

© 2015 IEEE. Personal use of this material is permitted. Permission from IEEE must be obtained for all other uses, in any current or future media, including reprinting/republishing this material for advertising or promotional purposes, creating new collective works, for resale or redistribution to servers or lists, or reuse of any copyrighted component of this work in other works.

# Performance Analysis of Self-Organising Neural Networks Tracking Algorithms for Intake Monitoring Using Kinect

Samuele Gasparri\*, Enea Cippitelli\*, Ennio Gambi\*, Susanna Spinsante\*  
and Francisco Flórez-Revuelta+

\*Dipartimento di Ingegneria dell'Informazione, Università Politecnica delle Marche,  
Ancona, Italy I-60131 E-Mail: {s.gasparri, e.cippitelli, e.gambi, s.spinsante}@univpm.it

+Faculty of Science, Engineering and Computing, Kingston University,  
Penrhyn Road, Kingston upon Thames KT1 2EE, UK; E-Mail: F.Florez@kingston.ac.uk

**Keywords:** Intake Monitoring, Behaviour Analysis, Kinect, Point Cloud, Tracking Algorithms.

## Abstract

The analysis of intake behaviour is a key factor to understand the health condition of a subject, such as elderly or people affected by diet-related disorders. The technology can be exploited for this purpose to promptly identify anomalous situations. To this end, the point cloud, provided by a depth camera placed on the ceiling in top-down view, is used as input to three self-organising algorithms. The output are three different models that represent the monitored person during intake activities. Starting from these models, the nodes representing the head and the hands are selected. They are useful to identify most of the actions performed by the person while having a meal. In the experimental section, the positions of these nodes are compared with a ground truth and the performance of the proposed algorithms are evaluated in terms of distance error.

## 1 Introduction

People who do not follow a correct diet can undergo some very problematic consequences. One of these is obesity, as stated by the World Health Organization: in 2014, more than 1.9 billion adults were overweight and 600 million were obese [1]. Water assimilation habit is another key factor determining a healthy lifestyle. A quantity of 1.25 liters for men and 0.75 liters for women is the minimum daily suggested amount of water to reduce the risks of stroke, kidney failure or even breast cancer [2]. These serious diseases must be taken into account with more attention if the person is older. Indeed, elderly are more vulnerable due to some issues correlated to their age, such as the decreasing thirst sensation. Technology improvements in the last years have allowed the prevention and detection of possible wrong dietary habits. In particular, the living room or the kitchen in a house, where usually the principal meals are had, can become smart environments, where the feeding habits of the subject may be monitored. For example, a RGB camera

can provide information regarding to the volumes of food in a meal and the associate calories [3]. Using a depth stream, the dependency on RGB can be overcome. For this reason, in this work we adopt the Kinect sensor alone, to generate the depth streams processed by different machine learning algorithms. A data-fusion approach applied to food intake monitoring is presented in [4] where the information provided by Kinect and on-body inertial sensors have been analysed for 15 participants over 5 meals. The system, to which the analysis presented in this paper apply, is conceived to be as much unobtrusive as possible. To this aim, the Kinect sensor is installed on the ceiling to get a top view and only the depth stream is exploited. This way, at the same time, the visual and physical impact of the system in the room is strongly reduced and the privacy of the subject is preserved. Using this configuration, it is not possible to exploit the skeleton algorithm provided by Microsoft [5] to track people's movements, so other solutions, which directly use the depth stream, are necessary. A tracking systems based on two particle filters has been proposed by *Migniot et al.* [6]. The same idea is used by *Bednařk et al.* in [7], with good results in terms of performance on a dataset of 10 people. Unfortunately, the system requires a starting pose phase to identify the anthropometric parameters of the person.

In this work, the use of the depth stream provided by the Kinect from a top view, in combination with three different unsupervised self-organizing algorithms, has been studied, for the first time, to monitor the movements performed by a person during food intake activities. By experimental comparison, the best performing algorithm is identified, and deemed suitable for food intake monitoring in the proposed system configuration.

