

A Model-based Algorithm to Estimate Body Poses using Stereo Vision

Quirin Mühlbauer, Kolja Kühnlenz and Martin Buss

Abstract—Estimating the human body pose is of great interest for many tasks, such as human robot interaction, people tracking and surveillance. During the recent years, several approaches have been presented, which still have weaknesses regarding occlusions or complex scenes. In this paper, we present a novel algorithm for human body pose estimation using any three-dimensional representation of the environment, like stereo vision. The presented algorithm is able to leave out body parts and is therefore able to deal with occluded body parts. In a first step, possible humans need to be detected, e.g. by using a skin color filter. A disparity map containing depth information is computed using a stereo matching algorithm. It leads to a three-dimensional representation of the scene. Starting with the detected skin parts, our algorithm segments this point cloud into smaller clusters. The possible matches are then verified, and the body pose is estimated using a kinematic human model with 28 degrees of freedom. As our algorithm is capable of dealing with arbitrary three-dimensional representations, it can easily be adapted to use a three-dimensional laser range finder instead of a stereo camera system.

I. INTRODUCTION

A robot working in close collaboration with humans must be able to interact safely. To increase its acceptance, the robot should have the ability to recognize the gestures, actions and the intention of its companions, so it is essential to estimate the human body pose in real time. Main area of application of the algorithm developed in this paper is the interaction between a human and the *Autonomous City Explorer* (ACE)[1], developed at our institute. The ACE project has the goal, to develop a robot which is capable to find its way to a certain target without any prior knowledge. Inspired by the behavior of humans in unknown environments, ACE must find its way by asking pedestrians. When asked for the way, a pedestrian will point in the direction the robot has to go to. In order to estimate this direction, the body pose of the pedestrian has to be known. Another application is the estimation of the pedestrian's moods (perhaps he is frightened of talking to a mobile robot) and to react in an adequate way, or the tracking of pedestrians to plan future movements and to prevent collisions.

Many approaches use markers to estimate the body pose. Obviously, in our application it is not suitable to equip every pedestrian with such markers. Hence, a markerless approach has to be found. A skin detector filter is used to find all skin parts in a picture. In the next step, a three-dimensional representation of the scene is created by using a stereo

camera and computing a disparity map. This yields to a colored point cloud, which can be segmented. For every skin part found in the step before, one segment is created. Only those segments, where the detected skin part belongs to the head, will be used for further processing. Starting with the head, a human model with 28 degrees of freedom is fitted into the segments. To allow a wide variety of body poses, different typical postures of the model will be used as initial postures for the fitting. During the fitting process, the pose of the model will be adjusted iteratively to achieve a good estimation. For each typical model pose, an error metric will be computed. The poses will then be verified and the one with the smallest error metric will be chosen. The novelty of this approach is the independency of the sensor, as it is based on a three-dimensional representation of the environment. This representation can be obtained by numerous sensors, like laser scanners or stereo vision systems.

The remainder of this paper is organized as follows: An outline of the state of the art will be given in Section 2. The human model and the algorithms used for the body pose estimation will be presented in Section 3, while Section 4 shows some experimental results. A conclusion and an outlook to future works will be given in Section 5.

II. STATE OF THE ART

Developed for realistic animation of the human body in Hollywood movies, the first applications for the reconstruction of human body poses have been motion capture methods. An actor was equipped with markers, so his movements could easily be recorded. On the other hand, markers are uneligible for outdoor scenarios. A mobile robot communicating with pedestrians must use available sensors, such as cameras or laser range finders. The field of camera-based body pose estimation can be divided into various approaches which are summarized in the following.

Some use monocular vision systems, such as [2], where an image of the body with several landmarks serves as input, or shape descriptors are extracted from image silhouettes [3] [4] [5]. Probabilistic models are used in [6]. To increase the robustness, [7] uses semi-supervised learning. In [8] prior knowledge about the human movements is necessary, which can be stored in a motion library [9]. [10] maps typical postures into two-dimensional images.

