

Action Capture with Accelerometers

Ronit Slyper and Jessica K. Hodgins

Carnegie Mellon University

Abstract

We create a performance animation system that leverages the power of low-cost accelerometers, readily available motion capture databases, and construction techniques from e-textiles. Our system, built with only off-the-shelf parts, consists of five accelerometers sewn into a comfortable shirt that streams data to a computer. The accelerometer readings are continuously matched against accelerations computed from existing motion capture data, and an avatar is animated with the closest match. We evaluate our system visually and using simultaneous motion and accelerometer capture.

Categories and Subject Descriptors (according to ACM CCS): I.3.7 [Computer Graphics]: Three-Dimensional Graphics and Realism – animation. I.3.6 [Computer Graphics]: Methodology and Techniques – interaction techniques.

1. Introduction

Accelerometers are tiny, accurate, and low-cost MEMS devices. These properties make them ideal for a wearable performance animation system. The obvious approach of integrating acceleration to compute position fails because of drift. We present a system for performance animation that harnesses the power of accelerometers in a different way: accelerometer readings are matched against a database of accelerations computed from motion capture, and the closest clip is played. We call our system “action capture” to delimit its strengths and weaknesses: motion reconstruction works well when the user is performing repeatable, easily recognizable motions, such as martial arts moves or jumping jacks. It does less well on motions where the accelerations do not have a consistent pattern, e.g. casual gestures of a lecturer.

Our secondary goal in this work is to demonstrate how *comfortable* performance animation garments can be built easily and cheaply using materials from the emerging field of e-textiles, and microcontrollers from hobby electronics. The action-capture shirt we created, with five accelerometers sewn in, was made for about \$200 using only off-the-shelf parts.

We will next discuss work related to our performance-animation solution, and describe the techniques that we

have borrowed from the field of e-textiles. Section 3 details our hardware construction and our software system with its wavelet-matching algorithm. Our results are best demonstrated by the accompanying video, but we also provide a numerical comparison to simultaneous motion capture accelerations in Section 4. In Section 5, we conclude with a discussion of the limitations of our system, and possibilities for future work.

2. Background

This section describes related work in motion capture, performance animation, and accelerometer-based capture.

Recent work has shown how we might build cheaper and more portable hardware for motion capture. Vlasic et al. [VAV*07] created an impressive, portable system with acoustic-inertial trackers: accelerometers, gyroscopes, and an acoustic subsystem to measure distances between limbs. The acoustic sensors are necessary to prevent the drift inherent in the double-integration of accelerations. Our system avoids this problem by comparing against motion capture accelerations (i.e. not integrating). Raskar et al. [RNd*07] used photosensing markers. The FootSee system [YP03] used an XSensor pad to capture foot-ground pressure distributions, and matched these to an existing database of simultaneous motion capture and foot pressure. Our goal in this work is to

test the capabilities of off-the-shelf hardware and accelerometers. Thus we use neither custom parts nor more expensive sensors, such as the inertial sensors used in commercial motion capture systems [Mov08].

The time required to prepare for a motion capture session has long motivated the idea of capturing motion from video, e.g. [CMC*06], [RSH*05], and [Org08]. For example, Chai et al. [CH05] used a small set of markers and two video cameras to reconstruct full body motion. Their performance animation system used a database of motion capture, and at runtime built a locally linear model of poses close to the control signals and to previous poses to find the best match to the current marker locations. We similarly use a database in our matching algorithm, but perform no interpolation.

Our focus is on reconstructing human motion, but a related problem is mapping the motion of a manipulated object to the actions of a character on the screen. The Swamped! exhibit [JWB*99] demonstrated a *sympathetic interface* – an inviting plush toy whose actions were mirrored in the virtual world. The toy was outfitted with gyroscopes, flex, squeeze, and other sensors, and communicated via a wireless transmitter. Dontcheva et al. [DYP03] created a capture system based on motion-captured Tinkertoys which were automatically mapped to DOFs of the character being controlled.

