Monitoring the Crus for Physical Therapy

Toon Van Craenendonck¹, Tim Op De Beéck¹, Wannes Meert¹, Benedicte Vanwanseele², and Jesse Davis¹

¹ Department of Computer Science, KU Leuven
Celestijnenlaan 200A, 3001 Heverlee, Belgium
toon.vancraenendonck@student.kuleuven.be
{tim.opdebeeck,wannes.meert,jesse.davis}@cs.kuleuven.be

² Department of Kinesiology, KU Leuven
Tervuursevest 101, 3001 Heverlee, Belgium
Benedicte.Vanwanseele@faber.kuleuven.be

Abstract. Capturing the movements of a patient performing a rehabilitation exercise currently involves an extensive lab setup. The goal of this study is to investigate whether a 3D camera, such as the Microsoft Kinect™, can be used to monitor patients locally. Specifically we are interested in the lower limbs since most 3D camera algorithms focus on the upper body while for rehabilitation, the lower body is crucial. This paper presents two particle-filtering based algorithms for accurate tracking. The first algorithm estimates the configuration of the lower limbs simultaneously while the second one estimates the configuration of one limb at a time. We compare our estimates with a gold standard and find that we are able to recognize most movement characteristics. Furthermore, our approach is better at tracking the height of the foot and yields more stable tracking results than the NITE skeleton tracker.

1 Introduction

Capturing the movements of a patient performing a rehabilitation exercise currently involves an extensive lab setup with a variety of sensors that are mounted in the floor or attached to the body. As a consequence, this technology is not available to most physiotherapists and patients. Fortunately, the development of more portable alternatives is empowering physiotherapists to monitor patients out of the lab and is enabling daily monitoring. The advantages of this evolution are threefold. Firstly, it will aid in diagnosing patients since objective tools for screening patients will become more readily available. Secondly, it provides the physiotherapist with objective information about the period in between physical therapy sessions. This is important because treatment plans prescribe additional exercises to be performed at home. Better insight into the quantity and quality of the exercises performed by a patient will enable physiotherapists to adjust treatment plans to the individual needs of a patient. Thirdly, such a portable system can be used to provide patients with automatic feedback. This is important since low quality executions of prescribed exercises can hinder the rehabilitation process and can even lead to new injuries.
The goal of this study is to investigate whether a 3D camera, such as the Microsoft Kinect™, whose output is a 3D point cloud (i.e., a set of points in a three-dimensional coordinate system), can be used to replace (some of) the traditional sensors in order to monitor patients out of the lab. In order to make the monitoring appropriate for uses in the field of physical therapy, high-accuracy tracking of joint positions is needed. However, current 3D camera algorithms focus on the body as a whole and are mainly meant to be used for playing games. New algorithms are being proposed for therapy purposes but they typically focus on the upper body. In this work, we focus on the lower limbs and specifically the crus, that is, the lower part of the leg, to increase the accuracy of the estimated knee and foot positions.

Focusing on the lower limbs is challenging since reflections on the floor add noise to the data points that are close to the floor. This makes tracking the feet and knees more difficult. However, accurate tracking of the knee and the foot as well as the relative positioning of the knee with respect to the foot is particularly important when monitoring popular exercises such as lunges or when monitoring landing technique in jumping sports such as netball. During lunges, the front leg’s knee should not pass the toes of the front foot because this causes greater stress on the knee [9,14]. To land safely after jumping, it is important to keep the knee above the foot as well [2]. Furthermore, tracking the height of the foot is useful to measure foot clearance, which is considered to be related to the risk of falling for elderly patients [1].

