Estimating human motion from multiple Kinect Sensors

Stylianos Asteriadis, Anargyros Chatzitofis, Dimitrios Zarpalas,
Dimitrios S. Alexiadis, Petros Daras
Information Technologies Institute,
Centre for Research and Technology, Hellas,
6th km Charilaou Thermi, GR-57001
Thessaloniki, Greece
{stiast, tofis, zarpalas, dalexiad, daras}@iti.gr

ABSTRACT

Human motion estimation is a topic receiving high attention during the last decades. There is a vast range of applications that employ human motion tracking, while the industry is continuously offering novel motion tracking systems, which are opening new paths compared to traditionally used passive cameras. Motion tracking algorithms, in their general form, estimate the skeletal structure of the human body and consider it as a set of joints and limbs. However, human motion tracking systems usually work on a single sensor basis, hypothesizing on occluded parts. We hereby present a methodology for fusing information from multiple sensors (Microsoft's Kinect sensors were utilized in this work) based on a series of factors that can alleviate from the problem of occlusion or noisy estimates of 3D joints' positions.

Categories and Subject Descriptors

I.2.10 [**Artificial Intelligence**]: Vision and Scene Understanding—*Motion*; I.4.8 [**Image Processing and Computer Vision**]: Scene Analysis—*Sensor Fusion*

General Terms

Experimentation, Reliability

Keywords

Kinect-based motion detection, multiple kinects, skeleton extraction

1. INTRODUCTION AND RELATED WORK

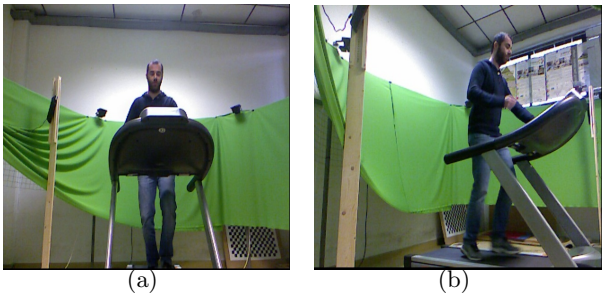
Human motion estimation is an active field of research, with a multitude of approaches following multifarious paths, both regarding the algorithms and hardware employed. Multi-camera systems, usually targeting 3D capturing, are among the most traditional methods [14]. One of the typical works is presented in [6], where motion reconstruction of freely moving humans employs Shape-from-Silhouette (SFS) for estimating shape from multiple structures. Multiple video streams are also used in [12], where the authors use partial 3D reconstructions from different cameras and translate the problem of motion parameters estimation into a minimization problem between pre-defined skin models and reconstructions. Another family of motion extraction methods relies on efficient marker-based tracking: markers can be visual or body-attached sensors (e.g. magnetic [11]). In [8], skeletal structures are acquired following a series of steps that segment markers in terms of their motion and joints' positions are inferred.

Recently, with the advent of the Microsoft Kinect sensor, a lot of attention has been focused on depth sensors. Kinect captures in real-time (30fps), and releases 2.5D data of resolution 640×480, accompanied with registered RGB data. One of the major components of the Kinect sensor, is its ability to infer human motion by extracting human silhouettes in skeletal structures. In particular, Fig. 1 shows the body parts (joints) a Kinect skeleton consists of. A lot of works have taken advantage of Kinect-based tracking for applications, mainly related to human activity recognition, interaction with objects [9], etc. However, most of these works employ single sensor solutions, especially due to interference with each other, when more than one sensors are used, resulting to erroneous or missing depth estimates. As a consequence, self-occluded body parts or conditions where part of the body is occluded by other objects are not handled.

This work is motivated by augmented reality applications, where distant users would like to interact in 3D environments performing specific sports. Thus, the scope is that one user can be indoors, (e.g. running on a treadmill), captured by Kinect sensors, and his/her reproduction (in the form of either an avatar animated by the user's actual movements or complete 3D reconstruction) is depicted to the other user. Multiple Kinect sensors are needed in this case in order to overcome the problem of (self)occlusions and produce reliable skeletons and 3D reconstructions.

