Towards 3D Human Posture Estimation Using Multiple Kinects Despite Self-Contacts

Andrew Phan
McGill University
aphan2@cim.mcgill.ca

Frank P. Ferrie
McGill University
ferrie@cim.mcgill.ca

Abstract

We present a marker-less human motion capture system that uses multiple RGB-D cameras to estimate the 3D posture of multiple people online at interactive rates in an indoor workspace measuring approximately 5 $m \times 5\ m \times 2\ m$. An interesting aspect of this work is how we handle the self-contact problem. We propose a novel multi-view voting scheme (MVS) to fuse measurements from different 2D or 3D algorithms. As a proof of concept, we present an MVS implementation that fuses optical flow images from each view and labels points in the current instance using the previously estimated posture. These labels allow us to trim invalid edges in a geodesic distance graph model and improve localization of geodesic extrema corresponding to the head, hands and feet for posture estimation. The system performs at ~8.3 Hz with a cumulative latency of ~570.40 ms and a projected median localization error of ~0.149 m. In addition, we propose a new multi-view Kinect and Vicon publicly accessible motion capture dataset for validation and benchmarks.

1 Introduction

The context for this work is a system, developed jointly under the project heading CHARM[1] for safely controlling a robotic assistant in a workspace shared by one or more human workers on a manufacturing assembly line. The focus of this research is on the human tracking module whose task is to use a commodity RGB-D camera network to detect, track and estimate the workers' posture at interactive rates.

Depth images simplify the foreground segmentation task and, instead of sparse marker-based features, provide dense point clouds of surfaces. However, if objects have the same depth and are touching, it is difficult to determine where one object ends and another begins using solely the depth data. Our goal is to leverage the complimentary strengths of both RGB and depth modalities in order to resolve ambiguities due to such contacts.

We define three classes for what we refer to collectively as the contact problem: 1) Subject-subject contacts occur when multiple subjects in the scene touch each other e.g. during a handshake; 2) Subject-object contacts occur when the subject touches objects in the scene e.g. manipulating tools; and 3) Self-contacts occur when one of the subject's body parts occludes or touches another body part e.g. hand on hip. These undesired and unavoidable contacts hinder accurate limb segmentation which in turn causes the posture estimation to deteriorate. Although the focus of this paper is on addressing the self-contact problem, our new publicly available dataset contains many instances of all three contacts.

[1]Collaborative, Human-focused, Assistive Robotics for Manufacturing, a joint project under the NSERC CRD program with the participation of General Motors Canada, the University of British Columbia, Laval University and McGill University.

1.1 Related work

In the literature, several papers use multiple RGB-D cameras for human motion capture [1] using techniques such as overlapping silhouette analysis [2], particle filters [3] and Kalman filters [4] to combine measurements across all the cameras. In this paper, we adopt a different data-driven approach based on the use of a geodesic distance graph (GDG), a popular graphical representation [5, 6, 7, 8] of the point cloud corresponding to the subject being tracked that, to the best of our knowledge, has only been used for 2.5D motion capture with a single RGB-D camera.

In a typical GDG, each vertex corresponds to a point in the subject's point cloud and edges connect neighbouring vertices. In this paper, we take the additional formulation as in [6, 8] such that each edge stores a weight corresponding to the Euclidean distance between the two vertices. Where the geodesic distance is defined as the distance between two points along the surface of an object, in the GDG the geodesic distance is approximately the sum of the edge weights along the shortest path from one body part to another. The geodesic distance is a useful metric between it is generally stable and posture invariant. Furthermore, the GDG is a useful representation because it facilitates locating geodesic extrema corresponding to the five extremities of the body (the hands, feet and head), useful for human posture estimation. Unfortunately, the self-contact problem causes the GDG construction to create invalid edges between vertices belonging to segments of the body that are not adjacent. When this occurs, geodesic extrema extraction may yield unexpected results and complicate localization of the desired extremities.

A number of recent papers use a naively constructed GDG in which edges are created between each vertex and all neighbours located within a maximum search radius. As a result, clever algorithms are required to deal with the contact problem such as using additional orientation and depth image patch descriptors [5], using a weighting strategy that depends on the quality of the feature extraction and of the local optimization [7], or using the Affine Scale Invariant Feature Transform (ASIFT) to track geodesic extrema [8].