This paper presents two algorithms that are based on particle filtering. To achieve high accuracy and to make the approach robust against noise caused by reflections, these algorithms make use of temporal information as well as measurements of the true lengths of a patient’s limbs since this information is typically readily available. The first algorithm estimates the configuration of the lower limbs simultaneously. The second estimates the configuration of one limb at a time. Next, the paper evaluates the proposed approaches with both the measurements of a gold standard measurement system as well as with the estimates of the NITE tracker algorithm. The former comparison shows that most of the movement characteristics can be recognized in our position estimates when tracking lunges. The latter comparison reveals that overall our approach is more stable than the NITE algorithm, and specifically in tracking the foot positions. This increase in accuracy comes at the cost of an increased computation time.

2 Background and Related Work

The algorithms used in this study are based on particle filtering. This section gives some basics related to particle filtering. The algorithm section will cover its specific implementation in this study.

2.1 Particle Filter

The goal of most pose estimation algorithms can be formulated as obtaining an approximation of the posterior $$p(x|z)$$, where $$x$$ is the pose and $$z$$ is the noisy
observation. The posterior will be approximated by a finite set of \( M \) samples \( X^i_1, \ldots, X^i_M \). The general outline of the particle filter algorithm consists of two steps: prediction and correction [6,12]. At every iteration \( k \), a new set of particles \( X_k \) will be constructed from the previous set \( X_{k-1} \) and the current observation \( z_k \). In the beginning, the new particle set \( X_k \) is empty. From every particle in the previous particle set \( X_{k-1} \) a new particle is generated (prediction) and its weight is calculated (correction). Generating a new particle from an existing one involves the system model: given this current state, what will be the next state of the system? More formally this prediction step can be written as

\[
x^i_k \sim p(x^i_k|x^i_{k-1})
\]

In this step \( M \) particles \( x^i_k \) are sampled from the distribution \( p(x^i_k|x^i_{k-1}) \) and are added to a temporary particle set \( \overline{X}_k \). \( p(x^i_k|x^i_{k-1}) \) can be calculated using the system model, a function describing the evolution of states over time. Assigning a weight means determining the likelihood of the current measurement \( z_k \), given that the underlying state is indeed \( x^i_k \). This can be formulated as

\[
w^i_k \propto p(z_k|x^i_k)
\]

This correction step calculates a weight \( w^i_k \) for every particle \( x^i_k \) by evaluating \( p(z_k|x^i_k) \), what can be calculated by assigning penalties and rewards to each particle given the current observation.

In the next step, which is called importance sampling, the new particle set \( X_k \) is constructed from the temporary set \( \overline{X}_k \) by sampling \( M \) particles with probabilities proportional to their weights. This ensures that particles which are more likely to be correct, given the current observation, are more likely to be transferred to the new particle set.

### 2.2 Related Work

Particle filtering is a widely used method to do skeleton tracking, and often the skeleton configurations are represented by kinematic trees. Typically, these approaches focus on real-time methods to achieve an interactive experience. To be real time, most methods simplify some parts of the model. For example, in gesture recognition it is often sufficient to track only the upper body [11,10]. Or multiple cameras are used to compensate for the efficient but approximative techniques [15,5].

While real time tracking is an important issue, it is a challenging objective and a common problem with particle filtering as this is computationally intensive. To overcome the high dimensionality of the search space for the kinematic tree, some modifications have been proposed such as ‘Progressive Particle Filtering’ [4]. In a progressive particle filter, estimates of the person’s global position, torso, upper arms and legs, and lower arms and legs are made separately. So a particle does not represent an entire skeleton configuration, but only a part of it. The approach used in this work is similar with a stronger focus on the crus instead of the upper body. Furthermore the previous frame as well as knowledge about the movements of a body that are physically possible are taken into account.
2.3 Relationship to Decision Tree approaches

Most commercial solutions use decision trees learned from a training set [13]. Although training is computationally expensive the skeleton configuration can be estimated relatively accurately in real time. The pose is estimated from a single frame and no temporal information is taken into account.

In this work we compare to the NITE tracking software that is used in the OpenNI framework. This algorithm is chosen because it is available for all platforms and can be used with other 3D cameras as well. The accuracy of NITE is comparable to the accuracy obtained with the Microsoft Kinect API [3].

