A Mobile Low-Cost Motion Capture System based on Accelerometers

Jan-Phillip Tiesel Jörn Loviscach

Universität Bremen Hochschule Bremen

Abstract. Low-cost accelerometers can be employed to create a motion-capture solution for below US$ 100. It may be used in mobile settings employing an MP3 player or a similar recording device to capture the analog data of 15 degrees of freedom. The solution is integrated with standard 3D animation software. We introduce methods to extract and tweak kinematical as well as timing data from these acceleration sensors, which are attached to an actor's limbs. These methods take care of the fact that the measured acceleration data alone can neither provide complete nor accurate information to satisfactorily reconstruct the captured motion. Particular emphasis is placed on the ease of use, in particular concerning the calibration of the system.

1 Introduction

To create believable and lifelike motion manually by keyframing poses requires highly skilled animators and generally consumes hours of work for seconds of animation. In contrast to that, motion capture technology is theoretically able to provide such data in real time. However, commercial motion capture systems are too expensive for small film or game studios, not to speak of amateurs. These systems must be operated by trained professionals and usually take an hour or longer to be set up and calibrated, which renders them essentially immobile.

Inertial sensors in the form of micromachined accelerometers and miniaturized gyroscopes are widely available today. In order to fathom the capabilities of an ultra low-cost approach, we decided to exclusively use accelerometers as sensors. Note that whereas mathematically one could integrate the acceleration data twice to uncover position data, practically such an integration would blow up the data within seconds due to tolerances and noise.

This work demonstrates two approaches to how sensor data of accelerometers can be interpreted and edited in a standard animation software package:

1. The orientation of bones is estimated using acceleration sensors that are attached to an actor's limbs. By using the fixed direction of gravitational acceleration, profound assumptions on the posture of body segments can be made if the actor acts out the motion with no jerky movements.
2. The magnitude of a sensor's acceleration is used to retime existing keyframe animation.

Both approaches can be combined to first capture the orientation in slow-motion and then capture the timing in a second pass done at actual speed. As the interpretation of the accelerometer data will most likely contain errors and will not yield an accurate and complete description of the initial motion, we equip the user with ways to refine and augment the recorded data.

The overall capturing process can be broken down into five steps:

1. Accelerometer signal range calibration (once in the lifetime of a sensor)
2. Calibration of the orientation of the accelerometers relatively to the limbs (once per capturing session)
3. Real-time capture of estimated limb orientations (only slow-motion movements to minimize errors due to manual acceleration)
4. Smoothing of capture data and creation of the kinematical chain representation in the animation software
5. Re-enacting the previous motion with proper timing to retime the kinematical data (only if fast movements occurred during step 3)

The main contributions of this paper to the state of the art in motion capture are the mobile low-cost motion capturing system, the easy-to-use calibration methods, the drift-free and robust user-assisted interpretation of orientation data, and the retiming process assisted by acceleration data.

This paper is structured as follows: Section 2 provides an overview of related work. Section 3 introduces the special hardware we created. The calibration of the system is covered in Section 4. Section 5 describes the process employed to compute orientation data and apply them to a 3D model. How acceleration data can be employed to retime animations is treated in Section 6. Section 7 presents results; Section 8 concludes this paper.

2 Related Work

An overview of the research in vision-based motion capturing is given by Moeslund and Granum [1]. Fewer cameras or alternative low-dimensional controllers have been successfully used to synthesize new animations from existing motion capture data [2, 3]. Whereas they succeed in creating realistic motion representations, they lack the ability to operate low-cost solutions with unlimited measurement volume. The decreased complexity in the hardware setup often leads to an increase in the complexity of system operation (e. g., the need for building an extensive motion capture database before the capturing process). The overall price of the used hardware could be lowered but is still in the range of several hundred to several thousand US$.

Mechanical motion capture employs sensors attached to the actor's body to estimate joint angles. Those systems are free of occlusion effects and often utilize wireless data transfer to permit mobile applications. While the *ShapeWrap*™ system by Measurand Inc. uses a flexible tape that bends in accordance with the actor's limb movements, inertial sensors are used by Animazoo in their *Gypsy-Gyro* suit. Both manufacturers state that the setup and calibration time as well

as the essential user training are minimal compared to vision-based solutions. Up to date those systems are not available for less than US$ 20,000 and US$ 60,000, respectively.