In [6], Schwarz et al. present a method for trimming invalid edges in the GDG. If extremities go missing due to self-contacts, then the optical flow is used to estimate the current positions of missing segments based on their last known positions. Specifically, an edge is removed if one vertex lies on the occluding segment and the other lies on the occluded segment. Unfortunately, their concept of occluding and occluded segments does not apply to the the multi-view 3D case and their Matlab implementation performs at 2 Hz for Kinect data and 6 Hz for lower resolution Time of Flight data. In this paper, we extend their approach to multiple views and improve the performance to interactive rates for multiple Kinects. Furthermore, we make available a novel and comprehensive dataset containing the raw RGB-D data from four Kinects, complete which cal-

ibration parameters, and the skeleton ground truth provided by a commercial marker-based Vicon motion capture system.

The remainder of the paper begins in Section 2 with a description of a novel algorithm for computing an improved GDG that specifically addresses the self-contact issue. This is followed in Section 3 by a set of experiments using our new dataset that demonstrates the preliminary results of our algorithm in a real-world context. Finally, the paper concludes in Section 4 with some observations and a pointer to future work.

2 The Human Tracking Pipeline

Using multiple orthogonal views reduces occlusions, requires simpler a priori assumptions and allows the subject the freedom to face any direction. Unfortunately, many 2.5D algorithms do not apply directly to 3D and the use of multiple views complicates the data acquisition and online processing performance of the system. Using calibrated RGB-D cameras and commodity hardware, we propose an online multi-view voting scheme (MVS) that operates at interactive rates, combines measurements from multiple sources and generates a refined geodesic distance graph (GDG). This new GDG is robust to self-contacts, improves the localization of extremities and ultimately improves the posture estimation. As shown in Fig. 1, the algorithm is implemented as a pipeline containing a feedback loop where the previously estimated posture is used in combination with the optical flow in the intensity images to first label points in the current point cloud and then reject edges that fail a segment adjacency constraint.

In our sensor network, we use four Microsoft Kinect cameras placed around the work cell in an approximately orthogonal configuration. For ground truth, we use the data provided by the Vicon marker-based motion capture system [9]. Offline, we calibrate the network so that we can register all the data into a common reference frame using the same approach as Kramer et al. [10] and using standard stereo camera calibration techniques [11]. As such, we divide the calibration task into three steps. First, we intra-calibrate the Kinects by covering the IR emitters so that the checkerboard corners can be accurately located in the IR camera and computing both the intrinsic and extrinsic parameters of the RGB and IR cameras of each Kinect. Next, with the IR emitters uncovered, we world-calibrate as many Kinects as possible by obtaining the extrinsic parameters between the RGB camera of each Kinect and the world reference frame. The Vicon is also world-calibrated to the same world frame at this time by carefully aligning the Vicon calibration wand with the checkerboard at the world origin. Finally, we inter-calibrate the Kinects by obtaining the extrinsic parameters between the RGB cameras of Kinect pairs.

When the system is first powered on, the first online step is to model the background in order to reduce the number of pixels that need to be projected into the world reference frame. We initialize and update the background model in each depth image frame using the Minimum Background algorithm [12]. Next, using both the background model and the calibration parameters, we segment and register the foreground by projecting the foreground pixel from each Kinect into the common world reference frame. Then we downsample the resulting foreground point cloud by applying a 3D voxel grid and perform euclidean clustering [13].

If there are two subjects, then we expect to obtain two relatively large clusters corresponding to each subject. In practice, due to noise, other objects in the scene or subject-subject contacts, the number of clusters varies. To help identify human blobs, we compute the three principal axes of inertia lengths and the convex hull volume [14] of each cluster. Then we track and label blobs with stable features as human if they compare to previously measured values. Although principal axes of inertia are primarily for rigid bodies, we find that they are relatively stable and well suited for identifying human blobs, especially when the subject adopts an upright or standing posture such as during the initial posture.