Berger *et al.* [2] have studied the effect of using more than one Kinect for motion capturing and they propose those conditions under which effects of interference can be undervalued to the benefit of performance. Authors in [15] also employ a multi-Kinect solution to human performance capture. In particular, they propose a three hand-held Kinects solution for reconstructing skeletal poses in a joint optimization framework of camera parameters and human shape templates. In Caon et al. [4], the authors also employ a multiple-Kinect set up for user posture and gesture estimation, in 3D

Mirage 2013 June 06 - 07 2013, Berlin, Germany



(a) (b)

Figure 1: Example of occluded human motion. The treadmill's bars and console are severelly occluding the runner's body.

ID	joint
1	Head
2	Neck
3	Left Shoulder
4	Left Elbow
5	Left Hand
6	Right Shoulder
7	Right Elbow
8	Right Hand
9	Torso
10	Left Hip
11	Left Knee
12	Left Foot
13	Right Hip
14	Right Knee
15	Right Foot

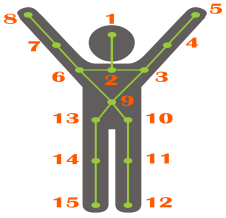


Figure 2: Joints tracked by the Kinect Sensor.

smart environments. To our knowledge, this is probably the only research work proposed for direct fusion of Kinect-based skeletons, based on confidence values: new skeletons are constructed based on weighted averages of the original ones. Weights are defined based on the ability of the sensor to return joint positions.

In the proposed work, joints' positions are inferred following the maximization of an Energy Function of randomly sampled candidate positions. Since one or more sensors may return false detections, we utilize a series of confidence values (based on expected expressivity, posture and depth measurements), focusing on positions close to actual measurements. The use of multiple Kinect sensors is targeting environments where (self)occlusions are frequent (Fig. 1) and can be utilized for robust activity recognition, 3D reconstruction of humans and other domains where human motion can give significant information.

The structure of the rest of the paper is the following: Section 2 gives details of the multiple skeleton fusing algorithm, Section 3 presents a series of experiments and the impact of different parameterizations on the final outcome. Section 4 concludes the paper.

2. METHOD OVERVIEW

Constructing a motion tracking scheme, as a combination of joints from different Kinect-based skeleton structures, or building a new skeleton, following a weighting scheme, for known applications and activities [5] would provide an efficient framework for training expected joint positions and discarding those that are far from what would be expected. For instance, knowing that a person is performing a certain sport would automatically provide with a knowledge base for training appropriate motion-related or structural models. However, estimating joints positions for unknown movements is independent from such domain knowledge. For this reason, we hereby present a local approach, following a confidence-based logic regarding expected posture, expressivity and motion history for each body part. More specifically, in the proposed work, expected 3D position $\mathbf{x}_j$ of joint j is based on previous locations, expressivity, posture and relation to each Kinect's k depth data, and is calculated following the maximization of the sum of energy functions E_j^k over a set of candidate positions $\mathbf{p}$:

$$x_j = \arg\max_{\mathbf{x}} \left(\sum_{k \in K} E_j^k(\mathbf{p}, \mathbf{x}) \right) \quad (1)$$

with K being the set of all kinect sensors. $E_j^k(\mathbf{p}, \mathbf{x})$ is a weighted combination of distance kernels of candidate positions from tracked joints, multiplied with the output of a Mamdani Fuzzy Inference Scheme (FIS). The FIS shceme has been chosen to fuse the abovementioned factors, in an effort to model a joint's probability of representing a true estimate.

