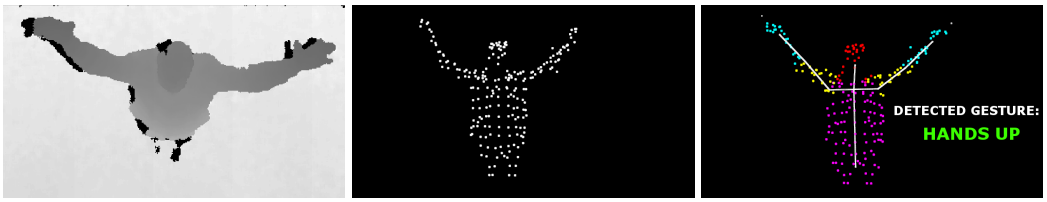
Excel@FIT 2015
http://excel.fit.vutbr.cz

Human gesture recognition using top view depth data obtained from Kinect sensor

Jan Bednařík*, David Herman**



Abstract
In this paper we present a system suitable for real-time human tracking and predefined human gestures detection using depth data acquired from Kinect sensor installed right above the detection region. The tracking part is based on fitting an articulated human body model to obtained data using particle filter framework and specifically defined constraints which originate in physiological properties of the human body. The gesture recognition part utilizes the timed automaton conforming to the human body poses and regarding tolerances of the joints positions and time constraints. The system was tested against the manually annotated 61-minutes-long recording of the depth data where ten different people were tracked. The 92.38% sensitivity was reached as well as the real-time performance exceeding 30 FPS. No a priori knowledge about the tracked person is required which makes the system suitable for seamless human-computer interaction solutions, security applications or entertainment industry where the position of sensors must not interfere with the detection region.

Keywords: Human gesture recognition — Human tracking using depth sensor — Human tracking from top view — 3D human body pose — Human model fitting — Articulated human model — Depth sensor — Microsoft Kinect for Xbox 360 — Human-Computer interaction using Kinect — Particle filter based human pose estimation — Tracking people

Supplementary Material: Demonstration Video

*jan.bednarik@rcesystems.cz, *Faculty of Information Technology, Brno University of Technology*
**david.herman@rcesystems.cz, *RCE systems s.r.o.*

1. Introduction

For many years the human tracking and gesture recognition have been the subjects of an extensive research. With the advent of depth sensors capable of 3D scene reconstruction the solutions proposing configuration of an articulated human model configuration estimation started to emerge.

Most recent solutions rely on the side view mounted sensors, i.e. a standing person must directly face the sensor. However, some applications might require the sensor to be placed outside the scene which the user moves within. The security surveillance applications where the sensor unobtrusiveness and wide view ranges are required, and the human-computer interaction applications might be mentioned as the examples. This paper thus introduces a novel solution which enables the high precision and reliability in human tracking and human gesture recognition using top view depth data.

2. Gesture recognition: related work

The design of human tracking using depth sensor has been addressed several times in the past. Among all the most prominent is the Microsoft Kinect for Xbox 360 hardware and software solution[1] based on learning a randomized decision forest on over a million training examples and mapping depth images to body parts[1]. This approach however requires the side view sensor placement (see Figure 1) thus making the sensor interfere with the detection region.

To process the top-view data Yahiaoui[2] uses a stereo pair of RGB cameras that detect a human and perform the tracking of the head part with Kalman filter based prediction of position. Migniot [3] proposes a system combining 2D model used for human localization and a 3D articulated human model based on particle filter to estimate a pose from top-view depth data. Gasparrini [4] presents a fall detection system based on a priori known background model, depth blob clustering and human recognition from the blobs using complex anthropometric features and relationships. A precise human pose is not estimated since it suffices to detect that a central point of the tracked blob reaches a given height threshold implying a possible fall. Rauter [5] then introduces Simplified Local Ternary Patterns, a new feature descriptor which is used for human tracking based on the head-shoulder part of the human body.

The solution proposed in this paper is based on the bootstrap particle filter recursively evaluating the model-data correspondence with regards to the common physiological properties of the human body which is represented as the articulated human model. Under specific conditions the penalization is applied to certain estimated poses in order to overcome common tracking imprecisions. The core of the predefined gestures recognition subsystem is based on the timed automaton with conditional transitions.