On the first pass, we construct the GDG naively as described in Section 1. Compared to the 2.5D approach, we use a 3D search radius in metres instead of a 2D search radius in pixels. Regarding extracting geodesic extrema in 3D, the geodesic distance thresholding approach by Schwarz et al. [6] only works if the blob's centroid is added to GDG along with the necessary edges that connect the centroid to points on the torso. However, we prefer not to rely on the centroid because a) it is highly sensitive to vertical hand positions, b) the search radius must be tuned to the subject's torso and c) we want to minimize the number of edges created for performance reasons. Instead, we use the Accumulative Geodesic EXtrema (AGEX) [5] algorithm to locate exactly five geodesic extrema corresponding to the extremities similar to Brandao et al. [8].

Similar to Schwarz et al. [6], we label the five geodesic extrema called primary landmarks corresponding to the head, hands and feet. Assuming a human kinematic model with 15 segments of fixed and pre-determined segment lengths totalling 32 degrees of freedom (DOF), we initialize the posture estimation when the subject adopts an unambiguous T-pose where the arms are out the side and slightly offset to the front for left-right side disambiguation. Then we obtain additional landmarks called secondary landmarks corresponding to wrists, elbows, knees, ankles and neck by computing the centroid of the ring of points obtained by considering the points located at a pre-determined geodesic distance offset from the respective primary landmark.

To estimate the skeleton posture, we first assume a shape model where each segment is a capsule defined by a radius. Compared to other geometric shapes such as cylinders, capsules allow the quickest and simplest intersection detection. Then we perform the Levenberg-Marquardt nonlinear least squares minimization algorithm [15] to minimize the cost function, which currently consists of a landmark fitting term and a segment intersection term. Formally,

$$\varepsilon(\mathbf{q}, t) = \sum_{l=1}^{L} \|\mathbf{p}_l^t - f_l(\mathbf{q})\|_2^2 + \sum_{i=1}^{S} \sum_{\substack{j=1 \\ j \neq i}}^{S} c_{ij}(\mathbf{q})^2, \qquad (1)$$

where L is the number of landmarks, S is the number of segments, $\mathbf{q}$ is the vector of parameters, $\mathbf{p}_l^t$ is the position of the l-th landmark at time t, $f_l : R^d \rightarrow R^3$ is the forward kinematic function that computes the position of the l-th landmark given $\mathbf{q}$, $c_{ij}(\mathbf{q}) = r_i + r_j - d_{ij}(\mathbf{q})$ if segments i and j intersect otherwise $c_{ij}(\mathbf{q}) = 0$, r_i is the capsule radius for segment i and $d_{ij}(\mathbf{q})$ is the distance between the two capsule segment axes. The posture obtained can be used in higher level processes such as gesture or activity recognition. The remaining steps, described next, detail the novel MVS feedback loop for producing a refined GDG that is robust to self-contacts.

