

Bewegung als präemptive Präsentationsvariable in einem interaktiven System

Roland Jesse Felix Ritter
Thomas Strothotte
Otto-von-Guericke Universität Magdeburg

Zusammenfassung

Eines der zentralen Probleme bei der Modellierung eines interaktiven Systemes und der Gestaltung seiner visuellen Komponenten liegt in der Repräsentation zeitlicher Zusammenhänge. Dieses gilt in besonderem Maße für interaktive Systeme, dessen Systemstatus während der Ausführungszeit durch Nutzerinteraktionen verändert wird. Dieses Papier skizziert die Verknüpfung eines 3D-Puzzles mit Bewegungsmethoden zur Repräsentation temporaler Abhängigkeiten. Verschiedene Möglichkeiten der Bewegungsdarstellung werden ebenso vorgestellt wie die Modellierung ihrer Parameterisierung. Anhand einer praktischen Fallstudie wird die Anwendbarkeit demonstriert sowie auf einige Implementierungsdetails eingegangen.

1 Motivation

In virtuellen Trainingsumgebungen ist das Erwerben von Hintergrundwissen über ein geometrisches Modell eine zentrale Aufgabe [HSS00]. Zur Erlangung dieses Wissens stellen Ritter u.a. in Form eines 3D-Puzzles ein Werkzeug zur Unterstützung von Lernprozessen mittels interaktiver Exploration vor [RSDP01]. Dabei wird ein Lerneffekt erreicht, indem Benutzer das präsentierte Modell selbständig aus seinen Einzelteilen zusammensetzen. Vom System werden sie dabei in verschiedener Hinsicht unterstützt. So werden beispielsweise Objekte mittels aufblinkenden Farbveränderungen hervorgehoben sobald ein Benutzer mit ihnen agiert.

Insbesondere in komplexen 3D-Szenarien sind Objekte, die für eine Interaktionsaufgabe relevant sind, oftmals von anderen Objekten verdeckt. Blinken ist als präemptives Hervorhebungsmerkmal nicht ausreichend adäquat, wenn es nicht möglich ist, den Betrachterstandpunkt mittels automatischer Kameraführung so anzupassen, dass Objektverdeckungen vermieden werden können. Bartram motiviert den Einsatz von Bewegung als eigenständige Ausdrucksdimension [Bar98]. Diesen Ansatz weiterführend, stellen Jesse und Strothotte Bewegungsmethoden vor, die für Visualisierungen eingesetzt werden können [JS01].

Bewegungen sind als Hervorhebungsmethoden einsetzbar, die sowohl Anwenderwünschen als auch -fähigkeiten gerecht zu werden helfen. Dabei ist darauf zu achten, jedwede Bewegungsdarstellung weitestgehend parameterisierbar zu modellieren. Rechnung trägt dieses dem Umstand, dass Bewegungen als eigenständige Ausdrucksdimension nicht notwendigerweise direkte Eigenschaften des bewegten Objektes repräsentieren. Gleiches gilt hingegen auch für andere Präsentationsvariablen im Sinne Noiks [Noi94]. Um den Einsatz von

Bewegungen zu flexibilisieren, ist er auf einem formalen Modell aufzubauen. Gemäß des zeitlichen Charakters von Bewegungen liegt ein besonderer Schwerpunkt für ein solches Modell auf der Berücksichtigung temporaler Parameter.

Die Erstellung eines formalen Modelles zur temporalen Bewegungsparameterisierung weist mehrere Vorteile auf: Es dient der minimalen Einschränkung der Vielseitigkeit aller potenziellen Einsatzfelder für die Bewegungsdarstellung ebenso wie der Ermöglichung einer automatischen Abbildung nicht inhärent temporaler Weltaspekte auf Bewegungsparameter. Das Modell bildet damit die Grundlage für ein Zeitnetzwerk, welches analog eines Visualisierungsnetzwerkes (oftmals auch bezeichnet als Visualisierungspipeline) für das visuelle Modell, die automatische Generierung eines temporalen Modelles zur Systemlaufzeit ermöglicht.

Der verbleibende Teil dieses Beitrages gliedert sich wie folgt: Abschnitt 2 untersucht die Eignung von Bewegungsdarstellungen zum einen in Hinsicht auf die Entsprechung der Bedürfnisse des Anwenders, die sich in seiner Zielfunktion gegenüber dem System ausdrücken. Zum anderen werden kognitive Aspekte von Bewegungen heraus gestellt, die sowohl Stärken als auch Schwächen von Bewegungen aus wahrnehmungsorientierter Sicht beleuchten. In Abschnitt 3 werden die verschiedenen Teilbereiche der Parameterisierung von Bewegungsdarstellungen vorgestellt. Es wird eine Unterscheidung zwischen methodischer und temporaler Parameterisierung ebenso eingeführt wie die schematische Darstellung der formalen Spezifikation einer Parameterisierungsfunktion für Bewegungsdarstellungen. Der praktische Einsatz von Bewegung als Hervorhebungsmethode in einem interaktiven System wird exemplarisch in Abschnitt 4 betrachtet. An dieser Stelle werden ebenfalls einige Hinweise auf Implementierungsdetails aufgeführt. Die Vorstellung verwandter Arbeiten in Abschnitt 5 rückt die Betrachtungen in ihren Kontext. Abschließend gibt Abschnitt 6 eine kurze Zusammenfassung und geht auf offene Aufgabenstellungen ein.

2 Bewegungsdarstellungen

Die Verwendung von Bewegungen in einem interaktiven System kann und sollte nicht reflexionslos erfolgen. Sie ist ebenso wie der Einsatz jeder anderen Präsentationsvariable kontextbedingt auszuwählen. Abschnitt 2.1 geht ein auf die Entsprechung der repräsentierten Welt durch die visuelle Darstellung im System. Kognitive Aspekte des Nutzermodelles und Schlussfolgerungen bezüglich des Einsatzes für Bewegungsdarstellungen werden in Abschnitt 2.2 heraus gestellt.