Gaming with the Wii Remote [Nin08] can be regarded as a performance animation experience. The Wii Remote uses an off-the-shelf ADXL330 accelerometer, $\pm 3g$, the same used in our prototype. The Wii Remote allows the player to act *with* an object, performing and gesturing as if the object were an extension of his arm. For example, the player swings the Wii Remote as if he were playing tennis or launching a bowling ball. Our system is wearable and captures more degrees of freedom.

Gaming is not the only field to take advantage of the low cost of accelerometers. The medical community has used them to measure gait [RM02] [LCL*07], recognize falls in the elderly [BCHN07], and evaluate physical activity in obese children [PKB*08]. Similarly, in Wearable Computing, accelerometers are used to identify discrete activities [LWJ*03] [TIL07]. When actions are performed slowly, the orientation reading, rather than the movement, dominates, and poses and gestures can be inferred from these readings (modulo rotation around the axis of gravitation) [FBRA07] [TL06] [FDC03]. Lee et al. [LH99] propose a compensation technique for faster motions; they place two accelerometers at different points on a limb, and use geometry to subtract the acceleration due to motion. Realtime visual systems with accelerometers can also give feedback in motion training, e.g. in martial arts [KG05]. Our hardware framework would fit well in many of these applications.



Figure 1: Left. *The outside of the shirt shows the location of the accelerometers (circled in red).* Right. *The traces on the inside of the shirt are laid out to minimize crossings.*

2.1. E-textiles

E-textile research seeks to merge electronics with textiles to create comfortable and computational clothing. The field is closely allied with Wearable Computing, but takes the “wearable” one step further by not merely attaching computational devices, but *sewing* them into clothing with conductive thread and fabric. Post et al. [POR00] describe projects from the MIT Media Lab, including a fabric keyboard and musical MIDI jacket. The paper describes the materials and technique of “E-broidery”: embroidered textile circuits designed with CAD tools and sewn with conductive thread in embroidery machines. Buechley [Bue06] designed and made commercially available an e-textile kit called the “LilyPad” consisting of small sensors and a microcontroller with sewable leads [Spa08]. Her innovations in e-textiles have made the field accessible; she uses the kit as an educational tool to teach electronics to children. We use the commercially available accelerometers and conductive thread from her kit, as well as her sewing construction techniques.

The Georgia Tech Wearable Motherboard project [PMJ02] pioneered the paradigm of “fabric is the computer”. Their Smart Shirt is woven with optical fiber in a programmable network that can detect bullet wounds and transmit readings from other sensors for combat or medical monitoring. Other e-textiles platforms, e.g. [JLT03] and [HART08], have followed.

3. Methods

We have constructed a hardware and software system to demonstrate the feasibility of using accelerometers embedded in clothing for motion capture. The shirt streams data to the software system in real time, while the software does continuous searching of the motion capture database to control an avatar onscreen. We now describe our hardware construction and software matching algorithms.

3.1. Hardware

We used a popular electronics prototyping platform called Arduino, designed to be accessible to people with only min-

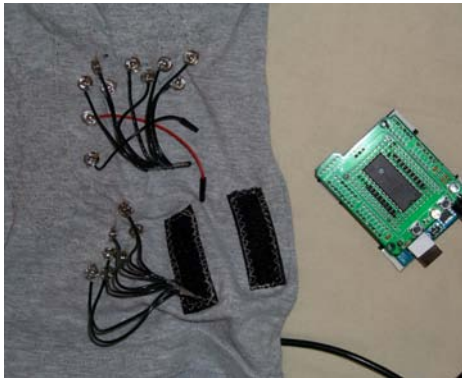


Figure 2: A close-up of the “hard” hardware. Wires are snapped onto the shirt, and the board secured with black strips of velcro. The arrangement is secure and removable for washing.

imal electronics experience. Arduino is open source, low cost, and easy to use. It consists of a microcontroller board with input and output pins, and a C++-like programming environment. Boards connect to a computer via USB, serial, or Bluetooth. Our project uses the \$35 USB board (Sparkfun SKU: DEV-00666). One could easily substitute a board with wireless connectivity, at higher cost. The Lilypad project sells a sewable version of the Arduino.