Other approaches, like the one presented in this paper, focus on multi-view systems, such as stereo vision, which provide additional depth information. The fitting of the model can be described as an optimization problem [11] and can easily be combined with tracking people [12] [13]. [14] uses a learning algorithm, where training data is recursively classified into

Q. Mühlbauer, Kolja Kühnlenz and M. Buss are with Institute of Automatic Control Engineering, Technische Universität München, D-80290 Munich, Germany (qm@tum.de, kolja.kuehnlenz@ieee.org, m.buss@ieee.org)

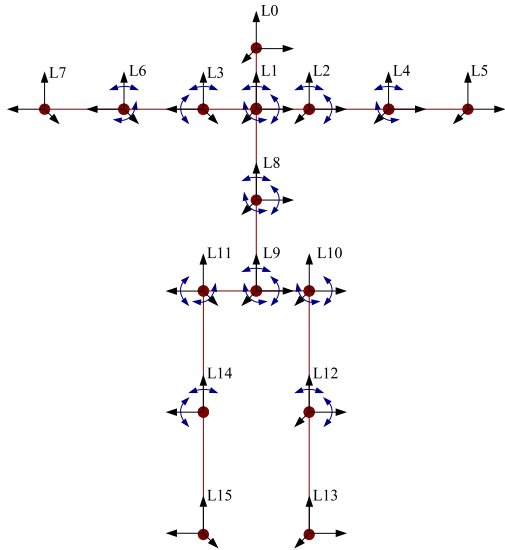


Fig. 1. Reduced human model with 15 links, the corresponding coordinate systems and the degrees of freedom.

several clusters with silhouette and depth images.

Another approach is the use of an image stream, where features can be tracked between the images [15]. Through the two-dimensional tracking of the features, their three-dimensional positions can be computed [16]. [17] uses twists and exponential maps to recover high degree-of-freedom articulated configurations of the human body. Of course, the image stream can consist of monocular or stereo images. [18] uses an image stream with multiple views to fit a human model into the recovered scene. Three-dimensional voxel data created from multiple views is used in [19].

III. HUMAN BODY POSE ESTIMATION

This section shows the algorithms used for the body pose estimation, starting with the human body model and its poses. In the next section, the algorithm used to segment the three-dimensional representation is presented. A description of the algorithm used to detection the skin color segments can be found in [20].

A. Human Model

Figure 1 shows a schematic view of the human model with 15 links and 28 degrees of freedom. Each link provides one to three degrees of freedom, and is rotated around the axis of the coordinate system of the link it is connected to. Table I shows the links, the number of degrees of freedom, the hierarchical structure and the length of the link.

As the hands and the feet are too small to be detected robustly by the stereo matching and they are not relevant to estimate the pointing direction, they have not been taken into account. Hence, this model is reduced by 10 degrees of freedom. To increase the accuracy of the estimated body pose, 27 typical poses have been considered. As we are especially interested in the configuration of the arms, the poses differ mostly in the arms. To be able to recognize the pose from every point

TABLE I
LINKS OF THE HUMAN MODEL

Link	Name	Connected to Link	DOF	Length (in m)
0	Start (Head)	-	0	-
1	Neck	0	0	0.25
2	Shoulder Left	1	3	0.25
3	Shoulder Right	2	0	0.25
4	Upper Arm Left	2	2	0.375
5	Lower Arm Left	4	2	0.375
6	Upper Arm Right	2	2	0.375
7	Lower Arm Right	6	2	0.375
8	Upper Back	1	3	0.5
9	Lower Back	8	3	0.5
10	Hip Left	9	3	0.25
11	Hip Right	9	0	0.25
12	Upper Leg Left	10	2	0.5
13	Lower Leg Left	12	2	0.5
14	Upper Leg Right	11	2	0.5
15	Lower Leg Right	14	2	0.5

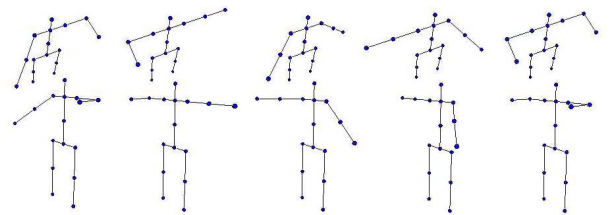


Fig. 2. Five typical body poses of the reduced human model from different points of view.

of view, the whole body is rotated in steps of 33.3° . Figure 2 shows some of the different poses.

To validate the human model, the link lengths and the angles between the links are considered. For each link, a minimal and a maximal value for each of the parameters have been estimated. As they are coupled, the left and right shoulder are treated specially. The left shoulder has three degrees of freedom and whenever it is moved, the right shoulder is limited in its movement and will show a similar movement. Hence, Table I shows no degrees of freedom for the right shoulder.

