Online Full Body Human Motion Tracking Based on Dense Volumetric 3D Reconstructions from Multi Camera Setups*

Tobias Feldmann[1], Ioannis Mihailidis[2], Sebastian Schulz[1], Dietrich Paulus[2], and Annika Wörner[1]

[1] Group on Human Motion Analysis, Institute for Anthropomatics
Department of Informatics, Karlsruhe Institute of Technology
feldmann@kit.edu, s.schulz@kit.edu, woerner@kit.edu
[2] Working Group Active Vision (AGAS), Institute for Computational Visualistics
Computer Science Faculty, University Koblenz-Landau
janni@uni-koblenz.de, paulus@uni-koblenz.de

Abstract. We present an approach for video based human motion capture using a static multi camera setup. The image data of calibrated video cameras is used to generate dense volumetric reconstructions of a person within the capture volume. The 3d reconstructions are then used to fit a 3d cone model into the data utilizing the Iterative Closest Point (ICP) algorithm. We can show that it is beneficial to use multi camera data instead of a single time of flight camera to gain more robust results in the overall tracking approach.

1 Introduction

The estimation of human poses and the pose tracking over time is a fundamental step in the process chain of human motion analysis, understanding and in human computer interaction (HCI). In conjunction with artificial intelligence (AI) pose estimation and tracking are the sine quibus non of motion recognition which in turn is a fundamental step in scene and motion analysis with the goal of automatic motion generation on humanoid robots.

A lot of research has taken place, to automatically exploit information for pose estimation and tracking in video data. The video based techniques can here be separated into two major approaches. First: Approaches based on markers placed on the observed subjects. Second: Marker-free approaches based on image features, on geometrical or on statistical information. The latter approaches usually replace the knowledge about markers with knowledge about a human body model. In HCI tasks it is usually desirable to avoid markers to allow a less intrusive observation of persons.

* This work was supported by a grant from the *Ministry of Science, Research and the Arts of Baden-Württemberg*.

R. Dillmann et al. (Eds.): KI 2010, LNAI 6359, pp. 74–81, 2010.

Marker-less image based human pose tracking is usually started by the extraction of features like points, edges [1], contours, colors [2], segments [1], boundaries, 2.5d [3] or 3d points [2,4], etc. from one or more given input images. In an additional step a correspondence problems has to be solved to assign the found features to an a priori given kinematic human body model. To obtain the pose of the person within the images, the body pose problem has to be solved, which can be done by using e.g. probabilistic approaches like particle filters [1], geometric approaches like Iterative Closest Point (ICP) in the Voodoo framework [3], iterations of linear approximations [5] or learning approaches, e.g. based on silhouettes [6]. An extensive survey of current approaches can be found in [7].

A key point is the definition of the distance function to assign correspondences. Often, color segments of a priori known color distributions are (skin color, foreground colors, etc.) to determine regions of interest. The color segments can be used to directly calculate 3d information via triangulation, e.g. by the use of stereo information [2] or in form of fore-/background estimation for a subsequent dense 3d reconstruction by utilizing a multi camera setup and the shape from silhouette approach [4].

2 Contribution

We present a purely image based marker-less online capable tracking system based on a human cone model with 25 degrees of freedom (DOF). We focus on the shape from silhouette approach for a volumetric 3d reconstruction and combine it with ICP based tracking. We demonstrate the applicability of the overall approach for online human body tracking in static multi camera scenarios.

The approach works as presented in Fig. 1. Given the input images of n calibrated cameras, the foreground has to be segmented by background subtraction. We use a more complex color model to enhance the segmentation results and, hence, shift the calculations to the graphics processing unit (GPU) to gain online capable performance. The segmentation results are used as input for a dense binary voxel reconstruction. The voxel carving can be parallelized very easily by dividing the volume into subvolumes. Thus, we use multithreading to increase the speed of the reconstruction. In case of m cores we use $m-1$ threads for the dense reconstruction. The results of the reconstruction are then used as input for an ICP based tracking framework, which runs in a separate process on the last unused core of the CPU.