2.1 Calibration of Skeletons

Human torso is the most reliably detected area, even under heavy occlusions, as it can be accurately estimated based on other features' 3D positions. Thus, it is used as a reference pattern for registering skeletons with each other, on a common coordinate system. In particular, the transformation under which triangles ($\mathbf{x}_1^k$,$\mathbf{x}_2^k$,$\mathbf{x}_3^k$) formed by the joints corresponding to the left/right shoulders and the torso joint (as named in OpenNI) of skeleton k are rigidly aligned on a reference skeleton k_0 is found [3]. Initially, matrix H is calculated, using (2), and the rotation matrix is extracted by applying Singular Value Decomposition (SVD) on H (3). Finally, the 4×4 transformation matrix T is extracted using (4). Every point on skeleton k is then transformed to the coordinate system defined by k_0 using T (in homogeneous coordinates).

$$H = \sum_{l=1}^{3} (\mathbf{x}_l^{k_0} - \overline{\mathbf{x}^{k_0}})(\mathbf{x}_l^k - \overline{\mathbf{x}^k})^T \quad (2)$$

$$[U, S, V] = svd(H) \quad (3)$$

$$T = \begin{pmatrix} I_{3\times3} & \overline{\mathbf{x}^{k_c}} \\ \mathbf{0} & 1 \end{pmatrix} \begin{pmatrix} \mathrm{VU}^T & \mathbf{0} \\ \mathbf{0} & 1 \end{pmatrix} \begin{pmatrix} I_{3\times3} & -\overline{\mathbf{x}^{k_0}} \\ \mathbf{0} & 1 \end{pmatrix} \quad (4)$$

2.2 Initialization of candidate joint positions

A uniformly sampled swarm of N possible joint positions $\mathbf{P}_j = \{\mathbf{p}_j^1 \ldots \mathbf{p}_j^N\}$ is initialized in the 3D neighborhoods $\mathbb{W}$ of all joints $\mathbf{s}_j^k$ (j denotes the joint id and k the skeleton/sensor it belongs to) 3D positions. Each candidate point is assigned the id $k \in K$ of the skeleton it was initialized with and is subsequently accordingly translated in the 3D space. In subsequent frames, each point $\mathbf{p}_j^n$ is assigned a specific weight.

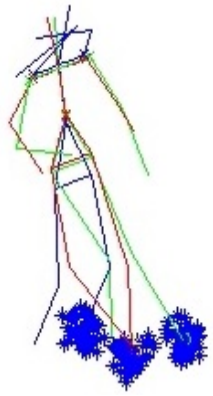


Figure 3: Initialization of candidate positions for left foot (joint id = 12) from three different Kinect Sensors.

The weight depends on two main factors. Firstly, on its distances from $\mathbf{s}_j^k$, for every $k \in K$ and, secondly, on a series of confidence values. These confidence values are modelled as non-linear functions of corresponding joints expressivity parameters, their distances on the z-axis from the corresponding values of sensor's depth map and the possibility that $\mathbf{s}_j^k$ returns natural body parts postures. The following subsections thoroughly describe the computation of these parameters.

2.3 Confidence Estimates

For each joint $\mathbf{s}_j^k$, a series of factors are taken into account and subsequently merged to produce an overall estimate of the probability that it can constitute a reliable joint position for the final skeleton. More precisely, the following confidence estimates are utilized:

2.3.1 Joint Expressivity

In those cases when a joint is occluded, positions are roughly estimated, often resulting to jerky movements, with unnaturally high amounts of overall activation. This leads to highly noisy movements, with very low amounts of fluidity, in comparison to those corresponding to non-occluded body parts. The concept behind the fluidity expressivity parameter [7] is that it models the smoothness of single gestures, seeking to capture the continuity between movements. Considering a time window of T frames, the "quantity" of movement (inverse fluidity) at time t, $F_{j,t}^k$ for $\mathbf{s}_{j,t}^k$ is computed as the standard deviation of its Overall Activation $O_{j,t}^k$ from the following equations:

$$O_{j,t}^k = \sum_{t' \in (t-T,T)} \| d\mathbf{s}_{j,t'}^k / dt \|$$
$$F_{j,t}^k = \sqrt{E[(O_{j,t-T..t}^k - \overline{O_{j,t-T..t}^k})(O_{j,t-T..t}^k - \overline{O_{j,t-T..t}^k})]} \quad (5)$$

In the following and to favour brevity, index t will only be referred to when necessary.