The paper is organized as follows: Section 2 describes the pre-processing phase to calculate the PC from the depth frame and the person's orientation. In Section 3, these information is exploited as input to the algorithms used to track the movements performed by the subject. The performance comparison is presented in Section 4. Finally, Section 5 draws the main conclusions of the work.

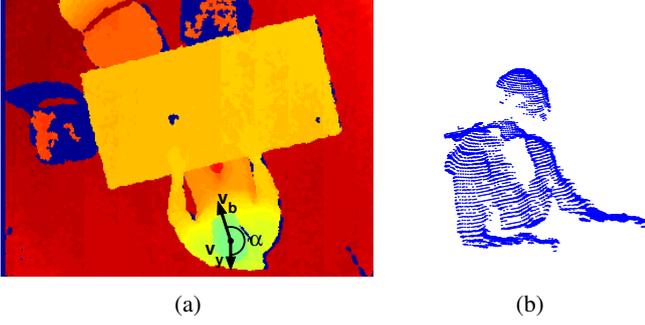


Figure 1: Raw depth frame provided by the Kinect sensor (a). Point cloud calculated from the blob of the person (b).

## 2 Processing of the depth frame

The sensor used in this work is the Kinect V1, placed on the ceiling at a height of 3 m from the floor. It provides five types of streams: depth, infrared, colour, skeleton and audio. The proposed algorithms exploit only the first input.

Figure 1a shows a depth frame where cold colours are used to identify the closer surfaces to the sensor and the warm ones for the farther objects. When the colour is blue, it means that the sensor has not been able to estimate the distance for that specific pixel. In particular, in the scene a person is present, sitting by the side of the table. The blob of the person is identified with the same approach for foreground detection used in [8]. The arrow labelled as  $(v_b)$  indicates the body orientation of the person while  $(v_y)$  is the standard direction.  $\alpha$  is the angle between the two vectors and it will be used in all the tracking solutions described in the next section. The direction of the person is automatically found exploiting geometric characteristic of the table and the hypothesis that the subject is sitting parallel to one side of the table.

Figure 1b shows the PC calculated from the depth blob using the depth camera's intrinsic parameters. The PC, calculated for each frame provided by the sensor, is the input data to the three algorithms described in the next section.

## 3 Self-organizing algorithms for intake monitoring

A self-organizing neural network is composed of  $D$  nodes. Each node  $n_i$  ( $i = \{1, 2, \dots, D\}$ ) has associated a reference vector  $(w_i)$  belonging to the input space. Nodes are connected by edges establishing the topology of the network. In this work, the goal is to track the human upper body (head and arms) with different algorithms to obtain the position of three joints: head ( $J_{hd}$ ), left hand ( $J_{hl}$ ) and right hand ( $J_{hr}$ ). In order to do that, the input space is the  $3D$  manifold formed by the coordinates of the points in the PC. The topology establishes a neighbourhood relation between nodes, where  $N(n_i)$  is equal to all the neurons connected to  $n_i$  by an edge.

### 3.1 Self organizing map

The first algorithm tested to track the movements of the person is the Self Organizing Map (SOM). It is an unsupervised machine learning algorithm proposed by *Kohonen* [9]. The main characteristic of SOM is that both the number of nodes and their connectivity are pre-established and fixed throughout learning. Figure 2a shows the adapted model to the person's PC. A single node ( $J_{hd}$ ) represents the head while two left and right linear SOMs model the arms, with  $(J_{hl})$  and  $(J_{hr})$  set as the end nodes.  $J_{hd}$  must be disconnected from the other nodes, otherwise it can be attracted by them and shifted to the chest's PC. Comparing Figure 1b and Figure 2a some points in the PC have been deleted. In particular, the chest must be discarded from the PC as input to the SOM because the model does not take into account this part of the body. Points in the chest are those inside a bounding box, placed in the centre of the chest, whose dimensions are proportional to the fitted model. Without this pre-processing phase, the body of the person can attract some nodes and consequently, the model would not be able to track the subject's movements.