2.1 Entsprechung gemäß der Zielfunktion des Anwenders

Ein System zur interaktiven Exploration spiegelt mehrschichtige Repräsentationsmodelle wieder. Auf der höchsten Ebene findet sich die *repräsentierte Welt*, welche vollständig die dem System zu Grunde liegende Semantik repräsentiert. Diese Ebene wird bezeichnet als $world_{rep} = \{user, scene, relations, operations\}$. Sie beschreibt somit eine Menge bestehend aus dem Nutzermodell, dem Szenenmodell, Beziehungen zwischen verschiedenen Bestandteilen der Szene sowie Operationen, die in der repräsentierten Welt definiert sind.

Ein typisches Nutzermodell ergibt sich als Menge $user = \{percept, actions, scope\}$. Die Wahrnehmung eines Anwenders sowie seine Aktionen werden durch die Mengen *Percept* sowie *Actions* abgebildet. Der *Scope* beschreibt eine Menge von Parametern des Nutzermodelles, welche die individuelle Zielfunktion eines Anwenders in Bezug auf das System ergeben. Er ist somit gegeben als $scope = \{s_1, \dots, s_n\}$. In Abhängigkeit der möglichen Belegungen aller Parameter liefert diese Zielfunktion die Motivation und Notwendigkeit zur Parameterisierung von Bewegungsdarstellungen, die als eigenständiges Ausdrucksmittel eingesetzt werden.

Die Auswertung der Zielfunktion des Anwenders erfolgt in Analogie zu Reddy [Red01]. Mit dem Ziel der Laufzeitoptimierung seines Systems, wertet Reddy Parameter des Nutzermodelles aus, um einen Interessensbereich zu ermitteln, auf den einzelne Aspekte des Renderings konzentriert werden. Bei der Parameterisierung von Bewegungsdarstellungen geht es darum, die Zielfunktion des Anwenders dahingehend auszuwerten, dass mittels Bewegungen eine bestmögliche Interaktionsunterstützung bereit gestellt werden kann, die eine Unterstützung bei der Objektlokalisierung anbietet. Hierbei ist ausdrücklich die Restriktierung jedweder Bewegung gemäß des verbleibenden Systemzustandes eingeschlossen.

2.2 Kognitive Aspekte von Bewegungsdarstellungen

Schon frühe Studien belegen, dass die Gruppierung von semantisch verknüpften Daten visuell einfacher erfasst wird, wenn sie mit Hilfe mehrerer ähnlicher Bewegungsmuster an Stelle von statischen Darstellungen ausgedrückt werden [Joh64]. Weiterhin schildert Ware eine Studie von Johansson [Joh73], in der dargestellt wird, dass auch sehr abstrakte Bewegungsmuster von den Probanden als Entsprechungen für konkrete Sachverhalte erkannt werden [War00, 237f]. Visuelle Metaphern sind selbst dann auflösbar, wenn die Bewegung nur für Bruchteile einer Sekunde dargestellt wird. Ebenso ist der kausale Zusammenhang von Daten durch Benutzer leicht erkennbar, wenn er entweder mittels gleicher oder zeitlich miteinander verknüpften Bewegungsmustern dargestellt wird [War00, 235f]. Bartram u.a. zeigen, dass Objektbewegungen schneller und fehlerfreier erkannt werden als Änderungen der Farbe oder Form eines Objektes [BWC01].

Die oben geschilderte Notwendigkeit zur Parameterisierung von Bewegungsdarstellungen wird unterstützt durch das begrenzte Wahrnehmungspotenzial des Anwenders. Der Einsatz von Bewegung als Hervorhebungsmerkmal ist mit Sorgfalt einzusetzen, um den Benutzer kognitiv nicht zu sehr zu belasten. Pylyshyn u.a. zeigen, dass Bewegungen ein ausdrucksstarkes Visualisierungsmittel sind und besonders leicht Informationen vermitteln [PBF⁺93]. Sie bringen allerdings ebenfalls zum Ausdruck, dass nicht mehr als fünf verschiedene Bewegungsmuster voneinander unterschieden werden können. Mit Sorgfalt dosiert, kann Bewegung somit hilfreich zur Interaktionsunterstützung sein. Für komplexe Nutzeraktionen kann es unter Umständen sinnvoll sein, auf bewährte präemptive Präsentationsvariablen, wie das Blinken mittels Farbveränderungen, zurück zu greifen.

3 Parameterisierung von Bewegungsdarstellungen

Die Parameterisierung von Bewegungen erfolgt auf zwei Ebenen. Zum einen werden verschiedene Bewegungsmethoden unterschieden. Zum anderen sind Bewegungen inhärent zeitabhängig und somit temporal parameterisierbar. Die Abschnitte 3.1 sowie 3.2 gehen auf diese beiden Parameterebenen näher ein. Abschnitt 3.3 beschreibt eine formale Spezifikation der Bewegungsparameterisierung. Damit wird zum einen die Grundlage für eine einfache Modellverifikation gegeben und zum anderen liefert sie die formale Basis möglicher Implementierungen. Im Sinne der Motivation zur Entwicklung von Referenzmodellen für verschiedene Visualisierungsaspekte gemäß Robertson und Ferrari liefert dieses Spezifikation auch Anregungen für weitere Arbeiten [RF94, 287].

3.1 Methodenbasierte Bewegungsparameterisierung

Jesse und Strothotte stellen die Verwendung von Bewegung für Visualisierungszwecke vor [JS01]. Die möglichen Bewegungsarten teilen sie ein in drei Basiskategorien:

Translation (T) beschreibt eine andauernde Veränderung einer Objektposition. Da verschiedene Szenenobjekte in einem räumlichen Zusammenhang zueinander stehen, helfen oszillierende Bewegungen dabei, eine Invalidierung der Szenenkohärenz zu verhindern.

Rotation (R) erfolgt um variabel wählbare Rotationszentren. Drehungen um das Objektzentrum verursachen nur geringfügige Szeneninkohärenzen, sind allerdings auch kognitiv schwach und erfordern somit eine hohe Wahrnehmungsleistung des Anwenders. In Analogie zur Translation können auch Rotationen oszillierend erfolgen.

