Human motion capture using 3D reconstruction based on multiple depth data

Wassim Filali, Jean-Thomas Masse[*,†], Frédéric Lerasle[*], Jean-Louis Boizard[*] and Michel Devy
CNRS, Laboratoire d'Analyse et d'Architecture des Systèmes
7 avenue du colonel Roche, F-31400 Toulouse, France.
{wfilali, jtmasse, lerasle, boizard, michel}@laas.fr

[*] Université de Toulouse, UPS, LAAS
F-31400 Toulouse, France.

[†] Magellium
F-31520 Ramonville Saint-Agne, France.

Abstract— **Human motion is a critical aspect of interacting, even between people. It has become an interesting field to exploit in human-robot interaction. Even with today's computing power, it remains a difficult task to successfully follow the human's motion from image processing alone. New sensors were introduced, bringing depth sensing at low or no cost. Using this new technology, this paper presents a new methodology to see space with multiple depth sensors, using machine-learning technique, and features in voxel space to learn to reconstruct humans' joints in single, fused acquisitions. We back up and validate the procedure with ground truth acquired from commercial Motion Capture, and prove the approach to perform particularly well on an expansive set of motion and poses, and compare with current standard software on single depth sensors.**

Keywords- human posture reconstruction, depth sensing, sensor fusion, machine learning, voxel

I. INTRODUCTION AND FRAMEWORK

Human Motion Capture (MoCap in short) was until recently reserved for animation and biological studies. It has now entered home entertainment, such as Motion Controllers in consoles such as the Nintendo Wii and PlayStation 3. More recently, the Kinect sensor of the Microsoft Xbox was introduced. It had a resounding commercial success [1]. Currently, similar devices providing depth and RGB images exist, such as the Asus Xtion Pro Live [2]. PrimeSense's technology of structured light sensing [3] powers both the Kinect and the Xtion.

Several SDKs such as OpenNI [4] and the proprietary Kinect SDK exist. They provide access to the sensors' data, as well as algorithms to track movement and gestures, both in forms of events and real-time skeleton joints positions. The skeleton algorithms work on the device's depth maps. They exploit machine learning, an optimization-based approach, to segment body parts. This choice allows the delivery of an optimal stream of information and super-real-time performances.

The OpenNI SDK's closed-source middleware NiTE [5], from PrimeSense [6], is what produces the whole-body skeletons in OpenNI. It only needs depth maps independent from the sensor. Therefore, we could OpenNI with a Kinect, a stereo or even a time-of-flight camera. That is why - in this paper - we will call the Asus Xtion we used to acquire our data a *sensor*.

978-1-4799-0652-9/13/$31.00 ©2013 IEEE

Even if the literature has expansively evaluated the sensor's depth accuracy [7] [8], the skeleton algorithms have been seldom compared to physical ground truth, even though simulated data was extensively used to circumvent menial labeling of the training data for the learning algorithms. The available software (and the algorithms they are based on) focuses on entertainment applications. It tends to overlook the user not facing the sensor. It does not handle big changes in view angle nor strong occlusion and it has a limited range.

That is why our focus is two-fold: (1) to reformulate the pioneering work of Shotton et al. [9] in a multi-sensor framework, and (2) to couple such a system with a commercial MoCap in order to provide ground truth. This ground truth is used for learning and for quantifying improvements in testing.

The first following section contains related work. The section III after it describes the multi-sensor and MoCap platform we used for data acquisition. Section IV describes our approach, while section V considers implementation details. Finally, evaluations and discussions come in section VI before conclusion and perspectives.

II. RELATED WORK

Literature on Motion Capture by classical camera and computer vision techniques is very rich, as the survey from Moeslund et al. shows [10]. Common sense is enough to say that multi-ocular approaches perform better than monocular strategies. Best self-occultation resistance starts with three or four cameras [11]. HumanEva [12] for example, describes and publishes a multi-view datasets for human motion estimation with ground truth.

More recently, budget active sensor technology [3] bloomed. Such depth maps combined with advanced learning algorithms such as random forests [9], [13], can detect the full human body structure in single images, as demonstrated by [9]. However, all the single sensor approaches are limited, even those using depth. The issue related to investigating a multiple depth sensor approach is due to the lack of a large-scale, public RGB-D database with ground truth.