Prior to use the SOM algorithm, the model must be rotated from the standard direction  $(v_y)$ , to the specific one exhibited by the person  $(v_b)$ . Because of this, in the first useful frame, the  $3D$  initial model is centred in the PC, in particular,  $J_{hd}$  is set to the highest point in the manifold of Figure 1b. All the other nodes in the initial model are then rotated using Equation (1) to the final position  $(w_i^{rot})$ .

$$\begin{aligned} w_i^{rot_x} &= \cos \alpha (w_i^x - J_{hd}^x) - \sin \alpha (w_i^y - J_{hd}^y) + J_{hd}^x \\ w_i^{rot_y} &= \sin \alpha (w_i^x - J_{hd}^x) - \cos \alpha (w_i^y - J_{hd}^y) + J_{hd}^y \\ w_i^{rot_z} &= w_i^z \end{aligned} \quad (1)$$

Now, it is possible to use the SOM to adapt the rotated model to the PC. This rotation stage is only carried out for the first frame of a sequence. For latter frames, the last trained model is used as initial configuration. In detail, the algorithm randomly selects one  $3D$  point in the manifold per time, until all the elements are evaluated. At each single step ( $p$ ), the SOM finds the closest node in the model to the specific input point ( $\xi$ ), using the Euclidean distance (*competition phase*) as evaluation metric. This specific node ( $w_{BMU}$ ) is called best matching unit (BMU). Equation (2) is used to update the position of the BMU. The node is attracted by  $\xi$ , and the displacement, weighted by the  $\varepsilon^p$  factor, is proportional to the distance between them.

$$\Delta w_{BMU} = \varepsilon^p (\xi - w_{BMU}) \quad (2)$$

The weight factor of the BMU is defined as:

$$\varepsilon^p = \varepsilon_i \left( \frac{\varepsilon_f}{\varepsilon_i} \right)^{p/p_{max}} \quad (3)$$

where  $\varepsilon_i$  and  $\varepsilon_f$  are respectively the initial and final values for the learning rate, while  $p_{max}$  is the total number of points in the PC as input to the SOM.  $\varepsilon^p$  is used to reduce the displacement of  $w_{BMU}$  to  $\xi$  in order to have a network that, during the

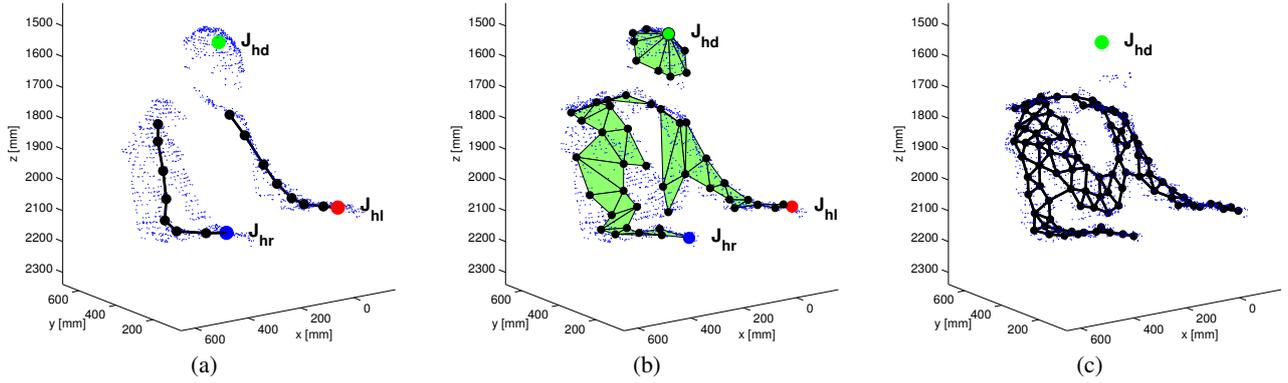


Figure 2: Results after the training of SOM (a), SOM.Ex (b) and GNG (c) algorithms.

training phase considers the whole manifold and not only the last input point.