Formveränderung (D) drückt sich entweder in wiederholenden Skalierungen oder in morphenden Formwechseln aus. Während hierbei die Darstellung jedes einzelnen Objektes manipuliert werden kann, wird der räumliche Bezug zwischen verschiedenen Objekten nicht modifiziert. Ordnet sich die Bild- der Szenenkohärenz unter, bieten Formveränderungen ein ausdrucksstarkes Mittel der präemptiven Objekthervorhebung.

Der Bewegungskatalog ergibt sich somit als $cat = \{T, R, D\}$. Zur Bestimmung einer konkreten Bewegungsmethode ist zum einen die Kombination der einzelnen Bewegungsarten des Kataloges möglich und zum anderen eine Belegung des Parameterraumes $P = \{p_1, \dots, p_m\}$ zu definieren. Eine konkrete Methode zur Bewegungsdarstellung ergibt sich somit als $method = \{C, P\}$ mit $C = \{c_1, \dots, c_n\}; c_i \in cat, 1 \leq i \leq n$.

3.2 Zeitbasierte Bewegungsparameterisierung

Eine der zentralen Herausforderungen bei der Modellierung von Systemen, die das Verhalten von Objekten wiedergeben, ist die Repräsentation von Zeit. So wird beispielsweise die Verwendung jeder einzelnen Bewegungsmethode zeitlich beschränkt, um die Anzahl gleichzeitig verwendeter Bewegungsdarstellungen zu limitieren. Die Abbildung von Zeit

auf den Parameterraum für Bewegungsdarstellungen ist Gegenstand dieses Abschnittes. Dabei umreißt Teilabschnitt 3.2.1 die grundlegenden Begriffe und Zusammenhänge. Auf Abbildungsfunktionen für Zeitereignisse geht Unterabschnitt 3.2.2 ein. Abschließend wird in Teilabschnitt 3.2.3 ein Ebenenmodell zur Veranschaulichung der Verwaltung temporaler Intervalle vorgestellt.

3.2.1 Modellgrundlage

Allen schildert verschiedene Möglichkeiten zur Modellierung von Zeitrepräsentationen [All91]. Für den Einsatz der Bewegungssteuerung in einem interaktiven System erfolgt die Zeitmodellierung datumsbasiert. Damit ist die Erzeugung einer vollständigen linearen Ordnung aller Ereignisse gewährleistet. Die Menge aller Zeitintervalle (\mathcal{I}) ergibt sich aus der Summe aller modellierten Intervalle (\mathcal{I}_{model}) und der zur Laufzeit zu gewinnenden Intervalle ($\mathcal{I}_{runtime}$), welche sich aus Nutzerinteraktionen ergeben. Es ist $\mathcal{I} = \mathcal{I}_{model} + \mathcal{I}_{runtime} = \{I_1, \dots, I_n\}; n \in \mathbb{N}$.

Als eine Alternative schildert Allen die direkte Modellierung mit Hilfe von Zeitdauerangaben in [All91, 348ff]. Zeitnetzwerke, die auf dieser Basis aufgebaut werden können, bieten eine feinere Modell-Granularität im Sinne von [BJW00, 7]. Inkrementelles hinzufügen neuer Ereignisse zum Netzwerk ist allerdings eine teure Operation, da sie unter Umständen jedes Mal eine Neuberechnung des gesamten Netzwerkes erfordert. Nutzerbasierte Ereignisse, die erst zur Laufzeit anfallen, wären somit schlecht in das Modell zu integrieren.

3.2.2 Abbildung von Zeitereignissen auf Parameterisierung

Sei als Modellprimitive ein Intervall $I = \{e_{begin}, e_{end}\}$ mit einer impliziten Intervalldauer von $e_{end} - e_{begin}$ gegeben. Im Sinne der einfachen Schreibweise können dabei die Intervallgrenzen jeweils mit I^b und I^e direkt bezeichnet werden. Die Menge aller Ereignisse, die während der Laufzeit des Systems bekannt sind, sei mit E bezeichnet. Zur Bestimmung der Zugehörigkeit eines gegebenen Zeitereignisses $e \in E$ zu einem Intervall $I \in \mathcal{I}$ sei die Funktion $\psi : E \rightarrow \mathcal{I}$ gemäß Gleichung (1) definiert.

$$\psi(e) = \begin{cases} I_0 & \text{Einzelintervall} \\ I_i & \iff I_i^b \leq e \leq I_i^e \quad | i = [1, 2] & \text{Doppelintervall} \\ I_j & \iff I_j^b \leq e \leq I_j^e \quad | 1 \leq j \leq n & \text{sonst} \end{cases} \quad (1)$$

Hierbei bildet $\psi(e)$ den Ereigniszeitpunkt mit Hilfe einfacher Vergleichsoperationen direkt auf seinen zugehörigen Intervall ab. Gemäß Abschnitt 3.2.1 bezeichnet n die Anzahl der momentan definierten Intervalle und somit die Dimension von \mathcal{I} .

Mit Hilfe einer Interpolationsfunktion $\tau : I \rightarrow \mathcal{I}$ ist ein Intervall diskretisierbar, um einen fließenden Übergang unterschiedlicher Parameterbelegungen zu ermöglichen. Damit werden Beschleunigungen, Verzögerungen, sowie das Ein- beziehungsweise Ausblenden (Fading) einzelner Bewegungsdarstellungen ermöglicht.

Die Funktion δ zur Bestimmung einer konkreten Bewegungsparameterisierung ergibt sich nun als Tupel $(\mathcal{M}, \mathcal{I}, \psi, \tau)$, wobei \mathcal{M} die Menge aller verfügbaren Methoden zur Bewegungsdarstellung gemäß Abschnitt 3.1 bezeichnet, \mathcal{I} die Menge aller momentan definierten Zeitintervalle, ψ und τ entsprechen den oben aufgeführten Abbildungsfunktionen.