3. Articulated human model

The tracked person is modeled with an articulated human model based on the kinematic chain which consists of twelve joints and ten links (see Figure 2) representing the upper part of the human body. In order to reach the closest approximation to a real human skeleton, each joint is assigned a specific operation range (minimum and maximum rotation represented by Euler angles) and number of degrees of freedom with regards to human body physiology.

[1]Introduction of Microsoft Kinect for Xbox 360 - https://msdn.microsoft.com/en-us/library/hh855355.aspx

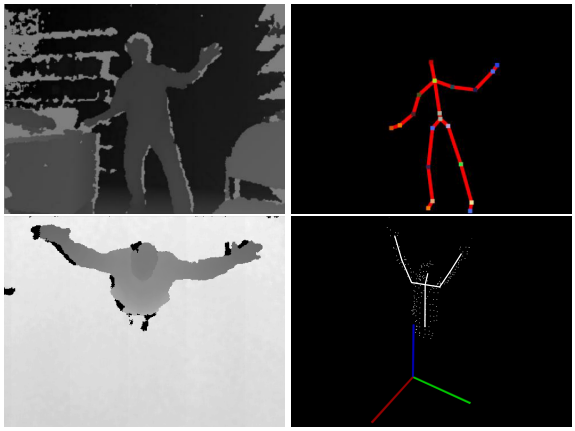


Figure 1 Comparison of depth data obtained when using side view (upper two figures)[6] and top view installed (bottom two figures) Kinect. It is evident that the top view data are substantially less descriptive as far as the human pose is concerned.

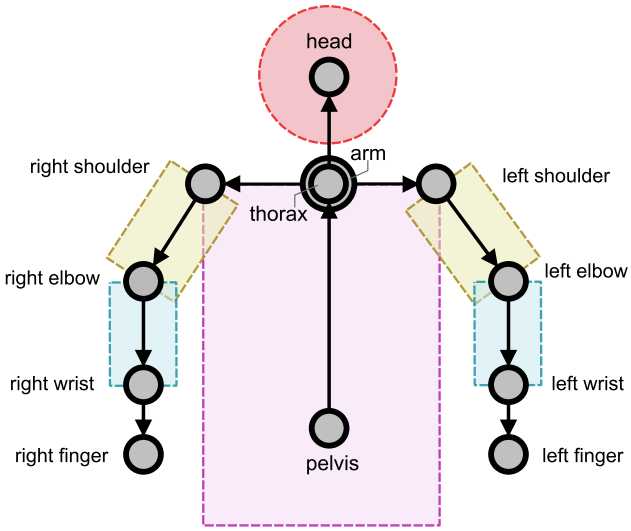


Figure 2 The Figure depicts the articulated human model designed as the kinematic chain (grey circles and arrows) and the body parts envelopes. A torso, an arm and a forearm are modeled as a cylinder, a head is modeled as a sphere.

Considering the evaluation function used by the particle filter (see Section 4) it is convenient to transform the obtained depth data from a coordinate system of the sensor to a coordinate system of each joint. For this purpose the kinematic chain is designed as the composition of transformation matrices where a single joint can be located by applying the Euclidean transformation on the position of the joint it is dependent on. For instance, the transformation matrix M_{LE} of a `left elbow` joint can be derived as:

$$M_{LE} = M_{LS} T_{LE} R_{z_{LE}} R_{x_{LE}} R_{y_{LE}}, \quad (1)$$

where M_{LS} denotes the transformation matrix for `left shoulder` joint, $R_{[x|y|z]}$ denotes the rotation

matrix (how the joint itself is rotated about each axis), and T_C denotes the translation matrix (how the joint is translated from the position of its parent joint – the `arm` joint in this case).

Since in a top view the upper limbs are the most descriptive body parts for estimating human's pose, similarly to [3] a wrist, an elbow and a shoulder correspond to separate joints in the articulated model while the whole spine is simply modeled as a single link between two joints, a `pelvis` and a `thorax`.

A single joint's rotation controls no more then one link. Thus, even though both the `thorax` and `arm` joints are dependent on `pelvis` and always share the same position, they are not linked to each other.