Zhu and Zhou [4] used MARG sensors (measuring *m*agnetic field, *a*ngular *r*ate and *g*ravity) to determine rigid body orientation. An extensive amount of research was done using inertial sensors for medical applications like gait analysis [5]. Mayagoitia et al. [6] presented a combination of accelerometer and gyroscope measurements to obtain kinematic parameters such as knee angle or thigh linear acceleration. Luinge and Veltink [7] investigated the maximum amount of information that can be extracted from a tri-axis accelerometer.

Hansson et al. [8] pointed out two of the major limitations of orientation estimation by the exclusive use of accelerometers. First, rotation about the line of gravity cannot be detected; second, the ability to measure rotations about the object's axes depends on its overall orientation. The work of Giansanti et al. [9] indicated that the reconstruction of both orientation and position from only accelerometer data does not yield feasible results. Rehbinder and Hu [10] proposed fusing data from rate gyroscopes and accelerometers to give long-term, drift-free attitude estimates. A combination of computer vision motion tracking and inertial tracking using the Kalman filter was presented by Welch [11].

One of the main challenges that animators face is getting the timing of the overall motion to be as lifelike and believable as possible. Terra and Metoyer [12] presented an approach that allows the user to draw a sketch of the desired motion using the proper timing. They then used the timing information of the user's sketch to retime an existing keyframed animation. Thorne et al. [13] implemented an animation system based entirely on sketching out the motion and using the kinematical and temporal information to reproduce the completed motion sequence.

3 Sensor Hardware and Data Flow

The sensor hardware (see Figure 1) consists of a battery-powered central circuit board wired to five boxes; each box contains an accelerometer chip LIS3L02AQ from STMicroelectronics. This micromechanical device measures acceleration along three perpendicular axes and outputs three corresponding voltages.

The central circuit board contains a clock generator chip (NE 555), a counter (MOS 4024), and an analog 16:1 multiplexer (MOS 4067). The 15 output voltages of the accelerometers are connected to the inputs of the multiplexer. A reference voltage, stabilized by an operational amplifier (one quarter of RC 4136), is connected to the 16^{th} input. The timer and the counter are used to continuously cycle the multiplexer's output through the 16 inputs, so that each appears 60 times per second.

The output is decoupled through another operational amplifier (one quarter of RC 4136) and fed into the left audio input channel of the PC's sound card, see Figure 2. The most significant (i. e., most slowly varying) bit of the counter is fed into the right channel. On the PC, our software synchronizes with this 60 Hz

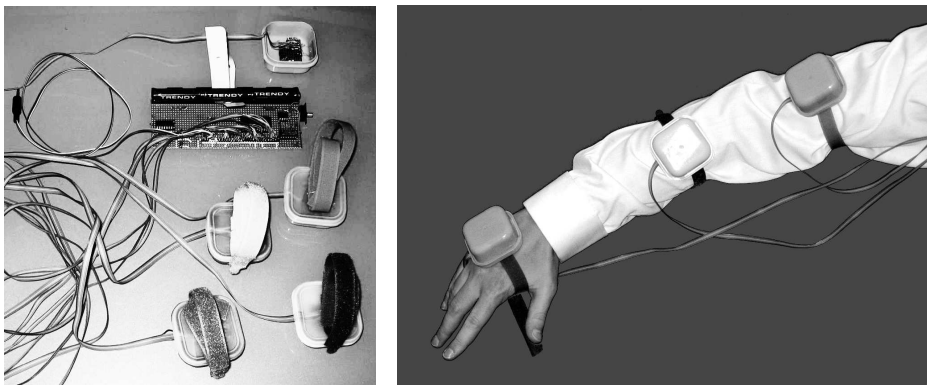

Fig. 1. The hardware interface consists of five tri-axis accelerometers and a main board connectable to an analog stereo audio input

signal and processes the switched signals from the left channel in real time. Per cycle, the software forms 16 time slots over which the signal is averaged. This helps to suppress the overshooting and ringing transients introduced by the Nyquist low-pass filter of the sound card. As standard PC sound cards do not reproduce DC voltages, we recover them by forming differences with the measurement for the reference voltage, which forms the 16^{th} input.

