

Otto-von-Guericke-Universität Magdeburg  
Fakultät für Informatik



Diplomarbeit

# Visualisierung von Bewegungen in Liniengraphiken

Ronny Schulz

Institut für Simulation und Graphik

1999



Otto-von-Guericke-Universität Magdeburg  
Fakultät für Informatik  
Institut für Simulation und Graphik



Diplomarbeit

# Visualisierung von Bewegungen in Liniengraphiken

Verfasser: Ronny Schulz  
geboren 29. Januar 1974 in Lübz  
Matrikel-Nummer: 150130

Prüfer: Prof. Dr. Thomas Strothotte

Betreuer: Dipl.-Inform. Maic Masuch  
Dipl.-Inf. Stefan Schlechtweg

Beginn: 7. Oktober 1998  
Abgabe: 17. März 1999



# Zusammenfassung

In computergenerierten Liniengraphiken fehlen derzeit Möglichkeiten, Bewegungen in einem einzelnen Bild darzustellen. Diese Arbeit behebt dieses Defizit. Durch Analysen traditionell erstellter Bilder werden geeignete Stilmittel zur Bewegungsdarstellung für den Einsatz in Liniengraphiken ermittelt.

Es werden Algorithmen zur computertechnischen Umsetzung der gefundenen Stilmittel entwickelt, die die Geometrie- und Animationsdaten eines einem Bild zugrundeliegenden Modells nutzen. Dabei wird ein Konzept zur Einbindung dieser Algorithmen in einen Liniengraphik-Renderer entwickelt, so daß sowohl alle Stilmittel einzeln als auch in Kombination eingesetzt werden können.

Das vorgestellte System gibt dem Benutzer die Möglichkeit, die einzusetzenden Stilmittel frei zu wählen und ihr Aussehen zu beeinflussen. Die Ausgabe erfolgt vektororientiert und ist somit völlig auflösungsunabhängig.

## Abstract

Currently, facilities to represent motion in a single image are missing in computer generated line drawings. The present thesis fixes this deficiency. By analyzing pictures that have been created in a traditional way, applicative techniques for the use in line drawings are being worked out.

Algorithms are developed to make these techniques applicable for the use with computers. In order to do this the geometry and animation data of a model that underlies a picture is used. A concept is developed that integrates these algorithms into a line drawing renderer in such a way that every technique can be used separately as well as in combination with any other techniques.

The system presented allows the user to choose arbitrarily a technique he wants to apply and to modify its look. The output is vector based and therefore completely resolution independent.



# Vorwort

An dieser Stelle möchte ich allen danken, die mich während und vor der Erstellung dieser Arbeit hilfreich zur Seite standen.

Zuerst danke ich meiner Mutter und meiner Oma, die mich während dieser Zeit finanziell stark stützten und ohne deren liebevolle Unterstützung das Studium überhaupt nicht möglich gewesen wäre. Ich danke außerdem meinem Vater für die unerhoffte Finanzspritze.

Ich bedanke mich bei meinen Betreuern Stefan Schlechtweg und Maic Masuch sowie bei Bert Freudenberg für ihre kritischen Hinweise und die wertvollen Anregungen. Dabei möchte ich insbesondere Stefan für die selbstlose Hilfe bei diversen technischen Schwierigkeiten während der gesamten Diplomphase danken.

Weiterhin möchte ich mich bei meinen Freunden bedanken, die mich bei der Erstellung dieser Arbeit unterstützt haben – besonders bei Peter Krüger für die exzessive Nutzung seines Scanners, bei Thomas Fuchs für die Hilfe bei der Erstellung von Modellen und bei Axel Hoppe für das Aufzeigen und die Sicherung des Lichts am Ende des Tunnels.

Ich danke weiterhin Jens Elkner für die problemlose Behandlung der erforderlichen Erhöhung der Disc-Quotas sowie Andreas Raab für die guten Denkanstöße zu Beginn der Arbeit.

Ich versichere, daß ich die vorliegende Arbeit allein und nur unter Zuhilfenahme der angegebenen Quellen angefertigt habe.

Magdeburg, den 17. März 1999

Ronny Schulz



# Inhaltsverzeichnis

<b>1</b>	<b>Einleitung</b>	<b>5</b>
<b>2</b>	<b>Grundlagen</b>	<b>7</b>
2.1	Begriffsdefinition . . . . .	7
2.2	Bewegungsdarstellung . . . . .	8
2.3	Methoden zur Darstellung von Bewegungen . . . . .	12
2.3.1	Speedlines . . . . .	12
2.3.2	Pfeile . . . . .	14
2.3.3	Konturwiederholungen . . . . .	15
2.3.4	Andere Methoden zur Bewegungsdarstellung in Einzelbildern .	16
2.3.5	Eignung der Methoden für den Einsatz in Liniengraphiken . .	17
2.4	Verwandte Arbeiten . . . . .	18
<b>3</b>	<b>Konzept</b>	<b>21</b>
3.1	Möglichkeiten der algorithmischen Umsetzung . . . . .	21
3.1.1	Unter Einbeziehung von Geometriedaten . . . . .	21
3.1.2	Ohne Einbeziehung von Geometriedaten . . . . .	22
3.1.3	Vergleich . . . . .	23
3.2	Technische Voraussetzungen und Randbedingungen . . . . .	24
3.2.1	Konzept eines Renderers für Liniengraphiken . . . . .	24
3.2.2	Gestaltung der Linien . . . . .	25
3.2.3	Erweiterung des Renderers um Bewegungsdarstellungen . . . .	25

<b>4</b>	<b>Realisierung</b>	<b>27</b>
4.1	Bewegungspfade . . . . .	27
4.2	Bewegungslinien . . . . .	29
4.2.1	Untersuchungen zur Plazierung von Bewegungslinien in traditionell erstellten Bildern . . . . .	30
4.2.2	Heuristiken zur Generierung von Bewegungslinien . . . . .	31
4.2.3	Bestimmung der Ausgangspunkte von Bewegungslinien . . . . .	31
4.2.4	Interpolation fehlender Eckpunkte . . . . .	35
4.2.5	Umgehung von Überdeckungen . . . . .	38
4.2.6	Einfluß verschiedener Linienstile . . . . .	40
4.2.7	Zusammenfassung des Algorithmus zur Berechnung von Bewegungslinien . . . . .	42
4.3	Konturwiederholungen . . . . .	43
4.3.1	Abgeschwächte Konturen . . . . .	45
4.3.2	Bildung von Teilkonturen . . . . .	46
4.3.3	Zusammenfassung des Algorithmus zur Generierung von Konturwiederholungen . . . . .	49
4.4	Bewegungspfeile . . . . .	49
4.4.1	Pfeildarstellung durch Linienstile . . . . .	50
4.4.2	Pfeildarstellung durch separates Zufügen einer Pfeilspitze . . . . .	51
4.4.3	Zusammenfassung des Algorithmus zur Berechnung von Pfeilen . . . . .	52
<b>5</b>	<b>Beispiele</b>	<b>55</b>
5.1	Translationen graphischer Primitive . . . . .	55
5.2	Einfache Bewegungen einfacher Objekte . . . . .	56
5.3	Bewegungen komplexerer Objekte . . . . .	57
5.3.1	da!l!-Animation . . . . .	57
5.3.2	Animation eines Rennwagens . . . . .	58
5.3.3	Animation eines fliegenden Drachens . . . . .	59
5.3.4	Animation eines Zuges . . . . .	59
5.4	Animation von Teilen eines komplexeren Modells . . . . .	60

---

5.5	Verbesserung der Ergebnisse . . . . .	62
5.5.1	Hinweise zur Modellierung . . . . .	62
5.5.2	Hinweise zur Anwendung des Programmes . . . . .	64
<b>6</b>	<b>Diskussion</b>	<b>65</b>
6.1	Kritik . . . . .	66
6.1.1	Bewegungslinien . . . . .	66
6.1.2	Pfeile . . . . .	68
6.1.3	Konturwiederholungen . . . . .	68
6.2	Weiterführende Betrachtungen . . . . .	69
6.2.1	Darstellung von relativen Bewegungen . . . . .	69
6.2.2	Nutzung der entwickelten Methoden in Animationen . . . . .	70
6.3	Fazit . . . . .	71
<b>A</b>	<b>Bildersammlung</b>	<b>77</b>
A.1	Photographie . . . . .	77
A.2	Konturwiederholungen . . . . .	78
A.3	Pfeile . . . . .	79
A.3.1	Wissenschaftliche Beispiele . . . . .	79
A.3.2	Praktische Beispiele . . . . .	85
A.4	Speedlines . . . . .	86
A.5	Kombinationen . . . . .	88
A.6	Andere Methoden . . . . .	89



# Einleitung

---

Dreidimensionale Computergraphik ist in den letzten Jahren zu einem Begriff geworden, der den meisten Menschen vertraut ist. Der Einsatz spektakulärer Computereffekte in Kino- und Fernsehfilmen fasziniert Millionen von ihnen. Doch dies ist nur eine Seite der Computergraphik, der Photorealismus.

Eine andere Seite ist in der Liniengraphik zu finden. Sie ist ein wichtiges Teilgebiet des Gegenstücks des Photorealismus, des Nicht-Photorealismus. Dabei wird versucht, eine graphische Ausdrucksform zu finden, die nicht primär der möglichst realistischen Abbildung gewidmet ist, sondern andere Ziele verfolgt. So können in einer nicht-photorealistischen Graphik zum Zwecke der Übersichtlichkeit unnötige oder störende Details weggelassen und auf diese Weise das Bild vereinfacht werden. Weiterhin kann der Fokus durch illustrative Techniken auf wichtige Teile einer Szene gesetzt werden. Das Ergebnis dieser Techniken ist ein Bild, dessen Aussehen immer noch realistisch ist, aber Abweichungen aufweisen kann.

Liniengraphiken finden in vielen Bereichen unseres Lebens Verwendung. Vor allem Comics und technische Illustrationen, wie etwa Gebrauchsanleitungen, sind ein häufiges Einsatzgebiet. Dabei befinden sich liniengraphische Darstellungen auf einer höheren abstrakten Ebene als photorealistische. Außerdem sind sie für einen Zeichner schneller und leichter zu erstellen. Liniengraphiken werden auch im täglichen Leben ständig angefertigt: Um einen Sachverhalt klarzustellen reicht normalerweise die Erstellung einer Skizze, also einer Liniengraphik.