3 Skeleton Estimation Algorithm

Our approach consists out of two phases, which both make use of a particle filter. The first phase is to localize the person in the scene and the second phase is to estimate the skeleton configuration of this person’s lower limbs. Since our priority is to achieve a more accurate position for a certain patient and use this approach in a rehabilitation setting where the user is measured beforehand, we communicate this information to the algorithm.

3.1 Localizing the Person in the Scene

A number of particle filter iterations is performed for every frame to localize the person in the scene.

State Representation. A particle or state sample $x_k$ is represented by a cylinder. These cylinders are defined using a $(x, y, z)$ coordinate and a predefined width and height based on the measurements of the current user.

System Model. Because of the sufficiently high frame rate of the camera, we can assume that a person does not travel large distance between frames. Therefore, the system models the transition from one frame to another by adding Gaussian noise in every direction to each particle.

Weight Calculation. The weights of the particles are computed from the observed point cloud generated by the 3D camera by adding one reward ($r_1$) and two types of penalties ($p_1$ and $p_2$) per particle and by normalizing over all particles.

$$ W_k^t = \left[ \sum_{p \in \text{points}} r_1 \cdot \mathbb{1}_{c_k^1}(p) - p_1 \cdot \mathbb{1}_{c_k^2 \setminus c_k^1}(p) - p_2 \cdot d(c_k^1, \text{head}) \right] $$ (3)

1. Particles are rewarded for points of the point cloud that are inside the cylinder $c_k^1$.
2. A penalty is introduced for points outside the cylinder $c_k^1$ that are inside a larger cylinder $C_k^1$ with the same central axis. This favors particles that capture the entire person instead of just part of it.
3. After $n$ iterations, cylinders with a large distance between their top and the head of a person will be further penalized. Hence the top of the cylinders will align with the person’s head. Introducing the penalty earlier on would make finding good particles a lot harder. During the initial iterations the best particles only contain part of the points. Therefore the additional penalty would possibly reduce the weight of the good particles to 0 as well. Experiments have shown that $n = 7$ results in a good performance.

The person’s position can be determined by first calculating a new cylinder as the weighted average of all cylinders, followed by calculating the center of gravity of all points that are inside that cylinder.

### 3.2 Estimating the Skeleton Configuration

Once a person’s position inside a frame is determined, the skeleton configuration is estimated. This section covers two distinct approaches that accomplish this: one that is non-hierarchical and one that is hierarchical.

**Non-Hierarchical Approach**

*State Representation.* A particle or state sample $x_k$ is represented by a kinematic tree. In a kinematic tree, the position of a joint is given by a transformation relative to the previous joint in the tree. The skeleton structure that we use looks as follows:

```
           root
         /    \    
    left hip  right hip
     /   \   /   \
left knee right knee
    /     \     
left foot right foot
```

The state is not defined by the actual 3D coordinates of the hip, knee or foot but by transformations relative to the parent element in the tree. To go from one element to the next a rotation $\theta$ is performed followed by a translation $T$. The actual 3D coordinates are then obtained by starting at the origin and performing a series of transformations.

Defining all the rotation and translation parameters for every element in the tree requires a 42-value state vector (seven elements in the trees, each requiring three rotation and three translation parameters). However, we made the following assumptions to reduce the state space:

1. The torso is always in the upright position, so the root will not be rotated.
2. The relative position of the hips with respect to the root remains unchanged.
3. To generate candidates for the knee and foot elements biomechanical limitations impose that the rotation along the \( z \)-axis is always zero.

4. Exercise specific characteristics as well as joint specific limitations allow to impose additional restrictions on the rotation angles.