B. Segmentation of vision data

The three-dimensional representation of the scene acquired by the stereo vision system contains a large colored point



Fig. 3. Image with detected skin parts (left). This image has been taken with a stereo camera with a high aperture and has already been rectified. The right side shows the three-dimensional reconstruction of the scene using a stereo matching algorithm.

cloud. With the skin detector, we know what parts of the point cloud may belong to a human body. Consequently the remaining parts of the human body have to be found. Starting with the detected skin parts, the segmentation algorithm searches for locally adjacent structures and will create one cluster for each detected skin part. As the skin detector may deliver false positives and two or more skin parts of the same human are detected, the clusters have to be validated. The left part of Figure 3 shows the detected skin parts in a scene, while the right side shows the three-dimensional reconstruction of the same scene. A stereo matching algorithm running on NVIDIA’s CUDA has been used to compute the disparity map.

1) *Method for segmentation:* Algorithm 1 shows the algorithm used for the segmentation. Starting with a given point p , all neighbors n are considered. A neighbor n_i is included in the cluster, if the distance $d = \|p - n_i\|_2$ is below a certain threshold. In the next step, all neighbors of the next point of the cluster are considered. This function is repeated, until no valid neighbors are found. As the main goal is the detection of humans, the algorithm adds only those points to the cluster, which are included in a cylinder describing the area, which can be reached by a human. As they offer very competitive lookup times for radially bounded queries, kd-trees¹ have been chosen as spatial data structure.

Algorithm 1 Segmentation

```

 $p$  = detected skin part
 $i = 0$ 
repeat
  Detect neighbors
  for  $j = 1$  to number of neighbors do
     $d_j = \|p - n_j\|$ 
    if  $d_j \leq d_{max}$  then
      add  $n_j$  to cluster
    end if
  end for
  increase  $i$ 
   $p$  = next point in cluster
until  $i < i_{max}$ 

```

2) *Validation of the Segment:* After the clusters have been created, they have to be validated. To discard invalid clusters, the following filters are applied:

- The number of points in the segment is taken into account. If the number of points is too small, the cluster is most likely caused by a false positive.
- Again, if too many points have been found, the skin detector has found something else than a human, for example a wall.
- The fitting of the model into the cluster is starting with the head, so only the clusters are valid, where the detected skin part equals the head. To check this, the centroid of the cluster is computed and compared to the starting point.

¹<http://www.cs.umd.edu/~mount/ANN/>

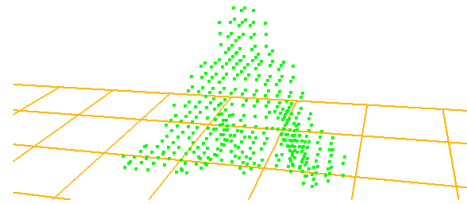


Fig. 4. Detected segments in the scene.

- Large clusters with a low density of points are most likely false positives, so they will be rejected.

After these filters, almost all invalid clusters will be rejected. The remaining invalid guesses will be removed in the next step, when no valid body pose can be found. Figure 4 shows the detected segments of the scene.

C. Extracting the body pose

After one cluster has been computed for each human in a scene, they can be used to extract the body poses. The algorithm will be executed once for each cluster, so the accurate number of humans can be found. For each of the possible humans, the algorithm is executed to fit all 27 typical body poses and computes an error metric for each pose. The pose with the lowest error will be selected as winner. As the algorithm should be able to deal with all different types of colors and clothes, color provides little useful information and is consequently not used. The segmentation and estimation of the body pose is performed by using only the position of the points. First the method will be described, followed by the error metric and the validation of the estimated body poses.

1) *Method for fitting a body pose:* Starting with the head, the algorithm will try to fit the attached links iteratively. The order of the links is the same as described in Table I. If a link is not part of the image or is occluded by an object in front of it, no valid fit can be made and the link and all links attached to it will not be included in the resulting human model.

The links are fitted based on the following method: start point of the link will be the end point of the previous link. Based on the model, the algorithm knows, where the end of the link should be placed in an ideal case, the reference point. In a real case, the end of the link will be placed somewhere near this reference point. The algorithm will search the points n_p around the reference point for possible ends of the link. An end point can only be valid, if the restrictions of the link (e.g. link length or the angles between the previous links) are not broken. Furthermore, the point density of the cluster along the link must not fall below a certain limit. After the link has been fitted, the error metric will be updated and the next link can be computed. This will be repeated iteratively until all links have been fitted.