3.2.3 Intervallebenen

Zur Veranschaulichung der unterschiedlichen Intervallzusammenhänge und ihrer Auswirkungen auf die temporale Bewegungsbeeinflussung stellt Abbildung 1 ein Ebenenmodell vor, welches unterschiedliche Parameterisierungen für Bewegungen auf der Basis der Anzahl verfügbarer Intervalle bestimmt. Die Intervalle beziehen sich dabei auf einzelne Bewegungsmuster und nicht auf die gesamte Systemdauer.

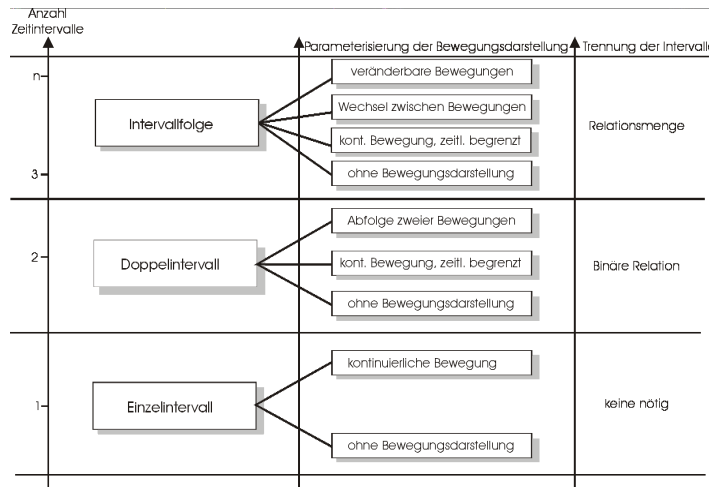


Abbildung 1: Intervallebenen für zeitlich parameterisierte Bewegungsdarstellungen

Die Ebene des *Einzelintervalles* drückt die einfachste Modellkomplexität aus. Das Applizieren einer jedweden Präsentationsmethode – wobei Bewegungsdarstellungen explizit mit eingeschlossen sind – ist nicht zeitlich beschränkt. Die Auswirkung der Methode wird für die gesamte Intervalldauer angezeigt. Endet der Intervall, das heißt an Punkt I_1^e , wird eine möglicherweise aktive Bewegungsdarstellung deaktiviert.

Auf der Ebene des *Doppelintervalles* sind zwei verschiedene Möglichkeiten der Bewegungsrepräsentation möglich: zeitlich begrenzte, kontinuierliche Bewegungen ebenso wie die Sequenz zweier aufeinander folgender Bewegungsmuster. Ihre Zeitdauer ergibt sich aus den Intervallgrenzen von I_1 und I_2 . Gilt $I_1^e = I_2^b$, wird eine kontinuierliche Bewegungsdarstellung beschränkt durch $I_1^e - I_1^b$, welches der vollständigen Zeitdauer des ersten Intervalles entspricht. Die verbleibende Zeit bleibt nutzbar analog der Ebene des Einzelintervalles.

Differenzierte Bewegungsmuster sind auf der Ebene der *Intervallfolge* anwendbar. Zunächst sind die Variationen der ersten beiden Ebenen ebenfalls auf dieser Ebene applizierbar. Explizit eingeschlossen ist dabei die Verwendung ausschließlich klassischer Präsentationsvariablen oder der kombinierte Einsatz dieser mit Bewegungsmethoden. Die Parameterisierung von Bewegung auf dieser Ebene ist in zwei Kategorien einteilbar: Einsatz aufeinander folgender Bewegungsmuster oder Einsatz modifizierbarer Bewegungsmuster.

3.3 Spezifikation der Parameterisierungsfunktion

In klassischer Darstellung sei die Signatur spezifiziert als Tupel $SIG = \langle S, OP, REL \rangle$ bestehend aus der Menge S , der Mengenfamilie $OP = (OP_{\omega,s})_{\omega \in S^*, s \in S}$, sowie der Mengenfamilie $REL = (REL_{\omega})_{\omega \in S^*}$. Die Elemente von S geben *Sorten (Typen)* an. S^* bezeichnet die Menge aller Sortenketten. Für jedes $\omega \in S^*$ und jedes $s \in S$ ist $OP_{\omega,s}$ die Menge, deren Elemente op die möglichen *Operatoren* bezeichnen. Es sind somit Abbildungen $op : S^* \times S \rightarrow S$. Für jedes $\omega \in S^*$ bezeichnet REL_{ω} die Menge der möglichen Relationen.

Im Sinne der einfacheren Lesbarkeit und zur Besseren Übersichtlichkeit wird diese redundanzbelastete Syntax vereinfacht und im Folgenden eine schematische Darstellung gewählt. Die Sorten werden hinter dem Schlüsselwort *sorts* aufgelistet. Operationsdeklarationen folgen dem Schlüsselwort *opns* und Relationsdeklarationen dem Schlüsselwort *rels*. Bezeichnet λ das leere Wort, kommt vereinfachend hinzu, dass statt $c \in OP_{\lambda,s}$ einfach $c : \rightarrow s$ geschrieben wird, statt $op \in OP_{\omega,s}$ (mit $\omega \neq \lambda$) einfach $op : \omega \rightarrow s$. Ähnliches gilt für Relationen. Aus $rel \in REL_{\omega}$ wird $rel : \omega$.

Die Spezifikation eines Modelles sei gegeben als Tupel $SPEC = \langle SIG, X, REQ \rangle$. Sie besteht aus einer Signatur SIG , einem Variablenvorrat X sowie einer Menge von Formeln $REQ \subset F_{SIG}(X)$, die die Algebra der Spezifikation festlegen. Für die schematische Darstellung der Spezifikation wird die syntaktische Darstellung der Signatur übernommen und weitergeführt. Der Variablenvorrat wird hinter dem Schlüsselwort *vars* aufgelistet, die Anforderungen der Algebra hinter dem Schlüsselwort *reqs*.