2.3.2 Occlusion Handling

Another confidence indicator for joints, able to report cases of possible occlusions is the error D_j^k between the z-coordinate of skeleton k joint j and the corresponding depth value reported by the Kinect sensor at that location, on the 2D depth map. These measurements were acquired by using the standard OpenNI framework [13] unprojection functions for the Microsoft Kinect Sensor.

2.3.3 Expected body part posture

Angles formed by adjacent body limbs have also been considered, as an indicator of expected poses of different body parts. To this aim, the angles corresponding to the inner product of the unit vectors $\overrightarrow{v_0}$ and $\overrightarrow{w_0}$ defining left/right leg and thigh and the inner product between the unit vectors of left/right upper and lower arm, were calculated. Expected (natural) poses for knees and elbows were between 0 and π, while unnatural angles were between π and 2π. The indicator considered here is described as the inverse probability of a body part having naturally looking postures:

$$A_j^k = 1 - P(pose|\mathbf{x}_j^k) \quad (6)$$

with *pose* being the set of expected joint angles, following a normal distribution with mean $\mu{=}\pi/2$ and standard deviation $\sigma{=}\pi/6$.

2.4 Fuzzy Inference of sensor's noise

For an overall estimate of the noise λ_j^k attributed to a kinect sensor tracking a specific joint, a function mapping F_j^k, D_j^k and A_j^k (subsection 2.3) to parameter λ_j^k is searched for. An intuitive way to model noise from the above values, is to construct a Fuzzy Inference System (FIS) with inputs F_j^k, D_j^k and A_j^k and output λ_j^k at each iteration. A FIS can model most of the continuous functions mapping n-dimensional spaces to $\mathbb{R}$. Employing FIS engines is more straightforward than typical mathematical representations, since domain knowledge can be expressed in the form of *if...then* rules. One of the most widely used types of Fuzzy Inference Systems is the Mamdani [10] Fuzzy Engine, which consists of *if-then* rules that combine degrees of memberships $\mu_r^{input_j^k}$ of different variables to fuzzy sets S, specific for every rule. Outputs $f_{r,j}^k(F_j^k, D_j^k, A_j^k)$ for each rule r are implicated based on $\mu_r^{input_j^k}$ and are further aggregated to an overall output function which is then used for calculating the crisp value of λ_j^k.

Fig. 4 illustrates the FIS architecture. Each input variable for a certain joint is normalized from 0 to 1 using its current maximum value for all sensors. This is done at each iteration and, thus, the inputs can be modelled by three fuzzy sets, namely **low**, **medium** and **high**. Fuzzy sets representing **medium** values were modelled by Gaussian membership functions (MFs), while Sigmoidals were used for **low** and **high** sets. The rules built for inference are described below:

1. if D_j^k is **low** AND F_j^k is **low** AND A_j^k is **low** then λ_j^k is **low**
2. if D_j^k is **high** OR F_j^k is **high** AND A_j^k is **high** then λ_j^k is **high**
3. if D_j^k is **medium** AND F_j^k is **low** AND A_j^k is **low** then λ_j^k is **medium**
4. if D_j^k is **low** AND F_j^k is **medium** AND A_j^k is **low** then λ_j^k is **medium**
5. if D_j^k is **low** AND F_j^k is **low** AND A_j^k is **medium** then λ_j^k is **medium**