Seit Anfang der 90er Jahre beschäftigt man sich in der Computergraphik intensiver mit den „nicht-photorealistischen“ Graphiken. Es entstanden erste prototypische Programme, die aus 3D-Modellen, wie sie für photorealistische Anwendungen verwendet werden, Liniengraphiken erzeugen. Dabei orientiert sich die Gestaltung dieser Graphiken an den Stilmitteln, die von Zeichnern benutzt werden. Den computergenerierten Liniengraphiken fehlt dabei bis heute noch die Möglichkeit, Dynamik und Bewegungen darzustellen, wie es in Comics beispielsweise üblich ist.

In dieser Arbeit werden Verfahren entwickelt, die dieses Defizit beheben. Durch die Anwendung verschiedener stilistischer Mittel, wie etwa Bewegungslinien, Pfeile oder Konturwiederholungen, wird dem Betrachter ein dynamischer Kontext vermittelt. Dabei sollen die verwendeten Stilmittel vom Erscheinungsbild nicht zu sehr von gewohnten Darstellungen abweichen.

Diese Arbeit betritt Forschungsneuland, deshalb ist sie etwas unkonventioneller aufgebaut, als vergleichbare Untersuchungen. So wurde, da es kaum Literatur zu dem Thema gibt, auf eine große Bildersammlung zurückgegriffen, um die von Zeichnern verwendeten Techniken zu analysieren. Diese Analyse führt im Kapitel 2 zur Kategorisierung geeigneter Stilmittel, die daraufhin genauer betrachtet werden. Anschließend werden verwandte Arbeiten vorgestellt.

In Kapitel 3 werden verschiedene Möglichkeiten untersucht, wie die in Kapitel 2 vorgestellten Möglichkeiten für Bewegungsdarstellungen computertechnisch umzusetzen sind. Dabei wird zuerst auf die technischen Voraussetzungen eingegangen und anschließend ein Konzept für einen erweiterten Liniengraphik-Renderer vorgestellt.

Die Realisierung der einzelnen Methoden wird in Kapitel 4 vorgestellt. Dabei werden die entwickelten Algorithmen detailliert beschrieben und durch Beispielbilder veranschaulicht.

Kapitel 5 enthält eine Sammlung von Bildern, die durch den erweiterten Liniengraphik-Renderer erzeugt wurden. Außerdem werden Hinweise zur Handhabung des Renderers gegeben, die die Qualität der Resultate verbessern können.

Abschließend werden die Ergebnisse der Arbeit in Kapitel 6 kritisch betrachtet und Lösungsansätze für auftretende Probleme gegeben, woraufhin weiterführende Gedanken zur Verbesserung und Weiterentwicklung des Programms formuliert werden.

Im Anhang sind typische Vertreter konventionell erstellter Abbildungen zu finden, die Bewegungsdarstellungen zeigen. Sie dienen unter anderem als Grundlage dieser Arbeit und zur Extrahierung der eingesetzten graphischen Mittel.

# Grundlagen

---

In diesem Kapitel wird der Begriff *Bewegung* untersucht und für die weitere Arbeit definiert. Eine Bildanalyse soll traditionelle Wege zur Darstellung einer Bewegung kategorisieren. Die gefundenen Methoden werden daraufhin auf ihre Anwendbarkeit in liniengraphischen Bildern untersucht. Abschließend werden Arbeiten vorgestellt, die sich mit verwandten Themen beschäftigt haben.

## 2.1 Begriffsdefinition

Bevor mit der Untersuchung von Bewegungsdarstellungen begonnen werden kann, muß der Begriff der *Bewegung* definiert werden. Die Wissenschaft kennt bereits mehrere Definitionen. Stellvertretend seien drei von ihnen angeführt werden:

- **Physik:** Ortsveränderung eines Körpers in bezug auf einen anderen Körper oder ein anderes Bezugssystem. Bei der ebenen B. liegt die Bahn in einer Ebene; andernfalls spricht man von einer räuml. B. Bei der gleichförmigen B. ist die Geschwindigkeit des Körpers konstant; bei einer ungleichförmigen B. tritt eine Beschleunigung auf. Bewegen sich die einzelnen Punkte des Körpers gleichförmig auf parallelen Geraden, so handelt es sich um eine fortschreitende B. (Translation). Behält ein einzelner Punkt oder eine Gerade des Körpers seine feste Lage im Raum bei, so spricht man von einer Dreh-B. (Rotation). Eine period. B. liegt vor, wenn der Körper immer wieder in seine Ausgangslage zurückkehrt. [Lex95]
- **Biologie:** passive oder aktive Orts- bzw. Lageveränderung eines Organismus oder von Teilen eines Organismus. Unter passiver B. versteht man alle Ortsveränderungen von Organismen, die ohne Eigenleistung unter Ausnutzung von Umweltenergie erfolgen, z.B. Samenverbreitung durch Wind, Wasser oder Tiere, Schwebeflug und Segelflug von Vögeln. Die aktive B. bezieht sich auf die Lage- oder Ortsveränderung eines Organismus oder seiner Teile, wobei der Organismus die benötigte Energie selbst aufbringen muß. auch Fortbewegung. [Lex95]
- **Mathematik:** kongruente Abbildung, eine eindeutige Abb. des Raumes auf sich, bei der alle Längen erhalten bleiben; Original- und Bildfigur sind kongruent, z.B. Parallelverschiebung (Translation), Drehung um einen Punkt bzw.

eine Gerade (eigtl. B.), Spiegelung an einer Geraden bzw. Ebene (uneigtl. Bewegung). [Lex95]

Jede dieser Definitionen ist speziell angepaßt an das jeweilige Fachgebiet und dort entsprechend anwendbar. Die physikalische Definition der Bewegung kommt dabei am ehesten dem nahe, wovon im weiteren Verlauf ausgegangen werden soll. Trotzdem erscheint eine allgemeine Definition für die Computergraphik angebracht.

**Def.:** *Eine Bewegung ist die Positions- bzw. Lageveränderung eines Objektes relativ zu einem Bezugspunkt, sowie jegliche Art von Volumenänderungen.*

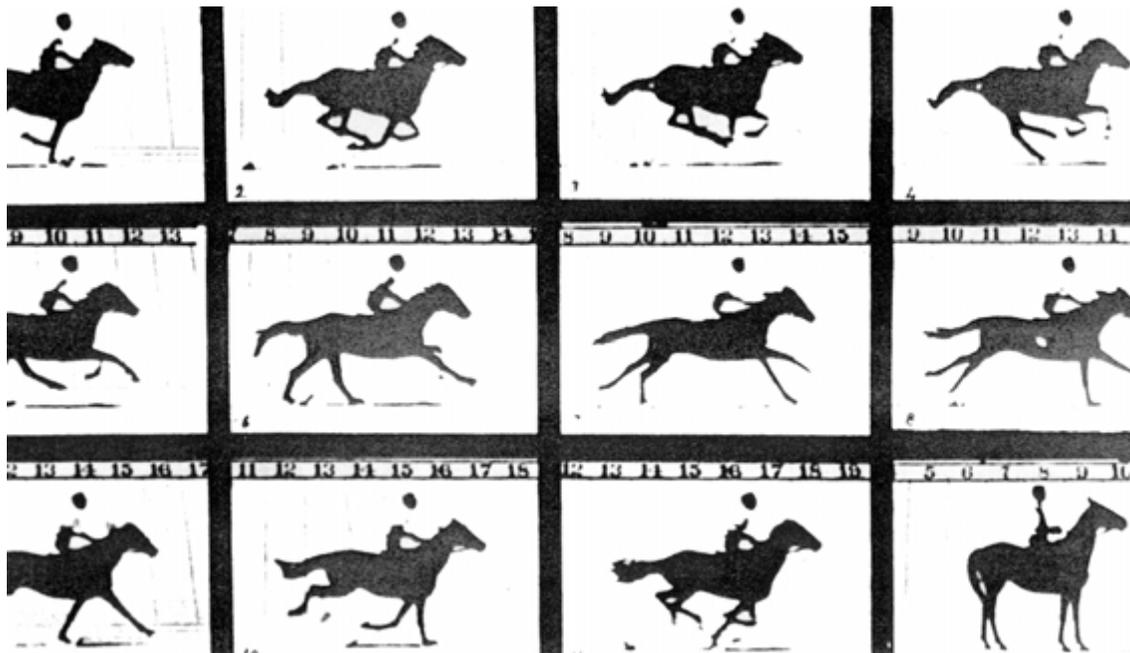
Für diese Arbeit gilt dabei der Zusatz, daß der Bezugspunkt ein feststehender Punkt im Weltenkoordinatensystem sei. Skalierungen und Morphing, also volumenverändernde Methoden der Animation, treten weniger häufig auf, gelten aber auch als Bewegungen.

## 2.2 Bewegungsdarstellung

Bewegungen können mit vielen Sinnen wahrgenommen werden, sie können gesehen, gehört, gefühlt und manchmal auch gerochen werden. Für diese Arbeit ist aber lediglich das Sehen von Bewegungen relevant. Fast alle Lebewesen nehmen Bewegungen von Geburt an wahr. Viele von ihnen können eine Bewegung schneller erkennen als eine Farbänderung. Die gute Erkennbarkeit von Bewegungen stammt dabei aus der Entwicklungsgeschichte. Jedes Lebewesen bewegt sich, und somit deutet eine Bewegung immer auch auf Leben hin. Um das eigene Leben vor einem sich nähernden Feind zu schützen und die Chancen auf eine Flucht zu wahren, ist die schnelle Wahrnehmung einer Bewegung von evolutionärer Bedeutung.

Menschen beschäftigen sich schon sehr lange mit der Darstellung von Bewegungen, jedoch fehlten ihnen lange Zeit geeignete Mittel. Viele Zeichnungen, angefangen bei alten Höhlenmalereien, stellen Szenen dar, in denen sich etwas bewegt. Es wurde das gemalt, was die Menschen beschäftigt, so anfänglich auch viele Jagdszenen. Dabei wurden oft typische Posen abgebildet, wie etwa das Ausholen zum Wurf eines Speers oder das Spannen eines Bogens. Diese Momentaufnahmen waren so aussagekräftig, daß der Betrachter genau wußte, worum es ging, und in seiner Phantasie das Bild mit Leben erfüllte.