Most multi-sensor works focus on optimizing a human body surface to the sensed point cloud [14]. Our approach rather combines multiple sensors raw data to output a voxel-

space. It also uses a new descriptor to feed the machine learning process. The learning and the validation of the output data use ground truth data captured using Motion Capture equipment. We believe that all these points of our approach will make it a strong proposal.

III. Our Multi-sensor Platform Setup and Data Acquisition

One of our objectives is to estimate the precision in the single-sensor system's estimation of the human body joints' positions. Therefore, we decided to use the Motion Capture system from Motion Analysis [15].

This acquisition serves three purposes:

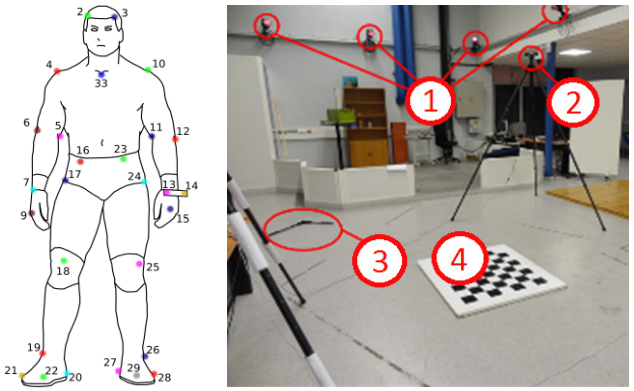
- First, to evaluate the performance of NiTE from several angles: frontal (as it was designed), and also sideways and behind;
- Second, to create a dataset to exploit in a learning algorithm;
- Third, to create another dataset to evaluate our learning and other algorithms.

We placed and oriented the MoCap reference frame on the RGB calibration chessboard, respectively ③ and ④ in Figure 1(b), in order to localize all the sensors in MoCap reference frame. Localizing the device's RGB sensor was sufficient because the Depth-to-RGB registration is factory-determined and readily available using OpenNI. Works show that significant improvement can be made, but only in the close range of the sensor [16].

Time synchronization was subsequently achieved by MoCap marker projection in RGB frames. On such a frame, a fast marker would leave a blurry trail. Since MoCap frames were 10 times as frequent, we fine-tuned the time synchronization to project the marker on the middle of the blur. That makes all the depth streams synchronized to MoCap since all the frames are time-stamped. We also considered sequences short enough for time-slide to be inconsequential.

Acquired data consist of two sets. The labeled set, IRSS35, has a full set of MoCap 35 markers to allow skeleton building. The NS-CAP13, has a reduced set of 13 markers, for quick testing. Both feature several sequences covering exercise moves, sport moves, regular moves, poses from regular life, and extremely varying random poses for completion. IRSS35 has nine sequences, some of them recorded twice, totaling 16 minutes or upward of 21569 workable depth frames.

It must be noted that each sequence is not a simple movement, but a mix and match of actions such as weight lifting, running, squats, as well as more mundane movement such as dancing, broom handling, and moving furniture.



(a) (b)

Figure 1. Precision measurement experiment.
(a) Motion Capture markers setup [15], and (b) Motion Capture (① are 4 of the 10 IR or NIR cameras) and Xtion (②) setup, with MoCap reference frame (③) and 1 m² camera calibration chessboard (④) also present.

IV. Description of Our Approach

Our approach uses any sufficient number of depth streams, but at least three. The first step is the foreground segmentation. The second is the labeling of all pixels belonging to every user in the field of view. Then a voxel mapping is built inside a volume encompassing the width, height and breadth of the body's point cloud. After that, each the body voxels are labeled as different body parts, such as the hand, forearm and elbow. Then each body part voxel contributes to find the associated joint using the Mean Shift algorithm.

Our approach is inspired from that of Shotton et al. [9], but as it is using a voxel space, it has the advantage of being a viewpoint-insensitive method. This makes the learning not influenced by the perspective effect, as it is the case in depth images. Figure 2 shows the two methods' workflows side by side.

We describe each step in further subsections.

A. Segmentation and voxellization

The segmentation step separates users from the background as well as from each other. It is a foreground segmentation. Thus, the reconstruction algorithms can work only on pixels belonging to a single person. We use the same segmentation as NiTE.