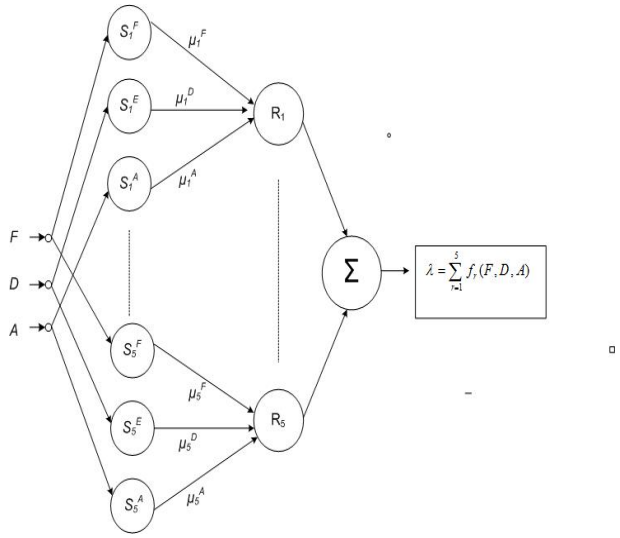


Figure 4: The structure of the FIS engine.

Inference is done using a Mamdani [10] type Fuzzy Logic System, with *min* and *max* t-norm and s-norm used for the AND and OR operators. For every Rule, implication is done using the Algebraic Product, while Algebraic Sum is employed for aggregation. Finally, defuzzification is done using the Centre of Gravity of the output. Figure 5 shows two examples of normalized inputs for a certain joint of a sensor and the noise produced, using the above described FIS.

2.5 Energy function estimates

The Energy function for each candidate joint position, $\mathbf{p}_j^n$ (n=$\{1..N\}$), at time t, is calculated using the following equation:

$$E^{\mathbf{p}_j^n} = \sum_{k \ni K} \{\lambda_j^k\}^{-1} e^{-|\mathbf{s}_j^k - \mathbf{p}_j^n|} \tag{7}$$

and is normalized with the sum of Energy Functions of the whole population of candidate positions:

$$P(j|\lambda_j^1 ... \lambda_j^K, \mathbf{p}_j^n) \propto \frac{E^{\mathbf{p}_j^n}}{\sum_{n \ni N} E^{\mathbf{p}_j^n}} = \frac{\sum_{k \ni K} \{\lambda_j^k\}^{-1} e^{-|\mathbf{s}_j^k - \mathbf{p}_j^n|}}{\sum_{n \ni N} \sum_{k \ni K} \{\lambda_j^k\}^{-1} e^{-|\mathbf{s}_j^k - \mathbf{p}_j^n|}} \tag{8}$$

After each iteration t, a roulette wheel selection scheme [1] is followed. Roulette wheel selection is a fitness-proportionate selection procedure, and has been extensively utilized in Genetic Algorithms, for the selection of parental chromosomes in future iterations. Every individual candidate position (in our case), is given a chance to breed future generations of possible positions, according to a survival probability, derived from $P(j|\lambda_j^1 ... \lambda_j^K, \mathbf{p}_j^n)$. To each individual (candidate position), a part of an imaginary roulette wheel is attributed. This part is proportional to the probability an individual has, of representing joint j; thus, more likely individual positions occupy more space on the roulette wheel. By spinning the wheel N times, a new set of possible solutions is generated and used for iteration $t+1$. Following this procedure, different candidates are selected multiple times, attributing them higher probabilities of being selected in future iterations. If there is a particularly fit candidate in the whole population, it would be expected to be more successful at producing offspring than a weaker rival. Consequently, candidate positions with high probabilities give rise to a higher number of their future instances with the algorithm, however, leaving space for less likely positions to be considered. This way, momentary positions that, in future iterations, are likely to be assigned higher probability, are kept in the cycle of possible positions and are only discarded after a long number of iterations, during which they exhibited continuously low probabilities of being considered as significant candidates. The algorithm converges to a few candidates after a number of iterations, which give the final joint position, through averaging.

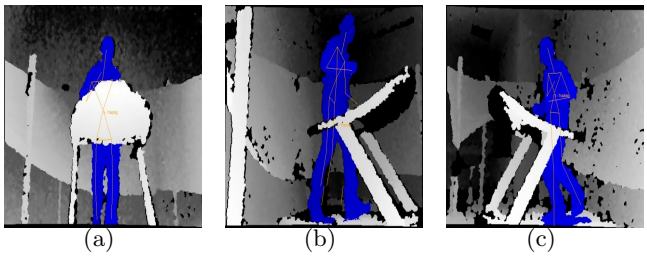


Figure 6: Sequence A: User running on a treadmill with a lot of occlusions.