As the acquired depth data correspond to a human body surface, main body parts are modeled as different non-polyhedra. Considering the trade-off between the computational efficiency and the precision, a sphere was chosen to approximate a shape of a head and a cylinder was chosen to approximate a torso, an arm, and an forearm.

4. Human detection and tracking using particle filter

With regards to the Kinect sensor operating range [7] [8] the device must be installed in the height range cca 3.8 to 5.5 meters above the ground, and the Kinect's optical axis angle of incidence should reach up to 18° for better tracking precision (see Figure 3). The viewing angles of the sensor delimit the detection region within which a tracked human is allowed to move so that the whole body could still be seen by the sensor.

First, a person must perform the *initial pose* which serves the purpose of obtaining the information about a human which the system has no prior knowledge about. This specific pose – upright posture with arms spread out – was chosen since it is relatively easy to detect (see Section 4.1) and since it forces a person not to raise the upper limbs above the head which would prevent the human height from being estimated correctly. Once the initial pose is detected the more demanding tracking algorithm based on a particle filter can be started (see Section 4.2). The system architecture as the conceptual schema can be found in Figure 4.

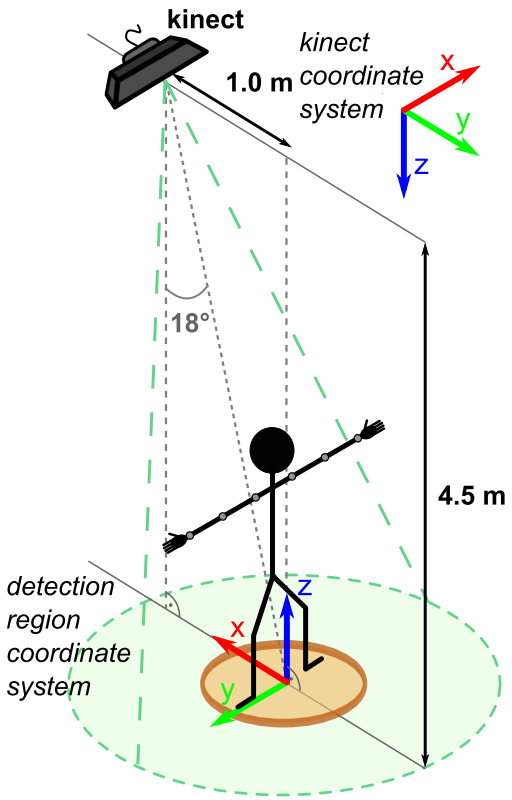


Figure 3 The figure depicts the required installation of the depth sensor Kinect and correspondingly delimited detection region (green circle). A person must perform an initial pose (arms spread out) in order for the tracker to start. A person's view direction is oriented along X axis.

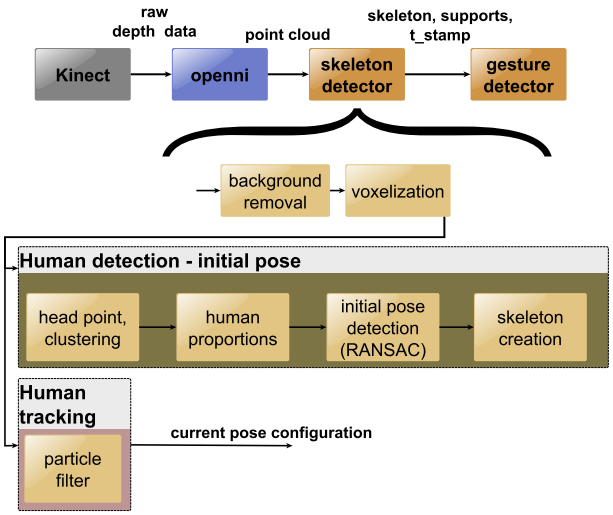


Figure 4 The conceptual schema of the gesture recognition system architecture.

4.1 Preprocessing - human detection

In the first step a background subtraction is performed by culling the depth points with Z-coordinate exceeding a user-specified threshold (the ground in the detection region is expected to be flat). In order to increase the computational speed the point cloud is voxelized and clustering is applied. Only the cluster containing the point closest to the global coordinate system origin (in XY plane) is kept while the other people and/or objects are filtered out.

