Augmented-Reality Environment for Locomotor Training in Children with Neurological Injuries*

Angelos Barmpoutis[1], Emily J. Fox[1], Ian Elsner[1], and Sheryl Flynn[2]

[1] University of Florida, Gainesville FL 32611, USA,
{angelos,ian}@digitalworlds.ufl.edu, ejfox@phhp.ufl.edu
[2] Blue Marble Game Co, Altadena CA 91001, USA
sheryl@bluemarblegameco.com

Abstract. In this paper a novel augmented-reality environment is presented for enhancing locomotor training. The main goal of this environment is to excite kids for walking and hence facilitate their locomotor therapy and at the same time provide the therapist with a quantitative framework for monitoring and evaluating the progress of the therapy. This paper focuses on the quantitative part of our framework, which uses a depth camera to capture the patient's body motion. More specifically, we present a model-free graph-based segmentation algorithm that detects the regions of the arms and legs in the depth frames. Then, we analyze their motion patterns in real-time by extracting various features such as the pace, length of stride, symmetry of walking pattern, and arm-leg synchronization. Several experimental results are presented that demonstrate the efficacy and robustness of the proposed methods.

Keywords: Augmented-Reality, Locomotor training, Spinal-cord injury, Physical Therapy, Rehabilitation, Game, Kinect.

1 Introduction

Locomotor training is an activity based therapy that aims to promote recovery of walking, by activating the neuromuscular system[4]. Locomotor training optimizes task-specific sensory input during intense stepping practice to promote activity-dependent plasticity. While locomotor training was initially developed for persons with spinal cord injuries, it has been recently translated to children and can be used in general with various types of neurologic injuries [14]. In the case of children, during locomotor training, stepping and standing often are practiced on a treadmill for over an hour causing many children to lose motivation and become bored. As attention and focus wane, critical task-specific movements, such as upright trunk posture and reciprocal arm swing, become nearly

* This project was in part funded by the NIH/NCATS Clinical and Translational Science Award to the University of Florida UL1 TR000064, and the University of Florida Informatics Institute Seed Fund Award. The authors would like to thank the sponsors and the anonymous volunteers who participated in the pilot study.

C.A. Linte et al. (Eds.): AE-CAI 2014, LNCS 8678, pp. 108–117, 2014.

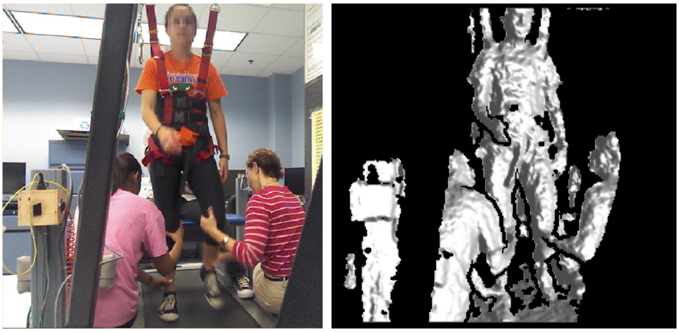

Fig. 1. A picture of our locomotor training environment (left), and the corresponding depth image (right) captured by a depth sensor located on the front of the treadmill.

impossible to evoke. Most importantly, a less intense and effective training session compromises the child's recovery. Incorporation of interactive and engaging video games is an innovative approach to enhance rehabilitation [6,2,5,11,13,7]. Although commercial games have demonstrated therapeutic effects when applied to children with neurological injuries, most games do not consider the specific impairments that are common in children with spinal cord injury and are not designed for use during locomotor training [6,7]. Therefore, our long-term objective is to design and develop an engaging and interactive game that enhances locomotor training for children with neurological injuries.

Our goal is to engage children to perform walking-related movements with their arms and legs in order to play the game. To accomplish this, we use a depth camera to detect and track movement in the unique locomotor training treadmill environment shown in Fig. 1 left. In leterature, there are several examples of methods or applications related to body tracking using depth cameras. A game-based rehabilitation system was presented in [9] using body tracking from RGB-D. Other applications include human detection [15], model-based 3D tracking of hand articulations [10], human pose recognition and tracking of body parts [12], and real-time 3D reconstruction of the articulated human body [3]. A detailed review of RGB-D applications using Microsoft Kinect sensor is presented in [8].