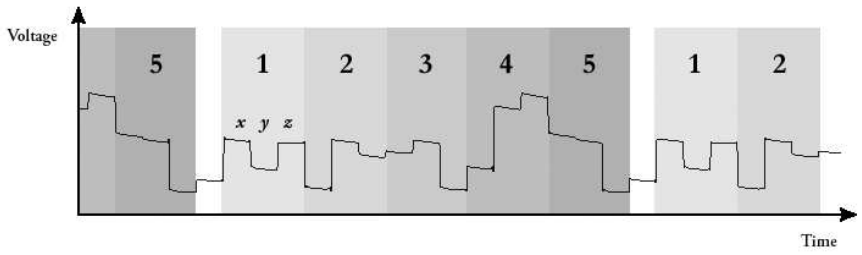

Fig. 2. The analog signal transmitted via one audio channel comprises the xyz values of the five sensors and a reference voltage

Optionally, the analog two-channel output signal can be recorded by an audio recording device such as an MP3 player that has recording capabilities for uncompressed wave signals. The data thus stored can be transferred to a computer after the capturing process is completed.

A stand-alone .NET-based software is responsible for the initial processing of the input data. During the real-time processing, the data are saved to a file that also holds general information about the capture process such as sampling rate, number of active sensors, and sensor naming. To import, edit and utilize the captured data with a standard 3D software package, a library of procedures

was created in MEL, the scripting language of Autodesk Maya®. It provides the user with a graphical interface that is fully integrated with the animation package.

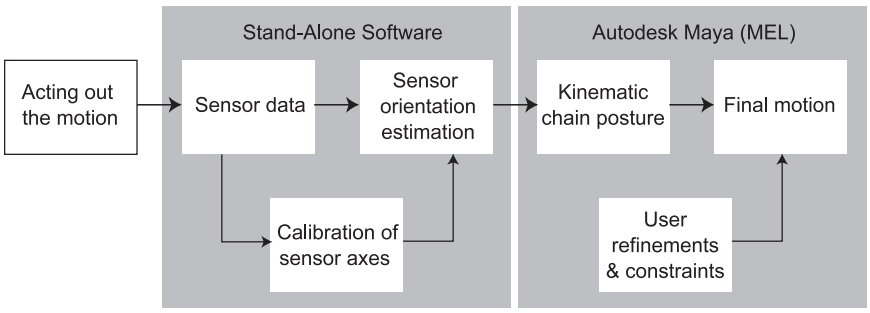

Fig. 3. A stand-alone data processor delivers the recorded motion to a set of MEL scripts

4 Calibration

Several types of coordinate frames occur within the system: There is a global one, the z axis of which points away from the earth's center; a coordinate frame for every limb or bone with its origin placed at the pivot point and its x axis pointing along the bone; and a coordinate frame for every sensor, the xyz axes of which are aligned to the sensor's ones, see Figure 4. Most of the computations deal with conversions between these three systems; the objective is to determine the relation between the limbs' systems and the global one. We use the subscripts "global," "limb," and "sensor" to specify the reference frame of a given vector quantity.

The accelerometers display combined offset and range errors of about 10 percent of earth's gravity. A huge part of this error can be corrected using a calibration step to be executed once in the lifetime of an accelerometer: The lowest and highest emerging values of the xyz components of the measurement are established while the user slowly rotates the sensor about all of its axes. Short peaks are suppressed to cancel quick motion and only capture the acceleration due to gravity. The resulting minimum and maximum values per component are used to linearly normalize the measurement range. After that, measurements done with a static sensor will yield values from the interval $[-1, 1]$, where 1 corresponds to a perfect alignment of the corresponding axis with the direction of gravity, that is, the negative direction of the global z axis.

The coordinate frames of the sensors typically do not match those of the limbs to which they are attached. This discrepancy requires a calibration that is to be executed every time the sensors are mounted. One has to establish a rotation matrix $R_{\mathrm{l}\leftarrow\mathrm{s}}$ that transforms the sensor's axes to the limb's axes.