In addition, also the neighbouring neurons of the BMU are moved. Equation (4) shows the update rule for  $N(w_{BMU})$ , where the main relevant difference respect to Equation (2) is that the weight factor is  $\varepsilon_N^p$ , with usually  $\varepsilon_N^p < \varepsilon^p$ .

$$\Delta w_i = \varepsilon_N^p (\xi - w_i), \quad \forall n_i \in N(n_{BMU}) \quad (4)$$

Sometimes, when one of the hands, for instance the left one, is very close to the face of the person,  $J_{hl}$  can be attracted too close to the head's PC. As consequence, this joint remains associated to the head even after the arm moves away. Equation (5) forces  $J_{hl}$  to stay close to its neighbours ( $N(J_{hl})$ ), and it is enabled only when the distance between these nodes exceeds a specific value. A similar process is performed with the right hand  $J_{hr}$ .

$$J_{hl} = N(J_{hl}) + \gamma \frac{J_{hl} - N(J_{hl})}{\|J_{hl} - N(J_{hl})\|_2} \quad (5)$$

Table 1 shows the final configuration parameters. Exhaustive tests have been implemented to identify the correct values for each one of them, with the aim to have a fast adaptation of the model to the PC.

Net. size	$\varepsilon_i$	$\varepsilon_f$	$\varepsilon_N^p$	$\gamma$
17	0.002	0.001	$0.5\varepsilon^p$	0.005

Table 1: Selected parameters for the SOM algorithm.

### 3.2 Extended SOM

An alternative to the SOM, called *Extended SOM* (SOM.Ex), used as tracking solution, has been recently proposed [10]. In the original SOM, the nodes have been used as the only fundamental components of the network. This feature can lead to an unstable model if the input to the SOM is a 3D space such as the PC. The most important innovation of the SOM.Ex is the introduction of new fundamental elements (segments and

planes) in the structure of the network. A segment is a pair of nodes connected by an edge while a plane is a group of three nodes placed in the vertices of a triangle. The equations used to train the model are very similar to Equations (2,3,4) but before using them, it is necessary to calculate the projection point of the input ( $\xi$ ) to these new elements. The *competition phase* will find the BMU between all the nodes and the projection points.

The evaluated network is characterized by 50 nodes, for a total of 47 planes, 9 of them are used to model the head. The trained model is visible in Figure 2b. The head cluster is separated from the torso, in this way the model is more similar to the distribution of the PC. The correct number of nodes has been tuned after a series of different tests with the aim to have an accurate tracking of the movements. Segments have not been used because they are not stable enough to represent in the correct way the different positions of the arm. A direct comparison with the previous SOM network highlights a factor 3 increase in the number of nodes. The chains, previously used in the SOM to model the arms, are now merged into a single group. Finally, the pre-processing phase to delete chest's PC is no longer necessary, because now it is considered in the model. The first time the SOM.Ex is used, the network must be correctly centred/rotated exploiting the same operations used for the SOM model, as described in Equation (1). For the next PCs, the model trained in the last frame is used as input to the SOM.Ex. In Table 2 the characteristic parameters used for the SOM.Ex are visible. The PC is under-sampled by a factor equal to 10, it allows reducing the computational time.  $\varepsilon_i$  and  $\varepsilon_f$  have the same meaning as the SOM parameters. They are two order of magnitude higher than the corresponding values used for the previous algorithm. This is justified by the greater number of nodes that now have less probability to become a BMU.

Net. size	PCStep	$\varepsilon_i$	$\varepsilon_f$	$\varepsilon_N^p$
50	10	0.5	0.4	$0.1\varepsilon^p$

Table 2: Selected parameters for the SOM.Ex algorithm.