Our accelerometer shirt is built out of an existing shirt, conductive thread, five sewable Lilypad accelerometers (built on triple-axis ADXL330, $\pm 3g$), and the microcontroller board. Our first prototype tested the assumption that accelerometers give noisy readings; it had two accelerometers per limb. We found that each accelerometer provided similar readings, and the redundancy was not necessary.

Our second prototype, shown in Figure 1, contains one accelerometer on each forearm and each upper arm, and one on the chest. The traces were sewn using conductive thread in the bobbin of a standard sewing machine. Extra thread was left at the start and end of each trace to make connections; on one end the thread was sewn to a standard snap, on the other to an accelerometer. The sewing of connections took the bulk of the time. The traces were coated with fabric paint to reduce erosion of the thread when washing the shirt and to prevent electrical shorts. The shirt is form-fitting but not unreasonably tight; the accelerometers are light – each about the weight of a small paperclip – and do not induce any pull on the sleeves.

We like to think of our “hard” hardware as a black box, where the current best hardware solution can be inserted. We used an Arduino USB board with a snapped-on proto-shield and a 16:1 mux. Wires are soldered to snaps on one end and header pins on the other which connect to sockets next to the mux, as shown in Figure 2. The mux’s output

is connected to the Arduino board’s 10-bit analog-to-digital converter. Twenty lines of code loop over the mux’s inputs and send the results, together with a time stamp, to the emulated serial port at 115200 baud. A 5ms delay is inserted at the end of the loop; the software gets a full sampling of the accelerometers about every 8ms.

The accelerometers are calibrated by pointing them upwards (registering $-1g$), then flipping them over ($+1g$). From these numbers we obtain the scaling factor per g, and the zero-g value. This step only needs to be done once.

3.2. Software Matching

Our motion capture dataset consists of 4.5 minutes of motion, recorded and saved at 120 frames/second from a Vicon system.

Our software system is written in C++ and will run on any Unix-like platform. It consists of three threads which run constantly throughout the life of the program: serial, search, and display.

3.2.1. Calibration

For each motion capture clip, we compute the accelerations at the points (virtual sensors) on the skeleton that are in the same locations as the accelerometers on the shirt. The matching that will happen later on is sensitive to the accuracy of the virtual sensor calibration – the axes of the virtual sensors *must* be well-aligned to those of the accelerometers. An action-capture garment thus needs to be calibrated to the motion capture dataset.

We perform this calibration by having the user mimic a prerecorded sequence (Figure 3) while viewing a motion capture clip. Any motion may be used for calibration; several may be selected if the user wishes to calibrate the accelerometers individually.

An optimization process is then run. The user can manually adjust the virtual sensors until the acceleration graphs match, or he can set the process to run automatically. In the latter case, the user indicates which limbs contain accelerometers, and the software adjusts the positions and orientations of the virtual sensors until the computed motion capture accelerations line up well with the mimicked accelerations. Because this process needs to be done only once, we opt for a brute force approach, trying all angles and positions and then refining.

3.2.2. Search

The motion capture data is preprocessed to speed up searching. The software scans through all sliding windows in the dataset of accelerations, taking sets of length 128 frames – a total of 30,000 records, each of length 128 x 15-dimensions from the five 3-axis accelerometers. For each set, a Haar wavelet transform is computed on each dimension separately