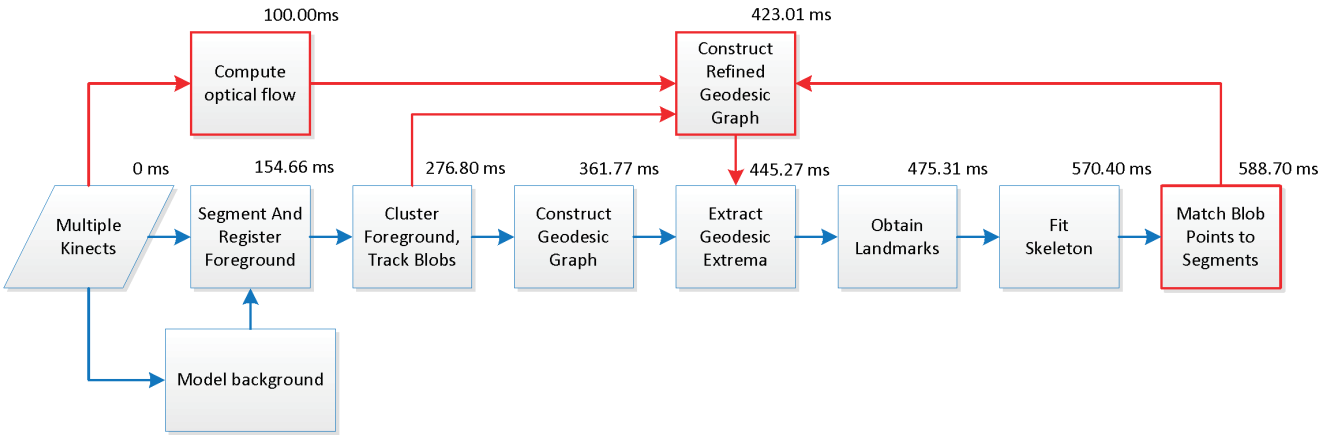

Figure 1. System data flow diagram

2.1 Multi-view voting scheme

The objective of the MVS is to label each point in the current blob with a segment (e.g. head, torso, etc.) using the previously estimated posture and while fusing measurements from multiple information sources (e.g. viewpoint or low-level vision algorithm). Each measurement or information source provides a single vote and votes are combined to produce a label. As a proof of concept, our preliminary MVS implementation fuses the optical flow images produced from each Kinect RGB camera.

We start by using the human shape model and the previously estimated posture to label each point in the previous blob with a segment. If a point lies within more than one segment candidate due to intersections, then we choose the closest segment as label. Note, some points may remain unlabelled because the shape model is a coarse approximation of the true shape or because of errors in the skeleton fitting. Next, for each Kinect RGB camera, we project all points of the current blob and all labelled points of the previous blob into the image reference frame. Then we use the chain of optical flow images from t_{prev} to t_{next}, which typically consists of 2-3 frames due to computational latencies, to estimate the current location of labelled points of the previous blob. Finally, we determine the closest segment for each point of the current blob and if the distance is below a threshold in pixels equal to a search radius (15 pixels) plus an uncertainty based on the number of flow frames used (2 times the number of frames), then a vote is registered otherwise the camera does not vote. When all the cameras have voted, the segment with the most votes is the point's label. In the case of a tie, the closest segment in 3D is chosen from among the candidates. If a point has no votes, then the closest segment in 3D is chosen as long as it is within a maximum distance.

2.2 Refined geodesic distance graph construction

To construct the refined GDG, we perform a number of checks before allowing an edge to be created. Explicitly, for each vertex, we search neighbouring vertices and reject edges if: a) both the vertex and its neighbour are labelled but not adjacent or do not belong to the same segment, b) the vertex is labelled but its neighbour is unlabelled (the reverse is accepted however) or c) if the vertex is unlabelled, its neighbour is labelled, but the distance between the two is greater than the distance between the vertex and another labelled neighbour with a different label. The result is a refined graph with fewer edges between non-adjacent segments as shown in Fig. 2. To close the feedback loop, we use the refined graph instead of the naive graph for locating geodesic extrema on the next iteration.

3 Experimental Results

To validate our algorithms and provide a public data set for benchmarks, we recorded 11 sequences with foreground and Vicon data and 8 sequences with the full and raw RGB-D data from four Kinects and Vicon data complete with calibration parameters that we make available online at [16]. While the former is easier to use because no calibration parameters are required, it only allows experimenting with 3D algorithms such as ICP or range flow. Conversely, the latter requires more work to register the data into the common world reference frame, but permits experimenting with the full range of 2D algorithms (e.g. stereo, SIFT, optical flow) in addition the 3D algorithms aforementioned. The sequences vary in the number of workers (1 or 2) and in the amount and type of contacts (subject, object and self).

First, we present our sensor network calibration results using the standard root mean squared (RMS) reprojection error metric in pixels. For the intra-calibration, we obtained an average RMS error of 0.301 pixels for the Kinect's RGB camera intrinsics, 0.260 pixels for the Kinect's IR camera intrinsics and 1.266 pixels for the extrinsics between the two cameras. For the subsequent world calibration error, we got an average RMS error 1.03 pixels. Finally, for the inter-calibration, the average RMS error was 0.988 pixels between Kinect RGB camera pairs. While the real-world error increases with the distance from the camera, we observe that a pixel error of 1 translates into approximately 1-2 cm in the real world for a typical workcell. Note also that to project the depth pixel into the world reference frame, the total reprojection error is the sum of the intra-calibration extrinsic error from IR to RGB and the world-calibration error from RGB to world.