In order to use the articulated human model the lengths of modeled links must first be estimated. The human body height is extracted from the depth point of the highest Z-coordinate. The lengths of all other

body parts are proportional to the height[9] and thus can be easily derived. If a human is performing the initial pose (arms spread out) it must be possible to fit a horizontal line to the depth points positioned in the expected height range corresponding to the chest line. RANSAC is used for this purpose.

Two most distant points among the set of inliers must comply to the expected arm span and the head centroid (i.e. the centroid of the depth points positioned in the expected height range of the head) projected to the line must be positioned approximately in the middle of that line. These measures prevent the tracker from being started in a situation when two or more people are present in the detection region close to each other.

4.2 Human tracking

In order to estimate the current state of the articulated human model (i.e. the configuration of all joints) with regards to the obtained depth data a bootstrap particle filter (BPF) is used [10]. At time t a state is represented by a particle $\{\vec{x_t^i}, w_t^i\}$ consisting of a vector of parameters $\vec{x_t^i}$ and assigned scalar weight w_t^i.

Vector $\vec{x_t^i}$ consists of sixteen parameters reflecting rotations and translations of the joints with regards to the given degrees of freedom (see table 1).

Table 1 The table summarizes which axes for rotation (R) and translation (T) are considered for the given joint.

joint	R [axes]	T [axes]
pelvis	X, Y, Z	X, Y, Z
thorax	X, Y	-
shoulder (left/right)	X, Y, Z	-
elbow (left/right)	X	-

Currently the `wrist` joints are not used since none of the pre-defined gestures reflects the hands positions and the `head` joint is not used since the head body part is modeled as a sphere which is invariant to rotations.

In each iteration the BPF aims to maximize the posterior probability $p(\vec{x_t}|\vec{y_t})$ (where y_t denotes the observation) and consists of resampling, propagation, evaluation, and best estimation step.

Resampling In this step the particles with high weights are duplicated while the particles with low weights are deleted. This enables for a fine-grained sampling of the parameter space in the subspace where it is most likely to find the closest approximation.

Propagation This step updates the values of particles' parameters. New parameters vector $\vec{x_{t+1}^i}$ is subject to $p(\vec{x_{t+1}}|\vec{x_t})$ which is modeled as a normal distribution for each (scalar) parameter. The limits which are set for parameters' values (maximum positive and negative joint rotation) comply with the common physiological properties of a human body[11], while standard deviations were estimated empirically.

Evaluation The step assigns a new weight w_{t+1}^i to each particle $\{\vec{x_t^i}, w_t^i\}$. The weight is given by the objective function $f_O(\vec{x}, \vec{d})$:

$$f_O(\vec{x}, \vec{d}) = \frac{1}{\sum_{i=1}^{D} \min_{bp} dist(\vec{d_i}, bp)} * penalty(\vec{x}), \quad (2)$$

where $\vec{d}$ denotes a vector of all D observed depth points, the function $dist(\vec{d_i}, bp)$ computes the shortest Euclidean distance between depth point $\vec{d_i}$ and a given non-polyhedra representing a body part bp and the function $penalty(\vec{x})$ evaluates the penalty (see Section 4.3) for the given parameters vector $\vec{x}$.

Since each body part can be represented by complex non-polyhedra arbitrarily translated and rotated in a global coordinate system, it is convenient to transform the depth points to the local coordinate system of body parts, so that the non-polyhedra would get aligned with the axes and the computation of $dist$ function would thus get less expensive. The transformed depth point is given as $\vec{d'} = M_{bp}^{-1} * \vec{d}$ where M_{bp}^{-1} is the inverse transformation matrix pertaining to body part bp as explained in Section 3.

Best estimation Maximum a posteriori (MAP) approach is used and thus a single particle with the highest weight is chosen to represent the best approximation.

As the `pelvis` joint represents the beginning of the kinematic chain a slight change in its global position can result in an significant change in the global position of the dependent joints along the kinematic chain (e.g. `shoulder` or `elbow`). Therefore the BPF works in two modes defined as follows:

Rough mode This mode is used to find the position of the tracked person within the detection region (i.e. X, Y and Z translation for the `pelvis` joint) and to roughly estimate a human pose. All of the sixteen parameters (as given in Table 1) are allowed to change their values.