Algorithm 2 shows a description of the used algorithm which is performed for every pose n . The actual link is denoted as i , the error metric as e_i , the vector holding all error metrics as

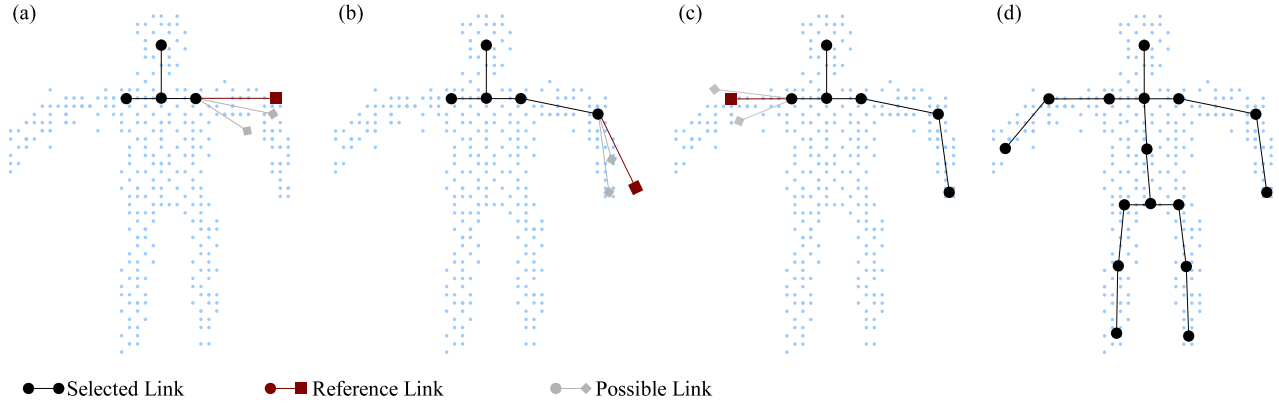


Fig. 5. Illustration of the link fitting algorithm. The red links illustrate the reference links, the gray ones the possible links detected by the find best fit algorithm.

Algorithm 2 Fitting a human body pose

Main Algorithm:

```

nr = Get Endpoints for Pose n
el = 0
for i = 1 to number of links do
  if ns[i - 1] != invalid then
    pr = ns[i - 1] + nr[i]
    ps = find best fit(pr)
    if ps is valid then
      ns[i] = ps
    else
      ns[i] = invalid
    end if
    ei = compute error metric(pr)
    el = el + ei
  end if
end fore[n] = el + compute error metric (ns)

```

```

ps = findBestFit(pr):
np = points near(pr)
for i = 1 to number of points in np do
  pt = np[i]
  e1 = compute point density near link
  e2 = compute link restrictions
  et[i] = e1 + e2
end for
m = compute best et
return np[m]

```

e, respectively. The reference end point for a link is denoted as p_r and the real end point as p_s . The list of the reference end points for the links is denoted as \mathbf{n}_r , the list with real end points as \mathbf{n}_s . The algorithm for the computation of the best fit of a link's end point can be found in the lower part of Algorithm 2. \mathbf{n}_p denotes the list of points near the reference point p_r , the resulting temporary error metrics are stored in \mathbf{e}_t . The computation of the error metric can be found in

Section III-C.2.

Figure 5 illustrates the link fitting algorithm after the left and right shoulder already have been fitted. Starting with the upper left arm (a), the lower left arm (b) and the upper right arm (c) are fitted. Figure 5 (d) shows the completed fitted body pose. In (b) the end point of the reference link would be placed outside the segment. The algorithm tries to find possible end points, which are placed inside the segment, and selects the best one.

2) *Error metric*: To select one of the 27 poses that have been fitted, an error metric has to be computed for every pose. This error metric tries to identify the best fitting pose. The error metric for a pose n is shown in equation 1

$$\mathbf{e}[n] = \alpha_1 \cdot e_l + \alpha_2 \cdot \sum_{i=1}^N a_i + \alpha_3 \cdot \left(1 - \frac{N_u}{N_c}\right) + \alpha_4 \cdot \sum_{i=1}^N s_i \quad (1)$$

and considers the following parameters, each weighted by a parameter α :

- *The number of links*. For each missing link i , a penalty a_i will be added. Links that have other links attached (like the shoulder) will lead to a higher penalty than links with no other attached links (like the lower arm). A successfully fitted link will have the penalty $a_i = 0$.