Bewegungsparameterisierung $\delta(\mathcal{M}, \mathcal{I}, \psi, \tau)$

sorts: $I, \mathcal{I}, E, \mathcal{M}, \mathcal{W}_{rep}$

opns: $\psi : E \rightarrow I$

$\tau : I \rightarrow \mathcal{I}$

$+$: $\mathcal{I} \rightarrow \mathcal{I}$

rels: $\langle, =, \rangle : I I$

$\subset, \subseteq, = : \mathcal{I} \mathcal{I}$

vars: $e_1, \dots, e_n : E$

$I_1, \dots, I_n : \mathcal{I}$

$m_1, \dots, m_n : \mathcal{M}$

$world_{rep}^1, \dots, world_{rep}^n : \mathcal{W}_{rep}$

reqs: $\psi(e_n) < \psi(e_m) \iff e_n < e_m$

$\tau : \forall I_i \in \mathcal{I} \implies I_1^b \leq I_i^b \wedge I_n^e \geq I_i^e$ (Intervallgrenzenbeschränkung)

$I_a < I_b \iff I_a^b \geq I_b^b \wedge I_a^e < I_b^e$

$I_a = I_b \iff I_a^b = I_b^b \wedge I_a^e = I_b^e$

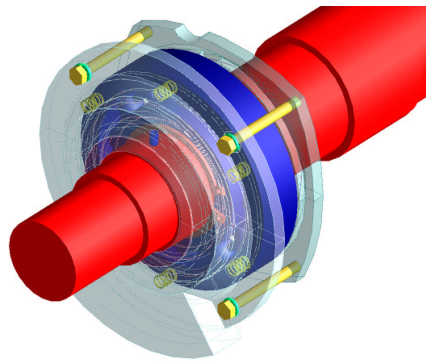
$I_a > I_b \iff I_a^b \leq I_b^b \wedge I_a^e > I_b^e$

$\mathcal{I} + \lambda = \mathcal{I}$

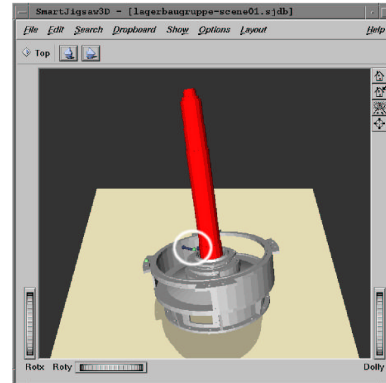
$\mathcal{I}_a = \{I_1, \dots, I_a\} \wedge \mathcal{I}_b = \{I_{a+1}, \dots, I_b\} \implies \mathcal{I}_a + \mathcal{I}_b = \{I_1, \dots, I_b\}$

(Intervallkontinuität)

$m_a \neq m_b \iff world_{rep}^a \neq world_{rep}^b$



(a)



(b)

Abbildung 2: Konstruktion einer Lagerbaugruppe aus ihren einzelnen Bestandteilen. Das vollständig zusammengesetzte Modell ist in 2(a) dargestellt. Eine Momentaufnahme während des frühen Konstruktionsstadiums ist in 2(b) wieder gegeben. Der Kreis kennzeichnet den Ort oszillierender Translationsänderungen einer anderweitig verdeckten Schraube.

4 Anwendung

Um die praktische Anwendbarkeit der bisherigen modelltheoretischen Betrachtungen prototypisch zu belegen, sei an dieser Stelle kurz der Einsatz von Bewegungsdarstellung in einem interaktiven 3D-Puzzle skizziert. Abschnitt 4.1 schildert ein Szenario zur Anwendung von Bewegungsdarstellungen zur Unterstützung der Nutzerinteraktion. Einige Hinweise zur Umsetzung der Bewegungsmethoden werden in Abschnitt 4.2 dargestellt.

4.1 Szenario

Vom Lernenden ist das Modell eines Generators iterativ zusammen zu setzen. Die einzelnen Iterationsschritte umfassen dabei die Konstruktion von Generatorbauteilen. Ein solcher Baustein ist eine Lagerbaugruppe, welche Gegenstand der exemplarischen Betrachtung im Rahmen dieses Beitrages ist. Diese Baugruppe setzt sich zusammen aus einer Welle, einem Flanschlagerschild zur Lagerung dieser Welle sowie stützenden Einzelteilen wie unterschiedlichen Scheiben und Schrauben. In Form einzelner geometrischer Figuren liegen diese Einzelteile in loser Form in der Szene vor und sind vom Lernenden (dem Anwender) interaktiv zu assemblieren. Abbildung 2 veranschaulicht dieses Vorgehen. Dabei zeigt Abbildung 2(a) die bildhafte Beschreibung der fertig konstruierten Lagerbaugruppe. In Abbildung 2(b) ist die Szene in einem frühen Konstruktionsstadium wiedergespiegelt. Um den Anwender weitestgehend automatisch von Seiten des Systemes unterstützen zu

können, liegt neben der rein geometrischen Szenenbeschreibung auch eine strukturelle Beschreibung vor. In dieser wird die Zusammengehörigkeit verschiedener Einzelbauteile ebenso modelliert wie die konkreten Positionen ihrer Verbindungsstellen. Das Zusammenfügen von Objekten basiert auf einer binären Operation. Zwei Objekte o_1 und o_2 werden mit Hilfe einer Verbindungsstelle V_{12} miteinander verbunden. Jede Verbindungsstelle setzt sich aus zwei Kontaktelementen ϑ_1 und ϑ_2 zusammen. Beschreibt \triangleleft die Zugehörigkeit eines Kontaktelementes ϑ zu einem Objekt o so gilt $\vartheta_1 \triangleleft o_1$ und $\vartheta_2 \triangleleft o_2$.