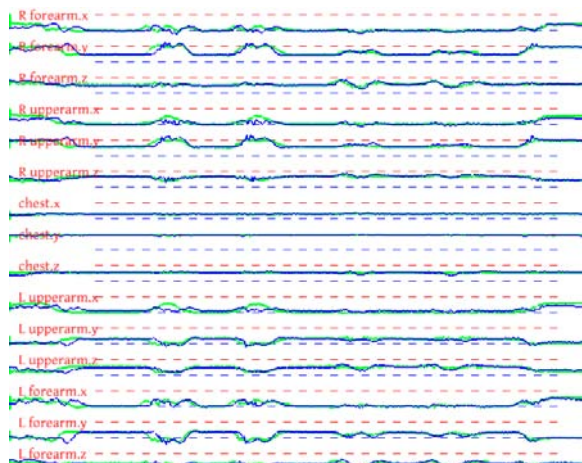


Figure 3: Calibration of the sensor locations. Time runs from left to right across the screen, 14 seconds total; graphs of the accelerometer axes are stacked vertically, with the dashed red and blue lines showing $+1g$ and $-1g$ for each axis. Blue lines show shirt accelerations from the actor performing the calibration motion. Green lines show the accelerations of the virtual sensors. The software (or user) adjusts the virtual sensors on the motion capture figure until the green lines overlay the pink, indicating that the shirt's accelerometers and the virtual sensors are aligned.

and the first 15 coefficients retained. These $15 \times 15 = 225$ coefficients are concatenated into a vector along with the joint positions of the pose in the 128th frame. This coefficient and pose vector will be compared against others using the L^2 -norm. We have found that including more than 15 coefficients from the wavelet transform yields little improvement in the search results. Including pose positions encourages continuity in the matching; we tuned the scaling factor on the positions so that continuity accounts for about 10% of the metric. The number 128 was selected based on fine-tuning; using too short a length loses the context of an action; too long limits the search. Preprocessing takes under a minute.

In the main program loop, the search takes the latest input accelerations, and computes the wavelet transform, as above, from the previous 128 frames. It concatenates onto this vector the joint positions of the last pose displayed, and then performs a linear scan through all of the preprocessed data to find the closest match. This computation completes in roughly 0.060 seconds with our dataset. Larger datasets would likely require a more sophisticated datastructure and search algorithm to maintain performance.

Once the search has found a best match, it sleeps until the total time elapsed between searches amounts to 10 frames (0.083 seconds). A delay in searching is key to reducing jerk-

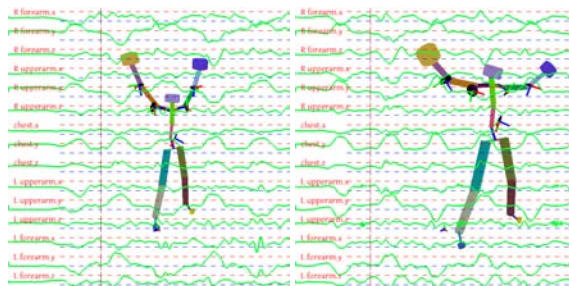


Figure 4: Two different motion capture subjects performing jumping jacks. Our approach relies on similarity of accelerations across subjects.

iness of the motion because it allows time for blending and prevents too frequent switching between matched activities.

When a new match is computed, the motion of the graphical character is blended across 10 frames to the new motion. The new motion clip is started at its 128th frame less 10 frames (the number that will play before the next search result comes in). The frame rate is set to accommodate the processing power of the computer; on a MacBook Pro 2.33 GHz, the system runs smoothly at 30 fps. When displaying the motion capture clips, we hold the legs fixed and zero the root position, as we are measuring and reconstructing only the torso and arm motion.

4. Results

Our system relies on the repeatability of accelerations, across motion capture sessions and across people. Figure 4 shows the motion-capture accelerations of two people doing jumping jacks, using the same virtual sensor calibration. The accelerations are similar, in both the left and right arms.

We used a Vicon motion capture system to test the accuracy of the accelerometer shirt. We performed a motion capture session while wearing the shirt; the results of overlaying the accelerations read from the shirt and computed from the motion capture database are shown in Figure 5.

The accompanying video shows our action capture system running with a database of 4.5 minutes of motions such as waving, jumping, and stretching. This corresponds to roughly 30,000 records that the search thread searches each loop; on a MacBook Pro 2.33 GHz, this takes about 0.060 seconds.

We have provided an “action capture” system that runs smoothly with a 0.083 second delay, which feels sufficiently reactive. This delay could be reduced to roughly 0.06 seconds, at the expense of jerkier motion and less blending.