We voxellize [17] using the segmentation information of the depth images, but also their depth information. A depth

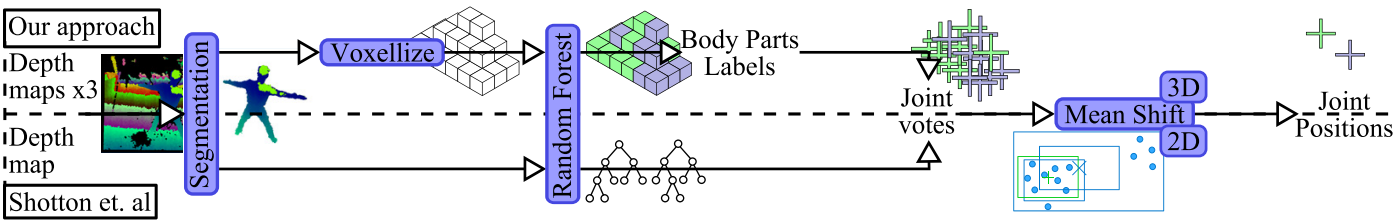


Figure 2. Our pipeline compared to Taylor et al. [13]. Our approach fully exploits the multiple depth sensors.

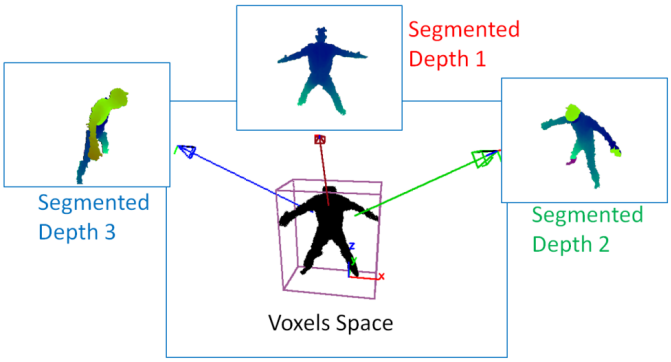


Figure 3. Human silhouette voxellization using multiple segmented depth streams

pixel considered as not belonging to the user eliminates all the voxels that project on it. Then, every depth pixel distance allows the elimination all voxels that are closer than that distance. Voxel size is 1 cm. We believe that this strikes a good balance between resolution, density, reliability, and speed.

Figure 3 shows an illustration of voxellization result.

B. *Random Decision Forest*

The Random Forest's role is to decide to which body parts each pixel or voxel belongs. Alternatively, the output can also be a geometrical offset leading to the probable location of the joint. Pure labeling is the simpler case where the offset is null, but requires post-processing to locate the joint.

A Random Decision Forest [18] is a learning algorithm that produces a forest of several decision trees. Each tree consists of split and leaf nodes. The number of trees is implementation-defined (see discussions in Section V).

Every node comprises a feature, and a test on the feature value of a sample. An optimization algorithm chooses both of these during the learning process. The optimization task is to maximize the selectivity of the feature/test pair for leafs, and to balance the tree at the splits level. The range of candidate features is a random subsample of all possibilities, which gives the algorithm its adjective. Splitting is stopped if there is not enough learning exemplars to further make significant statistics, or if the user has set a maximal depth to ensure speed. This depth and the quantity of features are also implementation-defined.

Random Forest split nodes lead to two other nodes, whereas leaf nodes contain a decision. To obtain the decision of a tree, one starts at the root with a dot and navigates the tree by following the result of each test leading to leaf node. The decision of the forest is the set of leaf nodes attained in each tree.

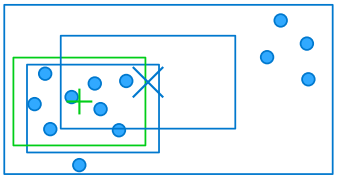
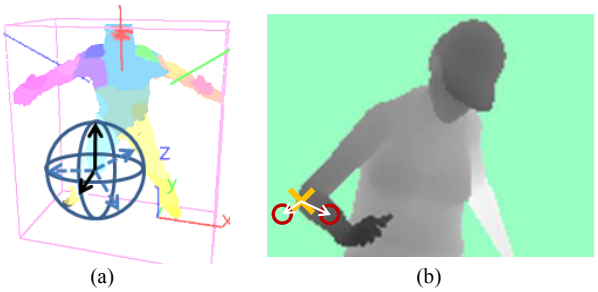


Figure 5. Mean-shift insensitivity to outliers.
× marks the initial barycenter with the neighborhood of the enclosing rectangle, after iterations, it converges to the + sign.