### 3.3 Growing Neural Gas Network

The Growing Neural Gas Network (GNG) was proposed by *Fritzke* [11] based on the work of *Martinetz et al.* [12] with the *neural gas network* (NG). The most important differences with the previous two algorithms are:

- The pre-defined model is not required, only the start position of the first two nodes must be randomly chosen in the input space. In view of this fact, in the initial phase of the algorithm, it is not necessary to rotate the model according to the subject’s position;
- A local error is associated to each node: when a new node must be added to the network, it is used to find the best position that reduces the global error;
- The number of nodes is not constant (limited by an upper bound), they can be added or deleted during the learning phase;
- Connections can be added/removed between nodes. In particular, for each step of the algorithm, the two closest nodes to  $\xi$  are identified (BMUs), and a connection is created if it is not yet present. An edge is deleted if its age exceeds a threshold. These features are an advantage with respect to the SOM, because now the network is more adaptable to the topology of the PC;
- The learning coefficients are constant, while in the SOM they decrease when the time grows.

The PC is pre-processed before using the algorithm. This is necessary to simplify the final part of the system, when the hand nodes must be identified. In particular, the head’s points, selected by Equation (6), are deleted.

$$\|PC_{hd}(t) - J_{hd}(t - 1)\| < th_{hd} \quad (6)$$

$PC_{hd}(t)$  represents the points of the head in the current frame,  $J_{hd}$  is the head joint calculated from the mean of all the elements in  $PC_{hd}$ .  $th_{hd}$  is a distance threshold used to find the head points at frame  $t$ , based on the position of  $J_{hd}$  in the previous frame. For the first PC,  $J_{hd}$  is equal to the highest point in the input space. Comparing the PC in Figure 1b and Figure 2c, the deleted points that belong to  $PC_{hd}$  can be easily identify. Table 3 summarizes the parameters used for the GNG. In detail, *Net. size* is the maximum number of nodes in the network. The value 100 represents a good trade-off between resolution of the network and computational time.  $\epsilon_w$  and  $\epsilon_n$  are the learning parameters used to weight the displacement of the winner nodes.  $a_{max}$  is the maximum age allowed for a connection before deleting it, while  $\lambda$  is the rate whereby the new nodes are added in the network. In this case, a new node is placed in the model every 10 inputs.  $\beta$  is the coefficient used to reduce the local error for each node to avoid an overflow in the variables. Finally,  $\alpha$  is exploited to reduce the errors of the two closest points to a new node so, the next element will be not placed in the same area. The network output from the GNG is shown in Figure 2c.

<i>Net. size</i>	$\epsilon_w$	$\epsilon_n$	<i>a_max</i>	$\lambda$	$\alpha$	$\beta$
100	0.1	0.01	15	10	0.5	0.0005

Table 3: Selected parameters for the GNG algorithm.

The original GNG has been modified to fulfil some constraints of the application. In Figure 3a a top projection of the GNG model after the training is shown. The square represents the specific input, while the triangles placed in the hands are the two BMUs to  $\xi$ .

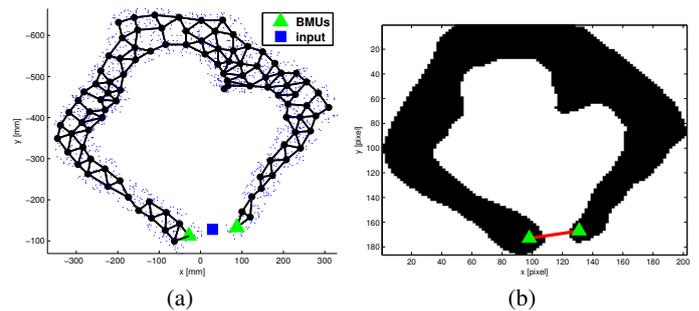


Figure 3: Trained network adapted to the PC (a), corresponding blob in the depth frame (b).