In the accompanying video, we show simultaneous motion capture and accelerometer-shirt capture of a longer sequence of 35 seconds. The RMS distance in acceleration

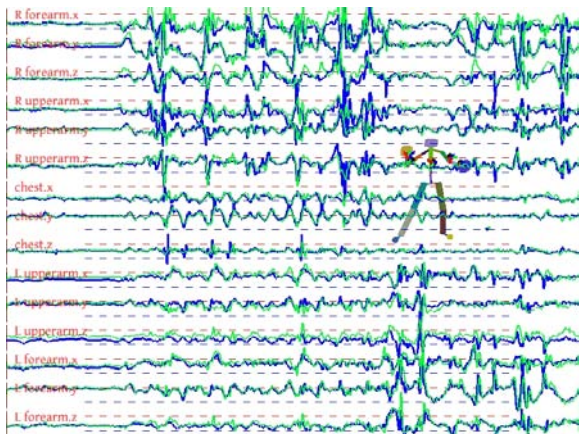


Figure 5: Accelerometer readings (blue) are shown overlaid with accelerations computed from simultaneous motion capture (green). The match is close, though not exact.

between the accelerometer readings and the motion capture accelerations is shown in Table 1. Alongside is shown the distance between the accelerometer readings and the accelerations of the motion capture segments that our system matched to these readings. Both are in the same range; slight variations occur from the matching algorithm, and from occasional jumps (shoulder pops) in the motion capture data.

In the third column of Table 1, we reduce the number of accelerometers used in the reconstruction to the two end effectors, the left and right forearms. We compare the shirt accelerations to the accelerations of the motion capture clips used in this reconstruction. The quality of the motion suffers; as expected, the RMS distance of the end effectors is smaller, but most of the other accelerations grow.

As the video shows, our system is good at picking out common static poses, and well-defined actions whose accelerations are repeatable. It performs poorly when actions are performed with different timing or orientation than those contained in the database. Without strictly enforced continuity, occasional jumps occur when accelerations are ambiguous. Nonetheless, we think that our system shows the power of the accelerometer approach.

5. Discussion and Conclusion

A natural question about this work is, will it extend to larger databases? We believe that as it stands, it will probably not, as adding more accelerations will muddle the search, since many poses and motions will have the same accelerations. For instance, when we added a few sitting motions to the dataset, flickering began to occur as poses from these motions were briefly chosen by the search. Perhaps this problem would be overcome by yet more data, or stronger continuity enforcement.

Unlike Chai et al. [CH05], we do not interpolate among several matches, but only play the best match. We believe that computing interpolation weights from measured accelerations (rather than marker positions as Chai did) might not provide good results because of the higher variability in accelerations when performing similar actions. Testing this hypothesis would be easy given our current system, and we plan to do so.

Using a limited dataset may also work to advantage in some situations. Gaming is one application of our system. Limiting the database effectively limits the actions the player can perform to those appropriate to the game. The speed of our system makes it possible to control virtual avatars, opening it to applications such as the online game *Second Life*.

Our system could be used in performance animation, perhaps in laying down animation tracks for input into software such as *Maya*. One could limit the dataset to those types of actions one wished to see, then use our software to lay them out with physically realistic timing. The hardware framework we have presented – promoting wearable comfort in addition to cost-effectiveness – would be useful in motion rehabilitation and training.

Our system does not reconstruct the root position and orientation of the character, except for the vertical axis; additional sensors would be needed to implement this. An IMU could be used to provide full root orientation. Position sensing would likely require either careful interpolation or an external sensor such as a set of cameras.

Several areas of future work present themselves. Additional sensors might allow for more accurate reconstruction of the user's motion. For example, flex sensors would provide approximate joint angles and pressure sensors would allow footplants to be identified. Our e-textile setup would readily extend to a full-body suit; searching may even benefit from the natural coordination of arms and legs to become more accurate. The data compression and search algorithms could be made more sophisticated, perhaps clustering the accelerations according to motion type to get a speed boost and increase continuity. Principal Components Analysis could reduce memory use by eliminating signal redundancy across dimensions.