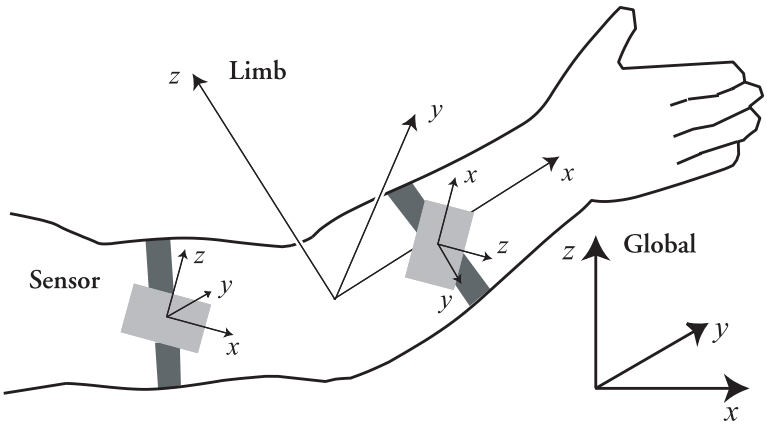

Fig. 4. Three types of reference frames are used in the computation

To understand how this rotation matrix can be found easily consider a sensor that is attached to an actor's lower arm. Measure the sensor data when the actor extends his arm horizontally and vertically one after the other. Let $\boldsymbol{a}_{\text{sensor}}$ and $\boldsymbol{b}_{\text{sensor}}$ be the corresponding results, corrected to become perpendicular unit vectors, what they should be in theory. The desired rotation matrix $R_{\text{l}\leftarrow\text{s}}$ transforms these vectors into the corresponding gravity vectors in the limb frame: $(0, 0, -1)^{\text{T}}$ and $(1, 0, 0)^{\text{T}}$, respectively. This matrix can be computed as

$$R_{\text{l}\leftarrow\text{s}} = \left[\boldsymbol{b}_{\text{sensor}}, -\boldsymbol{a}_{\text{sensor}} \times \boldsymbol{b}_{\text{sensor}}, -\boldsymbol{a}_{\text{sensor}}\right]^{\text{T}}, \tag{1}$$

where the matrix on the right hand side is formed from the three given column vectors.

The data gained in the calibration phase can be used to convert every acceleration measurement from the corresponding sensor frame to the limb frame. Note that since we are dealing with derivatives, we need not bother about the placement of the origin of a frame and are only concerned with its orientation.

5 Estimating and Using Limb Orientation Data

The main task in the motion capture process is to use the acceleration data given in the limb systems to determine the orientation of these systems within the global system. In this mode, we expect the actor to move slowly, so that mostly the acceleration due to gravity will be measured. This way, the accelerometers can be used as inclinometers. Even if the actor applies an acceleration of half the acceleration of gravity, the error in orientation is limited to 30 degrees and will mostly be about 15 degrees. On top of that, an acceleration that large cannot be sustained in linear direction for more than half a second, due to the limited position range of the body's limbs.

If one assumes slow motion, the calibrated measurement output of a sensor equals $\boldsymbol{a}_{\text{limb}}$ and has approximately a length of one. To adjust the limb in the 3D software, we're seeking a matrix $R_{\text{g}\leftarrow\text{l}}$ that transforms the global orientation into that of the limb. In particular, $R_{\text{g}\leftarrow\text{l}}$ has to map $\boldsymbol{a}_{\text{limb}}$ to the global direction of gravity $(0, 0, -1)^{\text{T}}$.

There is an infinite number of rotations with this property. This stems from the fact that static rotations about the global z axis do not change the gravity field. Thus, we aim to find the simplest possible rotation and offer the user a control on the remaining single degree of freedom. The simplest rotation is the (unique) one with the shortest turning angle. Its axis is perpendicular to both vectors $\boldsymbol{a}_{\text{limb}}$ and $(0, 0, -1)^{\text{T}}$ and thus can be found through their cross product. Its angle ϕ can be determined through

$$\phi = \operatorname{atan2}\left(\sqrt{a_x^2 + a_y^2}, a_z\right). \tag{2}$$

Typically, the sensors used during recording are part of a kinematical chain. By creating a skeleton rig according to the user's specifications, the connections of limbs and their according sensors are defined. The individual position of the root joint(s) and the length of every bone can be easily edited with the user interface provided by our scripts.

The aim constraint functionality of the Maya software is used to transfer the captured orientation information to the individual limbs. The first column of every matrix $R_{\text{g}\leftarrow\text{l}}$ specifies the global orientation of the corresponding limb's x axis. Adjusting these vectors by the limb's length and adding them along the kinematical chain provides the targets required as aim constraints.

Due to the choice of the simplest rotation, there is not yet a rotational component about the global z axis. To compensate for this, the user can manually animate the joint's rotation about the global z-axis. This is easily accomplished by concatenating this z rotation with $R_{\text{g}\leftarrow\text{l}}$.

6 Animation Retiming

In a second approach that can be used independently or in combination with the presented method of orientation estimation, we utilize the acceleration magnitude of a sensing unit to retime an existing motion. This can be either a hand-keyframed motion or a motion sequence previously captured by the accelerometers. The overall timing of the actor's motion is applied by fitting the acceleration curve of a given animated object over time to the measured acceleration information of the sensor(s). The instances of time where the acceleration reaches peak values serve as references for the fitting process.