Die Erfindung der Photographie trug anfänglich nur wenig zur Bewegungsdarstellung bei. Dies lag vorwiegend daran, daß zum Belichten eines Bildes mehrere Sekunden notwendig waren, um ein klares Bild zu erhalten. Da Photographien damals relativ teuer waren, wurde mit ihnen wenig experimentiert. Vielmehr empfand man es als unschön, wenn eine Bewegung auf einem Photo sichtbar war, da sie nur als unscharfes Etwas gedeutet werden konnte. Mit der Zeit entwickelte sich auch diese Technik weiter und die Belichtungszeit wurde kürzer und inzwischen setzen viele Photographen gezielt Bewegungsunschärfe als Stilmittel ein. [Web90]



**Abbildung 2.1:** Mit einer Photoserie bewies der englische Photograph Eadweard James Muybridge 1872, daß Rennpferde alle vier Hufe gleichzeitig vom Boden heben.

1872 machte der englische Photograph Eadweard Muybridge erstmals Serienaufnahmen einer Bewegung, die für das menschliche Auge nicht auflösbar war. Mittels mehrerer hintereinander ausgelöster Kameras nahm er eine Bewegungsstudie eines galoppierenden Pferdes auf. Er bewies mit den Bildern, daß Rennpferde alle vier Hufe gleichzeitig vom Boden abheben können. Abbildung 2.1 zeigt eine Reproduktion der historischen Aufnahmen. [Pat88]

Der wohl wichtigste Schritt in der Entwicklung der Bewegungsdarstellung war die Erfindung der „bewegten Bilder“, einer Vorstufe des Films. Dabei wurden gezeichnete oder photographische Bewegungsabläufe auf eine rotierende Scheibe mit einem Ring mit Sichtschnitten befestigt. Auf diese Weise konnten kurze Bewegungssequenzen dargestellt werden.

Die Darstellung der Bewegungen faszinierte die Menschen und die bewegten Bilder wurden zu einer großen Attraktion auf Jahrmärkten. Mit der Entwicklung des Films verloren die bewegten Bilder jedoch bald ihren Reiz. Etwa zur gleichen Zeit begannen einige Zeichner von Comics, sich Gedanken zu machen, wie sie Bewegungen in ihren Werken ausdrücken konnten. Dabei spielt sicherlich die zu der Zeit vorherrschende Art des Humors eine Rolle. Die Möglichkeit, Bewegungen festzuhalten, aber ohne Ton auskommen zu müssen, wurde von vielen Schauspielern, die das Publikum unterhalten wollten, vornehmlich zu Slapstick-Einlagen genutzt. Da die Comics eigentlich immer die Zeit widerspiegeln, in denen sie geschaffen werden, und da insbesondere in Amerika Slapstick sehr populär war, mußten Techniken geschaffen werden, den Witz, den die Filme verbreiteten, für die Comics nutzbar zu machen.



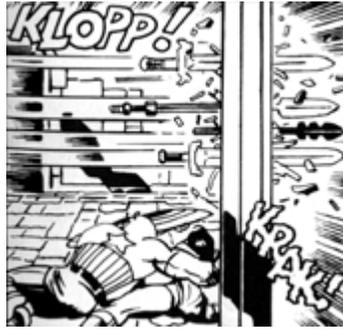
Abbildung 2.2: Die Anfänge der heute sehr populären Speedlines.

Durch starke Abstraktionen entwickelte sich ein Stilmittel, daß die Bewegungsdarstellung auch in Comics ermöglichte und das vom Betrachter nicht als Objekt eingeordnet wurde, die *Speedline*, eine einfache Linie. Abbildung 2.2 zeigt einen der ersten Comics, in denen Speedlines eingesetzt werden. Die Linien zeichnen dabei den Weg nach, den der zugehörige Teil des Bildes vorher zurücklegte. Anfänglich versuchte man, die Bewegungen physikalisch korrekt nachzuzeichnen. Mit der Zeit wurden aber auch diese Mittel immer weiter stilisiert. Heute sind Speedlines aus Comics kaum noch wegzudenken. Jeder Künstler hat seine eigene Art, diese Linien einzusetzen. Die Erscheinungsform der Speedlines ändert sich dabei modebedingt geringfügig. Abbildung 2.3 zeigt am Beispiel eines Comics von MARVEL aus [McC93] die deutlichen Veränderungen der Art der Darstellung dieser Linien.

Doch die Kreativität der Comic-Zeichner brachte noch weitere Methoden hervor, die das Ergebnis künstlerischer Abstraktionsprozesse sind. Aus der Idee heraus, so schnell zu sein, daß man an mehreren Orten gleichzeitig gesehen wird, entwickelte sich eine Technik, bei der ein extrem schnelles Objekt mehrmals in einem Bild gezeichnet wird. Diese Technik wurde frühzeitig mit dem Einsatz von Speedlines kombiniert und weiterentwickelt. Besonders beliebt ist bei den Comic-Zeichnern das „Kampfknäuel“, bei dem vom eigentlichen Kampf nicht sehr viel zu sehen ist, außer ein paar Fäusten und anderen schlagkräftigen Dingen (siehe Abbildung 2.4). Diese Technik wurde besonders häufig in den Cartoons der 40er und 50er Jahre eingesetzt.

Eine weitere, nicht nur in Comics angewandte Technik ist, wichtige Positionen einer Bewegung zu zeichnen und überflüssige Zwischenphasen auszulassen. Diese werden vom Betrachter automatisch interpoliert. Diese Technik wird von MCCLOUD in [McC93] als „closure“ bezeichnet. Wichtig ist dabei, daß alle Hauptphasen auf einmal sichtbar sind und nicht durch einen Seitenwechsel unterbrochen werden.

Filme leben von der Bewegung. Ein Film ist dabei aber nichts anderes als eine Aneinanderreihung von vielen Bildern. Diese Erkenntnis brachte die ersten Filme ohne Schauspieler hervor, die Trickfilme. In den Trickfilmen waren die Künstler nicht an die Gesetze der Physik gebunden. Entsprechend wurden Handlungen verfilmt, an deren Umsetzung normale Filme scheitern mußten. Dabei wurden auch viele populäre Effekte aus den Comics übernommen, wie zum Beispiel die Speedlines.



**Abbildung 2.3:** Die Anwendung der Speedlines verfeinerte sich immer weiter. Die neue Dynamik der Bilder kam beim Publikum sehr gut an. In: Scott McCLOUD, Understanding Comics



**Abbildung 2.4:** Ein Beispiel für die populären „Kampfknäuels“. Bugs Bunny wird von einem Chinesen herumgewirbelt. Bugs selbst ist in dem Bild nicht mehr erkennbar. In: That's All Folks, Aurum Press, 1994

Für die Anwendung im Trickfilm wurden sie an die neuen Bedürfnisse angepaßt. Abbildung 2.5 zeigt dies im Disney-Film „Dumbo, der fliegende Elefant“.

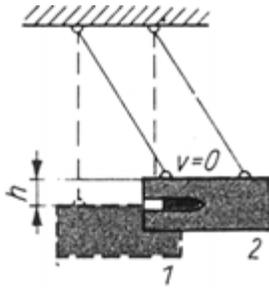
Doch nicht nur in Comics und Trickfilmen, auch in der Wissenschaft werden Bewegungen dargestellt. So werden beispielsweise im Maschinenbau rotierende Teile durch einen zusätzlichen Pfeil gekennzeichnet. Abbildung 2.6 zeigt ein Beispiel aus der Physik. Zu einer Übungsaufgabe wird ein erklärendes Bild angegeben. In diesem Bild werden Anfangs- und Endposition einer Bewegung dargestellt. Dabei können die verschiedenen Positionen durch die unterschiedliche Darstellung auseinandergehalten werden.

In der Geologie werden tektonische Bewegungen durch Pfeile dargestellt; das gleiche Mittel wird auch zur Verdeutlichung von Strömungen in mehreren Wissenschaftsbereichen benutzt. Eine ganz spezielle Art der Bewegungsdarstellung haben die Meteorologen entwickelt, um die Bewegung von Wetterfronten darzustellen. Um diese Darstellung zu verstehen, muß man sich jedoch intensiv damit beschäftigen. Im Anhang sind verschiedene Beispiele wissenschaftlicher Illustrationen aufgeführt, darunter auch eine Wetterkarte.

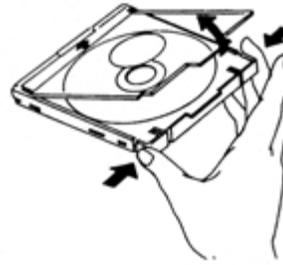
Schließlich sind Hinweise auf Bewegungen auch in Gebrauchsanweisungen zu finden. Hier werden Schritte in Bildern näher erklärt, die durch den reinen Text vielleicht unverständlich geblieben sind. Mögliche Bewegungen werden dabei durch Pfeile dargestellt (siehe Abbildung 2.7).



**Abbildung 2.5:** Dumbo der fliegende Elefant. In: The Illusion Of Life: Disney Animation, Hyperion, 1981



**Abbildung 2.6:** Physikalische Zeichnung zur Bemessung der Objektpositionsänderung. Anfangs und Endposition des Auffangbehälters sind unterscheidbar gezeichnet worden.