5. All limb lengths are fixed which predefines the translations.

Only the translation matrix for the root requires three values. However, these values are already estimated in the previous step. The pose estimation step can now be reduced to estimating a 8-value state vector.

\[
\begin{bmatrix}
\theta_{\text{leftknee},x}, \theta_{\text{leftknee},y}, \theta_{\text{rightknee},x}, \theta_{\text{rightknee},y}, \\
\theta_{\text{leftfoot},x}, \theta_{\text{leftfoot},y}, \theta_{\text{rightfoot},x}, \theta_{\text{rightfoot},y}
\end{bmatrix}
\]  

\( (4) \)

**System Model.** The system model is the same as the model used to localize the person in the scene.

**Weight Calculation.** The weights of the particles are determined by calculating a reward and a penalty for each particle.

\[
W_k = \sum_{p \in \text{points}} r_1 \cdot \mathbb{1}_{c_{1k} \cup c_{3k} \cup c_{4k}(p)} - p_1 \cdot \mathbb{1}_{\text{outside}_C}(p)
\]

where \( \text{outside}_C \) is equal to:

\[
\text{outside}_C = (C_{1k} \setminus c_{1k}) \cup (C_{2k} \setminus c_{2k}) \cup (C_{3k} \setminus c_{3k}) \cup (C_{4k} \setminus c_{4k})(p)
\]

and \( c_{1k}, c_{2k}, c_{3k}, c_{4k} \) are the four cylinders representing the upper and lower part of the left and right leg after \( k \) iterations.

1. Particles are rewarded for points that lie inside a cylinder with diameter \( d_1 \) of which the axis aligns with one of the upper legs \( (c_{1k} \text{ and } c_{2k}) \) or lower legs \( (c_{3k} \text{ and } c_{4k}) \).

2. Penalties are given for points that lie outside the cylinders with diameter \( d_1 \), but inside the corresponding cylinders with the same axis but with diameter \( d_2 > d_1 \), denoted with a capital “C”. The intuition is that we want to favor cylinders that capture all points in a local neighborhood corresponding to the entire part of the leg.

**Hierarchical Approach** Instead of estimating the skeleton configuration as a whole once the person is localized in the scene, this approach further splits the pose estimation step by searching for the configuration of one limb at a time, starting at the root and working its way down the kinematic tree.
State Representation. Since a particle corresponds to a possible configuration of one limb, a particle or state sample $x_k$ is represented by a 2-value state vector under the same assumptions as in the non-hierarchical approach. Such vectors consist of two rotations which are defined with respect to the previous link in the kinematic tree that was estimated.

System Model. The system model is the same as the previous two system models.

Weight Calculation. This approach calculates the weights of particles in the same way as the non-hierarchical approach. It differs in a sense that the particles only represent one limb at a time. Once the configuration of a limb is estimated the particle filtering process is restarted for the next limb down the kinematic tree. Hence the problem of estimating 8-value state vectors is now reduced to estimating four 2-state value vectors to determine the configuration of the legs. In each step the weights are calculated as follows.

$$W_k^i = \sum_{p \in \text{points}} r_1 \cdot 1_{e \cdot \text{limb}_k^i} (p) - p_1 \cdot 1_{c \cdot \text{limb}_k^i} (p)$$

4 Experiments

In this section we try to answer the two following questions.

Q1. How do the estimates obtained with both particle filters compare to the measurements of a gold standard measurement system?

Q2. How do both algorithms compare to the NITE tracker algorithm?

4.1 Data Collection

A first data set was collected by simultaneously recording lunge executions with both a Vicon system as well as a Microsoft Kinect™. Using eight infrared cameras and markers attached to the subject’s body, the Vicon system captures 3D coordinates of the different joints with a millimeter accuracy and can be considered as a gold standard. The Kinect™ returns a 3D point cloud representing the pose of the person in the scene. This point cloud consists of 640x480 points. Furthermore with increasing distance from the sensor the random error of depth measurements increases quadratically and the depth resolution decreases quadratically, therefore the ideal range in which to acquire data is 1-3 m from the camera [8]. An important issue to keep in mind while working with these point clouds is that they do not represent the human body directly. Instead the points corresponds to the surface of the side of the body that is facing the camera. In total four lunge executions of one subject were recorded. On average these recordings last 3.3 seconds and contain 99 frames each.