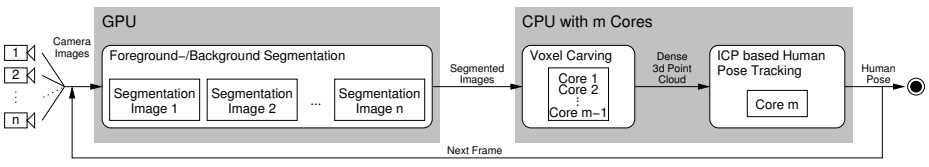


Fig. 1. The integrated pipeline of voxel based human pose estimation and tracking

3 Fore-/Background Segmentation

One key aspect in volumetric 3d reconstruction is the determination of the foreground in the camera images to utilize this information for dense volumetric reconstruction in a later step. A simple approach to distinguish between fore- and background is simple differencing between the camera images and a priori recorded images of the background to extract the silhouettes of the image foreground. The recording scenario is often influenced by changing light conditions e.g. if the recording room is equipped with windows. In realistic scenarios it is hence reasonable to use a color space which splits luminance and chroma. An often used color space which implicitly splits luminance and chroma is HSV. However, color spaces like HSV suffer from singularities near for example $V = 0$, which leads to large errors introduced by only small amounts of image noise in dark areas. Unfortunately, in realistic videos of indoor scenes a lot of dark areas exist below objects which occlude the illumination spread of lamps and windows. Hence, due to the robustness in realistic scenarios the computational color model of [8] is used. The idea is to separate chroma and luminance directly in RGB without previous color space transformations.

Given two colors $\boldsymbol{c_1}$ and $\boldsymbol{c_2}$ in RGB color space, two distances have to be calculated. First: The distance of brightness α. Second: The distance of chroma d (c.f. Fig. 2). The distance of brightness is defined as a scalar value α that describes the factor of brightness, the color $\boldsymbol{c_1}$ have to be scaled with to be as bright as the color $\boldsymbol{c_2}$. Geometrically, α defines the root point $\boldsymbol{r}$ of the perpendicular of the vector $\boldsymbol{c_1}$ through the point $\boldsymbol{c_2}$. The value of α can be calculated with the dot product as shown in eq. 1. If $\alpha < 1$, the color $\boldsymbol{c_2}$ is darker than the color $\boldsymbol{c_1}$, if $\alpha > 1$, the color $\boldsymbol{c_2}$ is brighter than the color $\boldsymbol{c_1}$.

$$\alpha = \frac{\|\boldsymbol{c_1}\|}{\|\boldsymbol{r}\|} \quad \text{where} \quad \boldsymbol{r} = \boldsymbol{c_1} + \langle(\boldsymbol{c_2} - \boldsymbol{c_1}), \boldsymbol{n}\rangle \boldsymbol{n} \quad \text{with} \quad \boldsymbol{n} = \frac{\boldsymbol{c_1}}{\|\boldsymbol{c_1}\|} \tag{1}$$

Using the normed direction vector $\boldsymbol{n}$ of $\boldsymbol{c_1}$ from eq. 1 the color distance d can be calculated as described in eq. 2.

$$d = \|(\boldsymbol{c_2} - \boldsymbol{c_1}) \times \boldsymbol{n}\| \tag{2}$$

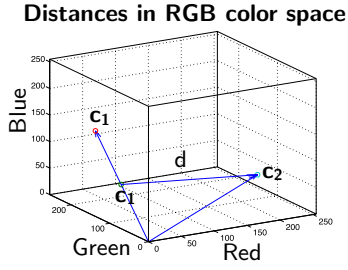


Fig. 2. Distance measure in RGB color space. The distance of two colors $\boldsymbol{c_1}$ and $\boldsymbol{c_2}$ is decomposed into the luminance distance α and the chroma distance d.

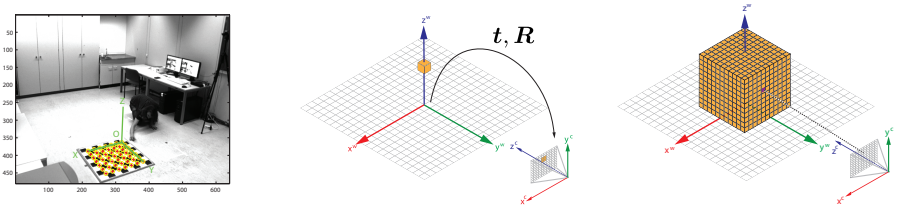
Due to online constraints and in contrast to [8] we are using static thresholds τ_α and τ_d for the discrimination of fore- and background and use the algorithm in parallel on different camera views all at the same time in a multi camera setup. The values are estimated empirically during runtime by using sliders. It is most important to adjust the τ_α correctly to be able to cope with light changes and shadows. Due to the computational complexity of the calculations and the aim of online applications we exploited the computational power of the GPU of the used computer system. By shifting the calculations to the GPU, we were on the one hand able to speed up the segmentation process. On the other hand, in this way we were able to keep the CPU free from segmentation calculations and, thus, use the CPU primarily for additional tasks like the dense volumetric 3d reconstruction and the model based pose tracking of the observed person.