The main challenge in our particular application, which is the main focus of this paper, is that generic body tracking algorithms fail to detect and analyze the patient's body motion due to its close proximity with other objects or human subjects in the locomotor training treadmill environment (see examples in Fig. 5). To overcome this problem we propose a model-free graph-based body segmentation algorithm that detects the arms and legs of the patient. Descriptive motion features are extracted from the segmented regions of the limbs and their movement patterns are analyzed by computing various motion indices that we proposed in this paper to capture the symmetry between the right and left leg kinematics, their pace, and the synchronization of the arm swing with the walking pattern. Several experimental results are presented using real and synthetic data that demonstrate the efficiency of the proposed framework.

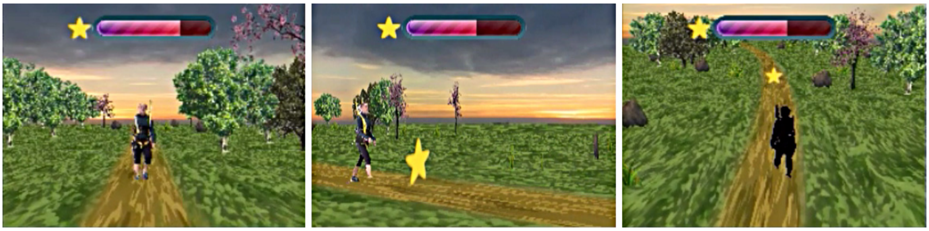

Fig. 2. Screenshots of the developed 3D augmented-reality environment taken from three different orientations to show the front, side, and back of the patient respectively.

2 Methods

In this section, we present our framework for computed-assisted locomotor training using augmented-reality gaming environments (see Fig.2). The framework has two main goals: 1) to enhance the traditional methodologies for physical therapy by exciting kids for walking using gaming technologies, and 2) to compute in real-time several motion-based quantities such as periodicity, synchronization, pace, and others in order to provide the therapist with a quantitative framework for monitoring the progress of a patient, evaluating the effectiveness of therapy, and automatically optimizing the parameters of the therapy.

Motion detection sensors such as infrared depth cameras offer an infrastructure that can facilitate the aforementioned goals and also offer a natural user interface for comunicating with computers without the need of remote controllers or other hand-held or warable electronic devices. Each data frame captured by a digital range camera is a two dimensional array of depth values (i.e., distance between the plane of the sensor and the depicted objects). The depth value of the pixel with coordinates (x, y) on a perticular depth frame is denoted by $d(x, y) \in \mathbb{R}^+$ (see an example of a depth frame in Fig. 1 right).

In the proposed framework, the acquired sequence of depth frames is processed by a graph-based segmentation algorithm that detects the regions of the patient's arms, legs, and torso in the depth images. Descriptive features of the segmented regions are then extracted and employed by motion pattern analysis algorithms, which are described in detail in the following sections.

2.1 Graph-Based Segmentation

Each depth frame is scanned horizontally (row by row) and segmented into line stripes that are smoothly-varing 1-pixel-wide regions defined as

$$\mathcal{L} = \{(x_s, y), \cdots, (x_e, y) : x_s < x_e, \left|\frac{\partial d(x,y)}{\partial x}\right| < \epsilon_1, \left|\frac{\partial^2 d(x,y)}{\partial x^2}\right| < \epsilon_2 \; \forall x \in (x_s, x_e)\}$$

where x_s and x_e denote the start and end pixel coordinates of the line segment. The length of a line segment can be easily computed by $length(\mathcal{L}) = x_e - x_s + 1$.