**Abbildung 2.7:** Handhabung eines Caddy. In: Plector UltraPlex Bedienungsanleitung

## 2.3 Methoden zur Darstellung von Bewegungen

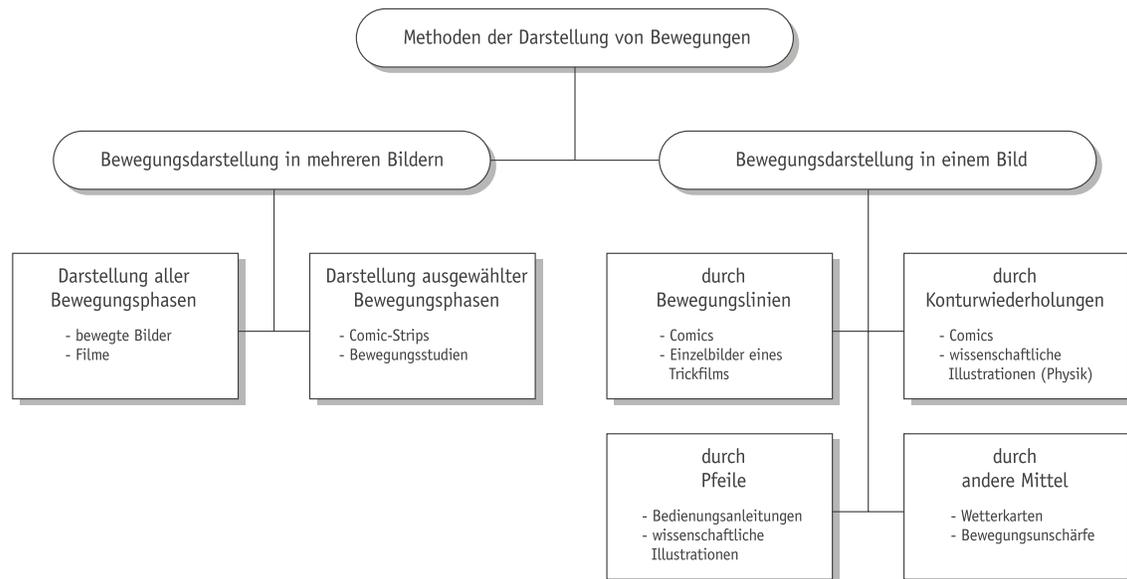
Der vorige Abschnitt beschrieb die Entwicklung der Methoden zur Bewegungsdarstellung. Dabei stellte sich heraus, daß Bewegungen sowohl in mehreren Bildern als auch in einem einzigen Bild visualisiert werden können. Die Bewegungen sind mittels mehrerer Bilder natürlich besser darstellbar als nur mit einem, entsprechend gibt es bereits computertechnische Umsetzungen für die Erzeugung von Film- oder Bildsequenzen. Der Film war dabei bis jetzt immer das große Vorbild in der Computergraphik und viele Effekte aus Filmen sind bereits per Computer zu erstellen.

Ganz anders sieht die Lage bei der Bewegungsdarstellung in einem einzigen Bild aus. Die in Photos auftretende Bewegungsunschärfe kann bereits per Computer automatisch berechnet werden (siehe [KB83] oder [PC83]). Durch Photomontagen können teilweise auch Konturwiederholungen umgesetzt werden, dies geschieht normalerweise aber per Hand. Die Wetterkarten der Meteorologen werden in der Regel mit Hilfe des Computers erstellt, ob dieser aber auch die Bewegungsdarstellung übernimmt ist unklar. Die anderen aufgeführten Methoden wie die Speedlines oder die Pfeile sind für computertechnische Umsetzungen jedoch noch nicht oder wenig beachtet worden. Entsprechend gibt es kaum Fachliteratur zu diesem Thema.

Abbildung 2.8 soll die Methoden der Bewegungsdarstellung kategorisieren und zusammenfassen. Die dabei aufgeführten Methoden zur Darstellung von Bewegungen in Einzelbildern werden im weiteren genauer untersucht.

### 2.3.1 Speedlines

Wie bereits erwähnt, sind Speedlines eine aus der Comicwelt entstandene Methode der Bewegungsdarstellung und sind so gut wie nirgendwo anders zu finden. Der Grund dafür ist, daß Comics lange Zeit als Trivilliteratur galten und entsprechend behandelt wurden. Niemand wollte Comics auch nur den geringsten Anspruch zugestehen, und ein Einsatz der in ihnen benutzten Stilmittel erschien vielen einfach ebenso anspruchslos.



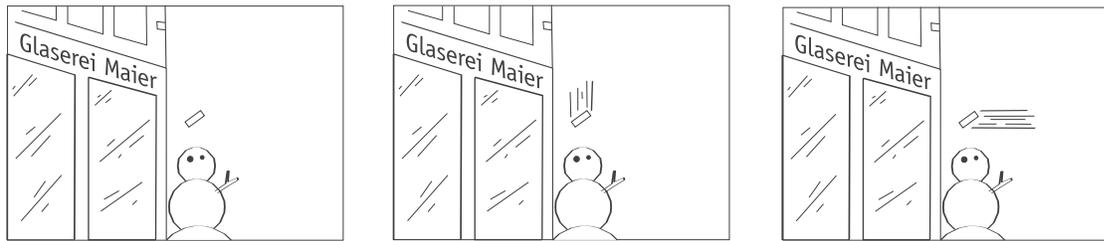
**Abbildung 2.8:** Übersicht über die Möglichkeiten der Darstellung von Bewegungen.

Heutzutage bilden Comics eine anerkannte Kunstform. Es gibt sogar Lehrbücher, die in Form eines Comics geschrieben sind. Das wohl bekannteste von ihnen ist das von Scott McCloud verfaßte Lehrbuch „Understanding Comics“, ein Comic über die Kunst, einen Comic zu erstellen.

Trotz der neuen Offenheit sind die in den Comics benutzten Stilmittel, deren Technik inzwischen sehr ausgefeilt ist, noch nicht zu einem weiteren Einsatz gekommen. Dabei sind sie sehr vielen Menschen durchaus vertraut und so veranlagt, daß sie von jedem zu verstehen sind.

Der Begriff *Speedlines* (deutsch: Geschwindigkeitslinien) entstand nach der Funktion, die sie erfüllten. Sie sollten dem Betrachter die Geschwindigkeit eines Objektes verdeutlichen. Eigentlich zeigen sie jedoch nicht die Geschwindigkeit. Statt dessen vermitteln sie dem Betrachter, daß sich ein Objekt überhaupt bewegt. Der Speedline kann man die genaue Geschwindigkeit des Objektes nämlich nicht entnehmen, sondern höchstens eine Aussage über die relative Geschwindigkeit zu einem anderen bewegten Objekt treffen. Weiterhin können auch Rückschlüsse auf die Bewegungsrichtung gezogen werden. Aus diesem Grund wird im weiteren Verlauf eher von *Bewegungslinien* als von Speedlines gesprochen, auch wenn die Begriffe im Grunde das gleiche beschreiben. Bewegungslinie trifft die eigentliche Funktion wesentlich besser.

Die Bewegungslinien zählen STROTHOTTE und STROTHOTTE in [SS97] zu der Menge der graphischen Symbole, deren Anwendung das jeweilige Bild als abstrakt-graphisch klassifiziert. Im Falle der Bewegungslinien machen die Autoren aber eine Ausnahme. Als Begründung wird angegeben, daß ein Comic auch ohne diese Linien zu verstehen sei und diese These wird an einem Beispiel gezeigt. Für einen Comic-Strip mag dies vielleicht stimmen, für ein einzelnes Bild kann eine solche allgemeine Aussage aber nicht gelten. Ein Gegenbeispiel soll verdeutlichen, daß Bewegungs-



**Abbildung 2.9:** Ein Ziegelstein befindet sich über dem Kopf eines Schneemanns und vor den Fenstern einer Glaserei. Wohin der Ziegel fliegt, kann im linken Bild nicht mit absoluter Sicherheit gesagt werden. Erst entsprechende Bewegungslinien geben die genaue Flugrichtung an und stellen sicher, ob die Scheibe der Glaserei zu Bruch geht, oder der Ziegel den Schneemann treffen wird.

linien durchaus disambiguierende oder gestalterisch ordnende Funktionen übernehmen können, die helfen, einen Comic richtig zu verstehen. Abbildung 2.9 illustriert, daß erst durch die zusätzliche Angabe von Bewegungslinien der richtige Sinn des Bildes erfaßt werden kann. Aus diesem Grunde kann keine allgemeingültige Klassifizierung der Comics getroffen werden. Es ist für jedes Bild einzeln zu entscheiden, ob ein abstrakt-graphisches Mittel den Sinn eines Bildes zum Ausdruck bringt oder nur schmückendes Beiwerk ist.

### 2.3.2 Pfeile

Auch Pfeile werden von STROTHOTTE und STROTHOTTE als graphische Symbole definiert. Bei der Verwendung von Pfeilen gibt es für sie nicht den geringsten Zweifel einer Klassifizierung des Bildes als abstrakt-graphisch. Dem kann aufgrund der vielfältigen Einsatzgebiete für Pfeile auch nicht widersprochen werden. Eine Klassifizierung dieser graphischen Hilfsmittel nach Art des Einsatzes wurde bereits in [Bö87] erstellt:

- *Pfeile als Mittel der Hervorhebung*  
Hierbei weist ein Pfeil auf ein Objekt hin, welches die Aufmerksamkeit des Betrachters erregen soll. Der Pfeil selbst ist dabei als Symbol zu sehen, dessen Bedeutung mit „Achtung! Wichtig!“ angegeben werden kann.
- *Pfeile zur Angabe von Bewegungsmöglichkeiten*  
Pfeile können auch als Symbol für die Beweglichkeit eines Objektes benutzt werden. Durch die Pfeile wird sowohl das Objekt gekennzeichnet, als auch jede sinnvolle Bewegung angegeben.
- *Pfeile zur Angabe einer Funktion*  
Bringt die Bewegung eines Objektes eine bestimmte Wirkung hervor, so kann dies ebenfalls durch Pfeile signalisiert werden. Dabei tritt die Bewegung an sich mehr in den Hintergrund. Der Pfeil gibt damit sowohl Möglichkeit und Richtung der Bewegung als auch den Hinweis auf die Funktion an.
- *Pfeile zur Kennzeichnung von physikalischen Phänomenen*  
Hat ein Objekt eine bestimmte Wirkung, die aber normalerweise nicht darge-

stellt werden kann, so bieten sich Pfeile als Symbol dieses Phänomens an. Auf diese Weise können beispielsweise Luftströmungen erkennbar gemacht werden. Als Beispiele seien hier die Abbildungen A.12, A.13 oder A.21 im Anhang angeführt.