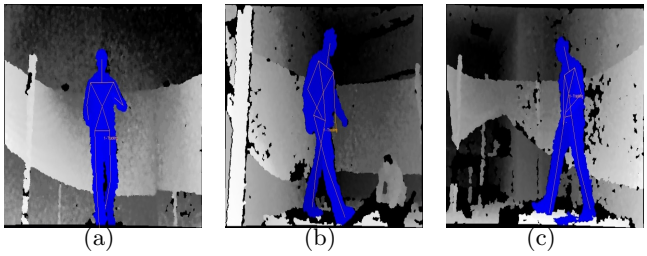


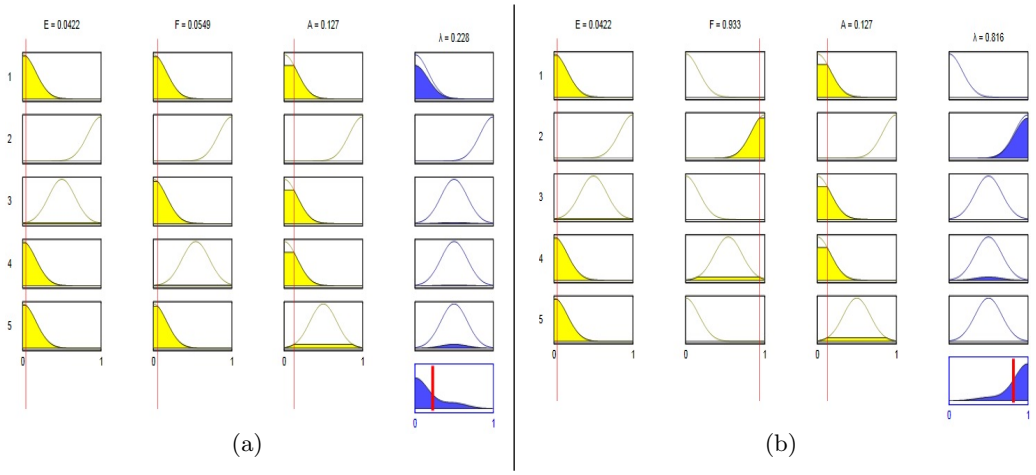
Figure 7: Sequence B: User running on a treadmill with self occlusions.

2.6 Adaptive re-initialization of candidate positions

The algorithm can be re-initialized multiple times, either at frequent intervals or adaptively. When significant body rotations occur, a threshold can be used for re-initializing candidate positions. In this way, candidate positions converge around the proper sensor in a scalable and motion-dependent manner. Employing the above step increased accuracy, as will be seen in the next Section.

3. EXPERIMENTAL RESULTS

For evaluating the proposed technique, two experiments with occlusions caused by objects and self occlusions were conducted: Two sequences (A and B) of a person on a treadmill were recorded. In each case, the user was asked to make rotational movements of about 45 degrees on his left and right, before starting to run on the treadmill. The difference between the two sequences is that the treadmill's console and handles were removed in the second case. Typi-



(a) (b)

Figure 5: Two examples of FIS input-output for a specifc joint. Rows correspond to the five rules of FIS, while the first three columns in both (a) and (b) to the three inputs (confidence values). The fourth column depicts the rule-based and the overall output. In (a) all inputs for the specific sensor have low values in comparison to the rest of the sensors, thus, resulting to a low noise for the specific Kinect. In contrast, figure (b) shows an instance where one input being high, results to a significantly noisy output λ_j^k.