4 Dense Voxel Based 3d Reconstruction

Based on the calibration data of the used calibrated multi camera setup and the silhouettes from the previous section 3, a dense volumetric 3d reconstruction is performed using a voxel carving approach derived from [4].

The camera setup is static and the intrinsic parameters (calibration matrix $\boldsymbol{K}$ and distortion coefficients kc_i with $i \in 1 \ldots 5$) and extrinsic parameters (rotation matrix $\boldsymbol{R}$ and translation vector $\boldsymbol{t}$) are determined using Bouguets Matlab Calibration Toolkit [9], (c.f. Fig. 3(a)) based on a checker board calibration target which defines the world coordinate system (WCS) and stay the same over the whole sequence. For each camera, the position in the WCS is determined in 6d based on rotations $\boldsymbol{R}$ and translations $\boldsymbol{t}$ (c.f. Fig. 3(b)). Therefore, the transformation of a 3d world point $\boldsymbol{p}^{\mathrm{w}}$ into the camera coordinate system (CCS) can be calculated by $\boldsymbol{p}^{\mathrm{c}} = \boldsymbol{R}\boldsymbol{p}^{\mathrm{w}} + \boldsymbol{t}$. The resulting 3d point $\boldsymbol{p}^{\mathrm{c}}$ has to be projected into 2d, distorted according to the distortion coefficients resulting in $\boldsymbol{p}^{\boldsymbol{d}}$ and finally linearly transformed by $\boldsymbol{K}$ with $\boldsymbol{p}^{p} = \boldsymbol{K}\boldsymbol{p}^{d}$ (c.f. [9]).

This projection of a single 3d point can be extended to a projection of a 3d volume grid as depicted in Fig. 3(c). The space of interest is divided into a uniform 3d grid of voxels. Due to the online constraints of our approach, we consider only the center of each voxel and project it into each camera image.



(a) Camera calibration (b) WCS to CCS (c) Voxel projection

Fig. 3. (a) Camera position based on WCS defined by calibration target. (b) Transformation of the orange voxel from WCS to CCS. (c) Center of each voxel of defined voxel space is projected into specific pixel coordinates of each camera.

The projection information is static as the camera setup is static and, thus, can be pre-calculated for each voxel and for each camera before the reconstruction process starts, which helps to speed up the projection process.

The idea of voxel carving is, to define two states of a voxel: The voxel is turned on or off. Initally, all voxels are off. By utilizing the silhouettes from section 3 an iteration over all voxels is started whereas for the voxel's projection to each camera a check is performed, whether the projection falls within the silhouette of the observed person in the camera image. If this is the case for all cameras, the voxel is turned on, otherwise the voxel remains off. After the iteration over all voxels, a discretized 3d reconstruction of the observed foreground has been created. This reconstruction is then repeated over a whole sequence of frames which contain synchronized video images of each camera.

5 Online Full Body Human Motion Tracking

Knoop et al. [3] created the public available[1] cone based *Voodoo* framework for human motion tracking in the *Cogniron* project[2]. The framework is able to track human poses online utilizing ICP and based on a human body model. Knoop et al. used the framework mainly with time of flight cameras and sensor fusion approaches by integrating 2.5d and 2d image data. We present our integration of data derived from a n-view camera setup into the Voodoo tracking framework to enable online tracking on pure video data. In contrast to 2.5d data taken from a single time of flight camera, our approach generates full 3d point clouds from different views and, hence, has the potential to achieve improved tracking results by solving ambiguities which result from occlusions.

One benefit of the Voodoo framework is it's flexible sensor integration, i.e. the possibility to add an arbitrary number of sensors as data sources. We integrated two additional sensors which use our own point cloud format to enable the data exchange between the 3d reconstruction of section 4 and Voodoo. The first sensor imports recorded sequences and enables Voodoo to replay these sequences for later analysis. The second sensor imports a continuous stream of point clouds from a connected reconstruction instance directly into Voodoo and in this way enables online tracking. The source code of the point cloud interface is available at: `http://hu-man.ira.uka.de/public/sources/libPtc.tar.bz2`.