Interagiert der Anwender mit Objekt o_1 , so ist dem System dessen semantische Zugehörigkeit von Objekt o_2 bekannt. Entsprechende Hervorhebungen, die auch den Einsatz geschilderter Bewegungsdarstellungen mit einschließt, können somit appliziert werden. Unterschreitet der räumliche Abstand zwischen ϑ_1 und ϑ_2 (und damit der Abstand von o_1 sowie o_2) einen vordefinierten Grenzabstand ϵ_{12} , so sind diese beiden Bauteile weiterhin vom System automatisch zusammenführbar. Die Anwendung dieses Schnappmechanismus führt zu einer Belegung aller $m_i \in \mathcal{M}$ so dass eine eventuell aktive Bewegungsdarstellung zur Hervorhebung eines der beteiligten Objekte inaktiv wird.

4.2 Umsetzung

Die verschiedenen Methoden $m_i \in \mathcal{M}$ von Objektbewegungen werden in Gestalt einer Algorithmenbibliothek umgesetzt. Diese baut auf Open Inventor™ auf. Ihre Struktur erlaubt sowohl das Einbinden anderer Bibliotheken der gleichen Abstrahierungsebene als auch die Integration in eben diese. Exemplarisch sei in diesem Zusammenhang auf einen modularen Werkzeugkasten zur Darstellung mit nichtfotorealistischen Renderingmethoden verwiesen, der derzeit entwickelt wird [Hal01].

Abbildung 3 zeigt den prinzipiellen Aufbau der einzelnen Bestandteile der Bewegungsbibliothek am Beispiel der Implementierung einer Oszillationsdarstellung. Die Abbildung der Repräsentation eines Objektes aus $world_{rep}$ wird in Form eines so genannten *Nodekits* implementiert. Darunter sei ein Teilszenengraph verstanden, welcher ein geometrisches Objekt repräsentiert, das mit Bewegung angereichert ist; Abbildung 3(a) veranschaulicht dieses. Die Kodierung der Kontrollparameter für die Bewegung in diesem Nodekit ist in die Beschreibung der herkömmlichen Präsentationsvariablen des Objektes integriert. Die Umsetzung der Parameterisierung einer konkreten Bewegungsmethode $m_i \in \mathcal{M}$ ist dabei in einem separaten Knoten gekapselt, welcher auch einzeln wiederverwendet werden kann. Die Belegung von unterschiedlichen $m_i \in \mathcal{M}$ für unterschiedliche $world_{rep}$ gemäß den Anforderungen der Spezifikation aus Abschnitt 3.3 bleibt dadurch gewährleistet. Abbildung 3(b) zeigt das Modell eines derartigen Knotens für oszillierende Bewegungen („Shuttle“). Als Stützpunkt für seine Bewegung dient eine einfache Strecke, die mit zwei Translationsangaben spezifiziert wird. Die Oszillationsfrequenz wird mittels eines Fließkommawertes angegeben, der die Zyklen je Sekunde festlegt. Andere Bewegungsdefinitionen sind analog in die Nodekitstruktur integrierbar.

Zur Kapselung der Bewegung gegenüber der restlichen Szene dient ein Separatorknoten. Dieser speichert die Statusmaschinenkonfiguration des Renderingsystems zwischen und stellt sie nach abgeschlossener Traversal des Nodekitszenengraphs wieder her. Die Szenenkohärenz bleibt somit gewahrt.

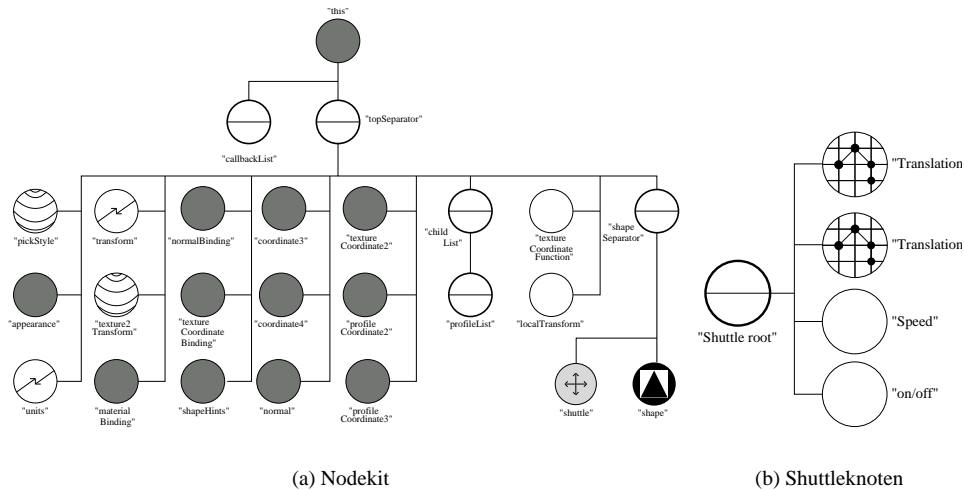


Abbildung 3: Entwurf der Szenengraphenstruktur eines Bewegungsknotens

5 Verwandte Arbeiten

Dieser Beitrag widmet sich der Parameterisierung von Bewegungsdarstellung zur Verwendung in einem interaktiven System. Ein besonderer Schwerpunkt liegt dabei auf der Berücksichtigung temporaler Zusammenhänge. Um die Arbeit in ihren notwendigen Kontext zu rücken, stellen die beiden folgenden Abschnitte zum einen bestehende Arbeiten zu Bewegungsdarstellungen und zum anderen grundlegende Arbeiten zur temporalen Modellierung vor.

5.1 Bewegungsdarstellungen

Objektbewegungen sind per se eine spezielle Form von Animationen. Allerdings dienen sie nicht der Zeitrepräsentation, bilden somit nicht notwendigerweise die Position von Objekten in Abhängigkeit der Zeit ab. Ihr Potenzial liegt vielmehr in der Wiedergabe von Objekteigenschaften.

Bartram u.a. untersuchen bewegte Icondarstellungen mit dem Ziel, die Effizienz der Informationsvermittlung und kognitiven Belastung verschiedener Bewegungsmuster zu identifizieren [BWC01]. Sie kommen zu dem Ergebnis, dass Bewegungen auf der einen Seite ausdrucksfähiger als statische Darstellungsformen sind; auf der anderen Seite aber auch Bewegungsmuster möglich sind, die den Anwender stärker ablenken als andere und somit Konzentrationsmindernder Natur sind.