(a) (b)

Figure 4. (a) An example of our 5-tests Voxels features $f_{v,\Delta_1..\Delta_5}$ (b) and of Shotton et al. 2-tests depth pixel features f_{p,Δ_1,Δ_2} [9].

Each leaf node stores a prediction model built upon the training set. This prediction model is either a probability mass function $p_l(c)$ over body parts $\{c \in C\}$ [9], which serve as an intermediate state, or a set of K weighted relative votes for all the limbs L $\{\mathcal{V}_{lj} = (\boldsymbol{\Delta}_{ljk}, w_{ljk})\}_{l \in L}$ for each body joint j [13]. It is constituted of a geometrical offset $\boldsymbol{\Delta}_{ljk}$ and a weight w_{ljk}. K is kept small for speed, without loss of precision. Our approach relies on the first kind of learning. We process body parts labels as an intermediate step, as [9], but in voxel space. We assume the current NiTE implementation is either [9] or [13].

The voxel descriptor $f_{v,\Delta_1..\Delta_5}(\mathbf{v})$ we use is simpler than the one on disparity images. We consider user voxels with a unit of 1 cm and not disparity pixels from a perspective viewpoint like features used by Shotton et al. Figure 4 shows them side by side. Pixel offsets-based descriptors $f_{p,\Delta_1,\Delta_2}(\mathbf{p})$ have to deal with different distances when pixel $\mathbf{p}$ is at differing depths, while f_v is insensible to perspective and computationally allows for more offsets. We chose the concatenation of five Boolean tests of the presence of the voxel at five geometrical offsets $\Delta_1..\Delta_5$ from voxel $\mathbf{v}$: To prevent overfitting, the offsets Δ_i are kept within a certain window limit (see section V for implementation details).

$$f_{v,\Delta_1..\Delta_5}(\mathbf{v}) = \sum_{i=1..5} 2^i \cdot V(\mathbf{v} + \Delta_i) \tag{1}$$

C. *Mean Shift*

The purpose of this step is to find the maxima of the joints votes' density. The Mean-shift algorithm [19] is a classic iterative solution adapted to the situation. Here we model the sampled votes for a joint probability. In both *cell* types, pixels or voxels, one applies Mean Shift on all the cells whose body part is associated with the joint being optimized. Supernumerary body parts increase accuracy by removing cells that are too far from a joint. For pixels, the resulting 3D location is the pixel's depth value pushed back by a predefined amount depending on the joint.

The method updates the maxima estimation x using a kernel K - Euclidian distance in our case - to weight all the samples x_i in the neighborhood $N(x)$ of x according to the formula

$$x \leftarrow \frac{\sum_{x_i \in N(x)} K(x_i - x) \cdot x_i}{\sum_{x_i \in N(x)} K(x_i - x)} \tag{2}$$

The neighborhood $N(x)$ is a very important step of the process, as it can significantly alter the quality of the convergence,

