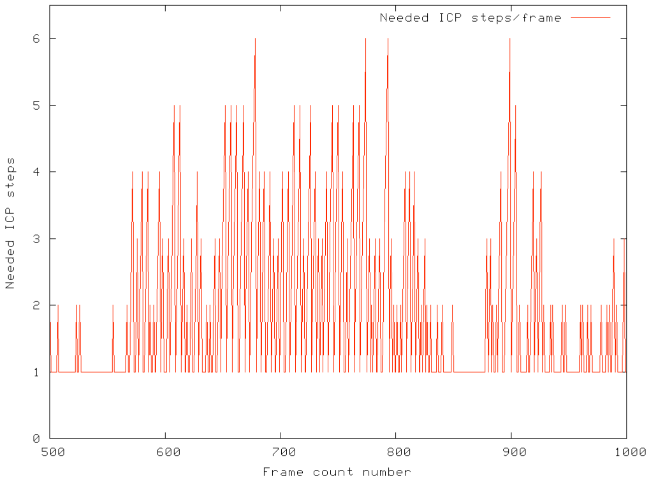

Fig. 5. Number of ICP steps needed for a typical tracking sequence

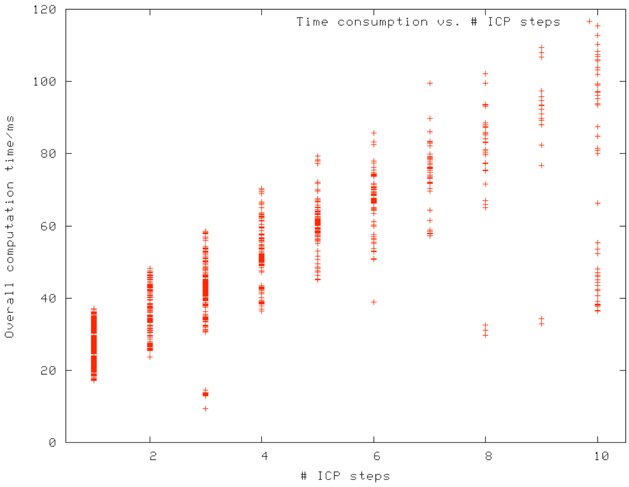

Fig. 6. Time consumption per ICP step vs. number of ICP steps

iterations again depends on the body displacement between two consecutive frames. Fig. 5 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. 6. A maximum number of 6 ICP steps has turned out to be a good tradeoff between time consumption per frame and tracking accuracy. This leads to a frame period of 20 – 70 ms, which corresponds to a framerate of 14.2 to 50 Hz. The maximum framerate in our framework is only constrained by the camera framerate, which is 30 Hz.

The relation between the number of body measurements

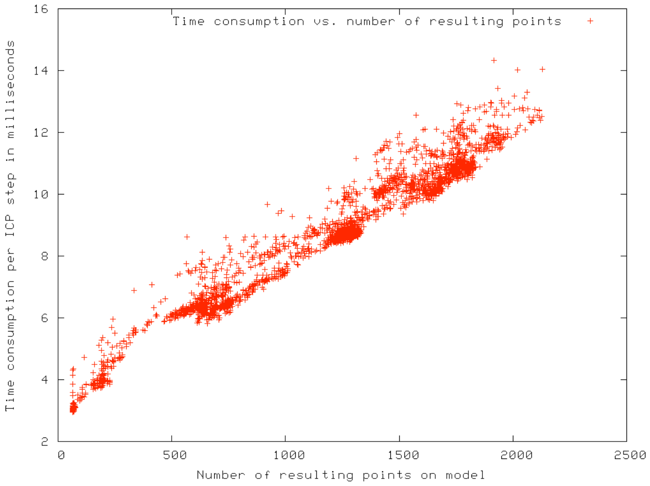

Fig. 7. Time consumption per frame vs. number of body measurements

and the computational effort for one ICP step is depicted in fig. 7. For each measurement of the target, several computations have to be carried out. This leads to the dependency in fig. 7. 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 framerate. To overcome this, one has to find a good tradeoff between accuracy and framerate. This compromize depends on the tracking target characteristics, as well as on the application which utilizes the Human Motion Capture data.

VII. Conclusion

This paper has proposed 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, and which are connected by an *elastic bands* joint model. The proposed approach runs in realtime. It has been demonstrated with a human body model for pose tracking.

The described way of adding 2d measurements to a 3d matching process is one the main innovation.

Future works will investigate methods to include valid ranges for joints via addition of artificial correspondences similar to the joint constraint model.

Acknowledgement

The work described in this paper was conducted within the EU Integrated Project COGNIRON (”The Cognitive Companion”) and funded by the European Commission Division FP6-IST Future and Emerging Technologies under Contract FP6-002020.

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*, Germany.

[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] S. Knoop, S. Vacek, and R. Dillmann, “Sensor fusion for 3d human body tracking with an articulated 3d body model,” in *Proceedings of the 2006 IEEE International Conference on Robotics and Automation (ICRA)*, Orlando, Florida, 2006.

[18] 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. [Online]. Available: http://www.h-anim.org/

[19] 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*. Tsukuba, Japan: IEEE-RAS, 2005.

[20] 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.

[21] 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.