Table 1: Successful selection of sensor for different re-initialization cycles

	t=50	t=100
Sequence A	87.5%	83.3%
Sequence B	91.7%	95%

cal instances are depicted in Fig. 6 and Fig. 7, respectively. Human motion was recorded with three Kinect sensors, with the one positioned in front of the user and the other two on his left and right. All three sensors had equal distances from the user. Sequence A is more challenging than B, as both self occlusions and occlusions of the hands, caused by the treadmill console and handles, take place. Table 1 shows the percentage of convergence to correct estimates of hands positions, for different numbers of iterations before re-initializing. Convergence of the majority of the population of candidate positions, to a Kinect sensor a priori known to be tracking the corresponding joint correctly, was considered as a successful one.

Fig. 8 shows the distribution of candidate positions along the three sensors used for Sequence A, at estimating the position of left and right hand, for 300 consecutive frames. As can seen in Fig. 6, the central Kinect sensor has to be rejected for both hands. However, some candidate positions coming from the right sensor, when tracking left hand do survive, although the left sensor is the one that prevails. The reason is that the hand is only partially occluded from this sensor and, thus, related information can still reconstruct motion reliably. For a population equal to $N = 2100$, the algorithm converges after about 20-30 iterations.

For increasing the robustness of the system, experiments were carried out with the algorithm re-launching when the silhouette is rotated significantly with regards to last initialization. Thus, a threshold of silhouette rotation is defined, for declaring the possibility that more confidence should be placed on a new sensor for tracking a joint. Considering a threshold equal to $pi/3$, an average of 85 and 103 iterations resulted for the parts of Sequence A and B, respectively, when the user was asked to turn right and left. Success rates have increased, managing to converge to correct positions, due to the ability of the system to re-launch and end up to the most appropriate sensor for a specific joint, whenever large body rotations occur.

In the experiment, candidate positions were set to $N = 2100$, while a spherical 3D area of radius equal to 100 millimeters was used for initializing candidate positions around each joint. Standard deviation of all membership functions of the Fuzzy Inference system was set to σ=0.3.

4. DISCUSSION AND CONCLUSIONS

Estimating human motion in non-intrusive environments is a crucial component of activity recognition, while it can act supportively for the transmission of 3D information over the network, 3D reconstruction of humans, etc. Microsoft's kinect sensor comes with a built-in mechanism of motion estimation, in the form of human skeletons. However, every day applications usually require that partial occlusions, caused either by objects or body posture, are to be taken into consideration. Moreover, noisy estimates of non-occluded parts can result in unnatural structures or gestures. The proposed work described a method for fusing, in a user and activity-agnostic manner, factors that distinguish among sensors those that deliver reliable information for joint position tracking. Future work will concentrate on training user and activity dependent models. Expressivity and posture vary among different users and activities and, thus, building proper models is expected to increase accuracy. Moreover, the system's ability to be part of a skeleton-based 3D reconstruction framework, under heavy movements and occlusions will be examined, while accurate motion and skeletal information is expected to play a key role in virtual environments, for a realistic mapping of human's actions on virtual worlds.

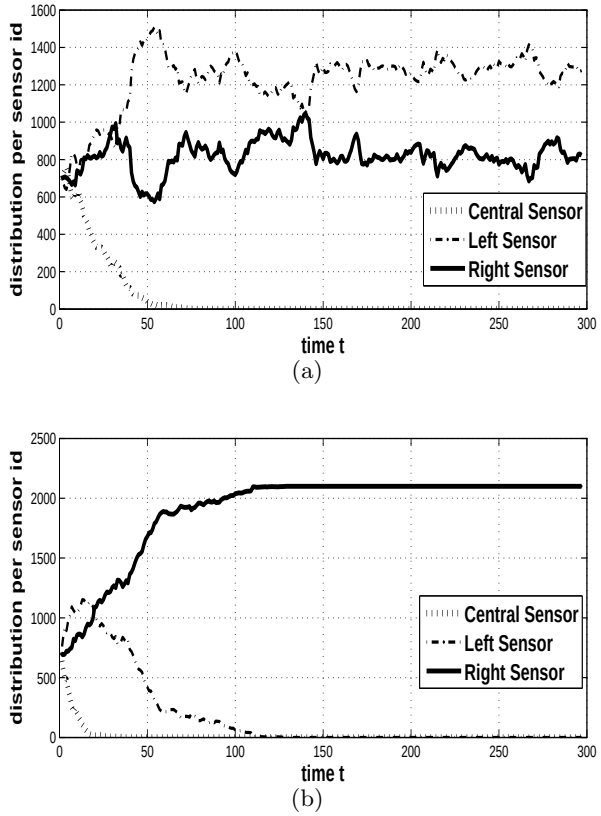


Figure 8: Distribution of candidate positions for (a) left and (b) right hand along three different sensors.