Fine mode In this mode it is expected that a position and a coarse human pose was already found (by the `rough mode`) so the more precise estimation of

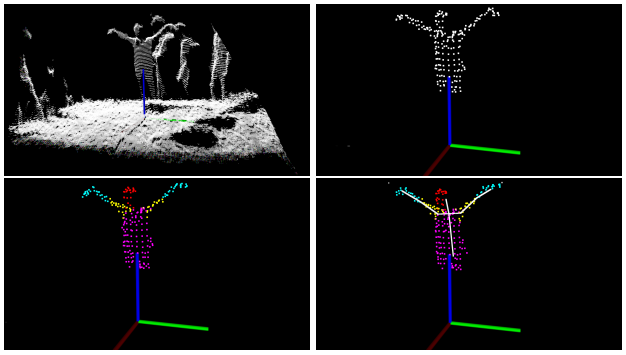


Figure 5 The figures depict the process of transforming the obtained raw point cloud (upper left figure) into down-sampled and clustered data (upper right figure), assigning the points to separate body parts (bottom left figure), and finally estimating the best matching articulated human model (bottom right figure).

the upper limbs configuration can be done. Only the parameters pertaining to `shoulder` and `elbow` are allowed to change their values.

All the steps necessary for processing each data frame including preprocessing and the estimation of the human model configuration are shown in Figure 5.

4.3 Penalty function

To reinforce the correct human pose estimation the penalty function (denoted as $penalty : \vec{x} \mapsto p,\ p \in \mathbb{R},\ p \in [0,1]$) is used to reduce the weight of the particles representing human poses unlikely to occur at the given time. An empirically estimated penalty value is assigned to each wrong pose. Two of such examples are given below:

Jumping Even though an ordinary human starts a jump by first swinging both arms down, the system tends to keep the articulated model at the stable position and only fits the upper limbs to the higher positions as if the user raised both arms. This event is recognized as a too rapid change of the upper limbs positions resulting in raised arms.

Squatting A natural squat consists of simultaneous leaning forward and moving the hips backwards. As the system tends to only model leaning forward, this pose is penalized or the tracking is temporarily stopped if the user happens to get under a reasonable height level.

5. Gesture recognition

A single gesture can be thought of as an ordered sequence of human poses defined by the orientations of the body parts, i.e. each joint is assigned the appropriate Euler angles. An essential part of the gesture

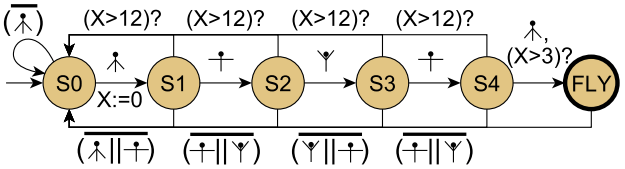


Figure 6 This figure depicts the timed automaton corresponding to the gesture `fly`. The transitions are denoted by the pose icon and time constraint (given in seconds). A horizontal line above an icon represents a complement of that pose (i.e. an arbitrary pose except the one displayed).

recognition subsystem is thus a timed automaton with conditional transitions where each state represents a human pose with given tolerance (reflecting the fact that each person might perform a gesture in a slightly different way) and time constraints. The timed automaton can only make a transition if both the time conditions of the current state and the orientation conditions of the body parts in the next state are met. For each gesture a separate timed automaton is specified (refer to Figure 6 which exemplifies the gesture `fly`).

Even though the Euler angles are convenient for intuitive definition of the human poses, they suffer from a significant drawback – the visual similarity of two poses does not necessarily correspond to their numerical similarity. This issue corresponds to the gimbal lock phenomenon[12] and effectively causes the fact that a single pose can be expressed by multiple combination of Euler angles values. To overcome this issue the positions of the upper limb joints are transformed to a spherical coordinate system. Since the problem still prevails for the poses where the inclination of the `elbow` and `wrist` joint is close to its minimum/maximum value (an arm straight up/straight down), the azimuth is forcibly set to 0° once the inclination reaches the given threshold.