- *The distance between the reference end points and the detected end points of each link* is also considered. High distances may lead to distorted body poses. This error is stored in e_l .

- *The number points of the cluster that are not nearby the pose*. When a pose does not use all points of a cluster, it is almost certain that one or more links couldn't be fitted correctly. The number of points used is denoted as N_u , the number of points in a cluster as N_c .

- *Density of points near a link*. If the density becomes locally low, it may be an invalid link. Every time the density near link i falls below a threshold, a penalty is added to s_i .

3) *Validation of the body pose*: After all links have been computed for a body pose, it has to be validated to avoid invalid configurations. Again, the lengths of the links are analyzed. Contrary to the validation of a single link, all link

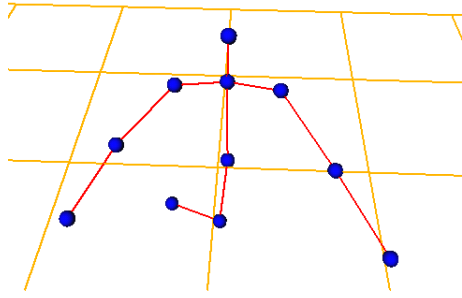


Fig. 6. Computed body pose

lengths are analyzed simultaneously. Equation 2 computes the median s of the scales of the estimated link lengths l^c compared to the reference link lengths l^r . N denotes the number of links used.

$$s = \frac{1}{N} \sum_{i=1}^N \frac{l_i^c}{l_i^r} \quad (2)$$

After the median scale has been created, it is compared to the scales of the single links. If the difference is larger than twice the standard deviation, the difference between the link lengths are considered too large and the pose is invalid. Furthermore, the rotations between the links are considered. The human body is subject to certain restrictions of the movement of the links. Many configurations are impossible or futile. To avoid these configurations, a minimal and a maximal angle have been considered for every degree of freedom. All necessary angles will be computed and if one exceeds the defined interval, the whole body pose will be considered as invalid.

D. Complexity

As described above, our algorithm will be used on a mobile platform, so the computational cost should be as low as possible. The resolution of the colored point cloud as well as the number of detected skin parts will have an influence on the computation time. The detection of the skin parts scales with the resolution of the input image. The complexity is $O(n)$, where n denotes the number of pixels in an image. Kd-trees have a complexity for the construction of $O(n \cdot \log(n))$ and the expected complexity for a nearest neighbor search is $O(\log(n))$. One kd-tree has to be computed for the whole scene and one for each detected cluster. The strongest influence on the computation is the resolution and therefore by the number of points in a cluster. The complexity for the segmentation for each detected skin part is $O(n \cdot n_e \cdot \log(n))$, as it scales directly with the number of points n in the scene and with the expected number of neighbors n_e of each point. Fitting a body pose for a segment has again a complexity of $O(n \cdot n_e \cdot \log(n))$.

IV. EXPERIMENTAL RESULTS

The estimated body pose of the image given in Figure 3 is shown in Figure 6. To show the capability of the algorithm to compute the correct body pose in all three

dimensions, the body pose is shown from above. The results of the skin detection and segmentation have already been shown in Figure 3 and Figure 4, respectively. Figure 7 shows the results from different scenes. The first row shows the undistorted images as seen from the camera, with the detected skin parts. In the next two rows the estimated human body poses are shown from two different points of view.

As a stereo matching algorithm is used to create the point clouds, similar problems as in stereo matching are encountered. The stereo matching algorithm is unable to find a disparity for large areas of the same texture, so only silhouettes and no full representation can be found. Furthermore, some invalid stereo matches may lead to unpredictable behavior.

An experiment has been constructed to give a qualitative evaluation of the algorithm. Users have been asked to point in a direction, and the measured angle of the direction they are pointing to has compared to the computed one. 150 measurements have been recorded and the algorithm was able to estimate 80.2 % of the postures correctly, while the median error was around 6.8° . Only 3.2 % of the false positives could not be detected. As the algorithm will leave occluded body parts out, it is not able to compute the pointing direction when the user is pointing away from the camera and the arm can not be seen. Almost half of the body poses, the algorithm was not able to estimate in the experiment, can be explained by this problem. The other invalid matches can be explained by noise in the point cloud obtained by stereo vision or with other objects, which have been mixed up with body parts. The use of color may be a useful extension to increase the robustness of the algorithm.