In conclusion, we have shown that with simple hardware and simple searching, cheap accelerometers can create a workable action capture system by leveraging the power of existing motion capture databases.

6. Acknowledgments

The authors would like to thank Moshe Mahler for his work on the video, and Professor Mark Gross of the Carnegie Mellon University CoDe lab for his encouragement. Ronit Slyper was supported in part by an NSF Graduate Research Fellowship. This work was supported in part by NSF grants CCF-0702556 and IIS-0326322.

RMS error (g's) comparison of accelerometer readings to:

Axis	motion capture	standard reconstruction	end-effector reconstruction
<i>R forearm.x</i>	0.31	0.30	0.28
<i>R forearm.y</i>	0.54	0.51	0.43
<i>R forearm.z</i>	0.22	0.24	0.22
R upper arm.x	0.22	0.24	0.27
R upper arm.y	0.15	0.21	0.32
R upper arm.z	0.19	0.22	0.28
chest.x	0.16	0.18	0.28
chest.y	0.19	0.19	0.31
chest.z	0.14	0.15	0.26
L upper arm.x	0.35	0.28	0.28
L upper arm.y	0.44	0.43	0.41
L upper arm.z	0.55	0.49	0.53
<i>L forearm.x</i>	0.30	0.29	0.29
<i>L forearm.y</i>	0.24	0.25	0.26
<i>L forearm.z</i>	0.25	0.28	0.28

Table 1: A comparison of the RMS error between the accelerometers and accelerations computed from simultaneously recorded motion capture (with manual calibration); accelerometers and the accelerations of motion capture clips used in its reconstruction; and accelerometers and the accelerations of motion capture clips used in its reconstruction, where only the end effectors are used in the matching. The first two columns are similar, as the visual quality of the matched motion is similar to the original motion capture. Matching with only the end effectors (italicized) causes a visually poorer result; the last column shows an increase in the error of the chest and right upper arm.

References

- [BCHN07] BOISSY P., CHOQUETTE S., HAMEL M., NOURY N.: User-based motion sensing and fuzzy logic for automated fall detection in older adults. *Telemedicine and e-Health 13* (2007), 683–694.
- [Bue06] BUECHLEY L.: A construction kit for electronic textiles. In *10th IEEE International Symposium on Wearable Computers* (2006), pp. 83–90.
- [CH05] CHAI J., HODGINS J. K.: Performance animation from low-dimensional control signals. In *ACM Transactions on Graphics* (2005), vol. 24, pp. 686–696.
- [CMC*06] CORAZZA S., MUNDERMANN L., CHAUDHARI A. M., DEMATTIO T., COBELLI C., ANDRIACCHI T. P.: A markerless motion capture system to study musculoskeletal biomechanics: visual hull and simulated annealing approach. *Ann Biomed Eng 34* (2006).
- [DYP03] DONTCHEVA M., YNGVE G., POPOVIC Z.: Layered acting for character animation. In *ACM Transactions on Graphics* (2003), vol. 22.
- [FBRA07] FARELLA E., BENINI L., RICCÀŠ B., ACQUAVIVA A.: Moca: A low-power, low-cost motion capture system based on integrated accelerometers. In *Advances in Multimedia* (2007).
- [FDC03] FONTAINE D., DAVID D., CARITU Y.: Sourceless human body motion capture. In *Proceedings of Smart Objects Conference* (2003).
- [HART08] HARMS H., AMFT O., ROGGEN D., TROSTER G.: Smash: A distributed sensing and processing garment for the classification of upper body postures. In *Third International Conference on Body Area Networks* (2008).
- [JLT03] JUNKER H., LUKOWICZ P., TROSTER G.: Padnet: Wearable physical activity detection network. In *7th IEEE International Symposium on Wearable Computers* (2003), IEEE, pp. 244–245.
- [JWB*99] JOHNSON M. P., WILSON A., BLUMBERG B., KLINE C., BOBICK A.: Sympathetic interfaces: using a plush toy to direct synthetic characters. In *ACM SIGCHI Conference on Human Factors in Computing Systems* (1999).
- [KG05] KWON D. Y., GROSS M.: Combining body sensors and visual sensors for motion training. In *ACM SIGCHI International Conference on Advances in Computer Entertainment Technology* (2005), pp. 94–101.
- [LCL*07] LEE J.-A., CHO S.-H., LEE J.-W., LEE K.-H., YANG H.-K.: Wearable accelerometer system for measuring the temporal parameters of gait. In *IEEE Engineering in Medicine and Biology Society* (2007), pp. 483–486.
- [LH99] LEE J., HA I.: Sensor fusion and calibration for motion captures using accelerometers. In *IEEE*