5. ACKNOWLEDGMENTS

This work was supported by the EU funded project 3DLIVE, GA 318483.

6. REFERENCES

[1] J. E. Baker. Reducing bias and inefficiency in the selection algorithm. In *Proceedings of the Second International Conference on Genetic Algorithms on Genetic algorithms and their application*, pages 14–21, 1987.

[2] K. Berger, K. Ruhl, Y. Schroeder, C. Bruemmer, A. Scholz, and M. A. Magnor. Markerless motion capture using multiple color-depth sensors. In *Proceedings of Vision, Modeling and Visualization (VMV) 2011*, pages 317–324, 2011.

[3] P. J. Besl and N. D. McKay. A method for registration of 3-d shapes. *IEEE Trans. Pattern Anal. Mach. Intell.*, 14(2):239–256, Feb. 1992.

[4] M. Caon, J. Tscherrig, E. Mugellini, O. A. Khaled, and Y. Yue. Context -aware 3d gesture interaction based on multiple kinects. In *First International Conference on Ambient Computing, Applications, Services and Technologies (AMBIENT)*, pages 7–12, 2011.

[5] A. A. Chaaraoui, P. Climent-Pérez, and F. Flórez-Revuelta. A review on vision techniques applied to human behaviour analysis for ambient-assisted living. *Expert Syst. Appl.*, 39(12):10873–10888, Sept. 2012.

[6] K. Cheung, S. Baker, and T. Kanade. Shape-from-silhouette of articulated objects and its use for human body kinematics estimation and motion capture. In *Proceedings of IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR)*, volume 1, pages 77–84, June 2003.

[7] B. Hartmann, M. Mancini, and C. Pelachaud. Implementing expressive gesture synthesis for embodied conversational agents. In *Proceedings of the 7th International Gesture Workshop*, pages 188–199, 2006.

[8] A. G. Kirk, J. F. O'Brien, and D. A. Forsyth. Skeletal parameter estimation from optical motion capture data. In *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 782–788, 2005.

[9] H. S. Koppula, R. Gupta, and A. Saxena. Human activity learning using object affordances from rgb-d videos. *CoRR*, abs/1208.0967, 2012.

[10] E. H. Mamdani and S. Assilian. An experiment in linguistic synthesis with a fuzzy logic controller. *Int. J. Hum.-Comput. Stud.*, 51(2):135–147, Aug. 1999.

[11] J. F. O'Brien, R. E. Bodenheimer, Jr., G. J. Brostow, and J. K. Hodgins. Automatic joint parameter estimation from magnetic motion capture data. In *Proceedings of Graphics Interface*, pages 53 –60, 2000.

[12] K. Ogawara, X. Li, and K. Ikeuchi. Marker-less human motion estimation using articulated deformable model. In *Proceedings of the IEEE International Conference on Robotics and Automation*, pages 46 –51, 2007.

[13] OpenNI organization. *OpenNI User Guide*, November 2010. Last viewed 19-01-2011 11:32.

[14] T. Popham. *Tracking 3D Surfaces Using Multiple Cameras: A Probabilistic Approach*. PhD thesis, University of Warwick, November 2010.

[15] G. Ye, Y. Liu, N. Hasler, X. Ji, Q. Dai, and C. Theobalt. Performance capture of interacting characters with handheld kinects. In *Proceedings of the 12th European Conference on Computer Vision (ECCV)*, 2012.