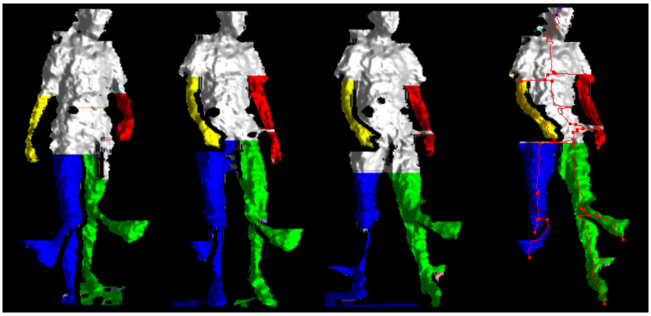

Fig. 3. Representative results of our body segmentation algorithm in different instances during the walking cycle. The generated tree graph is superimposed on the right plate. The partial inclusion of the assistants' arms in the regions of the subject's legs does not affect the estmated motion features as shown in Fig.4, which demonstrates the robustness of the proposed framework to such outliers.

The computed line segments are organized in the form of a directed graph, which is constructed simultaneously with the segmentation of the line segments. In such graph each line segment $\mathcal{L}$ can be connected with line segments in the previous row of pixels that form the set of $parents(\mathcal{L})$ defined as

$$\mathcal{L}' \in parents(\mathcal{L}) \Leftrightarrow \exists(x,y) \in \mathcal{L}, \exists(x,y-1) \in \mathcal{L}' : \left|\frac{\partial d(x,y)}{\partial y}\right| < \epsilon_1. \qquad (1)$$

Equivalently, each line segment can be connected with line segments in the next row of pixels by defining the set $children(\mathcal{L})$ as the inverse of Eq. 1 as follows:

$$\mathcal{L}' \in children(\mathcal{L}) \Leftrightarrow \mathcal{L} \in parents(\mathcal{L}'). \qquad (2)$$

The graph produced by Eqs. 1 and 2 may contain cycles. To enforce the creation of non-cyclic graphs we define the set $father(\mathcal{L})$ as the subset of $parents(\mathcal{L})$ that contains the largest line segment:

$$father(\mathcal{L}) = \underset{\mathcal{L}' \in parents(\mathcal{L})}{\arg\max}\ length(\mathcal{L}'). \qquad (3)$$

The above process segments a given depth frame into several regions that are computed as independent disconnected graphs and typically correspond to different objects in the field of view. In most applications the subject of interest corresponds to the graph with the largest number of pixels, and in general can be easily isolated from the rest of the objects in the scene (see Fig. 3).

Each graph can be further segmented into smoothly varying regions by constructing sets of connected line segments with coherent structural characteristics as follows:

$$\mathcal{S} = \{\mathcal{L}_1, \cdots, \mathcal{L}_n : \mathcal{L}_i = father(\mathcal{L}_{i+1}), |children(\mathcal{L}_i)| = 1\ \forall i \in [i, n-1]\}. \qquad (4)$$

The line segments $\mathcal{L}_i$ in Eq. 4 form a sequence of descendants without simblings, which corresponds to a linear graph. The set of segments $\mathcal{S}$ can also be organized into a graph by defining the $father(\mathcal{S})$ and $children(\mathcal{S})$ using the connections defined in $father(\mathcal{L}_1)$ and $children(\mathcal{L}_n)$ respectively. An example of a graph of segments is shown in Fig. 3 (the network of the graph is visualized on the right).

In our application, the regions of the legs and arms of the depicted subjects can be found by performing simple graph searches. More specifically, the legs can be found by searching for the segment with the largest sum of distances from the top of the older ancestor and from the bottom of the youngest descendant. Such distances can be easily computed by accumulating the height of each segment in the corresponding path of the graph given by

$$height(\mathcal{S}) = \max_{\forall(x,y)\in\mathcal{L}_i,\forall\mathcal{L}_i\in\mathcal{S},} y - \min_{\forall(x,y)\in\mathcal{L}_i,\forall\mathcal{L}_i\in\mathcal{S},} y + 1. \qquad (5)$$

The left and right children of the solution correspond to the right and left legs respectively. Finally, the left or right arms can be found as the largest right or left children in the graph respectively, which are not already marked as legs. Examples of estimated segments are shown in Fig. 3 with different color-coding.