Gupta und Ghosh verwenden Bewegungsdarstellungen zur Abbildung temporaler Eigenschaften von Kundendaten eines E-Commerce Datenbestandes [GG01]. Mit Methoden des Data-Mining werden die Ausgangsdaten analysiert und Muster von Wanderungen verschiede-

dener Kundenströme extrahiert. Die saisonalen Abhängigkeiten dieser Wanderungen werden abgebildet auf Bewegungsmuster. Die Parameterisierung dieser Muster ist abhängig von der Natur, Anzahl und Qualität der Eingangsdaten für die einzelnen Kunden. So entstehen Oszillationen als Ergebnis sich entsprechend wiederholender Wanderungen eines Kundenstromes in aufeinanderfolgenden Saisonen.

5.2 Modellierung temporaler Abhängigkeiten

Rechenzeitmangel führt dazu, dass Änderungen am Systemzustand nicht immer in Realzeit dem Benutzer mitzuteilen sind. Dix und Abowd bezeichnen sich daraus ergebende Verzögerungen in der Darstellung als temporale Inkohärenz [DA96]. Um dem entgegen zu wirken, schlagen sie vor, einen zeitlichen Toleranzintervall d einzuführen, der schon in der Systemmodellierung berücksichtigt wird. Einen bewussten Einsatz der Verzögerung als Ausdrucksmittel wird in diesem Zusammenhang nicht berücksichtigt.

Die Möglichkeit zur Auswertung zeitlicher Aktionen eines Benutzers wird von Palanque und Bastide mit Hilfe von Petrinetzen thematisiert [PB96]. Sie beziehen sich dabei auf die Möglichkeit der Modellierung eines interaktiven Systems auf der Basis von Benutzerinteraktionen. Ein besonderer Bezug auf ein dem System zu Grunde liegendes geometrisches Modell erfolgt hingegen nicht.

Allen führt die Notation von 13 Basisrelationen ein, die den zeitlichen Zusammenhang von Intervallen beschreiben [All83]. Diese basieren auf der ausschliesslichen Betrachtung von Zeit in Form von Intervallen. Die von Allen vorgestellten Relationen sind vollständig. Sie ermöglichen die Beschreibung jedweder Relation zwischen zwei gegebenen Intervallen. Dabei werden grundsätzlich nur relative Zeitangaben und keine festen Datumsangaben verwendet.

Dieses Modell wird von Freksa aufgegriffen und um den Begriff des „Semi-Intervalls“ erweitert [Fre92a]. Dieser zeichnet sich dadurch aus, dass er lediglich durch Anfangs- und Endpunkt gekennzeichnet ist. Auf diesen Intervallgrenzen sind nun die Relationen $<$, $=$ und $>$ fest zu legen. Diese Relationen sind dadurch charakterisiert, dass der Beginn eines Ereignisses (Intervalles) grundsätzlich vor seinem Ende liegt ($\alpha < \omega$, $A < \Omega$) und dass sie transitiv sind. Um die Verhältnis zwischen zwei beliebigen Intervallen zu beschreiben, reichen somit zwei Relationsangaben aus. Weiterhin ermöglicht diese Modellkonstruktion, lückenhaftes Wissen über eine Menge von Intervallen auf weit einfachere Weise als [All83] modellieren zu können. Dieses Relationsprinzip weitet Freksa auf eine Verknüpfung temporaler mit räumlichen Vergleichskriterien aus, die er als *konzeptionelle Nachbarschaft* bezeichnet [Fre92b].

Weiterführende Intervallzusammenhänge untersucht Weber. Er führt *hierarchische Zeitintervalle* ein, um der menschlichen Wahrnehmung von Zeit gerecht zu werden [Web00]. Ein besonderes Augenmerk legt Weber auf die darauf aufbauende Integration von visuellen und nichtvisuellen Benutzungsoberflächen.

Galetto und Viola¹ schildern, dass die Umsetzung der Darstellung zeitlicher Veränderungen ausschließlich mittels der zwei Operationen *Hinzufügen* und *Deaktivieren* möglich ist

¹Hier referenziert mittels [Ren97].

[GV94]. Ist eine neue Version darzustellen, so wird zuerst die alte deaktiviert und anschließend die neue Version angezeigt. Um eindeutige Informationen über die zeitliche Geschichte eines Objektes zu bewahren, ist diese Vorgehensweise zu erweitern. Möglichkeiten dafür zeigt Langran² auf, indem er eine Vorgänger-Nachfolger-Beziehung für jedes Tupel aufstellt [Lan93].

6 Zusammenfassung und Ausblick

Dieser Beitrag motiviert den Einsatz von Bewegungsdarstellungen für Hervorhebungszwecke in einem interaktiven System. Bewegungen werden dabei als eigenständige Ausdrucksform betrachtet und nicht als eine Form der Animation, die Positionsveränderungen in Abhängigkeit von der Zeit darstellt. Um der kognitiven Kapazität von Anwendern gerecht zu werden, wird ein Modell zur Parameterisierung von Bewegungsdarstellungen eingeführt. Damit wird es ebenfalls möglich, den Einsatz von Bewegung durch den Nutzer selbst gestalten zu können. Das Modell hilft somit, den Anwenderwünschen besser gerecht werden zu können.

Die Bewegungseigenschaften (Amplitude und Frequenz) sind im derzeitigen Entwicklungsstadium noch fest vorgegeben. In Entwicklung befindet sich eine skriptgesteuerte Parametrisierung der selben. Diese erlaubt zum einen, unterschiedliche Eigenschaften für unterschiedliche Objekte zu bestimmen. Weiterhin ermöglicht eine Skriptsteuerung, zeitliche Charakteristika für das Verhalten von Objekten zu spezifizieren. Ferner ermöglicht sie eine automatische Animationsabfolge, um beispielsweise bestimmte Lernszenarien vorführen zu können, die vom Lernenden anschließend nachzubilden sind.

