Entertainment Computing 5 (2014) 271–283

Contents lists available at ScienceDirect

Entertainment Computing

journal homepage: ees.elsevier.com/entcom

An interaction approach to computer animation ☆

Benjamin Walther-Franks *, Rainer Malaka [1]

Research Group Digital Media, Universität Bremen, Fb3, Bibliothekstr. 1, 28359 Bremen, Germany

ARTICLE INFO

Article history:
Received 31 March 2014
Revised 15 July 2014
Accepted 19 August 2014
Available online 2 September 2014

Keywords:
Motion design interfaces
Performance animation
Human–computer interaction
Design space

ABSTRACT

Design of and research on animation interfaces rarely uses methods and theory of human–computer-interaction (HCI). Graphical motion design interfaces are based on dated interaction paradigms, and novel procedures for capturing, processing and mapping motion are preoccupied with aspects of modeling and computation. Yet research in HCI has come far in understanding human cognition and motor skills and how to apply this understanding to interaction design. We propose an HCI perspective on computer animation that relates the state-of-the-art in motion design interfaces to the concepts and terminology of this field. The main contribution is a design space of animation interfaces. This conceptual framework aids relating strengths and weaknesses of established animation methods and techniques. We demonstrate how this interaction-centric approach can be put into practice in the development of a multi-touch animation system.

1. Introduction

Moving images are omnipresent – in cinema, television, computer games and online entertainment. Digital media such as text, images and film are nowadays produced by a diverse crowd of authors, ranging from beginners and laymen to professionals. Yet animation is still seen by most people as a highly sophisticated process that only experts can master, using complex interfaces and expensive equipment. However, consumer motion capture technology has recently enabled and created a mass-market for easy-to-use animation tools: computer games. In contrast to most professional animation tools, recent games employ full-body interaction for instance via Kinect, allowing users to control a virtual character instantaneously through their body. This trend is feeding back into the area of the "experts", with researchers investigating time-efficient interfaces for computer puppetry using the Kinect (e.g. [61,55]. Computer animation is currently seeing an influx of ideas coming from the world of easy-to-use game interface made for players with no prior training. Game designers in turn are informed by design knowledge and methods developed over decades of research in human–computer interaction (HCI).

It is thus time that computer animation be approached from an HCI perspective. This could aid describing and analyzing the vast spectrum of animation techniques ranging from very intuitive puppetry interfaces for computer games to highly sophisticated control in advanced animation tools. Our goal is to understand principles that underlie human–machine interactions in computer animation. With new ways of thinking about interactions with continuous visual media and a thorough investigation of new animation interfaces on a theoretical foundation, motion design interfaces can be made more beginner and expert friendly.

This can be achieved by embedding computer animation methods and interfaces in an HCI context. Trends in motion design interfaces can be connected with discussions on next generation interfaces in HCI. Theoretical frameworks can aid us in tackling the concrete user interface issues by a profound analysis, which can aid the process of designing new mechanisms for more natural and intuitive means of motion creation and editing.

This article approaches this goal in three main steps. We will first review related work from computer graphics, human computer interaction and entertainment computing from a user- and interface-centric perspective with a focus on methods, mappings and metaphors. In the second step we construct a design space for interfaces that deal with spatiotemporal media. In the third step, the utility of this conceptual framework is illustrated by applying it in designing a multi-touch interactive animation system.

2. Animation techniques: an interaction view

Computer-based frame animation is the direct successor of traditional hand-drawn animation, and still the main method. Advances in sensing hardware and processing power have brought

☆ This paper has been recommended for acceptance by Andrea Sanna.

* Corresponding author. Tel.: +49 421 218 64414.
E-mail addresses: bwf@tzi.de (B. Walther-Franks), malaka@tzi.de (R. Malaka).
[1] Tel.:+49 421 218 64402.

http://dx.doi.org/10.1016/j.entcom.2014.08.007
1875-9521/

entirely new possibilities. Motion capture records the live performance of actors, introducing a new form of animation more akin to puppetry than traditional animation. Programmed animation enables realistic simulations to provide interesting secondary motion and create more believable worlds.

Traditionally, in computer-based keyframe animation, only extreme poses or key frames need to be manually established by the animator. Each keyframe is edited using manipulation tools, which can be specialized for the target domain, e.g. character poses. Some manipulation tools allow influencing dynamics directly in the scene view. The most common means of specifying dynamics is by using global descriptions, such as time plots or motion paths. Spatial editing between keyframes can be achieved indirectly by editing interpolation functions or by defining a new key pose.

Motion timing is usually done via global descriptions of dynamics. However, some temporal control techniques directly operate on the target. Snibbe [58] suggests timing techniques that do not require time plots but can be administered by directly manipulating the target or its motion path in the scene view. As with spatial editing, the practicality of temporal editing with displacement functions depends heavily on the underlying keyframe distribution. Timing by direct manipulation in the scene view is also supported by the latest animation software packages. Tweaking motion trail handles allows for temporal instead of spatial translation; visual feedback can be given by changing frame numbers adjacent to the handle. Spatial control of time has also been proposed for video navigation [15].