Time constraints are hard on a mobile robot, the estimation of the body poses must be completed in nearly real time. Without optimization, the segmentation and human body pose estimation is performed in less than 250 ms. Together with the skin color detection and stereo matching, a frame rate of 2 fps is achieved on a standard pc. Using a dual core processor or another skin color filter could increase the computation speed up to 5 fps.

V. CONCLUSIONS AND FUTURE WORKS

A. Conclusions

We presented a novel approach to detect human body poses using any three-dimensional representation of the environment, like a standard stereo vision system. The algorithm uses point clouds and a kinematic human model, which can be represented by 27 typical human body poses. Our algorithm delivers reliable results and avoids false positives rather than computing an invalid pose. It is capable to detect an arbitrary number of humans in a scene and to deal with occlusions.

B. Future Works

As mentioned above, the execution time can be reduced. The algorithm can easily be parallelized by using dual core processors or a stream processor. No color information is considered for the fitting, although it might provide valuable

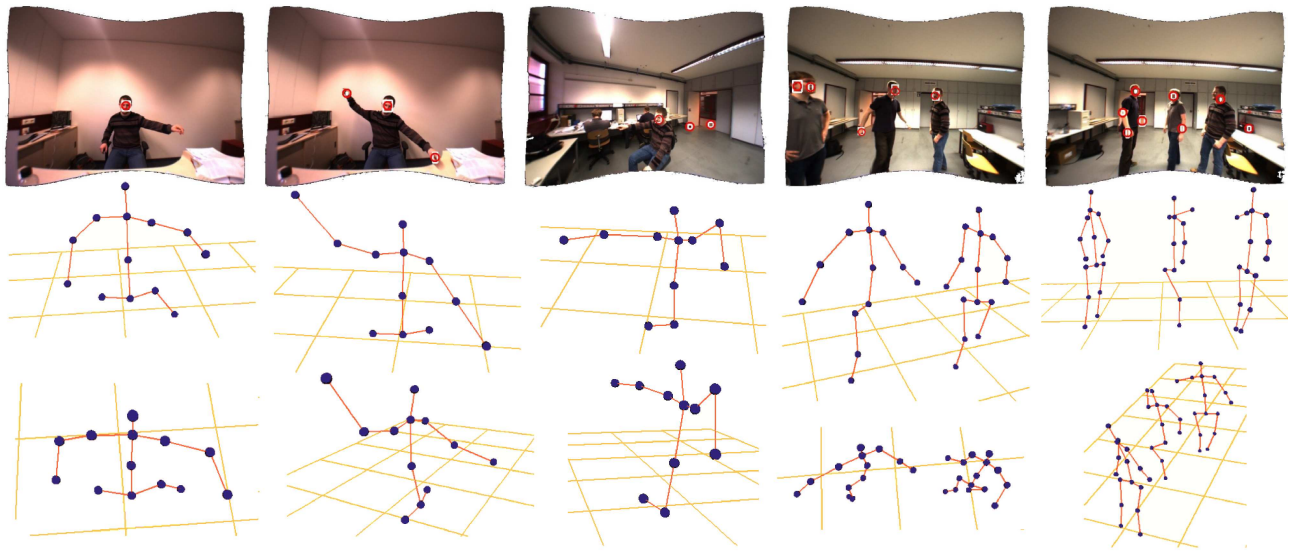


Fig. 7. Results of the human body pose estimation.

information. Another application could be the use of three-dimensional laser range finders instead of a stereo vision system. Our algorithm uses a single stereo image as input and every image is computed independently from its predecessors. Another useful feature, which could also increase the quality, could be the tracking of humans and the movements of each human. Initially developed for the recognition of human poses and validation of humans, the algorithm could also be extended to detect other, arbitrary objects.

VI. ACKNOWLEDGMENTS

This work is supported in part within the DFG excellence initiative research cluster *Cognition for Technical Systems – CoTeSys*, see also www.cotesys.org.