7 Danksagungen

Diese Arbeit wird gefördert von der DFG (STR 398/4). Unser Dank gilt in besonderem Maße Dr. Eberhard Blümel vom Fraunhofer IFF für die Bereitstellung des Geometriemodelles der Lagerbaugruppe. Den anonymen Gutachtern sei gedankt für ihre kritischen Kommentare sowie konstruktiven Vorschläge zur Erweiterung der vorgestellten Arbeiten. Interessante Denkanstöße für die präsentierte Arbeit lieferte auch Kees van Overveld in mündlicher Diskussion.

Literatur

- [All83] James F. Allen. *Maintaining Knowledge about Temporal Intervals*. Communications of the ACM, 26(11):832–843, 1983.
- [All91] James F. Allen. *Time and Time again: The Many Ways to Represent Time*. International Journal of Intelligent Systems, 6(4):341–355, 1991.

²Referenziert aus ebd.

- [Bar98] Lyn Bartram. *Enhancing Visualizations With Motion*. In: Hot Topics: Information Visualization 1998, S. 13–16, North Carolina, USA, 1998.
- [BJW00] Claudio Bettini, Sushil Jajodia und Sean X. Wang. *Time Granularities*. Springer-Verlag, Berlin Heidelberg New York, 2000.
- [BWC01] Lyn Bartram, Colin Ware und Tom Calvert. *Moving Icons: Detection And Distraction*. In: Proceedings of Interact 2001, Tokyo, 2001.
- [DA96] Alan Dix und Gegory Abowd. *Delays and Temporal Incoherence Due tu Mediated Status-Status Mappings*. SIGCHI Bulletin, 28(2):47–49, April 1996.
- [Fre92a] Christian Freksa. *Temporal Reasoning Based on Semi-Intervals*. Artificial Intelligence, 52(1-2):199–227, 1992.
- [Fre92b] Christian Freksa. *Using Orientation Information for Qualitative Spatial Reasoning*. In: A. U. Frank, I. Campari und U. Formentini, Hrsg., Proceedings Int. Conference on Theories and Methods of Spatio-Temporal Reasoning in Geographic Space, Berlin, 1992. Springer.
- [GG01] Gunjan K Gupta und Joydeep Ghosh. *Detecting Seasonal Trends and Cluster Motion Visualization for Very High Dimensional Transactional Data*. In: Proceedings of First International SIAM Conference on Data Mining (SDM01), 2001.
- [GV94] R. Galetto und F. Viola. *Time as the fourth dimension of the topographic databases*. In: Mapping and Geographic Information Systms. International Society for Photogrammetry and Remote Sensing, 1994.
- [Hal01] Nick Halper. *OpenNPAR – Ein modulares System für nicht-photorealistische Computergraphiken und -animationen*. Vortrag auf dem fmx/01 – Kongress für digitale Medienproduktion, Juni 2001.
- [HSS00] Axel Hintze, Marco Schumann und Stefan Stüring. *Interaktive Szenarien für die Ausbildung von Wartungs- und Instandhaltungspersonal*. In: Thomas Schulze, Peter Lorenz und Volkmar Hinz, Hrsg., Simulation und Visualisierung 2000, S. 225–237, Magdeburg, März 2000. Otto-von-Guericke Universität Magdeburg, SCS European Publishing House.
- [Joh64] Gunnar Johansson. *Perception of motion and changing form*. Scandinavian Journal of Psychology, 5(3):181–208, 1964.
- [Joh73] Gunnar Johansson. *Visual perception of biological motion and a model for its analysis*. Perception and Psychophysics, 14(2):201–211, 1973.
- [JS01] Roland Jesse und Thomas Strothotte. *Motion Enhanced Visualization in Support of Information Fusion*. In: Hamid R. Arabnia, Hrsg., Proceedings of International Conference on Imaging Science, Systems, and Technology (CISST'2001), S. 492–497. CSREA Press, June 2001.

- [Lan93] Gail Langran. *Issues of Implementing a Spatiotemporal System*. International Journal of Geographical Information Systems, 7(4):305–314, 1993.
- [Noi94] Emanuel G. Noik. *A Space of Presentation Emphasis Techniques for Visualizing Graphs*. In: Proceedings of Graphics Interface '94, S. 225–233, 1994.
- [PB96] Philippe Palanque und Rémi Bastide. *Time Modelling in Petri Nets for the Design of Interactive Systems*. SIGCHI Bulletin, 28(2):43–46, April 1996.
- [PBF⁺93] Zenon W. Pylyshyn, J. Burkell, B. Fisher, C. Sears, W. Schmidt und L. Trick. *Multiple parallel access in visual attention*. Canadian Journal of Experimental Psychology, 1993.
- [Red01] Martin Reddy. *Perceptually Optimized 3D Graphics*. IEEE Computer Graphics and Applications, 21(5):68–75, September/October 2001.
- [Ren97] Agnar Renolen. *Temporal Maps and Temporal Geographical Information Systems*, 1997.
- [RF94] Philip K. Robertson und Lisa De Ferrari. *Systematic approaches to visualization: is a reference model needed?*In: L. Rosenblum, R. A. Earnshaw, J. Encarnação, H. Hagen, A. Kaufman, S. Klimenko, G. Nielson, F. Post und D. Thalmann, Hrsg., Scientific Visualization, Kapitel 18, S. 287–305. Academic Press, 1994.
- [RSDP01] Felix Ritter, Thomas Strothotte, Oliver Deussen und Bernhard Preim. *Virtual 3D Puzzles: A New Method for Exploring Geometric Models in VR*. IEEE Computer Graphics and Applications, 21(5):11–13, September 2001.
- [War00] Colin Ware. *Information Visualization: Perception for Design*. Interactive Technologies. Morgan Kaufmann Publishers, 2000.
- [Web00] Gerhard Weber. *Temporale Modellierung multimedialer interaktiver Systeme*. Shaker Verlag, 2000.