6 Evaluation

For the evaluation we used a standard PC (Intel(R) Core(TM)2 Quad CPU Q6600, 2.40GHz and nVidia GeForce 8600 GTS) and a set of 3 to 8 Prosilica GE680C cameras with VGA resolution in a circular setup except as noted otherwise. In case of online human motion tracking, we connected 3 cameras to the PC via Gigabit Ethernet and used the image data directly for segmentation,

[1] `http://voodootracking.sourceforge.net/`
[2] `http://www.cogniron.org`

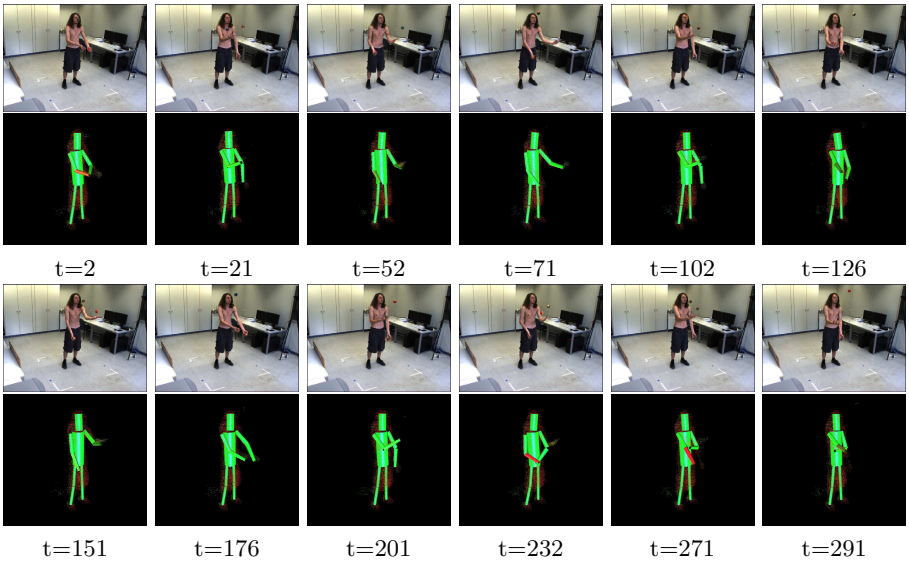


Fig. 4. Images, reconstructions and tracking results of the juggle sequence

3d reconstruction and motion tracking. In case of offline analysis, we recorded sequences with 8 cameras, created 3d point clouds in a second step and used the point cloud reader library to import the data into Voodoo for offline analysis. In the following we will present the results of the segmentation, the benefit of the GPU based calculations and the online and offline motion tracking results.

6.1 CPU versus GPU Based Segmentation

The performance increase of the GPU enabled version of the fore-/background segmentation has been measured on an Intel(R) Xeon(R) CPU X5450, 3.0GHz with a nVidia GeForce GTX 295.

The average processing time of an image of 640×480 pixels on the CPU in a non-optimized version using the linear algebra of the OpenCV library was 116.9ms, on the GPU it was 0.364ms. The time for images of the dimensions 1280×720 was 352.3ms on the CPU and 0.969ms on the GPU. The increase of speed is, hence around a factor of 320 to 360 depending on the image dimensions.

6.2 Full Body Motion Tracking: 8 Cameras (offline)

The first exemplary sequence contains a person juggling with balls in the laboratory (c.f. Fig. 4). The sequence has been recorded with 8 cameras with 50 frames per second. The voxel space dimensions had been $100 \times 100 \times 100$ with a resolution of 20mm per voxel. Despite the fast movements of the arms Voodoo is able to track the motions successful over 200 frames. In frame 232 an interchange of the left arm happens. The positions of the head, torso and legs has been tracked successfully over the whole sequence. The tracking of fast movements is, hence,

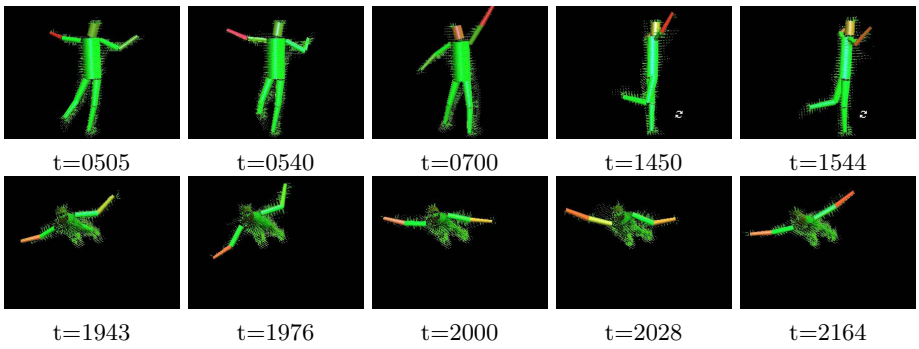


Fig. 5. Reconstructions and tracking results of the online tracking from different views. The average tracking speed is around 13 frames per second on a Intel Quadcore 2,4GHz and a nVidia 8600GTS.

no problem with our approach but the interconnection of extremities with other body parts can result in the loss of the correct pose of the affected body parts.