The original GNG [11] forces the creation of new connections between nodes. As a consequence, the topology of the body may not be preserved because there is an empty space between the hands. The information of the manifold must be exploited to find if the topology preservation is verified, as stated by *Flórez et al.* in [13]. In this case, two alternatives are possible:

- Check in the PC if the density of the points between the BMUs is enough to allow a new connection;
- Exploit the depth frame to assess if the connection is inside the blob (Figure 3b).

Comparing the two solutions, the second one is less costly in terms of number of operations to perform. Indeed in the first case, all the elements in the PC must be evaluated to identify the points in the region between the BMUs. In the other case, it is sufficient to control that the line between the BMUs in Figure 3b is inside the blob. As visible in the figure, this is not true and the connection is avoided. When the training phase is completed, it is possible to find the nodes that model the hands. Unfortunately, during the training phase of the model with different PCs, the position/number of connections of a specific point inside the network can change. For example, a node close to the hand position, in the next networks can move near the chest area, or even be deleted by the GNG. Hence, an appropriate post-processing algorithm has been implemented to recognize the hand joints from all the nodes in the networks.

As visible, in Figure 4a, the network output from the GNG has been rotated in the standard direction, reversing Equation

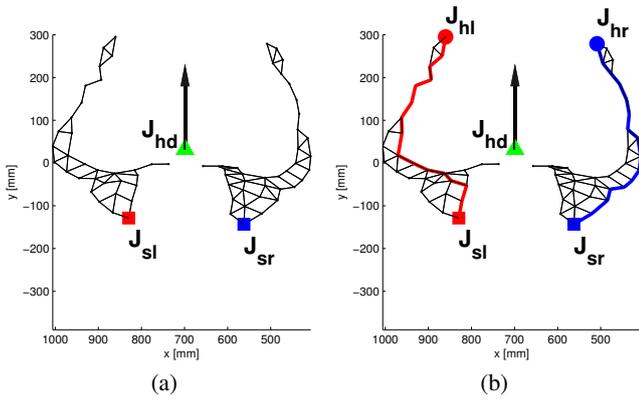


Figure 4: Algorithm used to find the hand nodes inside the network. Identification of the shoulder (a) and hand joints (b).

(1). Consequently, the network will be parallel to the  $y$ -axis, irrespective of how the person is oriented relative to the camera. The triangle highlights the joint  $J_{hd}$  calculated with Equation (6). The shoulder joints are selected respectively as the left/right node to  $J_{hd}$  with the lowest  $y$  coordinate from all the nodes in the network. These two joints ( $J_{sl}/J_{sr}$ ) are represented in Figure 4a by two squares. Assuming that the hand joints are the furthest points from the corresponding shoulder joints, the well-know *Dijkstra's* algorithm [14] can be exploited to find the appropriate nodes (Figure 4b).

## 4 Experimentation

All the algorithms previously described have been tested with a database of food intake activities [15]. The movements are performed by 35 different people, aged between 22 – 38 years and height in 1.62 – 1.97m, for a total amount of 48 tests. For each test, the volunteer enters in the scene and sits close to one side of the table. On the table are present a dish, some cutlery and a glass. No restrictions are imposed and the users are free to perform the intake movements in the way preferred. To promote the comparison between our results and those achieved by other research groups, all the networks coordinates for each one of the three algorithms and the ground truth are publicly available. For each frame, when a trained model is available, it is possible to evaluate the position of the three most important joints:  $J_{hd}$ ,  $J_{hl}$  and  $J_{hr}$ . These values are useful to identify most of the actions performed by the person during a meal. For example, the distances  $|J_{hd} - J_{hl}|$  or  $|J_{hd} - J_{hr}|$  are used to find when one hand is very close to the face. Alternatively, when some objects are present on the table, it is possible to monitor the interaction between them and the person. In order to find the algorithm that has the best accuracy, it is necessary to compare them with a ground truth. In particular, using a frame rate of 15 fps, for each useful PC, the position of  $J_{hd}^{GT}$ ,  $J_{hl}^{GT}$  and  $J_{hr}^{GT}$  has been manually assigned. When an area of interest is not visible, for example due to an occlusion, the joint is not labelled. The tracking process with each of the 3 models has been applied to the 48 sequences in the dataset, and the

error in Equation (7), i.e. the distance between the joint and the ground truth, has been calculated for each frame in which the network is available. For each joint, the mean ( $\mu$ ) and the standard deviation ( $\sigma$ ) have been evaluated.