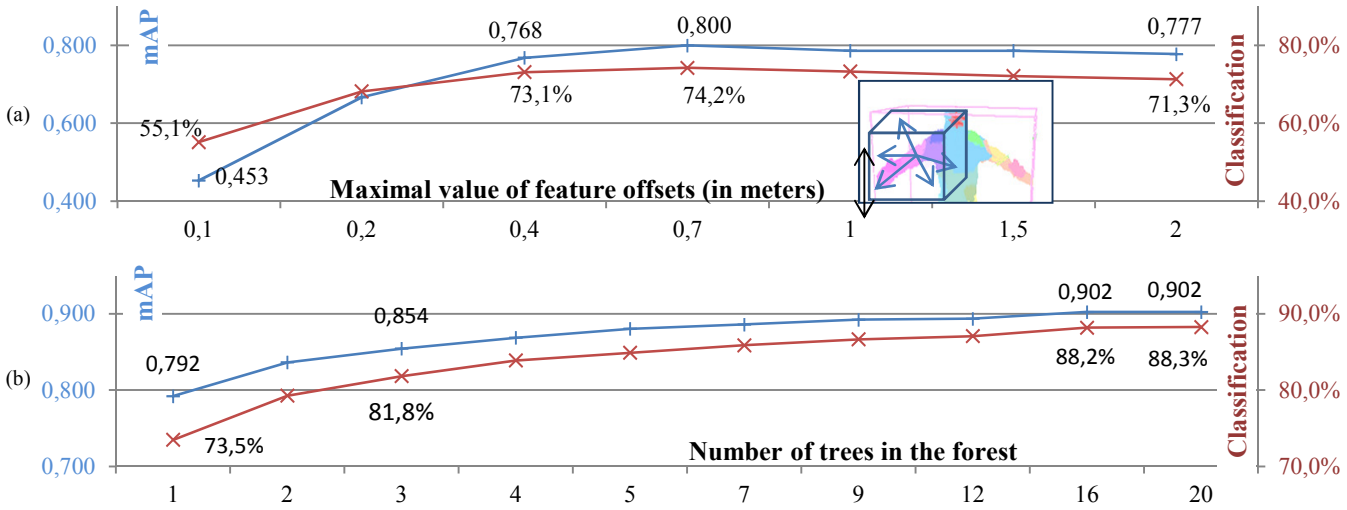


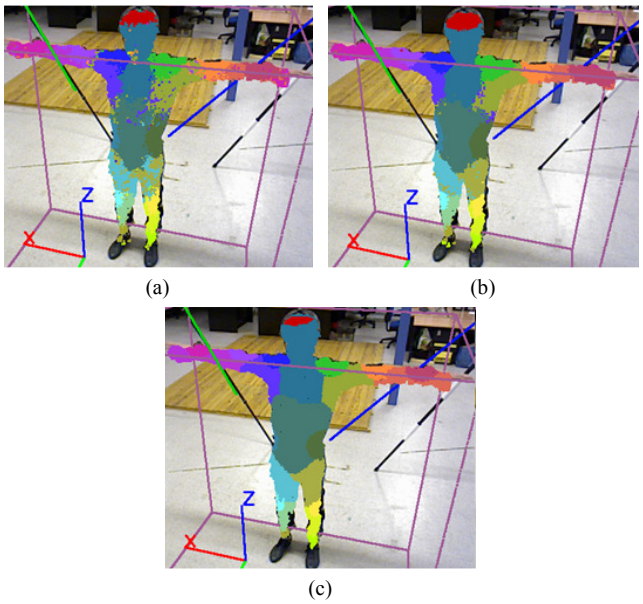
Figure 6. Plots of training variable impacts.
(a) shows the correlation between mean Average Precision (mAP, see subsection VI below) and classification with variable size of the window for voxel features. (b) is the same but varying the number of trees of the forest.

both in accuracy and speed. The trivial neighborhood is the whole set of votes. However, it is clearly outset by clusters of outliers, as often happens with voting techniques. Figure 5 shows an example of this situation and the benefit of limiting the neighborhood.

As an improvement, we additionally reduce the size of $N_t(x)$ over time: when we aim for a joint vote window of a target size, we first start at the maximum size. Then we gradually reduce the size of the window by a fraction until we have sufficient data compactness or when we have attained the target size. We step back if we have a compactness reduction to have the best center estimation of clouds that are empty inside.

V. IMPLEMENTATION

The first step in order to obtain a classifier based on a random forest is the training, which requires exemplar data. The acquisition of the database (see Section III above) provided depth as well as skeletal MoCap data, which we used as ground truth for the associated sequences.



(a) (b)

(c)

Figure 7. Voxel grid labelled by (a) 3-trees and (b) 20-trees voxel, compared to ground truth (c).

Every voxel gets a ground truth label (hand, head…). However, we cannot label all voxels within a given distance from the ground truth position of the Mocap marker. This would have mislabeled certain voxels where there were gaps. Such as the voxels of the torso that are very close to the hand. Therefore, we labeled voxels in a propagative manner. They are seeded with the joint-containing voxel, then the labeling propagates to the nearest connected and unlabeled voxel.

Many variables were free parameters. We summarize the results in Table I with their chosen optimal value. The most important variables are:

- Subsampling: We randomly chose some voxels out of the whole set for the learning. This is justified by the semi-continuous nature of the feature and observed phenomenon, a lot of the data is redundant.
- Maximal offset: That represents the farthest voxel that can participate in the descriptor of the voxel being labeled. It has to be limited to prevent overfitting to unrelated data. Large offsets could reach into very different body parts.