- *Pfeile zur Angabe von Größen*  
Zur Bemaßung von Objekten werden Pfeile in Zusammenhang mit Zahlen als Symbol zur Größenangabe benutzt. Ein alleiniger Einsatz von Pfeilen ohne Zusatzinformationen durch weitere Hilfsmittel erscheint aber nicht sinnvoll.
- *Pfeile als Mittel der Beschriftung*  
Schließlich können Pfeile auch als Symbol einer Zuordnung eines Textes zu einem Teil eines Objektes benutzt werden. Der Pfeil deutet dabei vom Text zum Objekt hin.

All diese Einsatzgebiete genügen der Definition eines abstrakt-graphischen Bildes in [SS97]: „In abstrakt-graphischen Bildern können Eigenschaften und Beziehungen in der Realität, die für Menschen nicht sichtbar sind, auf abstrakte Art und Weise präsentiert werden. Das Problem, unsichtbare Eigenschaften dem menschlichen Auge zu verdeutlichen, kann auf zwei Weisen gelöst werden: *durch Hinzufügen graphischer Symbole* oder *durch das Abbilden unsichtbarer Eigenschaften auf sichtbare Attribute.*“<sup>1</sup>

In Anbetracht der Möglichkeiten, Pfeile auch zur Darstellung von Bewegungen zu nutzen, wie dies beispielsweise in Abbildung 2.10 aus einem Jagdlexikon zu sehen ist, wird die Klassifizierung der Einsatzgebiete der Pfeile durch folgende Kategorie erweitert:

- *Pfeile zur Visualisierung einer zukünftigen Bewegung*  
Geht ein Pfeil von einem Objekt aus in eine bestimmte Richtung, so ist er als Symbol für eine zukünftige Bewegung des Objektes/Objektteiles in die angegebene Richtung zu betrachten.

Auch dieses Einsatzgebiet ist durch die Definition abstrakt-graphischer Bilder erfaßt.

Anders als Bewegungslinien finden Pfeile in Comics so gut wie keine Anwendung. Vielmehr ist ihr Einsatzgebiet eher in der Wissenschaft oder in Gebrauchsanweisungen zu sehen. Pfeile repräsentieren im Gegensatz zu Bewegungslinien eine zukünftige oder eine mögliche Bewegung eines Objektes.

### 2.3.3 Konturwiederholungen

Anders als Pfeile und Bewegungslinien lassen sich die eingesetzten Mittel bei Konturwiederholungen nicht als graphische Symbol definieren. Trotzdem wird dem mensch-

---

<sup>1</sup> Originalwortlaut: In abstract-graphical pictures, properties of and relations in reality which are invisible to humans can be presented in an abstract way. The problem of making invisible properties obvious to the human eye can be solved in two ways: *adding graphical symbols ...* [or] *mapping invisible properties onto visible attributes.* [SS97]



Um dem *temporal Aliasing* entgegenzuwirken, wurden verschiedene Techniken entwickelt. KOREIN und BADLER beschreiben in [KB83] Methoden, wie dem temporal Aliasing begegnet werden kann. Dabei werden zwei Algorithmen erläutert, wie speziell das Motion Blurring in computergenerierten Animationen erreicht werden kann. POTMESIL stellt in [PC83] ein Kamera-Modell vor, bei dem die Bewegung eines Objektes in der Zeit zwischen zwei Frames verfolgt wird und die Aufnahme durch eine Shutter-Kamera simuliert wird. Als Ergebnis wird ein äußerst realistisch aussehendes Motion Blurring erwartet.

Zwar wurden die Techniken wie bereits erwähnt unter einem anderen Vorwand entwickelt, doch das Ergebnis kann auch dazu genutzt werden, in Einzelbildern Bewegungen zu zeigen.

Weiterhin gibt es ganz spezielle Möglichkeiten, Bewegungen darzustellen, die aber fast alle nur sehr schwer verständlich sind, wie das bereits erwähnte Modell zur Darstellung von Bewegungen von Wetterfronten. Solche Methoden sind ungeeignet für eine allgemeinere Beschreibung von Bewegungen und werden deshalb nicht näher untersucht.

### 2.3.5 Eignung der Methoden für den Einsatz in Liniengraphiken

Liniengraphiken haben ganz bestimmte Eigenschaften, die sie von photorealistischen Bildern unterscheiden. So wird bei Liniengraphiken kein Anspruch auf Photorealismus erhoben, sondern eher Wert auf Erkennbarkeit gelegt. Auf Farben wird weitestgehend verzichtet, die meisten Liniengraphiken beinhalten nur Hintergrund- und Zeichenfarbe. Dadurch ist die Darstellung mittels Liniengraphiken automatisch abstrakter als eine Photographie des gleichen Objektes. Bevorzugt zur Wiedergabe von Liniengraphiken ist die Repräsentation als Vektorgraphik, also Pixel- und Auflösungsunabhängigkeit. Eine besondere Eigenschaft von Liniengraphiken ist hohe verlustarme Kopierbarkeit. Diese Eigenschaften sollten auf jeden Fall erhalten bleiben.

Die Anwendung von Bewegungslinien und Pfeilen erscheint problemlos, da es sich bei diesen Methoden selbst um liniengraphische Mittel handelt. Alle Eigenschaften des Bildes bleiben erhalten. Wenn Konturwiederholungen in der gleichen Art und Weise dargestellt werden, wie das Originalobjekt, sollten auch diese lediglich mittels Linien zeichenbar sein. Somit sind auch Konturwiederholungen als Mittel der Bewegungsdarstellung in Liniengraphiken einsetzbar.

Die Anwendbarkeit von Bewegungsunschärfe-Algorithmen in Liniengraphiken wird als Ergebnis der Untersuchungen in [Sch98] verneint. Unschärfe ist ein zu großer Widerspruch zu den Eigenschaften von Liniengraphiken. Somit ist Bewegungsunschärfe als ungeeignetes Mittel zu betrachten, wenn die Eigenschaften eines liniengraphischen Bildes erhalten bleiben sollen.

## 2.4 Verwandte Arbeiten

Der Darstellungen von Bewegungen in einem einzigen Bild wurde in der Computergraphik bis jetzt noch nicht sehr viel Aufmerksamkeit gewidmet. Entsprechend rar ist Fachliteratur zu diesem Thema.

WEBER zeigt in [Web90], wie Bewegungen mit Hilfe der Photographie festgehalten werden können. Er stellt fest, daß sich beim „Stehbild“ Bewegung nicht unmittelbar wiedergeben läßt und weist auf zwei Techniken hin, die in einem Photo den Eindruck von Bewegung erwecken.

Bei der ersten Technik wird mit einer für die Bewegungsgeschwindigkeit des Objektes zu langen Verschußzeit photographiert. Als Resultat wird das bewegte Objekt unscharf vor einem scharfen Hintergrund abgebildet.

Die zweite Technik erzielt bei einer relativ langen Verschußzeit durch das Mitziehen der Kamera parallel zur Bewegungsrichtung des Objektes eine scharfe Abbildung desselben vor einem verwischten Hintergrund. Die Anwendung dieser Technik erfordert aber viel Übung und eine gute Abschätzung der voraussichtlichen Bewegung des Objektes.

Beide Techniken erzielen im Endeffekt eine Bewegungsunschärfe, deren computertechnische Umsetzung bereits von KOREIN und BADLER in [KB83] oder auch von POTMESIL in [PC83] formuliert wurde.

Für Liniengraphiken ist die Bewegungsunschärfe ungeeignet. Die Anwendung anderer Mittel wurde bis jetzt so gut wie gar nicht untersucht. Es gab lediglich einen einzigen Versuch, Bewegungsdarstellungen in Liniengraphiken zu integrieren, der in [SPR<sup>+</sup>94] erwähnt wird. STROTHOTTE et al. beschreiben darin ein System, das stilistische Elemente, die der Betrachter nicht als Objekte interpretiert, zu einer computergenerierten Zeichnung dazufügt, um die visuelle Impression von Bewegung zu erzeugen. Es werden drei Möglichkeiten angegeben, dies zu erreichen:

- *Überlagerungen von bewegten Objekten*  
Bewegte Objekte werden zu vergangenen und zukünftigen Positionen gezeigt. Der Benutzer kann spezifizieren, ob ein gesamtes Objekt oder nur die vordere Kante gezeichnet wird.
- *Nutzung zusätzlicher Symbole*  
Nutzung von Linien, deren Dicke und Länge proportional zur Geschwindigkeit des Objektes ist.
- *Änderungen in der Objektgeometrie*  
Das Objekt wird beispielsweise am Beginn einer Bewegung mit hoher Beschleunigung langgezogen dargestellt.

Diese Kategorisierung ist weitgehend kompatibel zu der Kategorisierung, die in Abschnitt 2.3 vorgestellt wurde.

In das Versuchssystem in [SPR<sup>+</sup>94] wurden lediglich die ersten beiden Methoden implementiert. Bei der vorgestellten Umsetzung war der Eingriff des Benutzers insofern notwendig, daß für jedes bewegte Objekt ein Bewegungsvektor eingezeichnet werden mußte. Durch den zugrundeliegenden Linien-Renderer wurden die Angaben modifiziert und in die entsprechenden Methoden umgesetzt.

Über die Art der Umsetzung wurden keine weiteren Angaben gemacht. Dadurch, daß der Benutzer die Bewegungsrichtungen *im Bild* selber angibt, ist die Bewegungsfreiheit sehr eingeschränkt. Zu mehr als einfachen Translationen war das System nicht fähig. Darum mußten alternative Quellen gesucht werden.

Mit der Technik der Bewegungsdarstellung in nur einem Bild befaßten sich vorwiegend Zeichner von Comics. Obwohl in ihnen nicht näher auf die anzuwendende Technik eingegangen, haben sich die Bücher SCHNEIDERS [Sch94], THOMAS' und JOHNSONS [TJ81] und vor allem MCCLOUDS [McC93] als grundlegende Quellen herausgestellt. Die beiden ersten Werke beinhalten Beschreibungen über die Werke von Zeichnern bei Warner und Disney mit einem Blick auf die jeweilige Entstehung. Das Buch von MCCLOUD hingegen ist ein Lehrbuch zum Verstehen eines Comics und selbst als Comic gehalten. In ihm geht der Autor auf verschiedene Techniken zum Erschaffen eines Comics ein.