$$err_{pos}^{alg} = |J_{pos}^{alg} - J_{pos}^{GT}| \quad pos = \{hd, hl, hr\} \quad (7)$$

$$alg = \{SOM, SOM\_Ex, GNG\}$$

Figure 5 shows a portion of the complete error sequence for each one of the three joints. For  $J_{hd}$ , the curves do not differ considerably; this is due to the position of the head that, for the entire duration of the test, does not change significantly. For the left hand, the error for the SOM algorithm is bigger than the other two, with a peak in the first part due to the initial adapting phase of the algorithm. A similar peak is present for the right hand and in this case, the differences between SOM\_Ex e GNG are not so noticeable. Table 4 presents some statistic for the three joints. In all the situations, the GNG algorithm has a lower error than the SOM-SOM\_Ex. It can be justified with the higher number of nodes in the GNG, and with its better adaptation capabilities. In the SOM\_Ex results, the right hand has a significant worse accuracy than the left hand because in four tests the algorithm does not fit the model to the PC in the proper manner and, as consequence,  $J_{hl}$  is far from the right hand. This problem is less noticeable for the left hand. The statistical proof that there is a considerable difference in the error values is given using the ANOVA test (Analysis of Variance). In particular, the test has been applied to each joint, comparing the following pairs ( $err_{pos}^{GNG}, err_{pos}^{SOM}$ ), ( $err_{pos}^{GNG}, err_{pos}^{SOM\_Ex}$ ) and ( $err_{pos}^{SOM}, err_{pos}^{SOM\_Ex}$ ). Only for ( $err_{hd}^{GNG}, err_{hd}^{SOM}$ ) it is not possible to assert a significant difference.

## 5 Conclusion

In this work, three different unsupervised machine learning algorithms have been used to track person's movements during the food intake actions. After the training session, using in input the PC obtained from the depth frame, the network provided by the SOM and SOM\_Ex can be directly used to track the hand movements. This is due to the predefined networks for these two algorithms, where the nodes that model this body parts do not change during all the tests. This is not true for the GNG where a post-processing algorithm is necessary to identify the hand joints. The algorithms have been implemented in C++ language and tested on a PC with Intel i5 8GB RAM. The average processing time for the SOM is below 30 ms per frame while for SOM\_Ex and GNG, it is close to 60 ms. The GNG outperforms the other two algorithms in terms of accuracy of the tracking. Future works include enriching the dataset and the use of these networks to train supervised machine learning algorithms for the automatic recognition of intake activities.

## Acknowledgements

This work was supported by a STSM Grant from COST Action IC1303 AAPELE - Architectures, Algorithms and Platforms for Enhanced Living Environments.