REFERENCES

- [1] G. Lidoris, G. Lidoris, K. K. A. Bauer, T. Xu, K. Kuhlentz, D. Wollherr, and M. Buss, "The autonomous city explorer project: aims and system overview," in *Proc. IEEE/RSJ International Conference on Intelligent Robots and Systems IROS 2007*, pp. 560–565, 2007.
- [2] V. Parameswaran and R. Chellappa, "View independent human body pose estimation from a single perspective image," in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR)*, vol. 2, pp. II–16–II–22 Vol.2, 2004.
- [3] A. Agarwal and B. Triggs, "Recovering 3d human pose from monocular images," *Transactions on Pattern Analysis and Machine Intelligence*, vol. 28, no. 1, pp. 44–58, 2006.
- [4] A. Mittal, L. Zhao, and L. S. Davis, "Human body pose estimation using silhouette shape analysis," in *Proceedings of the IEEE Conference on Advanced Video and Signal Based Surveillance (AVSS)*, (Washington, DC, USA), p. 263, IEEE Computer Society, 2003.
- [5] A. Elgammal, C. Sminchisescu, and C.-S. Lee, "Inferring 3d body pose from silhouettes using activity manifold learning," in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR)*, vol. 2, pp. II–681–II–688 Vol.2, 2004.
- [6] R. Bowden, T. Mitchell, and M. Sarhadi, "Reconstructing 3d pose and motion from a single camera view," in *Proceedings of the British Machine Vision Conference*, vol. 2, (University of Southampton), pp. 904–913, September 1998.
- [7] A. Kanaujia, C. Sminchisescu, and D. Metaxas, "Semi-supervised hierarchical models for 3d human pose reconstruction," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 1–8, 2007.
- [8] N. R. Howe, M. E. Leventon, and W. T. Freeman, "Bayesian reconstruction of 3d human motion from single-camera video," in *Advances in Neural Information Processing Systems 12*, pp. 820–826, MIT Press, 2000.
- [9] M. J. Park, M. G. Choi, and S. Y. Shin, "Human motion reconstruction from inter-frame feature correspondences of a single video stream using a motion library," in *Proceedings of the ACM SIGGRAPH/Eurographics symposium on Computer animation (SCA)*, (New York, NY, USA), pp. 113–120, ACM, 2002.
- [10] B. Boulay, F. Bremond, and M. Thonnat, "Posture recognition with a 3d human model," in *The IEE International Symposium on Imaging for Crime Detection and Prevention (ICDP)*, pp. 135–138, 7-8 June 2005.
- [11] B. Allen, B. Curless, and Z. Popovi, "The space of human body shapes: reconstruction and parameterization from range scans," in *SIGGRAPH*, (New York, NY, USA), pp. 587–594, ACM, 2003.
- [12] S. Pellegrini and L. Iocchi, "Human posture tracking and classification through stereo vision and 3d model matching," in *Journal on Video and Image Processing*, 2007.
- [13] S. Knoop, S. Vacek, and R. Dillmann, "Sensor fusion for 3d human body tracking with an articulated 3d body model," in *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, pp. 1686–1691, 2006.
- [14] H.-D. Yang and S.-W. Lee, "Reconstructing 3d human body pose from stereo image sequences using hierarchical human body model learning," in *Proceedings of the 18th International Conference on Pattern Recognition (ICPR)*, vol. 3, pp. 1004–1007, 2006.
- [15] C. Malerczyk, "3d-reconstruction of soccer scenes," in *Proceedings of the 3DTV Conference*, pp. 1–4, 2007.
- [16] M. W. Lee and R. Nevatia, "Body part detection for human pose estimation and tracking," in *Proceedings of the IEEE Workshop on Motion and Video Computing (WMVC)*, pp. 23–23, 2007.
- [17] J. Bregler, C.; Malik, "Tracking people with twists and exponential maps," in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 8–15, 23-25 Jun 1998.
- [18] D. M. Gavrila and L. S. Davis, "3-d model-based tracking of humans in action: a multi-view approach," in *Proceedings of the 1996 Conference on Computer Vision and Pattern Recognition (CVPR)*, (Washington, DC, USA), p. 73, IEEE Computer Society, 1996.
- [19] Y. Sagawa, M. Shimosaka, T. Mori, and T. Sato, "Fast online human pose estimation via 3d voxel data," in *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 1034–1040, 2007.
- [20] S. S.K., C. D.S., V. M., and R. Singh, "A robust skin color based face detection algorithm," in *Tamkang Journal of Science and Engineering*, vol. 6, pp. 227–234, 2003.