In diesem Kapitel werden verschiedene Möglichkeiten der algorithmischen Umsetzung der ermittelten Methoden zur Bewegungsdarstellung betrachtet. Anschließend werden technische Grundlagen für die gewählte Umsetzungsmethode betrachtet und ein Konzept für einen erweiterten Liniengraphik-Renderer vorgestellt.

### 3.1 Möglichkeiten der algorithmischen Umsetzung

Im folgenden werden verschiedene Gedanken zur algorithmischen Umsetzung der genannten, für Liniengraphiken geeigneten, Methoden formuliert. Ausgegangen wird dabei von zwei völlig unterschiedlichen Ansätzen. Der erste setzt voraus, daß einem Bild ein geometrisches Modell und entsprechende Animationsdaten zugrunde liegen. Der zweite Ansatz ist rein bildorientiert. Beide Ansätze werden genauer analysiert und abschließend verglichen.

#### 3.1.1 Unter Einbeziehung von Geometriedaten

Liegt einem Bild eine nutzbare Geometrie zugrunde oder wird dieses Bild erst aus der Geometrie berechnet, bieten sich gute Ansätze für eine algorithmische Umsetzung aller drei Methoden an. Mit dem zugrundeliegenden Modell und den vorhandenen Animationsdaten kann jedes Objekt zu jedem Zeitpunkt der Animation verfolgt, berechnet und gezeichnet werden.

##### **Konturwiederholungen**

Konturwiederholungen sind durch die zugrundeliegenden Geometrie- und Animationsdaten relativ einfach umzusetzen. Das bewegte Objekt wird mehrfach berechnet, für jede Wiederholung neu. Um dies zu erreichen, wird das Objekt für die jeweilige Wiederholung so transformiert, daß es die Position einnimmt, in der es sich zu der korrespondierenden Zeit befand. Anschließend wird es auf die gleiche Weise in die 2D-Koordinaten projiziert, wie die Originalszene zum aktuellen Zeitpunkt.

Auf diese Weise wird für jede geplante Wiederholung ein weiteres Objekt berechnet, das anschließend in die Projektionsebene der Kamera des eigentlichen Frames projiziert wird. Durch verschiedene Zeichenmethoden können dann die verschiedenen Objekte entsprechend unterschiedlich abgebildet werden.

### **Bewegungslinien**

Für die Bewegungslinien müssen neue Methoden gefunden werden, da diese im Objekt selbst nicht vorhanden sind. Die Animationsdaten geben Aufschluß über die Bewegung eines Objektes. Die Transformationsmatrizen sind für alle Punkte eines Objektes gleich. Wenn also ein fester Punkt im Objekt über die vergangenen Frames verfolgt wird, hat man als Resultat dieser Verfolgung eine Bewegungslinie erhalten. Die sinnvolle Auswahl der Punkte ist Voraussetzung für realistisch wirkende und sich harmonisch in das Gesamtbild einfügende Bewegungslinien.

Die Bewegungslinien können, falls das Bild erst berechnet wird, zur gleichen Zeit generiert werden, ansonsten auch zusätzlich zu einem späteren Zeitpunkt. Dabei müssen entsprechende Methoden gefunden werden, wie beispielsweise überdeckte Linien gezeichnet werden.

### **Bewegungspfeile**

Pfeile können auf eine ähnliche Art erstellt werden wie Bewegungslinien. Die Verfolgung der Ausgangspunkte für die Linien muß aber für die folgenden Frames vorgenommen werden. An das Ende dieser „Zukunftsbewegungslinie“ muß dann noch eine Pfeilspitze angebracht werden. Es muß weiterhin darauf geachtet werden, daß die Pfeilspitze im Bild sichtbar ist.

All diese Algorithmen setzen natürlich voraus, daß die Modelle sinnvoll an die neuen Ansprüche der Animation angepaßt werden, denn es kann keinen Algorithmus geben, der alle Sonderfälle beachtet. Der Animator muß eine Vorstellung von dem haben, wie das Endergebnis aussehen soll.

Durch die zusätzlichen Informationen aus den Animationsdaten sollte es möglich sein, die Methoden zur Bewegungsdarstellung so umzusetzen, daß der Benutzer in das System nur marginal eingreifen muß und die Visualisierung der Bewegung automatisch nach den Vorgaben des Benutzers erfolgt.

## **3.1.2 Ohne Einbeziehung von Geometriedaten**

Liegt dem Bild keine Geometrie zugrunde, ist eine analytische Betrachtung des Bildes deutlich schwieriger. Hier ist ein manueller Eingriff des Benutzers nötig.

### **Bewegungslinien und Bewegungspfeile**

Die Generierung von Bewegungslinien kann durch den Computer unterstützt werden. Denkbar ist hier ein interaktives System, bei dem der Benutzer das zu bewegendes Objekt maskiert und anschließend einen Bewegungsvektor angibt.

Auf diese Weise können sowohl Bewegungslinien als auch Pfeile erzeugt werden. Durch die Ermittlung der Punkte, deren Höhe des Lotes auf den angegebenen Bewegungsvektor am größten ist, ist die Erzeugung der äußeren Bewegungslinien möglich. Diese könnten dann noch durch den Benutzer perspektivisch verzerrt werden. Die übrigen Bewegungslinien liegen auf jeden Fall zwischen den beiden bereits berechneten und sind somit interpolierbar.

### Konturwiederholungen

Konturwiederholungen stellen ohne eine zugrundeliegende Geometrie ein Problem dar. Bei einfachen Translationen könnten die Konturen des maskierten Objektes einfach versetzt neu gezeichnet werden. Schwierig gestaltet sich allerdings eine Rotationsbewegung des maskierten Objektes, die nicht nur um die z-Achse geschieht. Da keine 3D-Daten über das Objekt vorliegen, sind keine Informationen über das Aussehen aus anderen Perspektiven gegeben. Dadurch sind die anderen Konturen nicht zeichenbar.

Eine vollautomatische Umsetzung ist mittels dieser Methode nicht zu erreichen. Der Computer kann weder ermitteln, welches Objekt sich in einem Einzelbild bewegt, noch kann er die Art und Richtung der Bewegung berechnen. Eine Wiedererkennung eines bereits in einem früherem Bild vorhandenen und bearbeiteten Objekt ist nicht oder nur sehr schwierig möglich.

### 3.1.3 Vergleich

Für die Methode mit Einbeziehung der Geometrie müssen ein 3D-Modell der Szene und Animationsdaten erstellt werden bzw. vorhanden sein. Für die zweite Methode ist dies nicht nötig. Dafür muß der Anwender dort für jedes einzelne Bild stark interagieren. Eine automatische Bewegungsdarstellung ist sonst nicht möglich.

Ist jedoch eine Geometrie vorhanden, ist die Darstellung der Bewegung teilweise sogar vollautomatisch umsetzbar. Vorteil bei der Nutzung der Geometrie- und Animationsdaten ist die Berechnung perspektivisch korrekter Ergebnisse. Ohne solche Daten ist dies undenkbar.

Ob das ein Nachteil ist, kann in diesem Zusammenhang nicht entschieden werden, da normale in Comics eingesetzte Speedlines auch nicht unbedingt perspektivisch korrekt gezeichnet werden. Nachteilig ist aber auf jeden Fall, daß ohne Geometrie nicht alle Methoden der Bewegungsdarstellung umgesetzt werden können.

Nach Abwägung all dieser Fakten ist es recht deutlich, daß der Ansatz, bei dem Geometrie- und Animationsdaten mitgenutzt werden, für die weiteren Arbeiten vielversprechendere Ergebnisse erwarten läßt und somit der anderen Methode vorgezogen wird.

## 3.2 Technische Voraussetzungen und Randbedingungen

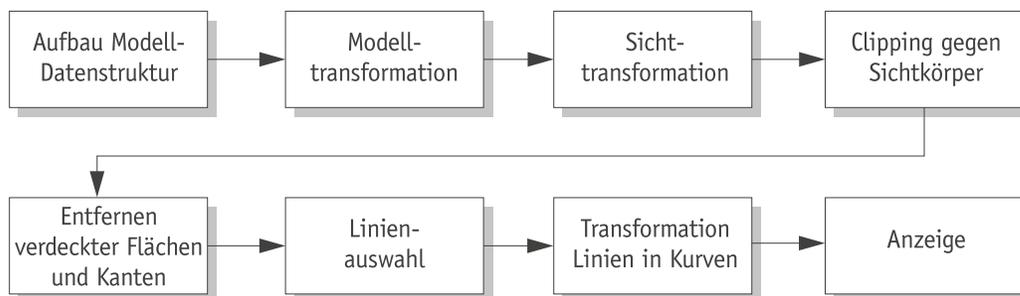
Um Bewegungsdarstellungen in Liniengraphiken algorithmisch umzusetzen, bedarf es einiger Grundvoraussetzungen. Zum einen muß ein Werkzeug vorhanden sein, mit dem aus einem 3D-Modell Liniengraphiken erzeugt werden können und das außerdem flexibel genug für Erweiterungen ist. Weiterhin wird eine spezielle Methode zur Darstellung von Linien benötigt, die es ermöglicht, das Aussehen der erzeugten Linien (Bewegungslinien, Pfeile, usw.) über eine Menge von Parametern zu steuern und an die Liniengraphiken anzupassen. In den folgenden Abschnitten werden diese Punkte detaillierter beschrieben, bevor sich Abschnitt 3.2.3 mit einem theoretischen Ansatz zur Erweiterung eines Liniengraphik-Renderers um die Darstellung von Bewegungen beschäftigt.

### 3.2.1 Konzept eines Renderers für Liniengraphiken

Ein am Institut für Simulation und Graphik der Universität Magdeburg entwickeltes Renderingsystem erfüllt die eben genannten Bedingungen. Es ist in der Lage, 3D-Studio-Modelle mittels Liniengraphiken darzustellen. Es wurde in Smalltalk implementiert und liegt im Quelltext vor. Somit ist dieser Renderer als Basis für die weiteren Arbeiten geeignet.