2.2 Motion Pattern Analysis

After segmenting the regions of the limbs in each depth frame, their motion is analyzed by tracking their motion patterns over time. Various features can be extracted from each segmented region such as the average X, Y, Z coordinate, the medial line, the orientation of the limb, however we found in our experiments that the average Z coordinate of the medial line is descriptive enough to be used in our motion pattern analysis. Hence, four sequences are computed and monitored over time $LL(t)$, $RL(t)$, $LA(t)$, $RA(t)$ that correspond to the average Z coordinate of the medial line of the left leg, right leg, left arm, and right arm regions respectively.

First, each of the sequences is smoothened using a median filter followed by a Gaussian filter to enhance the robustness of our calulations (see example in Fig.4A). Then the local extrema of each sequence are computed as

$$extrema(f(t)) = \begin{cases} 1 & \text{if } t = \arg\max_{t\in N(t)} f(t) \\ -1 & \text{if } t = \arg\min_{t\in N(t)} f(t) \\ 0 & \text{otherwise} \end{cases}, \qquad (6)$$

where $N(t)$ is a neighborhood in the time domain centered at t. The size of the neighborhood should be smaller than the duration of each pace to ensure that the extrema of each pace are calculated. Fig. 4B shows Eq. 6 computed from a real data sequence.

The duration between two consecutive same-type extrema (minima or maxima) is given by

$$gap(f(t), v) = \min_{\forall s\geq x:extrema(f(s))=v} s - \max_{\forall s<x:extrema(f(s))=v} s \qquad (7)$$

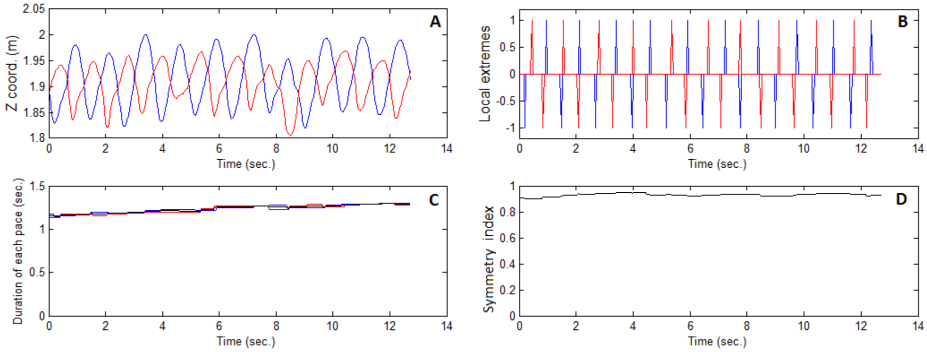

Fig. 4. Experimental results from a real data sequence of 10 paces. A) The estimated z-coordinates of the right (red) and left (blue) leg, B) The time signatures of the estimated local extrema of A, C) The estimated duration of each pace, D) The corresponding symmetry index.

Eq. 7 can be calculated for $LL(t)$ and $RL(t)$ for each type of maxima in order to compute the duration of each pace as a function of time:

$$pace(t) = \left[gap(LL(t),1) + gap(LL(t),-1) + gap(RL(t),1) + gap(RL(t),-1)\right]/4$$

In a walking pattern with constant speed it is expected that each of the four terms in Eq. 7 gives the same numerical value. However, in practice we compute the average of all four measurements in order to increase the robustness of the overall pace estimator. It should be noted that $pace(t)$ corresponds to the time of a full stride from the time when the recorded signal reaches a local maximum until the time when the same leg's signal reaches the next local maximum (see example in Fig. 4C). The corresponding length of a full stride can be easily computed as $pace(t)/speed$, where $speed$ is the speed of the treadmill, which is a known manually set quantity.