Next we compare the posture estimate to the ground truth skeleton provided by the Vicon (Fig. 4). For qualitative evaluation, we invite the reader to view the accompanying video demo available online [17]. For the quantitative evaluation, we compute the euclidean

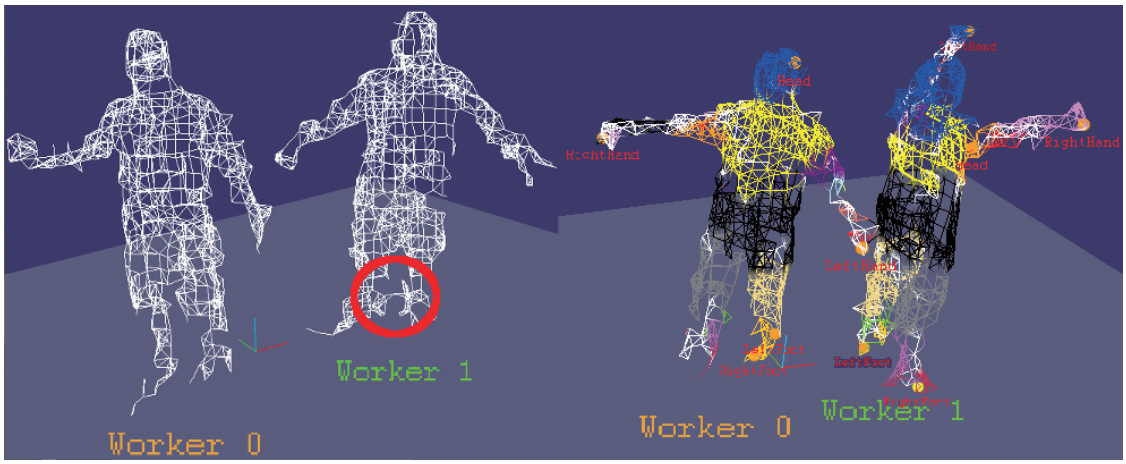

Figure 2. Geodesic distance graph edges: a) Naive (left, note the invalid edge near Worker 1's feet) b) MVS refined (right)

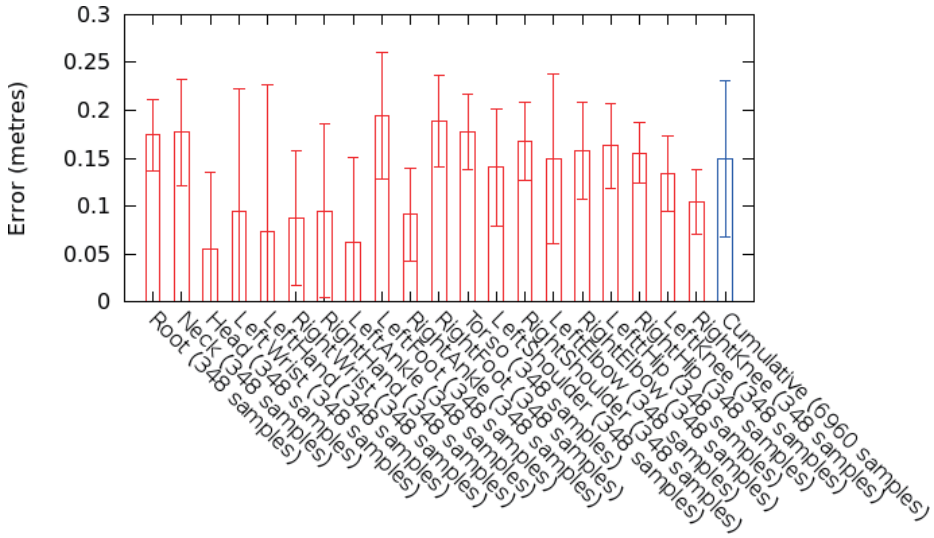

Figure 3. Median posture estimation errors compared to Vicon skeleton

distance between the Vicon skeleton's joints and our skeleton's joints and present our preliminary median error result in Fig. 3 for the simplest T-pose sequence, without undesired contacts, for which we obtain a median error of 0.149 ± 0.082 m. Note that because the raw and uncorrected Vicon skeleton lies on the surface of the body, our error includes a Vicon bias error approximately equal to the half the width of each segment and corresponding to the distance between skin and bone.