Im Kapitel 4 von [Str98] beschreiben SCHLECHTWEG und RAAB die Funktionsweise dieses Renderers. Grundlegender Unterschied zu einem photorealistischen Renderer ist, daß der Liniengraphik-Renderer analytisch und somit pixelunabhängig arbeitet. Dadurch werden sowohl andere Methoden für das Entfernen verdeckter Flächen als auch neue Schattierungsverfahren, die auf Linien basieren, erforderlich. Diese Methoden werden im genannten Kapitel ausführlich vorgestellt.

Abbildung 3.1 zeigt die Rendering-Pipeline des Liniengraphik-Renderers. In ihr sind alle Schritte vom Auswerten der Modelldaten bis zum Anzeigen der Liniengraphik dargestellt.



**Abbildung 3.1:** Rendering-Pipeline des Liniengraphik-Renderers

Eine besondere Eigenschaft dieses Ansatzes ist, daß alle Operationen im Objektraum ablaufen und aus diesem Grund viele Zwischenergebnisse, die bei photorealistischen Renderern irrelevant sind, aufgehoben und weiterbenutzt werden können. Dadurch ist es möglich, die Berechnung der Bewegungslinien und Pfeile komplett nach dem eigentlichen Rendering durchzuführen um dann letztendlich das gesamte Resultat auszugeben.

### 3.2.2 Gestaltung der Linien

Das Ergebnis des eigentlichen Renderings ist eine Menge von Linien bzw. Kurven. Durch sie ist die Position einer Linie genau beschrieben. Das Aussehen der Linien ist jedoch noch nicht vollständig bestimmt. Ohne jede Änderung würden die Linien überall gleich dick sein und dadurch extrem künstlich wirken. Die Gestaltung der Linien ist ein zweiter Freiheitsgrad, dessen Nutzung unabdingbar ist, um vertrauter wirkende Linien zu erstellen.

SCHUMANN stellt in [Sch97b] ein Linienmodell vor, das auf den beiden genannten Freiheitsgraden aufsetzt. Hauptanliegen der Arbeit ist die Gestaltung der Linien mittels verschiedener Parameter, die durch die Einführung von Linienstilen zusammengefaßt werden. Grundlage dafür ist die Definition eines Pfades, der durch polynomiale Kurven beschrieben wird und die Position der Linien festlegt. Der Pfad kann durch zusätzliche Informationen, wie etwa Druck oder Sättigung an den Kontrollpunkten und eine entsprechende Interpolationsmethode angereichert werden, und somit ganz gezielt Einfluß auf das Aussehen der Linie nehmen.

Der Linienstil hingegen legt das genaue Aussehen einer Linie fest. Er definiert eine deutlich größere Anzahl an Parametern als nur Helligkeit (Sättigung) und Breite (Druck), die alle gleichberechtigt behandelt werden. Die Anzahl der Stützpunkte kann dabei von der des Pfades differieren. Sowohl die Pfadparameter als auch die des Linienstils beeinflussen also das endgültige Erscheinungsbild. Dabei ist die Stärke des Einflusses der Parameter im Verhältnis zueinander definierbar.

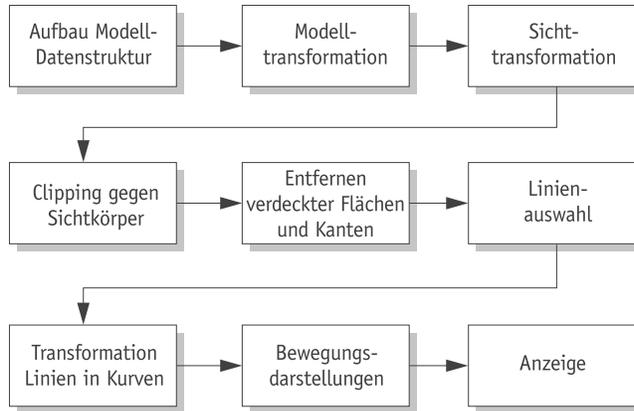
Da auch die Umsetzung dieses Linienmodells in Smalltalk implementiert und die Schnittstelle offen gehalten wurde, erscheint eine Nutzung des Modells für die weiteren Arbeiten ideal.

### 3.2.3 Erweiterung des Renderers um Bewegungsdarstellungen

Um Bewegungen in einem Bild darstellen zu können, muß der Liniengraphik-Renderer verändert werden. Dabei ist es notwendig, die bisherige Abfolge der einzelnen Phasen, die sogenannte Rendering-Pipeline, zu erweitern. Dabei müssen die Bewegungsdarstellungen noch vor der Anzeige in das Bild eingefügt werden. Da aber die

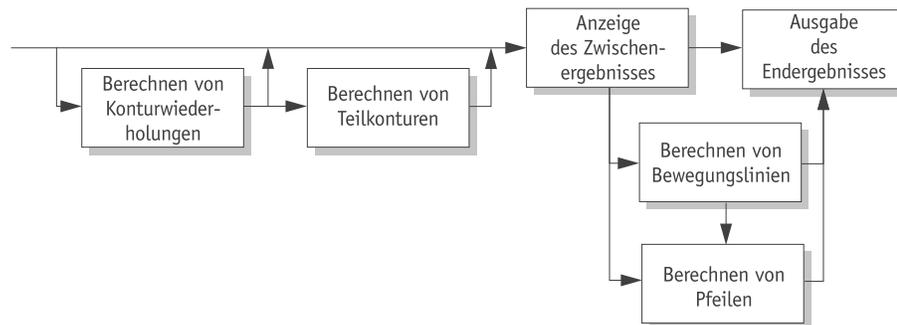
Kenntnis über das Aussehen der Objekte notwendig ist, müssen alle anderen Phasen vorher abgeschlossen sein.

Der Ansatz muß dabei so flexibel sein, daß er jede der geplanten Methoden zur Bewegungsdarstellung als auch Kombinationen daraus erlauben muß und auch weiterhin Bilder ohne jede Darstellung von Bewegungen möglich sind. Abbildung 3.2 zeigt die modifizierte Rendering-Pipeline.



**Abbildung 3.2:** Modifizierte Rendering-Pipeline des Liniengraphik-Renderers zur Realisierung von Bewegungsdarstellungen.

Die unterschiedlichen Darstellungsmethoden erfordern dabei ebenfalls eine feste Reihenfolge. Die Konturwiederholungen müssen noch vor den Bewegungslinien und Pfeilen berechnet werden, da an ihnen später noch Vergleiche zur Bestimmung der Sichtbarkeit vorgenommen werden. Bei der Ausgabe muß darauf geachtet werden, daß weiter zurück liegende Konturen eher gezeichnet werden müssen als aktuelle. Für die Erzeugung von Teilkonturen müssen vorher die Berechnungen zurückliegender Konturen durchgeführt werden. Abbildung 3.3 soll diese Zusammenhänge verdeutlichen.



**Abbildung 3.3:** Die Phase der Berechnungen zur Bewegungsdarstellung bis hin zur Anzeige des Endergebnisses gliedert sich in verschiedene Unterphasen auf.

Der vorgestellte Ansatz erlaubt neben jeder einzelnen Methode zur Bewegungsdarstellung auch Kombinationen und ist somit flexibel genug, um alle Anforderungen zu erfüllen. Die Realisierung der einzelnen Methoden wird im nächsten Kapitel geschildert.

# Realisierung

---

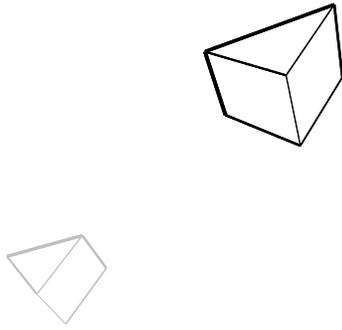
In diesem Kapitel wird beschrieben, wie die verschiedenen Möglichkeiten der Bewegungsdarstellung in Einzelbildern computertechnisch umgesetzt werden und welche Algorithmen dabei zum Einsatz kommen.

## 4.1 Bewegungspfade

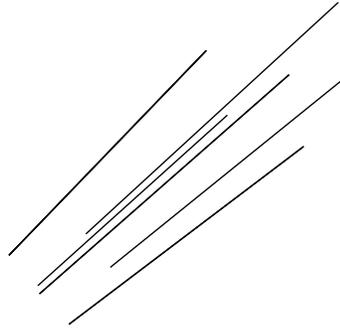
Als Grundlage für die Darstellung der Bewegung von Objekten ist Wissen über das Aussehen der Objekte, sowie über deren Bewegung erforderlich. Während ein Künstler dazu nicht unbedingt eine Vorlage braucht, sondern sich von seiner Kreativität, Erfahrung und Intuition leiten lässt, ist eine genaue Kenntnis dieser Daten für eine rechen-technische Umsetzung unabdingbar. Ein Modell wie es in Rendering- und Animationssystemen (z. B. Autodesk 3D-Studio) benutzt wird, enthält alle nötigen Informationen. Sowohl die Geometriedaten für die einzelnen Objekte als auch Daten über die Bewegung der Objekte sind im Modell vorhanden. Ein im 3D-Studio erzeugtes Modell eignet sich also hervorragend als Ausgangsbasis für die weiteren Arbeiten. Auf diese Weise können geeignete Modelle ohne große Schwierigkeiten erstellt bzw. bereits vorhandene Modelle genutzt werden. Der zu erweiternde Liniengraphik-Renderer ist in der Lage, 3D-Studio-Dateien zu lesen und auszuwerten.

Das 3D-Studio arbeitet mit polygonalen Modellen. Die Geometrie der Objekte wird dabei durch Dreiecksflächen beschrieben, die das Resultat einer Triangulierung der Objekte sind. Die Dreiecke werden durch ihre Eckpunkte (*Vertices*) definiert. Zusätzlich ist für jedes Objekt noch ein Pivotpunkt gegeben, der das Zentrum für alle Rotationsbewegungen und Skalierungen des Objektes darstellt. Der Pivotpunkt selbst wird also von Rotationen und Skalierungen nicht beeinflusst. Die Verfolgung der Position dieses Punktes über die Zeit stellt also die Translation des Objektes an sich dar. Verfolgt man jedoch die Bahnen der Eckpunkte, so werden diese auch durch Rotationen und Skalierungen der Objekte beeinflusst. Die Eckpunkte bieten sich also als Ansatzpunkt für die Darstellung komplexerer Bewegungen an, während der Pivotpunkt ein vielversprechender Ausgangspunkt für die Ermittlung einer Translation ist.