The symmetry of the walking patterns of the two legs as well as their synchronization with the motion of the arms are quantifiable indices that are especially useful for physical therapy. To calculate the level of symmetry of the walking pattern we first need to estimate the mid-point in the time domain between two same-type extrema given by

$$mid(f(t),v) = \left[\min_{\forall s \geq x : extrema(f(s))=v} s + \max_{\forall s < x : extrema(f(s))=v} s\right]/2 \tag{8}$$

and then compare two sequences $f(t)$ and $g(t)$ by computing the time difference between the extrema of one and the corresponding mid-point of the other as follows:

$$sym(f(t),g(t),v) = 1 - 2\left|mid(f(s),v) - s\right|/pace(s), \tag{9}$$

where $s = \max_{\forall r < t : extrema(g(r))=v} r$. The largest possible time difference is equal to the duration of half pace, hence the range of Eq. 9 is the interval of real numbers from 0 to 1.

Finally, Eq. 9 considers only one type of extrema of one of the two given sequences. To account for all combinations we compute the symmetry of the walking pattern as the sum of four terms given by

$$symmetry(LL(t), RL(t)) = \sum_{v=-1,1} \frac{sym(LL(t), RL(t), v) + sym(RL(t), LL(t), v)}{4}$$

where $LL(t)$ and $RL(t)$ denote the left and right leg sequences (see Fig. 4D). Similarly, the index that describes the sychnronization between the motion of the legs and the arms can be computed using the arm and leg sequences as follows: $[symmetry(LL(t), LA(t)) + symmetry(RL(t), RA(t))]/2$ (see Fig. 7 left).

2.3 Augmented-Reality Environment

The algorithms presented in Secs. 2.1 and 2.2 were implemented in Java using the J4K (Java for Kinect) Software Development Kit introduced in [3]. The input depth frames are processed in real-time using the proposed graph-based segmentation algorithm and the 3D image of the body of the patient is composed as a textured quadratic mesh by combining the segmented depth image with the corresponding video frame captured by a regular RGB camera.

The 3D image of the body of the patient is visualized within a 3D augmented-reality gaming environment that consists of a randomly generated scene with a walking path. The patient can watch herself walking in this synthesized environment from different 3D views (front, side, and back) that offer visual variability that makes the gaming environment more engaging and easier for the patient to understand the perspective and virtual surroundings (see screenshots in Fig. 2). The environment automatically animates in relation to the speed of the treadmill and new surrounding elements randomly appear as the game progresses.

A scoring system is also used in order to enhance the level of engagement by unlocking new features and environments based on the patient's score. Although in the current version of the game the scoring system is based on the number of items that the user collects from the virtual path, which is proportional to the distance walked, our goal is to introduce adaptive scoring mechanisms that will award good walking patterns based on the calculated symmetry and synchronization of the motion (Sec. 2.2).

3 Experimental Results

In this section, we present several quantitative experimental results obtained by using the proposed framework with real and synthetic data.

Locomotor training typically includes alternate bouts of approximately 5 minutes of treadmill stepping and standing practice to achieve a total of 60-minutes of practice. A team of trainers provide hands-on assistance as well as motivation for the child to step at an age-appropriate/normal speed, maintain an upright trunk posture and normal leg kinematics, and reciprocally swing his or her arms.

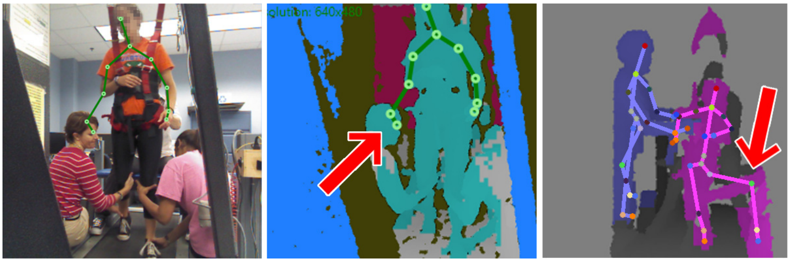

Fig. 5. Typical errors of existing skeleton fitting algorithms caused by the close proximity of the patient with other objects or human subjects in the clinical environment.