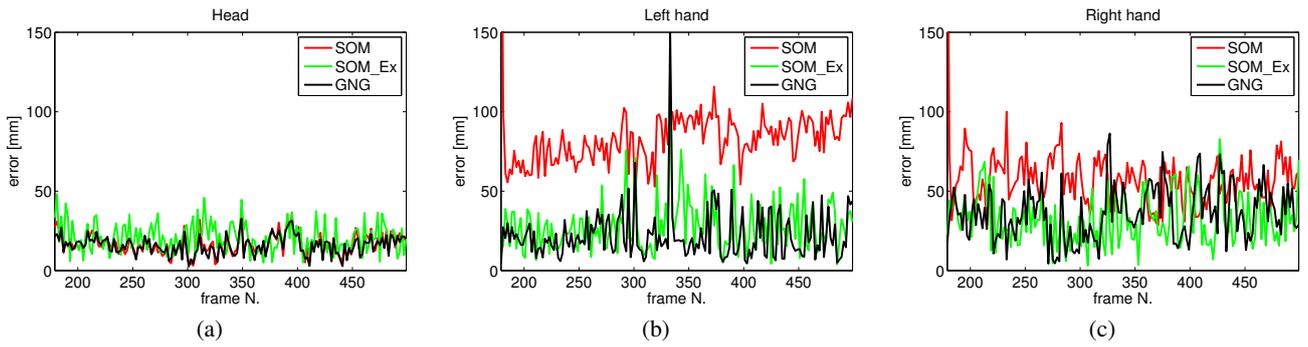


Figure 5: Error trend for the head (a), left hand (b) and right hand (c) joints.

<i>Joint \ Alg.</i>	SOM		SOM.Ex		GNG	
	$\mu$ [mm]	$\sigma$ [mm]	$\mu$ [mm]	$\sigma$ [mm]	$\mu$ [mm]	$\sigma$ [mm]
<i>Head</i>	33.3	12.4	38.4	28	32.2	13
<i>Left hand</i>	64.8	43.7	42.3	49	37	38.1
<i>Right hand</i>	66.35	35.13	78.2	148.8	36.9	49.2

Table 4: Mean and standard deviation for distance errors.

## References

- [1] <http://who.int/mediacentre/factsheets/fs311/en/>. accessed October 2015.
- [2] G. Howard and J. Bartram. “Domestic water quantity, service level, and health”. World Health Organization Geneva Vol. 1., 2003.
- [3] J. Dehais, S. Shevchik, P. Diem, and S.G. Mougiakakou. “Food volume computation for self dietary assessment applications”. In *Proc. IEEE 13th Int. Conf. Bioinformatics and Bioengineering (BIBE), 2013*, pages 1–4, November 2013.
- [4] V.D. Kakra, N.P. van der Aa, L.P.J.J. Noldus, and O. Amft. “A Multimodal Benchmark Tool for Automated Eating Behaviour Recognition”. In *Proc. Int. Conf. Measuring Behavior 2014, Wageningen, Netherlands, August 2014*.
- [5] 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 a single depth image”. In *CVPR. IEEE*, June 2011.
- [6] C. Migniot and F. Ababsa. Hybrid 3d-2d human tracking in a top view. “*Journal of Real-Time Image Processing*”, pages 1–16, 2014.
- [7] J. Bednařík and D. Herman. “Human gesture recognition using top view depth data obtained from kinect sensor”.
- [8] S. Gasparri, E. Cippitelli, S. Spinsante, and E. Gambi. “A depth-based fall detection system using a kinect<sup>®</sup> sensor”. *Sensors*, 14(2):2756–2775, Feb. 2014.
- [9] T. Kohonen. “*Self-Organizing Maps*”. Springer, 1997.
- [10] F. Coleca, A. State, S. Klement, E. Barth, and T. Martinetz. “Self-organizing maps for hand and full body tracking”. *Neurocomputing*, 147:174–184, 2015.
- [11] B. Fritzke. “A growing neural gas network learns topologies”. In *Advances in Neural Information Processing Systems 7, NIPS Conference, Denver, Colorado, USA, 1994*, pages 625–632.
- [12] T. M. Martinetz and K. J. Schulten. “A “neural gas” network learns topologies”. In *Proc. Int. Conf. on Artificial Neural Networks 1991* Espoo, Finland,.
- [13] F. Flórez, J.M. García, J. García, and A. Hernández. “Geodesic topographic product: An improvement to measure topology preservation of self-organizing neural networks”. In *Advances in Artificial Intelligence - 2004*, volume 3315, pages 841–850.
- [14] E. W. Dijkstra. “A note on two problems in connexion with graphs”. *Numerische Mathematik*, 1959, 1(1):269–271.
- [15] <http://www.tlc.dii.univpm.it/blog/databases4kinect>. accessed October 2015.