The system was originally developed as an entertainment solution enabling a human to control the dynamic movement of the glass kinetic installation (see Section 7). For this purpose three static and three dynamic gestures were proposed by the designer (see Figure 7). A static gesture consists of a single pose which does not change over time while a dynamic gesture corresponds to a time constrained sequence of poses. Currently supported gestures are defined as follows:

Hands up A static gesture. Both upper limbs must be held upwards creating imaginary letter ”V” for at least one second.

Hands down A static gesture. Both upper limbs must be held downwards creating imaginary upside down letter ”V” for at least one second.

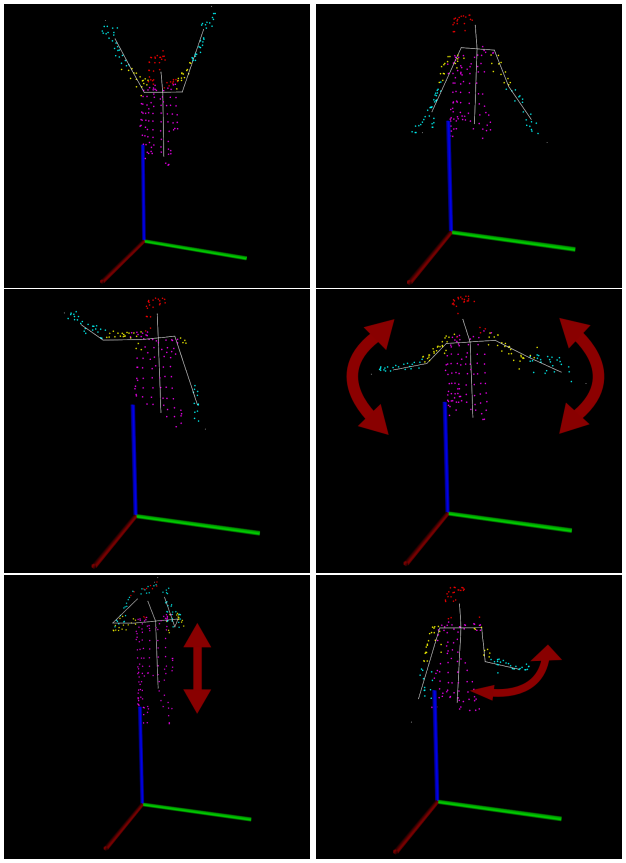


Figure 7 All six predefined gestures can be seen in the figure. From left to right and top to bottom : Hands up, Hands down, Selfie, Fly, Jump, Jedi.

Hands selfie A static gesture representing the act of taking a photograph of oneself. It can be performed by either of the upper limbs or by both of them simultaneously. Minimum duration is one second.

Fly A dynamic gesture. It mimics birds' act of waving their wings. The minimum and maximum durations are three and twelve seconds respectively.

Jump A dynamic gesture. An ordinary human jump where a head must reach at least 105% of a tracked human height.

Jedi A dynamic gesture. It mimics the act of swinging a (light) sword by either of both hands. An elbow should be close to a hip while a wrist follows one quadrant of a horizontal circle with the radius given by the forearm length.

6. Performance and experimental results

The proposed gesture recognition system was evaluated using five sequences of depth video data summing up to a 61.3 minutes runtime. The data was captured for ten different testing subjects (see Table 2) who, upon performing the initial pose, were instructed to repeat arbitrary body movements and perform occasionally some of the pre-defined gestures in a random way. The subjects were allowed to freely move within the detection region.

Table 2 For each test subject sex, body height, body shape and familiarity with the system is specified. High familiarity means the subjects had been already tested several times before, and it correlates with more precise performance of the gestures. Low familiarity means it was the first time for the subject to encounter the system, which often causes indistinct movements occurrence.

	sex	height [cm]	shape	familiarity
Jan	M	188	slim	high
David	M	178	slim	high
Katerina	F	155	round	low
Pavla	F	176	slim	low
Ales	M	181	slim	medium
Vojtech	M	189	slim	high
Vasek	M	174	slim	medium
Petr	M	175	round	low
Jana	F	162	round	low
Jiri	M	173	slim	high

For each recorded sequence the ground truth was specified with the manual annotation of the data. The particle filter was set to use the 640 particles, the system was tested against each of the sequence and the overall sensitivity given as

$$sensitivity = \frac{\sum true\ positive}{\sum ground\ truth}, \tag{3}$$

and precision given as

$$precision = \frac{\sum true\ positive}{\sum true\ positive + \sum false\ positive} \tag{4}$$

were evaluated.