A second data set was recorded to compare with the NITE skeleton tracking algorithm. NITE needs to calibrate before it can start tracking a subject. Unfortunately, it was impossible to calibrate the algorithm using the recorded data of
the first dataset since these sequences were too short. Therefore three new lunge executions were recorded for the same subject using the Kinect™, while making sure that the NITE algorithm had sufficient time to calibrate. On average these recordings last 3.83 seconds and contain 115 frames.

4.2 Methodology

In order to estimate the joint positions, the data points recorded with the Kinect™ were presented as input to our algorithms. However, before a side by side comparison with other systems could be made, some postprocessing was performed on the estimated poses. For each signal representing a coordinate over time, wavelet denoising was performed to reduce potential noise and the y and x-axis were shifted to align the approaches being compared. This was necessary to synchronize time and align the origin of the coordinates. It does not change the underlying signal and is only relevant for comparison purposes. The position of the knee was shifted from the centre of the base of the cylinder to the lowest point of the base as shown in figure 1. Experiments showed that the resulting knee position corresponds better to the gold standard. This makes sense since the point cloud only captures the front side of the leg. The cylinders are therefore situated a bit above the actual limb they represent.

Fig. 1: The corrected knee position. The position is shifted from the centre of the base of cylinder (blue dot) to the lowest point of the base (green dot).

For the non-hierarchical particle filter 200 particles were used. In each frame 12 iterations were performed to determine the center of gravity of the person in the scene, and five to find the positions of the limbs. For the hierarchical particle filter 200 particles were used as well. Here 12 iterations were also executed to determine the position of the person, but seven iterations were performed for each limb. These parameters were chosen experimentally since they provide us with satisfactory results while still having decent execution times.
Since calculating weights in our approaches reduces to counting points inside cylinders this computationally expensive step was parallelized. Since there are no dependencies between the checks for different points this could be optimized further to decrease the computation time.

In order to compare the different systems we will mostly look at the movement characteristics that can be detected in the graphs obtained by plotting the (estimated) positions of the knee and foot of the frontmost leg in the three planes of motion. We hope that the different lunge specific movements are reflected in these plots; that is, lifting one foot of the ground and lunging out, followed by lifting that same foot again and bringing it back to the starting position.

4.3 Results and Discussion

Experiment 1. We first compare our approach with the gold standard measurements obtained using the Vicon system. Four lunge executions were analyzed. Figures 2 and 3 show a comparison between the positions acquired by the Vicon system and the positions for the knee and foot obtained from the depth data using our algorithm for one of the lunge executions. The results of both the hierarchical and non-hierarchical implementations are shown.

Looking at the shapes of the graphs allows us to recognize the movement characteristics of a lunge. Figure 2 shows the knee movement. The subject is facing towards the Kinect™. When he starts the lunge, the knee is lifted and brought closer to the camera until a certain minimum is reached and roughly the same path is followed back to the initial position. This can be seen on the rightmost graph which shows the depth of the knee. Besides the depth also the height can be explained quite intuitively. First, the curve goes upwards as the leg is lifted up to move forward. Next, the curve goes down, which happens as the foot is being put back on the ground. The minimum of the graph is reached as the subject bends forward. The second peak corresponds to lifting up the leg to go back to the initial position. It is interesting to note the scales of these movements: the range of the depth variation is roughly 1 m, the range of the height variation roughly 15 cm and the range of the lateral variation is only 3 cm. The latter partly explains why approaching the curve for lateral variation is much more difficult. A similar description can be given for the foot movements (see Figure 3). For example, the first peak in the height plot can be explained by the lifting of the foot. The curve ends with a peak which again corresponds to lifting the foot to go back to the starting position.