The user chooses a keyframed 3D object. Its acceleration $d^2\boldsymbol{x}(t)/dt^2$ is determined numerically from its position $\boldsymbol{x}(t)$. Our MEL script determines peaks by looking at the slope of $|d^2\boldsymbol{x}(t)/dt^2|$ and inserts "peak markers" at those times. The script proceeds similarly for the measured acceleration data $\boldsymbol{a}_{\text{sensor}}$: It places peak markers at times where

$$\big||\boldsymbol{a}_{\text{sensor}}| - 1\big| \tag{3}$$

attains a locally maximal value. Note that no calibration of the sensors to the limbs is necessary since we are only interested in the magnitude of the acceleration. The subtraction of 1 approximately compensates for the fact that the measured acceleration data include gravity, in contrast to the acceleration data from the object's keyframed motion in the 3D software.

The user can manipulate the established peaks of both curves independently using the keyframe editing tools supplied by the 3D animation software. Once the peaks of the two curves are put into a one-to-one correspondence, the script can redistribute all of the object's keyframes using linear interpolation. Keyframes situated at acceleration peaks of the original motion curve are placed at the time of the corresponding peak of the sensor data curve.

7 Results

The calibration procedure for establishing the rotation matrices for the individual sensors is easy and fast. It typically takes just a few seconds per limb or even less if several sensors are part of the same kinematical chain and can be calibrated simultaneously (e.g., upper arm, forearm, hand).

We evaluated the system's precision using two different kinds of measurement. After an accurate calibration which secured perpendicular calibration vectors, the inclination angle of the estimated sensor orientation was compared to a manual measurement. This method showed a maximum error of 4 *deg* with the average error being smaller than 2 *deg*. The second method included manual and therefore less precise calibration by the actor wearing the sensor. This time, the estimated inclination angle was compared to angle measurements established by visual reference (digital photographs taken during the capturing process). As expected, the manual calibration resulted in higher error values. Regarding the quick and rudimentary calibration, the results are satisfying, yet amendable (average error of about 6 *deg* and maximum of 13 *deg*).

Several well-defined motion sequences of the upper torso with sensors attached to the actor's arm, forearm, and hand were executed and captured. The resemblance of the physical motion to the synthesized representation depended largely on the amount of rotation about the axis of gravity. No movements can be captured while limbs are moved on a plane perpendicular to the line of gravity (e. g., on a table), what requires a lot of user intervention to supply the missing rotation data. In contrast, movements along a vertical plane (e. g., swing of arms during walking, arm waving, arm reaching out for a glass of water, legs walking) are reproduced with good accuracy, see Figure 5.

Motion that implies high acceleration (e. g., ground contact during walking; fast or sudden movements in sports or action sequences) have to be acted out in slow motion. As no drift or offset error is introduced during periods of high acceleration, the user can either retime the captured slow motion sequence or act out the fast motion and fix kinematical errors that occur due to acceleration peaks in the Maya software.

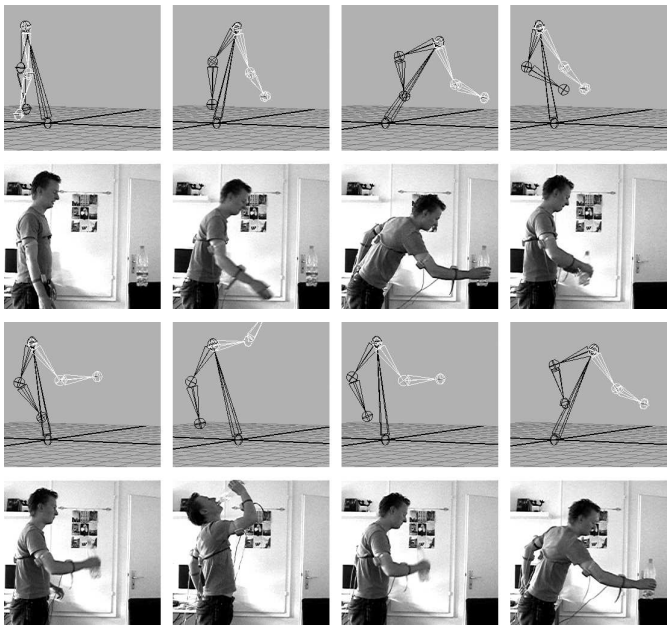

Fig. 5. Motion with vertical components can be captured well, often at its actual speed. The white arm in the 3D view is the one that is closer to the camera

The animation retiming method works reliably if the captured motion implies the occurrence of well-defined acceleration peaks (e. g., waving). Often, the actor has to exaggerate his movements in order to produce clearly identifiable acceleration peaks. In addition, the approach implies that the original keyframes need to have proper relative timing between acceleration peaks as the keyframes are distributed using linear interpolation.