TABLE I. EXPERIMENTAL RESULTS

Variable / Definition	Min tested	Optimal or first plateau value	Max tested
No. of candidate features	100	1500	5000
Data subsampling	0.5%	15%	100%
Trees' max depth	10	50	100
Maximal voxel offset (in cm)	10	70	200
Number of trees	1	16 (3*)	20

Optimal or threshold value was according to Body joints performance.
*: this value however severely cripples the speed. 3 trees are used instead.

TABLE II. EXAMPLE OF CONFUSION MATRIX ON A FRAME.

Head	RSh	RElb	RWrs	LSh	LElb	LWrs	RHp	RKn	RFt	LHp	LKn	LFt
597	0	0	0	0	0	0	0	0	0	0	0	0
0	1317	0	0	5	0	0	0	0	0	0	0	0
0	0	912	0	0	8	0	0	0	0	0	0	0
0	0	0	577	0	0	4	0	0	0	0	0	0
2	13	0	0	981	2	0	0	0	0	0	0	0
0	0	5	0	0	298	1	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	1612	0	0	13	0	0
0	0	0	0	0	0	0	0	1202	0	0	15	0
0	0	0	0	0	1	0	0	0	709	0	1	0
0	0	0	0	0	0	0	2	0	0	1380	0	0
0	0	0	0	0	0	0	0	3	0	0	1148	0
0	0	0	0	0	0	0	0	0	2	0	0	1117
2	0	2	72	0	15	79	2	1	3	6	3	20

Abbreviations: L: Left, R: Right, Sh: Shoulder, Elb: Elbow, Wrs: Wrist, Hp: Hip, Kn: Knee, Ft: Foot. Last line: Unknown (not labeled) body parts.. Left wrist was outside the voxel frame in this example.

- The number of trees in the forest: These trees will run in parallel and every one will have a vote.

We plotted the last two in Figure 6 against mAP (The red line "x" stand for the classification and the blue line "+" stands for the mAP). For more detail, mAP is presented in section VI Evaluations and Discussion in subsection B.

As we can see from Table I and Figure 6, always judging from mAP performance, more trees allow for a better decision. However, we noticed a high performance hit but only for a small improvement. As in any learning method, the approach is prone to overfitting. The Depth of a tree should be limited for performance, but doing so may introduce underfitting.

VI. EVALUATIONS AND DISCUSSION

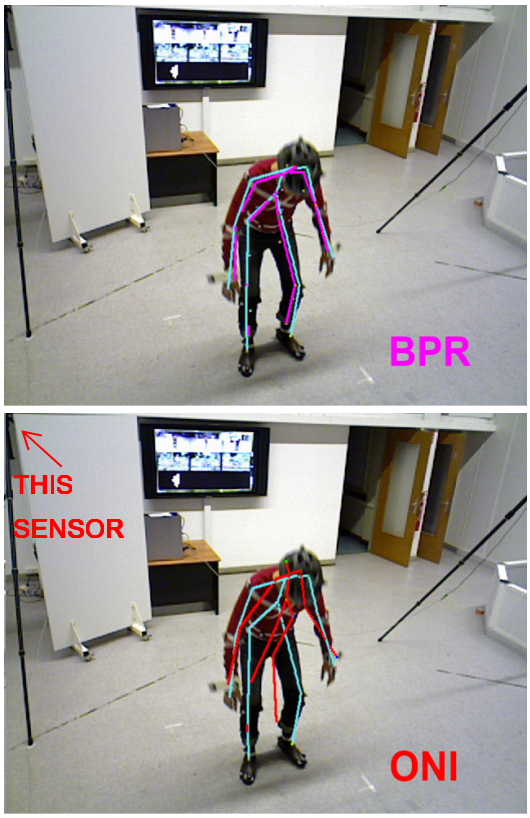
Test data comprises fifty common movements of the IRSS35 dataset, some of which are repetition but not identical to the hundreds that were used for training.

Training was run under cross-validation: the entire training set was split into three parts. Three trainings took place. Every one of them was trained on two parts and tested on a third part. We kept the training that resulted in the best cross-validation. This is called a three-fold cross-validation.

First, we present some quick qualitative results, then we present more quantitative results.