Table 3 manifests that the system is sensitive to how precisely the gesture is performed, which allows reaching both higher sensitivity and precision for testing subjects Jan, David, Vojtech and Jiri. High false positive rate evaluated for testing subjects Katerina and Petr occurred mainly due to the multiple indistinct upper limbs movements which the system incorrectly identified as either `jedi` or `selfie` gesture.

The per-gesture performance can be found in Figure 8. As for the sensitivity it is evident the system is more robust for dynamic gestures (`fly`, `jump` and `jedi`) which in general consist of the poses defined with higher tolerances. Low precision evaluated for gesture `jedi` is caused by the fact that a casual movement of upper limbs naturally performed by a human often overlaps with this gesture definition.

Table 3 The results of the system evaluation. For each depth video sequence the total length (T) is specified as well as the ground truth (GT), true positive count (TP), false positive count (FP), sensitivity and precision.

testcase	T [s]	GT	TP	FP	Sensitivity	Precision
Jan-1	137	16	16	0	100.00%	100.00%
Jan-2	181	20	19	0	95.00%	100.00%
David	99	16	15	1	93.75%	93.75%
Kater	634	43	39	14	90.70%	73.58%
Pavla	240	32	25	0	78.13%	100.00%
Ales	315	28	25	2	89.29%	92.59%
Vojt	421	31	31	0	100.00%	100.00%
Vasek	289	25	22	1	88.00%	95.65%
Petr	545	38	36	6	94.74%	85.71%
Jana	398	30	28	1	93.33%	96.55%
Jiri	420	36	35	0	97.22%	100.00%
SUM	3679	315	291	25	**92.38%**	**92.09%**

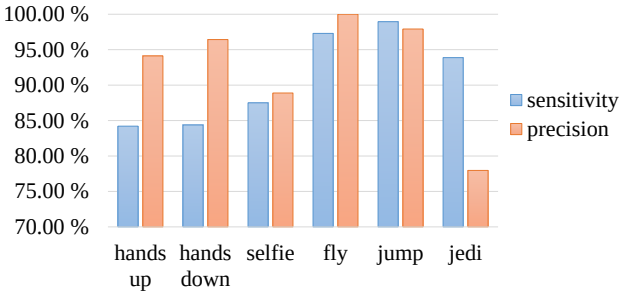


Figure 8 The evaluation of the gesture recognition system in a per-gesture manner.

The Figure 9 clearly shows that the system performance measured as the number of processed frames per second decreases and the quality of human tracking given as the average weight of the best estimations (see Section 4.2) over all test data frames increases as the amount of particles used by the particle filter is raised. The absolute value of the weight has no objective meaning, it is only used for relative comparison of the human tracking quality.

Once the number of particles reached the value of cca 600, no significant improvement was measured in the ability of the tracker to correctly estimate the model configuration for the given frame. On the other hand, if too few particles are used (approx. less then 400), the tracker loses the capability of following certain faster human body movements (as observed visually).

The frames per second (FPS) measure was obtained on optimized C++ implementation based on ROS framework[2] where most demanding part of the system – the *evaluation* step of the particle filter – was accelerated on GPU using CUDA framework. The hardware parameters of the PC under test were as follows: CPU Intel Core i5 4590 @ 3.3 Ghz x 4, GPU Nvidia GeForce GTX 760, 4GB RAM.

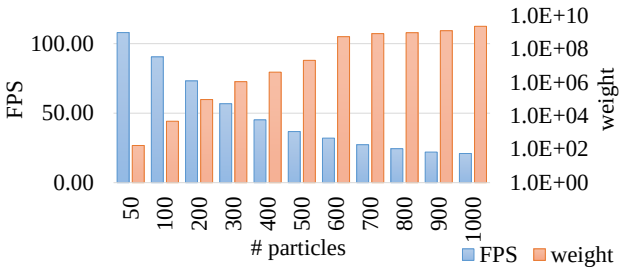


Figure 9 The figure depicts the decreasing system performance (`FPS`) and increasing quality of human tracking (`weight`) as the number of particles used by the particle filter grows.