We now discuss the cumulative latencies displayed at the top right corner of each major processing block in Fig. 1. From RGB-D frames to posture estimate, our multithreaded C++ implementation takes 570.40 ± 121.68 ms but is able to do this at a frequency of 8.37 ± 3.25 Hz on an AMD X4 640 3Ghz processor that was released in 2010. Also, we are able to obtain dense optical flow [18] frames at ~15 Hz × 4 views (60 Hz effective) with minimal border cropping thanks to a modern Nvidia Titan Black GPU released in 2014. Finally, note that the cumulative latency at the output of human blob tracking and labelling is ~276.80 ms while it is ~588.70 ms at the output of the labelling of previous blob points. Since the optical flow frame is obtained in intervals of ~100.00 ms, consequently we require 2-3 frames on average in the optical flow chain in order to perform the MVS.

Compared to the naive GDG construction, the MVS refined GDG has ~24% fewer edges of which ~3% were deemed to be invalid due to self-contacts. Furthermore, the MVS was able to determine an unambiguous label ~61% of the time. In ~13% of the time, there was a tie. These promising results demonstrate the validity of our approach and we hope to increase the percentage of points labelled by incorporating additional votes from other 2D (e.g. SIFT [19]) and 3D (e.g. range flow [20], ICP [21]) algorithms to further enhance the refined GDG construction.

4 Conclusion

We have presented a system to track and estimate the 3D posture of multiple subjects using multiple RGB-D cameras. To deal with the self contact problem, we proposed a novel multi-view voting scheme and have used it to fuse optical flow measurements as proof of concept. As such, we are able to obtain a refined GDG that contains ~3% fewer invalid edges for locat-

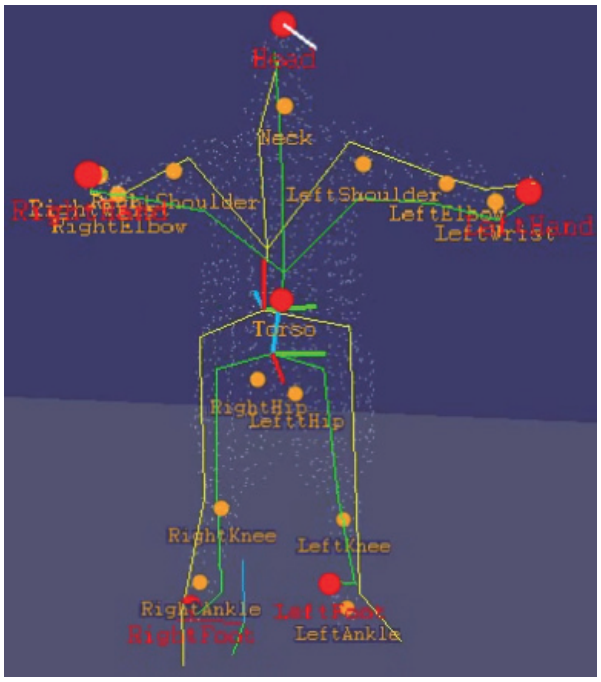

Figure 4. Landmark detection and skeleton fitting (green) compared to the Vicon skeleton (yellow)

ing geodesic extrema. The resulting system operates over a volume of approximately $5m \times 5m \times 2m$ at a rate of ~8.3 Hz, accommodating several individuals at a time. Although not as precise as a motion capture system, we are encouraged by the results obtained thus far. Current and future work will be aimed at refining the system further, particularly with respect to accuracy of localization.

Acknowledgment

This work was supported by Reparti and NSERC. We also thank colleagues Denis Ouellet (Université Laval) and Olivier St-Martin Cormier (McGill University) for their much appreciated assistance with the Vicon+Kinects datasets and the database, respectively. Finally, we thank NVIDIA for donating the Titan Black GPU.

References

[1] K. Berger, "A state of the art report on multiple rgb-d sensor research and on publicly available rgb-d datasets," in *Computer Vision and Machine Learning with RGB-D Sensors*. Springer, 2014, pp. 27–44.