Although the graphs obtained using the particle filters oscillate much more than the Vicon measurements, most of the previously discussed key characteristics can be found in these reconstructions. The similarity to the Vicon measurements can be seen in all the curves, except for the one showing the lateral movement of the foot. However, some distinctions can be made between the non-hierarchical and hierarchical particle filter. First, the hierarchical particle filter always results in lower knee height minima (middle column of Figure 2). Here the non-hierarchical particle filter agrees better with the Vicon data. On the other hand, the hierarchical particle filter is more consistent in the height
results of the foot. For example, in Figure 3 the non-hierarchical curve shows a drop between the two high peaks, which would mean the foot going lower than the floor. In this case the hierarchical particle filter seems to perform best.

Finally table 1 gives a quantitative comparison of both approaches with respect to the Vicon system using the mean squared error metric.

<table>
<thead>
<tr>
<th></th>
<th>X</th>
<th>Y</th>
<th>Z</th>
</tr>
</thead>
<tbody>
<tr>
<td>knee</td>
<td>784.8</td>
<td>4,260.5</td>
<td>268,632.5</td>
</tr>
<tr>
<td>hierarchical</td>
<td>1,412.5</td>
<td>6,894.7</td>
<td>292,512.5</td>
</tr>
<tr>
<td>foot</td>
<td>1,798.9</td>
<td>2,714.1</td>
<td>216,992.5</td>
</tr>
<tr>
<td>hierarchical</td>
<td>2,492.2</td>
<td>3,046.7</td>
<td>198,587.5</td>
</tr>
</tbody>
</table>

Table 1: Mean squared error of foot and knee positions in $mm^2$ w.r.t. to the Vicon system.

Fig. 2: Comparison of knee positions during a lunge execution (particle filter vs. Vicon). Top row: raw results, bottom row: smoothed using wavelet denoising. Left column: lateral position, middle column: height, right column: depth.
Fig. 3: Comparison of foot positions during a lunge execution (particle filter vs. Vicon). Top row: raw results, bottom row: smoothed using wavelet denoising.

Left column: lateral position, middle column: height, right column: depth

**Experiment 2.** As a second experiment we compare our algorithms to the results obtained with the NITE skeleton tracker, which is used in the OpenNI framework. The second data was used for this purpose in order to make sure that NITE was calibrated properly before recording. Three lunge executions were analyzed here. Figures 4 and 5 are similar to those in the previous section, except that the magenta curve now depicts the NITE results. A first observation is that most of the characteristics that were found in the graphs in the previous section are also found here. One difference between our algorithm and NITE is that the lowest knee height is consistently estimated lower by NITE. A more important difference however is found in the estimate of the foot position. Whereas our algorithm identifies the two large peaks corresponding to lifting the foot in the beginning and at the end of the lunge, the curve obtained with NITE does not match the expected shape at all.

Our particle filters were also compared visually with the NITE algorithm by looking at skeleton estimates in various frames in the sequences. The skeleton estimates of both approaches are plotted together with the point clouds of the sequences (from which the positions were estimated) in Figures 6 and 7. This
reveals some properties that are not clear from the previously shown graphs. Figure 6 shows frame 10, 30, 40, 50, 60, 70, 90 and 100 from the third lunge sequence with cylinder correction applied to our estimate. In blue it shows the skeleton estimate of NITE and in white it shows the fitted skeleton of the hierarchical particle filter. Figure 7 shows the same figures, but here the white skeletons are the estimates of the non-hierarchical particle filter.

In the previous graphs we always analyzed the leg that moves forward during the lunge, because that is the one that is expected to yield the most interesting curves. In these figures both legs can be compared. First, from Figure 6 it is clear that the hierarchical particle filter generally yields better estimates than the NITE tracker. This was not clear from the movement graphs, but several frames show estimates of the NITE algorithm that are far from the expected skeleton configuration. Such failures were not observed for the hierarchical particle filter. These observations are consistent over the three lunge sequences that were compared in the previous section: the NITE algorithm loses track in at least a couple of frames, whereas the hierarchical particle filter does not make wildly incorrect estimates.