A. *Qualitative results*

Figure 7 shows that the voxel classification is quite noisy, but well distributed. Therefore, it is robust enough to give good estimation of the body parts, as shown in Figure 9. Figure 9 shows results for every one of the three sensors ONI1, ONI2 and ONI3 as well as our fusion approach. We also used supplementary intermediate body parts, as in [9]. This provides more selectivity to the classifiers and allows Mean-shift to keep the parts correctly centered on the joints. The forearm between hand and elbow is one such body part.



(a)

(b)

Figure 8. Our skeleton reconstruction (BPR) (a) and the best OpenNI/NiTE reconstruction (b) where the upper body only is mostly correct.

The skeleton reconstruction in Figure 8(a) is also much more accurate, that is closer to MoCap skeleton in light blue than the reconstruction from the Kinect, Figure 8(b). It can even work in somewhat difficult poses.

More qualitative results, especially videos, can be found on the project's page[1].

B. *Quantitative Results*

The confusion matrix in Table II shows that the classification on a per-voxel basis is the most reliable as the most predicted class is the correct one, and the incorrectly accepted do not overwhelm the correct ones. We remind that the goal of a body joint detector is the body joints precision.

The metrics we employed in Figure 9 to evaluate the correct detection of a single body joint is the Average Precision [9]. For each joint, it is the sum of the distances to ground truth over all frames. When it is averaged over all the joints, it is the mean Average Precision, or mAP. Like a detector precision, it accounts the rate of correct classification among all tests, with failure rate being the complement to 100%.

For body joints, we use a threshold Euclidian distance to consider a correct classification. Plotting several mAP values over the threshold yields also the interesting result of which distance is yielding which precision.

[1] http://www.wassfila.com/BPR.

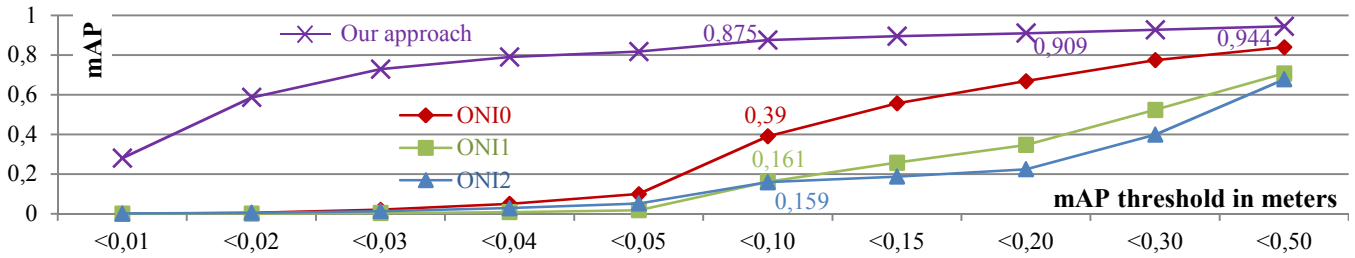


Figure 9. Comparison between our approach (BPR) and the three OpenNI/NiTE on single sensors (Sequence : IRSS35-C3). Threshold of 10 cm is standard from [9].

Our ground truth for joint position can actually only be derived from the markers of the MoCap as they obviously cannot be on the human model's real joints positions. This may have introduced uncertainties. It evaluates to an average of 4 centimeters does not exceed 7 centimeters.

The poor values we got for OpenNI's performance in Figure 9 are partly justified by our ground truth possibly differing from its own training. Our approach delivers performances reported in Figure 9 generally equivalent but for complex and self-occluded postures, surpassing the best-reported performances in [9] or [13]. The algorithm also performs the same under any angle and handles occlusion.

VII. Conclusion and Future Works

The contribution of this article is two-fold.

First, we presented a new way of exploiting a multi-Kinect, multi-Xtion or similar RGB-D sensor system. We used a voxel representation and new features to exploit the voxel data extracted from our system. We achieved critical performances under any orientation, which is of paramount importance in surveillance applications.

Second, we constituted a multiple RGB-D dataset with spatio-temporally calibrated Motion Capture ground truth. This allowed us to evaluate the previously mentioned approach's performances. In result, we compared favorably to detection algorithms that are currently available (Skeleton engine NiTE for OpenNI on each of the three Xtions).