To evaluate the feasibility and effectiveness of the proposed framework we organized a pilot study in which we recreated locomotor training sessions in a real clinical setting without employing real patients at this time. During the pilot sessions, depth data were acquired using the PrimeSense$^{\text{TM}}$depth sensor contained in the Microsoft Kinect$^{\text{TM}}$device. The sensor was placed in front of the treadmill and was connected to a 64-bit computer with Intel Core i5 CPU at 2.53GHz and 4GB RAM. The resolution of the depth camera was 320×240 pixels at 25 frames per second and it was calibrated so that it records depth in the range from 0.8m to 4.0m, which is suitable for capturing the patient's motion in our clinical setup. In this hardware setup the proposed algorithms were executed in real-time with average data processing time of 9.9889 ms/frame.

In order to compare our technique with other existing popular methods for body feature extraction from depth frames, we employed the skeleton fitting algorithm provided with the Microsoft Kinect SDK [1]. Our goal was to extract the location and/or orientation of the legs in order to be able to compute the pace and symmetry indices as described in Sec. 2.2. The skeleton fitting algorithm failed to provide full body skeleton for the majority of the depth frames sequences. Instead, upper body tracking was possible, which however does not track the user's legs. The failure of the skeleton fitting algorithm was caused due to the close proximity of the user's body with other objects in our clinical environment. Even in the rare cases in which the algorithm provided output, the obtained skeletons were often erroneous as it is shown in Fig. 5.

In contrast to the Kinect SDK algorithm, our proposed body segmentation algorithm (Sec. 2.1) was able to segment the user's body and provide results for the entire dataset. Fig. 3 shows the segmentation results obtained for four representative frames during the walking cycle. The segmented regions of the arms and legs are color-coded and in the right plate the underlying estimated graph is superimposed. It should be noted that the presence of the assistants' arms in the regions of the subject's legs did not affect our motion analysis because our feature extraction method is robust to such outliers as it is shown in Fig. 4.

Furthermore, we applied the proposed motion analysis framework (Sec. 2.2) to our segmentation results and the computed pace and symmetry indices are presented in the plots of Figs. 4, 6, and 7.

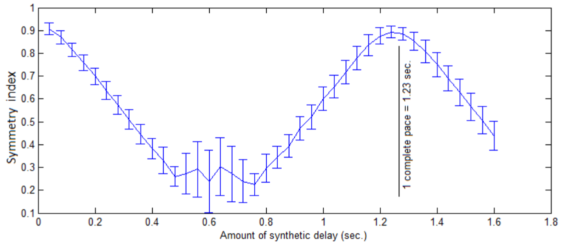

Fig. 6. Plot of the symmetry index versus the amount of synthetic delay introduced in the data sequence of the left leg.

Fig. 4 shows the data sequences extracted from the left and right legs for a period of 10 full strides, and the correspodning extrema, pace, and symmetry index. The red and blue lines correspond to the right and left legs. From the obtained results we can make the following observations: a) the extracted data sequences describe well the walking pattern, b) the pace estimated from the right leg deviates insignificantly from the one of the left leg, which demonstrates the robustness of our model, and c) the calculated symmetry index is very high as expected, which also demonstrates the effectiveness of the proposed index.

More specifically, the average duration of a full stride for a data sample of 60 sec. (Fig. 7) was found to be 1.2350±0.0360 sec. for the left leg and 1.2338±0.0407 sec. for the right leg, with the average for both legs at 1.2344±0.0351 sec. In order to assess the robustness of our algorithm we can multiply these results by the number of acquired frames per second. This will give us an estimated duration of pace of 30.8591 frames with standard deviation of less than a frame (0.8765), which concluively demonstrates that the stability of our estimator reaches the limit of the data acquisition frequency and hence cannot be further improved.

In order to demonstrate the behaviour of the proposed symmetry index in the case of abnormal leg kinematics we introduced various amounts of delays in the sequence of the left leg and for each case we computed the symmetry index. The results are presented in Fig. 6 and show that the symmetry index drops as the amount of delay approaches the half of the pace as expected.