7. Conclusion

In this paper a novel system capable of tracking a human and recognizing predefined gestures from top view depth data was introduced. The obtained data are fitted to the articulated human model representing the simplified skeleton of upper body part consisting of twelve joints and six rigid body parts.

Since the system requires no prior knowledge about a tracked human, the first step consists of detecting a human and estimating the essential human body attributes. Human tracking is based on a bootstrap particle filter and the quality of model-data fitting is reinforced by penalizing the unlikely poses given a current state. The core of gesture recognition subsystem is based on a timed automaton with conditional transitions.

The system was tested against more than one hour long depth video sequence and ten testing subjects with different body shapes. High sensitivity as well as the high precision was achieved reaching 92.38% and 92.09% respectively, and due to the GPU acceleration using CUDA framework the system runs in real time.

These results make the system perfectly capable of being deployed in the real world applications. This system was accepted for one of such applications – the entertainment solution enabling a user to control a glass kinetic installation which was exhibited at Euroluce 2015[3] in Milan, Italy, by Lasvit[4], a manufacturer of luxury glass installations and lightings based in the Czech Republic.

[2]Robot Operating System - `http://www.ros.org/about-ros/`

[3]Euroluce 2015 exhibition website - `http://salonemilano.it/en-us/VISITORS/Euroluce`

[4]Lasvit company website - `http://lasvit.com/`

8. Acknowledgement

This work was supported by the company RCE systems s.r.o.[5] which provided the development space and necessary hardware sources, and whose members were very helpful both during the development and testing processes.

References

[1] 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. In *Proceedings of the 2011 IEEE Conference on Computer Vision and Pattern Recognition*, CVPR '11, pages 1297–1304, Washington, DC, USA, 2011. IEEE Computer Society.

[2] Tarek Yahiaoui, Cyril Meurie, Louahdi Khoudour, and François Cabestaing. A people counting system based on dense and close stereovision. In *Image and Signal Processing*, volume 5099 of *Lecture Notes in Computer Science*, pages 59–66. Springer Berlin Heidelberg, 2008.

[3] Cyrille Migniot and Fakhreddine Ababsa. 3d human tracking in a top view using depth information recorded by the xtion pro-live camera. In *ISVC (2)*, volume 8034 of *Lecture Notes in Computer Science*, pages 603–612. Springer, 2013.

[4] Samuele Gasparrini, Enea Cippitelli, Susanna Spinsante, and Ennio Gambi. A depth-based fall detection system using a kinect sensor. *Sensors*, 14(2):2756–2775, 2014.

[5] Michael Rauter. Reliable human detection and tracking in top-view depth images. In *IEEE Conference on Computer Vision and Pattern Recognition, CVPR Workshops 2013, Portland, OR, USA, June 23-28, 2013*, pages 529–534, 2013.

[6] John MacCormick. How does the kinect work? [online]. http://users.dickinson.edu/~jmac/selected-talks/kinect.pdf, 2011.

[7] Microsoft Corp. Redmond WA. Kinect for windows sensor components and specifications, 2012. https://msdn.microsoft.com/en-us/library/jj131033.aspx.

[8] Microsoft Corp. Redmond WA. Coordinate spaces, 2012. https://msdn.microsoft.com/en-us/library/hh973078.aspx#Depth_Ranges.

[9] Inc. Fujitsu Microelectronics America. Human proportion calculator [online]. http://hpc.anatomy4sculptors.com/, 2014.

[10] Michael Isard and Andrew Blake. Condensation - conditional density propagation for visual tracking. *International Journal of Computer Vision*, 29:5–28, 1998.

[11] V. Janda and D. Pavlů. *Goniometrie: Učební text*. Institut pro daší vzdělávání pracovníků ve zdravotnictví, 1993.

[12] A.H. Watt and M. Watt. *Advanced Animation and Rendering Techniques: Theory and Practice*. ACM Press. ACM Press, 1992.

[5] Company RCE systems s.r.o. website - http://www.rcesystems.cz/