- Conference on Robotics and Automation* (1999), vol. 3, pp. 1954–1959.
- [LWJ*03] LUKOWICZ P., WARD J., JUNKER H., STAGER M., TROSTER G., ATRASH A., STARNER T.: Recognizing workshop activity using body worn microphones and accelerometers. *Technical Report GIT-GVU-03-32, Georgia Institute of Technology* (2003).
- [Mov08] MOVEN: *moven.com*. 2008.
- [Nin08] NINTENDO: *nintendo.com*. 2008.
- [Org08] ORGANICMOTION:
<http://www.organicmotion.com>. 2008.
- [PKB*08] PIETILÄÄINEN K., KAPRIO J., BORG P., PLASQUI G., YKI-JÄÄRVINEN H., KUJALA U. M., ROSE R. J., WESTERTERP K. R., RISSANEN A.: Physical inactivity and obesity: A vicious circle. *Obesity* 16 (2008), 409–414.
- [PMJ02] PARK S., MACKENZIE K., JAYARAMAN S.: The wearable motherboard: a framework for personalized mobile information processing. In *ACM DAC '02: Proceedings of the 39th conference on Design Automation* (2002), pp. 170–174.
- [POR00] POST E. R., ORTH M., RUSSO P. R., GER-SHENFELD N.: E-broidery: design and fabrication of textile-based computing. *IBM Syst. J.* 39, 3-4 (2000), 840–860.
- [RM02] R.E. MAYAGOITIA A.V. NENE P. H. V.: Accelerometer and rate gyroscope measurement of kinematics: an inexpensive alternative to optical motion analysis systems. *J Biomech.* 35 (2002), 537 – 42.
- [RNd*07] RASKAR R., NII H., DEDECKER B., HASHIMOTO Y., SUMMET J., MOORE D., ZHAO Y., WESTHUES J., DIETZ P., BARNWELL J., NAYAR S., INAMI M., BEKAERT P., NOLAND M., BRANZOI V., BRUNS E.: Prakash: lighting aware motion capture using photosensing markers and multiplexed illuminators. In *ACM Transactions on Graphics* (2007), vol. 26, p. 36.
- [RSH*05] REN L., SHAKHAROVICH G., HODGINS J., PFISTER H., VIOLA P.: Learning silhouette features for control of human motion. In *ACM Transactions on Graphics* (2005), vol. 24, pp. 1303–1331.
- [Spa08] SPARKFUN ELECTRONICS: *sparkfun.com*. 2008.
- [TIL07] TAPIA E. M., INTILLE S., LARSON K.: Real-time recognition of physical activities and their intensities using wireless accelerometers and a heart rate monitor. In *11th IEEE International Symposium on Wearable Computers* (2007).
- [TL06] TIESEL J.-P., LOVISCACH J.: A mobile low-cost motion capture system based on accelerometers. *Advances in Visual Computing* 4292 (2006).
- [VAV*07] VLASIC D., ADELSBERGER R., VANNUCCI G., BARNWELL J., GROSS M., MATUSIK W., POPOVIĆ J.: Practical motion capture in everyday surroundings. In *ACM Transactions on Graphics* (2007), vol. 26, p. 35.
- [YP03] YIN K., PAI D. K.: Footsee: an interactive animation system. In *SCA '03: Proceedings of the 2003 ACM SIGGRAPH/Eurographics Symposium on Computer Animation* (2003), pp. 329–338.