3motion

Payne, John; Keir, Paul; Elgoyhen, Jocelyn; Kenny, Tom; Naef, Martin

Published in:
ACM SIGGRAPH 2005 Sketches

DOI:
10.1145/1187112.1187209

Published: 01/01/2005

Document Version
Peer reviewed version

3motion: Untethered 3D Gesture Interaction

John Payne, Paul Keir, Jocelyn Elgoyhen, Tom Kenny, Martin Naef[1]

Digital Design Studio, Glasgow School of Art

The 3motion system enables the recognition of gestures based on 3D trajectories in space. It consists of both a software and a hardware component. The Software Development Kit (SDK) enables programmers to implement and control how they want to recognise movements in space such as a punch, a golf swing, a baseball pitch or even a dance move. Although the SDK can work with any positioning device from a 2D mouse to a 6 degree of freedom tracker we have developed our own low-cost hardware to provide wide-range wireless functionality. The main innovation of the 3motion system is the combination of a 3D curve matching algorithm with acceleration signatures from inexpensive inertial sensors.

1 The Hardware

Inertial sensing without a fiduciary system, such as ultrasonic positioning, is often rejected as an effective means of recording/tracking movements in space. Despite the inherent limitations of linear accelerometers, we successfully provide an effective interaction method for dynamic 3D gestures and movements such as writing in the air, casting spells, dancing, sporting movements and even combat gestures. Our 3 degree of freedom linear accelerometer provides a means to enable the low cost implementation of roaming interfaces involving such movements.

Our current prototype uses three orthogonal linear accelerometers to sense movement. The sensors are embedded within a single microchip, which implies that they can be placed in small props or even within clothing. Our current prototype is a handheld device that communicates via a bluetooth link to any device that that is running the 3motion enabled software such as a laptop, PDA or mobile phone. The current size of the prototype (Fig. 1) is mostly defined by batteries and debugging connectors. It is based fully on off-the-shelf components and could be mass-produced for well below $100 per unit.

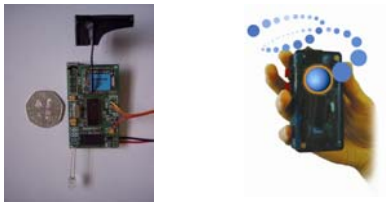

Figure 1: The 3motion hardware.

Our hardware transmits the inertial signature to the software for the gesture recognition process, as opposed to the derived position, since this is a more reliable data source that is not prone to the cumulative error effects of the double integration from inertia to velocity to position. This unique use of the data from these sensors provides a method that enables reliable 3D trajectories/gestures to be performed without the need for lengthy calibration procedures or sensor set-up.

2 3motion SDK

A device independent Software Development Kit (SDK) for gesture recognition has been created. The SDK uses a plug-in architecture and an expandable data structure framework to accommodate both current and future gesture input devices such as mice and 3D tracked points. Multiple and/or combined 2D or 3D signals from input peripherals can also be accommodated. A simple example would be a console controller with two analogue 2D inputs, giving two 2D trajectories or two 6DOF magnetic tracker sensors providing two trajectories in 3-space.

We define the term ‘gesture signature’ as the trajectory in space over time. The space is defined by the acceleration force components for the 3motion hardware. Typical gesture durations will be of the order of a few seconds. Our software algorithm first converts the collected ‘gesture signature’ data specified relative to Euclidian x, y and z axes, to an orientation-independent representation by using common vector calculus methods. The result is a series of values expressed in spherical coordinates. These values are set to thresholds to then provide the symbols for a Hidden Markov Model which is used as a final recognition step.

The programmer can define gestures to be saved in a library to which the user may add extra gestures to improve recognition reliability of their particular movements. After the actual gesture is performed by the user, it is assessed against the predefined gesture library and the best match is selected. The SDK allows the programmer to manipulate how the recognition is carried out, for example absolute position matching can be enabled or disabled to allow for a non-referenced inertial trajectory to be assessed.

3 Conclusions

This technology enables users to practice their golf swing while waiting for a train, play table tennis with friends miles away via mobile phone communication, or throw weapons at enemies while playing a roaming GPS gaming scenario. The software could even be utilised to recognise gestures from any other tracking input method such as video based tracking on mobile phone cameras.

These components work together to provide a platform to bring new and exciting interactive experiences to the end user since our hardware is small enough to be placed within any handheld prop or gadget such as a gun, lightsaber, magic staff, portable gaming device, or mobile phone. The freedom that this system allows means there is no longer considerable set-up times or reliability issues with implementing 3D gesture input within entertainment applications or virtual environments.

[1] [j.payne, p.keir, j.elgoyhen, t.kenny, m.naef]@gsa.ac.uk