Motion graphs are two-dimensional plots that map transformation values (vertical axis) against time (horizontal axis). With a 2DOF input device, such a graph thus allows integrated, simultaneous spatiotemporal control. In keyframe animation the motion editor is the standard way to manage keyframe value interpolation, typically by means of Bezier curve handles.

In contrast to keyframe animation, performance animation uses motion capturing of live performance of an actor or puppeteer by tracking a number of key points in space over time and combining them to obtain a representation of the performance. The recorded data then drives the motion of a digital character. The entire procedure of applying motion capture data to drive an animation is referred to as performance animation [44]. In a typical setup, an actor's motion is first recorded, then the data is cleaned, processed and applied to a digital character. Since the digital character can have quite different proportions than the performer, retargeting the motion data is a non-trivial task [24]. In this form of performance animation, capture and application of motion data to an animation are two separate processes, data handling is done offline. Online performance animation immediately applies captured data to a digital character, creating animation instantly, allowing the performer to react immediately to the results or to interact with an audience [59,24]. Processing limitations sometimes entail that performers can often only see a low-fidelity pre-visualization of the final rendering [44].

Many performance animation efforts aim to represent human motion accurately and limit the abstraction to a minimum and the motion capture performers use only the senses with which they have learned to act (e.g. kinaesthetic and proprioceptive feedback). For performance animation of stylized or non-humanoid characters it is desirable to control them in a less literal fashion. Such a style of performance control is often referred to as computer or digital puppetry [3,59]. Just as traditional puppeteers would rely on mirrors or camera feeds to adjust their performance, computer puppetry requires instant renderings of the applied input to allow performers to adjust their motions. Real-time mappings either use high bandwidth devices for coordinated control of all character DOF, or employ models based on example data or a physical simulation. One challenge is to control a high number of degrees of freedom (DOF) at the same time.

Real-time control of humanoid characters suggest literal mappings from the puppeteer's physique to the character's skeleton. Non-humanoid characters such as animals, monsters or animate objects are difficult since they have a vastly different morphology and motion style to humans. Seol et al. [55] address this by learning mappings through user's mimicking creature motion during a design phase. These learnt mappings can then be used and combined during online puppetry. In similar work, Yamane et al. [66] propose matching human motion data to non-humanoid characters with a statistical model created on the basis of a small set manually selected and created human-character pose pairs; however, this process is conducted offline. The technique for optimal mapping of a human input skeleton onto an arbitrary character skeleton proposed by Sanna et al. [67] manages without any manual examples and finds the best match between the two based solely on structural similarities.

For animation techniques on desktop input devices, however, typically less DOF are available. Recently this has been addressed by multi-touch input devices, which enable techniques for simultaneous rotation, scaling and translation (RST) for 4DOF control of a 2D target [26]. Reisman et al. [52] developed a technique for integrated rotation and translation of 3D content using an arbitrary amount of contact points on an interactive surface.

When input devices of lesser DOF than the object parameters are used, integrated control is not possible. This is a common problem in desktop interaction for navigating and editing 3D media, since most desktop input and display devices only have two DOF. Interface designers thus often face the problem of mapping two control DOF to a higher-dimensional target parameter space. A solution is to separate the degrees of control, i.e. splitting object DOF into manageable subsets [4]. With single-pointer input devices, this necessitates a sequential control of such subsets, e.g. through displays of multiple orthographic projections of the scene in one split screen or through spatial handles that are overlaid on top of the target object. [4].

If high-DOF devices are not available and temporal multiplexing is not desired, interface designers can choose to constrain the interaction to reduce required control DOF. A challenge for designers is that the model behind the constraint must be understood by the user, for instance by basing them on mechanisms already known from other contexts.

Yamane and Nakamura [64] present a pin-and-drag interface for posing articulated figures. By pinning down parts of the figure, such as the end-effectors (feet or hands) and dragging others, the whole character can be controlled with relative ease. Joint motion ranges, the current joint configuration and the user-set joint constraints (pins) thus allow constrained control of several character DOF with as few as two position input DOF for a 2D character. The various constraints are prioritized so that dragging constraints are always fulfilled and solved by differential kinematics that give a linear relationship between the constraints and the joint velocities.

Several research projects have attempted to leave the world of explicit mappings and enable low-to-high-dimensional control, bimanual interaction and multi-user interaction implicitly by simulating real-world physics. Frohlich et al. [20] let users kinematically control intermediate objects that are attached to target objects by springs. The spring attachment is also used by Agrawala and Balakrishnan [1] to enable interaction with a physically simulated virtual desktop, the Bumptop.

Limitations in the motion capture system or the performer's physiology to produce certain desired motions can be overcome by simulating parts of the body and their interaction with the environment. Ishigaki et al. [31] combine real-time full-body motion