![Comparison of knee positions during a lunge execution (particle filter vs NITE). Top row: raw results, bottom row: smoothed using wavelet denoising. Left column: lateral position, middle column: height, right column: depth](image)

**Fig. 4:** Comparison of knee positions during a lunge execution (particle filter vs NITE). Top row: raw results, bottom row: smoothed using wavelet denoising. Left column: lateral position, middle column: height, right column: depth
The conclusion for the non-hierarchical particle filter is comparable, although the hierarchical particle filter agrees better with the expectation most of the time. However, the non-hierarchical particle filter does also show some severe failures when it comes to tracking the rear leg.

The main disadvantage of our approach is the execution speed: the NITE algorithm runs in real time, whereas our approach requires about 9 seconds per frame (for the number of particles and iterations used in these experiments). Further optimization is therefore an important next step. When it comes to accuracy the particle-based approach outperforms NITE, as the figures in the previous section demonstrate. Another advantage of our approach is that it can be extended to incorporate exercise dependent knowledge to improve the accuracy. This is not easy to achieve with the NITE algorithm, since it is not open source and no details about it are published. Incorporating knowledge might be easier in the non-hierarchical approach. Here, every particle corresponds to an entire lower limb configuration. This might make it easier to embed certain constraints into the algorithm. Another advantage of our algorithm is that it does not require calibration.
Fig. 6: Skeleton estimates after cylinder correction, hierarchical particle filter (particle filter in white, NITE in blue).

Fig. 7: Skeleton estimates after cylinder correction, non-hierarchical particle filter (particle filter in white, NITE in blue)
5 Conclusion and Future Work

Being able to accurately monitor patients without the need for a lab would allow for better individualized and more efficient treatment of patients. In this study two algorithms based on particle filtering were presented to obtain accurate tracking of the lower limbs. The first estimates the configuration of all lower limbs at once while the second makes an estimate for one limb at a time. By first recording lunges, these two approaches were compared with gold standard measurements obtained with the Vicon camera system. This comparison allowed us to conclude that most key movement characteristics can be seen in the reconstructions obtained using particle filters. Only the lateral movement of the foot is a challenge because its range is close to the point cloud resolution. Currently, the accuracy of the lateral movement is insufficient for applications such as osteoarthritis monitoring but with next hardware generations using higher resolutions, we envision that this will get within reach of what is possible.

Compared to the NITE skeleton tracker the particle filter approach is better at tracking the height of the foot. Furthermore by overlaying the skeletons obtained with both methods it was found that the particle filter yields more stable tracking results. This improvement in tracking quality, however, comes at the cost of an increased computation time.

In future work we plan to generalize these algorithms further. Several extensions can be envisioned to make the estimation faster and more accurate. A first approach is to incorporate exercise dependent knowledge into the system model. This would result in a smarter distribution of samples, allowing for a reduced computation time or increased accuracy, or both. Second, the algorithm could be extended to use both RGB and depth data to make the algorithm faster and more robust. Third, the skeleton model may be improved by using more elaborate shapes instead of simple cylinders.

Since similarity is already found in the curves of the lateral knee movement, we plan to investigate whether the obtained accuracy with the next generation Kinect™ will be sufficient for applications such as osteoarthritis monitoring [7].

Acknowledgements

TODB is partially supported by the research fund KU Leuven (CREA/11/015), and EU FP7 Marie Curie Career Integration Grant (#294068). JD is partially supported by the research fund KU Leuven (CREA/11/015 and OT/11/051), and EU FP7 Marie Curie Career Integration Grant (#294068).

References


12. Orhan, E.: Particle filtering (2012), research note