6.3 Full Body Motion Tracking: 3 Cameras (Online)

The second exemplary sequence contains the results of an online tracking with the presented approach. The capture volume has been divided into 60^3 voxels with an edge length of 40mm per voxel. Using this configuration and three cameras with 25fps, the online tracking achieves an average speed of around 13 frames per second. The tracking results have been simultaneously captured via screen capture and a selection of the results is presented in Fig. 5. The first three frames of Fig. 5 show the tracking from behind. The point of view is then rotated to observe the person from the right in the next to frames 1450 and 1544. These two frames also demonstrate the advantages of a n-view capture method over a time of flight camera from the front as the bended knees (right knee in frame 1450, left knee in frame 1544) could be recognized and tracked correctly. The following frames 1943-2164 show the scene from above whereas it should be noted, that this is a completely artificial camera position as we did not place a real camera at the ceiling to observe the volume from above. The images show, that the arms could be tracked in all three dimension due to the 3d point cloud. Even though some tracking problems appear during rotation, i.e. the torso does not adapt to the movements of the arms in the shoulder areas. However, this can not be fixed by the data modality but has to be enhanced in the tracking framework. The full video sequence can be found at: `http://hu-man.ira.uka.de/public/videos/Tracking.mp4`.

7 Conclusion

We presented a marker-less system for online full body human motion tracking based on dense volumetric 3d reconstruction by utilizing the information of an

fully calibrated static multi camera system and fore-/background segmentation with a computational color model in RGB and a body model with 25 DOF. We found the tracking to benefit significantly from the integration of multi view information as presented by two exemplary sequences. Hence, if the scenario allows the integration of multiple views, this information source should be exploited. Although, we noticed difficulties in the tracking process during rotations which were based on not recognized and not adapted rotations of the torso element of the tracking framework. In further work, we will focus on solutions to cope with these kind of problems. Additionally, we will perform a quantitative error analysis based on offline recordings, which is not the focus of the current paper.

Overall, the presented results constitute a first fundamental step in a substantial 3d online motion tracking for succeeding motion analysis and understanding, which form the foundation of subsequent AI methods for online scene analysis regarding humans and motion generation for e.g. humanoid robots.

Acknowledgements

We would like to thank the *Humanoids and Intelligence Systems Laboratories* (HIS) headed by Prof. Dr.-Ing. R. Dillmann, and especially Martin Lösch, for their support regarding the Voodoo tracking framework.

References

1. Deutscher, J., Blake, A., Reid, I.: Articulated body motion capture by annealed particle filtering. In: CVPR 2000, vol. 2, pp. 2126–2133 (2000)
2. Azad, P., Ude, A., Asfour, T., Dillmann, R.: Stereo-based markerless human motion capture for humanoid robot systems. In: ICRA 2007, pp. 3951–3956. IEEE, Los Alamitos (2007)
3. Knoop, S., Vacek, S., Dillmann, R.: Sensor fusion for 3d human body tracking with an articulated 3d body model. In: ICRA 2006, pp. 1686–1691. IEEE, Los Alamitos (2006)
4. Cheung, G.K., Kanade, T., Bouguet, J.Y., Holler, M.: A real time system for robust 3d voxel reconstruction of human motions. In: CVPR 2000, vol. 2, pp. 714–720 (2000)
5. Rosenhahn, B., Kersting, U.G., Smith, A.W., Gurney, J., Brox, T., Klette, R.: A system for marker-less human motion estimation. In: Kropatsch, W.G., Sablatnig, R., Hanbury, A. (eds.) DAGM 2005. LNCS, vol. 3663, pp. 230–237. Springer, Heidelberg (2005)
6. Hofmann, M., Gavrila, D.M.: Single-frame 3d human pose recovery from multiple views. In: Denzler, J., Notni, G., Süße, H. (eds.) DAGM 2010. LNCS, vol. 5748, pp. 71–80. Springer, Heidelberg (2009)
7. Moeslund, T.B., Granum, E.: A survey of computer vision-based human motion capture. Computer Vision and Image Understanding 81, 231–268 (2001)
8. Horprasert, T., Harwood, D., Davis, L.S.: A statistical approach for real-time robust background subtraction and shadow detection. In: ICCV 1999 Frame-Rate Workshop, Kerkyra, Greece (September 1999)
9. Bouguet, J.Y.: Camera calibration toolbox for matlab (2010), `http://www.vision.caltech.edu/bouguetj/calib_doc/`