Our future works will concern the standalone skeleton detectors. Even if their performance is limited, we will consider using a spatiotemporal filter, such as a Particle filter. We will compare that filtering with our low-level fusion presented in this paper. In addition, we will apply our approach in more complex situations. We will consider object handling then a context where multiple people are interacting together.

References

[1] D. Alfonso, "Microsoft Investor Relations - Press Release," 27 January 2011. [Online]. Available: http://www.microsoft.com/investor/EarningsAndFinancials/Earnings/PressReleaseAndWebcast/fy11/q2/default.aspx. [Accessed 25 09 2012].

[2] ASUS, "ASUS Xtion PRO LIVE," [Online]. Available: http://www.asus.com/Multimedia/Xtion_PRO_LIVE/. [Accessed 23 01 2013].

[3] B. Freedman, A. Shpunt and Y. Arieli, "Distance-Varying Illumination and Imaging Techniques for Depth Mapping". United States Patent US 20100290698 A1 , 18 November 2010.

[4] WordPress, "OpenNI | The standard framework for 3D sensing," [Online]. Available: http://www.openni.org/. [Accessed 23 01 2013].

[5] PrimeSense, "NiTE Middleware - PrimeSense," [Online]. Available: http://www.primesense.com/solutions/nite-middleware/. [Accessed 07 05 2013].

[6] PrimeSense, "Home," 2013. [Online]. Available: http://www.primesense.com/. [Accessed 07 05 2013].

[7] D. Binney and J. Boehm, *Performance Evaluation of the PrimeSense IR Projected Pattern Depth Sensor,* London, England: University College London, 2011.

[8] M. R. Andersen, T. Jensen, P. Lisouski, A. K. Mortensen, M. K. Hansen, T. Gregsersen and P. Ahrendt, "Kinect Depth Sensor Evaluation for Computer Vision Applications," Aarhus University, Aarhus, 2012.

[9] J. Shotton, A. Fitzgibbon, M. Cook, T. Sharp, M. Finocchio, R. Moore, A. Kipman and A. Blake, "Real-time human pose recognition in parts from single depth images," *Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on,* pp. 1297 -1304, 2011.

[10] T. B. Moeslund, A. Hilton and V. Krüger, "A survey of advances in vision-based human motion capture and analysis," *Computer Vision and Image Understanding,* vol. 104, no. 2-3, pp. 90 - 126, 2006.

[11] J. Deutscher and I. Reid, "Articulated Body Motion Capture by Stochastic Search," *International Journal of Computer Vision,* vol. 61, no. 2, pp. 185-205, 2005.

[12] L. Sigal, A. O. Balan and M. J. Black, "HumanEva: Synchronized Video and Motion Capture Dataset for Evaluation of Articulated Human Motion," *International Journal of Computer Vision,* vol. 87, no. 1-2, pp. 4-27, March 2010.

[13] J. Taylor, J. Shotton, T. Sharp and A. Fitzgibbon, "The Vitruvian manifold: Inferring dense correspondences for one-shot human pose estimation," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2012.

[14] L. Zhang, J. Sturm, D. Cremers and D. Lee, "Real-Time Human Motion Tracking using Multiple Depth Cameras," *Proc. of the International Conference on Intelligent Robot Systems (IROS),* 2012.

[15] R. Maloney, "Movement Analysis Products," Motion Analysis Corporation, 4 January 2013. [Online]. Available: http://www.motionanalysis.com/html/movement/products.html. [Accessed 11 January 2013].

[16] D. Herrera C., J. Kannala and J. Heikkila, "Joint Depth and Color Camera Calibration with Distortion Correction," *IEEE Transactions on Pattern Analysis and Machine Intelligence,* vol. 34, no. 10, 2012.

[17] D. Cohen-Or and A. Kaufman, "Fundamentals of Surface Voxelization," *Graphical Models and Image Processing,* vol. 57, no. 6, pp. 453-461, 1995.

[18] A. Criminisi, J. Shotton and E. Konukoglu, "Decision Forests: A Unified Framework for Classification, Regression, Density Estimation, Manifold Learning and Semi-Supervised Learning," *Foundations and Trends® in Computer Graphics and Vision,* vol. 7, no. 2-3, 2012.

[19] K. Fukunaga and L. Hostetler, "The estimation of the gradient of a density function, with applications in pattern recognition," *Information Theory, IEEE Transactions on,* vol. 21, no. 1, pp. 32-40, 1975.