8 Conclusion and Future work

We have presented a motion capture system that requires only inexpensive special hardware and can easily be used in mobile settings. The system is robust in that errors do not accumulate in time or along kinematical chains. However, the usefulness of the orientation measurement depends largely on the type of motion to be captured.

An obvious solution to the indeterminacy of the rotation about the z axis would be to bundle each accelerometer with a tri-axis electronic compass and fuse the data of both sensors more or less intelligently. The compasses would raise the system cost significantly (by around US$ 60 per sensor).

Ways to improve the orientation sensing with or without compasses range from data filtering [14] to statistical approaches that determine the most likely orientation given past (and, if recorded, future) sensor data as well as typical

motion sequences. Such an approach may borrow many ideas from visual motion capture techniques such as [2].

In order to improve the presented animation retiming method, a curve matching algorithm such as Bruderlin's dynamic time warping [15] could be employed instead of redistributing the existing keyframes in a linear fashion.

References

1. Moeslund, T.B., Granum, E.: A survey of computer vision-based human motion capture. Computer Vision and Image Understanding: CVIU **81** (2001) 231–268
2. Chai, J., Hodgins, J.K.: Performance animation from low-dimensional control signals. ACM Transactions on Graphics **24** (2005) 686–696
3. Lee, J., Chai, J., Reitsma, P.S.A., Hodgins, J.K., Pollard, N.S.: Interactive control of avatars animated with human motion data. In: SIGGRAPH '02: Proceedings of the 29th annual conference on Computer graphics and interactive techniques, New York, ACM Press (2002) 491–500
4. Zhu, R., Zhou, Z.: A real-time articulated human motion tracking using tri-axis inertial/magnetic sensors package. IEEE Transactions on Neural Systems and Rehabilitation Engineering **12** (2004) 295–302
5. Sabatini, A.M.: Quaternion-based strap-down integration method for applications of inertial sensing to gait analysis. Medical and Biological Engineering and Computing **43** (2005) 94–101
6. Mayagoitia, R.E., Nene, A.V., Veltink, P.H.: Accelerometer and rate gyroscope measurement of kinematics: an inexpensive alternative to optical motion analysis systems. Journal of Biomechanics **35** (2002) 537–542
7. Luinge, H.J., Veltink, P.H.: Inclination measurement of human movement using a 3-d accelerometer with autocalibration. IEEE Transactions on Neural Systems and Rehabilitation Engineering **12** (2004) 112–121
8. Hansson, G., Asterland, P., Holmer, N., Skerfving, S.: Validity and reliability of triaxial accelerometers for inclinometry in posture analysis. Medical and Biological Engineering and Computing **39** (2001) 405–413
9. Giansanti, D., Macellari, V., Maccioni, G., Cappozzo, A.: Is it feasible to reconstruct body segment 3-D position and orientation using accelerometric data? IEEE Transactions on Biomedical Engineering **50** (2003) 476–483
10. Rehbinder, H., Hu, X.: Drift-free attitude estimation for accelerated rigid bodies. Automatica **40** (2004) 653–659
11. Welch, G.F.: Hybrid self-tracker: An inertial/optical hybrid three-dimensional tracking system (1995)
12. Terra, S.C.L., Metoyer, R.A.: Performance timing for keyframe animation. In: SCA '04: Proceedings of the 2004 ACM SIGGRAPH/Eurographics symposium on Computer animation, New York, ACM Press (2004) 253–258
13. Thorne, M., Burke, D., van de Panne, M.: Motion doodles: an interface for sketching character motion. ACM Transactions on Graphics **23** (2004) 424–431
14. Luinge, H.J., Veltink, P.H.: Measuring orientation of human body segments using miniature gyroscopes and accelerometers. Medical and Biological Engineering and Computing **43** (2005) 273–282
15. Bruderlin, A., Williams, L.: Motion signal processing. In: SIGGRAPH '95: Proceedings of the 22nd annual conference on Computer graphics and interactive techniques, New York, ACM Press (1995) 97–104