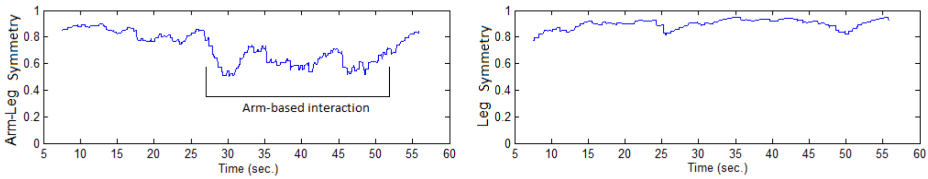

Fig. 7. Plots of the computed arm-leg symmetry (left) and the corresponding leg symmetry (right) from a real data sequence. As expected the motion pattern of the arms has more fluctuations compared to the more coherent symmetry index of the legs.

Finally, Fig. 7 show the arm-leg synchronization (left) in contrast to the symmetry of the leg motion (right). As expected the arm-leg index has higher values when the patient reciprocally swings his or her arms and drops otherwise, for example in the case of hand gestures or other type of arm-based interaction.

4 Conclusion

In this paper a novel framework was presented for locomotor training using an augmented-reality gaming environment, which is remotely controlled with natural user motions detected by a depth camera. The proposed algorithms analyze the motion patterns and compute various descriptive indices that provide the therapists with a quantitative framework for monitoring the progress of the patients. Several experimental results were presented that demonstrated the effectiveness and robustness of our methods. In the future we plan to employ this framework to enhance locomotor training and excite kids for walking.

References

1. Microsoft Kinect SDK, http://www.microsoft.com/enus/kinectforwindows/
2. Adamovich, S., Fluet, G., Tunik, E., Merians, A.: Sensorimotor training in virtual reality: a review. NeuroRehabilitation 25(1), 29–44 (2009)
3. Barmpoutis, A.: Tensor body: Real-time reconstruction of the human body and avatar synthesis from RGB-D. IEEE Trans. on Cybernetics 43(5), 1347–1356 (2013)
4. Behrman, A., Harkema, S.: Locomotor training after human spinal cord injury: a series of case studies. Phys. Ther. 80(7), 688–700 (2000)
5. Bryanton, C., et al.: Feasibility, motivation, and selective motor control: virtual reality compared to conventional home exercise in children with cerebral palsy. Cyberpsychol. Behav. 9(2), 123–128 (2006)
6. Dematteo, C., Greenspoon, D., Levac, D., Harper, J.A., Rubinoff, M.: Evaluating the nintendo wii for assessing return to activity readiness in youth with mild traumatic brain injury. Phys. Occup. Ther. Pediatr. (in print, 2014)
7. Deutsch, J.E., et al.: Use of a low-cost, commercially available gaming console (wii) for rehabilitation of an adolescent with cerebral palsy. Phys. Ther. 88(10) (2008)
8. Han, J., et al.: Enhanced computer vision with Microsoft Kinect sensor: A review. IEEE Transactions on Cybernetics 43(5), 1318–1334 (2013)
9. Lange, B., et al.: Interactive game-based rehabilitation using the Microsoft Kinect. In: IEEE Virtual Reality Workshops, pp. 171–172 (2012)
10. Oikonomidis, I., et al.: Efficient model-based 3d tracking of hand articulations using Kinect. In: Proc. of the British Machine Vision Association Conference (2011)
11. Reid, D.: The influence of virtual reality on playfulness in children with cerebral palsy: a pilot study. Occup. Ther. Int. 11(3), 131–144 (2004)
12. Shotton, J., et al.: Real-time human pose recognition in parts from single depth images. In: IEEE CVPR Conference, pp. 1297–1304 (2011)
13. Walker, M.L., et al.: Virtual reality-enhanced partial body weight-supported treadmill training poststroke: feasibility and effectiveness in 6 subjects. Arch. Phys. Med. Rehabil. 91(1), 115–122 (2010)
14. Willoughby, K.L., et al.: A systematic review of the effectiveness of treadmill training for children with cerebral palsy. Disabil. Rehabil. 31(24), 1971–1979 (2009)
15. Xia, L., et al.: Human detection using depth information by Kinect. In: IEEE Conference on Computer Vision and Pattern Recognition Workshops, pp. 15–22 (2011)