[2] K. Berger, K. Ruhl, Y. Schroeder, C. Bruemmer, A. Scholz, and M. A. Magnor, "Markerless motion capture using multiple color-depth sensors." in *VMV*, 2011, pp. 317–324.

[3] L. Zhang, J. Sturm, D. Cremers, and D. Lee, "Real-time human motion tracking using multiple depth cameras," in *Intelligent Robots and Systems (IROS), 2012 IEEE/RSJ International Conference on*. IEEE, 2012, pp. 2389–2395.

[4] C. Morato, K. N. Kaipa, B. Zhao, and S. K. Gupta, "Toward safe human robot collaboration by using multiple kinects based real-time human tracking," *Journal of Computing and Information Science in Engineering*, vol. 14, no. 1, p. 011006, 2014.

[5] C. Plagemann, V. Ganapathi, D. Koller, and S. Thrun, "Real-time identification and localization of body parts from depth images," in *Robotics and Automation (ICRA), 2010 IEEE International Conference on*. IEEE, 2010, pp. 3108–3113.

[6] L. A. Schwarz, A. Mkhitaryan, D. Mateus, and N. Navab, "Human skeleton tracking from depth data using geodesic distances and optical flow," *Image and Vision Computing*, vol. 30, no. 3, pp. 217–226, 2012.

[7] A. Baak, M. Müller, G. Bharaj, H.-P. Seidel, and C. Theobalt, "A data-driven approach for real-time full body pose reconstruction from a depth camera," in *Consumer Depth Cameras for Computer Vision*. Springer, 2013, pp. 71–98.

[8] A. Brandao, L. A. Fernandes, and E. Clua, "M5aie a method for body part detection and tracking using rgb-d images," in *Computer Vision Theory and Applications, 2014 International Conference on*, 2014.

[9] Vicon t-series. Retrieved 2014-04-19. [Online]. Available: http://www.vicon.com/System/TSeries

[10] J. Kramer, M. Parker, H. C. Daniel, F. Echtler, and N. Burrus, *Hacking the Kinect*. Springer, 2012.

[11] Opencv camera calibration and 3d reconstruction. Retrieved 2014-04-19. [Online]. Available: http://docs.opencv.org/modules/calib3d/doc/camera_calibration_and_3d_reconstruction.html

[12] K. Greff, A. Brandão, S. Krauß, D. Stricker, and E. Clua, "A comparison between background subtraction algorithms using a consumer depth camera." in *VISAPP (1)*, 2012, pp. 431–436.

[13] Pcl euclidean cluster extraction. Retrieved 2014-05-20. [Online]. Available: http://www.pointclouds.org/documentation/tutorials/cluster_extraction.php

[14] Pcl convexhull class template reference. Retrieved 2014-05-20. [Online]. Available: http://docs.pointclouds.org/1.7.0/classpcl_1_1_convex_hull.html

[15] S. Agarwal, K. Mierle, and Others, "Ceres solver," http://ceres-solver.org.

[16] O. St-Martin Cormier, A. Phan, and F. P. Ferrie. Mcgill-reparti artificial perception database. Retrieved 2014-12-17. [Online]. Available: http://www.cim.mcgill.ca/~apl/database

[17] A. Phan and F. P. Ferrie. Mva 2015 video demos. Retrieved 2014-04-18. [Online]. Available: http://www.cim.mcgill.ca/~aphan2/mva2015/

[18] Opencv video analysis farneback optical flow. Retrieved 2014-04-18. [Online]. Available: http://docs.opencv.org/modules/ocl/doc/video_analysis.html

[19] D. G. Lowe, "Distinctive image features from scale-invariant keypoints," *International journal of computer vision*, vol. 60, no. 2, pp. 91–110, 2004.

[20] J.-M. Gottfried, J. Fehr, and C. S. Garbe, "Computing range flow from multi-modal kinect data," in *Advances in Visual Computing*. Springer, 2011, pp. 758–767.

[21] Pcl iterative closest point class template reference. Retrieved 2014-12-18. [Online]. Available: http://docs.pointclouds.org/trunk/classpcl_1_1_iterative_closest_point.html