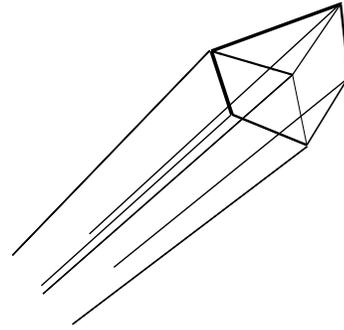
Verfolgt man diese Punkte über die gesamte Animationszeit, so entstehen Kurven, anhand derer die Gesamtbewegung der Punkte zu erkennen ist. Eine solche Kur-



**Abbildung 4.1:** Ausgangs- (hell) und Endposition (dunkel) eines Prismas. Die Bewegung wird von einer Kamera von schräg oben gezeigt.



**Abbildung 4.2:** Bewegungspfade aller Eckpunkte des Prismas.



**Abbildung 4.3:** Zum besseren Verständnis wurde das Prisma selbst an der Endposition des Bewegungspfad es eingefügt.

ve sei *Bewegungspfad* genannt. Dieser wird zur Veranschaulichung in die jeweilige Kameraperspektive projiziert (siehe Abbildungen 4.1, 4.2 und 4.3).

**Def.:** *Die Kurve, die aus der Abbildung der Positionen eines Punktes über die Gesamtzeit der Animation auf den dreidimensionalen Raum entsteht, wird Bewegungspfad dieses Punktes genannt.*

Anhand des Bewegungspfad es eines Punktes können zu jedem Zeitpunkt der Animation Aussagen über zukünftige und bereits verlassene Positionen des entsprechenden Punktes getroffen und Rückschlüsse aus den Animationsdaten gezogen werden:

- Besteht der Bewegungspfad eines Punktes auch nur aus einem Punkt, so bewegt sich der Punkt über die gesamte Animationszeit nicht.
- Sind die Bewegungspfade aller zu einem Objekt gehörender Punkte jeweils Punkte, so bewegt sich das ganze Objekt nicht.
- Ist unter den gesamten Bewegungspfaden aller Punkte eines Objektes genau ein Pfad ein Punkt, so ist der zugehörige Punkt der Pivotpunkt des Objektes und die Bewegung an sich besteht einzig aus Rotationen und Skalierungen.

Weiterhin sind die Aussagen über die vergangenen und zukünftigen Positionen der einzelnen Punkte für die Generierung von Bewegungslinien und Pfeilen nutzbar. Eine Sonderstellung nehmen jeweils die zeitlichen Randbereiche (Anfang bzw. Ende einer Animation) ein. An diesen Stellen können keine Aussagen über weitere zukünftige bzw. vergangene Bewegungen getroffen werden, weil keine entsprechenden Animationsdaten mehr gegeben sind. Dazu mehr in den Abschnitten 4.2 und 4.4.

Die Bewegungspfade an sich können, ähnlich der Animationspfade<sup>1</sup> in verschiedenen Animationssystemen, Aufschluß über die Bewegung der einzelnen Objekte geben. So

<sup>1</sup> Ein Animationspfad stellt für ein animiertes Objekt die Bewegung des Pivotpunktes als Kurve in der aktuellen Szene dar. Mittels Manipulation des Animationspfades kann die Animation geändert werden. Bewegungspfade geben nur die endgültige Bewegung eines Objektes an und haben auf die Animation keinen Einfluß.

kann vor einer längeren Renderingphase kontrolliert werden, ob die Objekte auch wirklich die Bahn verfolgen, die der Benutzer wünschte. Anhand der Kurven können Fehler in der Animation, wie etwa plötzliche Sprünge eines Objektes, erkannt werden. Außerdem läßt sich mittels der Bewegungspfade feststellen, ob eventuelle Richtungsänderungen harmonisch verlaufen. Bewegungspfade sind somit für eine schnelle Evaluierung einer Animation sehr hilfreich. Des weiteren ermöglicht das Gegenüberstellen von mehreren Bewegungspfaden eines Objektes, die jeweils durch unterschiedliche Programme berechnet wurden, eine Aussage darüber, ob die Programme genügend kompatibel zueinander sind (siehe auch [Hop98]), was aber für den weiteren Verlauf dieser Arbeit irrelevant ist.

## 4.2 Bewegungslinien

Mit Hilfe von Pfeilen und Bewegungslinien sollen zukünftige und vergangene Bewegungen visualisiert werden. Aufgrund der traditionellen Anwendungsgebiete bietet es sich an, die Pfeile für zukünftige Ereignisse zu nutzen, und die Bewegungslinien für bereits vergangene. Das bedeutet:

- *Bewegungslinien* stellen die Bewegung eines Punktes oder Objektes in der Vergangenheit dar.
- *Pfeile* stellen die Bewegung eines Punktes oder Objektes in der Zukunft dar.

Objekte, für die Bewegungslinien erstellt werden sollen, müssen bereits alle Transformationen bis zu dem aktuellen Zeitpunkt durchlaufen haben. Der Zeitpunkt darf nicht direkt am Beginn einer Animation gewählt werden, da es in dem Fall keine früheren Animationsdaten gibt, diese aber zwingend erforderlich für die Generierung der Bewegungslinien sind.

Die Erzeugung einer Bewegungslinie besteht aus mehreren Schritten. Zuerst wird ein Vertex<sup>2</sup> des Objektes bestimmt, das eine Bewegungslinie nach sich ziehen soll. Dann wird das Vertex der Reihe nach in die Positionen transformiert, die es in der Zeit einnahm, für die die Bewegungslinie dargestellt werden soll. Anschließend werden diese Positionen in die Kamerakoordinaten des aktuellen Frames projiziert. Die daraus entstandenen Punkte sind die Stützpunkte der Bewegungslinie. Mit Hilfe verschiedener Interpolationsmethoden können die Punkte dann verbunden werden.

Anhand der Bewegungslinien ist später erkennbar, aus welcher Richtung sich das Objekt bewegt hat. Darüber hinaus können aufgrund dieser Linien Spekulationen über die weitere Bewegung des Objektes getroffen werden. Über die genaue Geschwindigkeit der Bewegung lassen sich jedoch keine Aussage machen, aber Relationen bezüglich der Geschwindigkeit zwischen mehreren Bewegungen können erkannt werden. Andere Darstellungsmittel, wie etwa unterschiedliche Liniendicken oder Hellig-

---

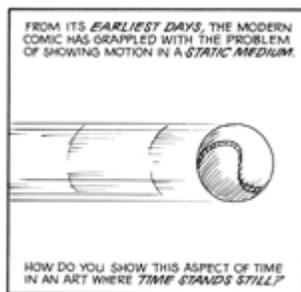
<sup>2</sup> Die Bezeichnungen Vertex und Eckpunkt beschreiben das Gleiche. Im weiteren Verlauf werden beide Bezeichnungen genutzt.

keit, können zusätzlich benutzt werden, um unterschiedliche Geschwindigkeiten des Objektes an sich zu verdeutlichen.

### 4.2.1 Untersuchungen zur Plazierung von Bewegungslinien in traditionell erstellten Bildern

Wie man die Bewegungslinien am günstigsten plaziert, ist nicht durch einen Algorithmus zu bestimmen, sondern vielmehr eine intuitive Entscheidung des Zeichners. Um bei der automatischen Generierung von Bewegungslinien möglichst nicht zu stark von der Darstellung der Speedlines in Comics abzuweichen, sollen zunächst Untersuchungen zur Plazierung von Speedlines in Comics durchgeführt werden.

Bewegungslinien werden von Künstler zu Künstler unterschiedlich angesetzt und haben sich über die Jahre hinweg leicht verändert. Es gibt jedoch Gemeinsamkeiten, die fast alle Zeichnungen aufweisen. Dazu zählt die Tatsache, daß bei Translationen die höchste und tiefste Bewegungslinie relativ zur Bewegungsrichtung immer auch dem höchsten und tiefsten Punkt des Objektes entspricht. Dies zeigt deutlich, daß die Linien auch wirklich zu diesem Objekt gehören. Weiterhin beginnen die Bewegungslinien größtenteils erst nach dem Objekt, sie schneiden es nicht oder nur selten. Dadurch bleibt die Kontur unverändert erhalten und es bleibt eine Trennung zwischen dem eigentlichen Objekt und der Bewegungslinie erhalten. Außerdem werden in vielen Zeichnungen markante Bereiche eines Objektes als Ansatz für Bewegungslinien gewählt. Die Anzahl der Linien darf weder zu groß noch zu klein sein, noch dürfen die Abstände der Bewegungslinien zu gleichmäßig sein. Diese Punkte müssen bei der Bestimmung der Ausgangspunkte beachtet werden. Abbildung 4.4 soll dies anhand der Darstellung eines fliegenden Baseballs verdeutlichen.



**Abbildung 4.4:** Darstellung eines fliegenden Baseballs mit Hilfe von Speedlines. Die oberste/unterste Speedline ist auf Höhe des obersten/untersten Punktes des Baseballs. In: Scott McCloud, Understanding Comics.

Abbildung 4.5 zeigt, wie Bewegungslinien aussehen, wenn wahllos eine unbestimmte Anzahl von Eckpunkten eines Objektes als Ansatzpunkte für die Darstellung der Bewegung selektiert werden. Die Bewegungslinien schneiden das Objekt und sind entweder zu wenig oder zu viele, gemessen an den oben genannten Kriterien. Weiterhin ist zu erkennen, daß es bei Häufungen von Eckpunkten auch zu Häufungen von Bewegungslinien kommt. Dies sollte unbedingt vermieden werden, da Häufungen von Eckpunkten nicht unbedingt auf einen optisch markanten Bereich hindeuten.



**Abbildung 4.5:** Zwei Bilder, bei denen wahllos Eckpunkte des Objektes für die Generierung von Bewegungslinien selektiert wurden. Fast alle Bewegungslinien schneiden das Objekt. An einigen Stellen häufen sich Bewegungslinien.

## 4.2.2 Heuristiken zur Generierung von Bewegungslinien

Auf Basis der Beobachtungen im vorhergehenden Abschnitt werden folgende Heuristiken formuliert:

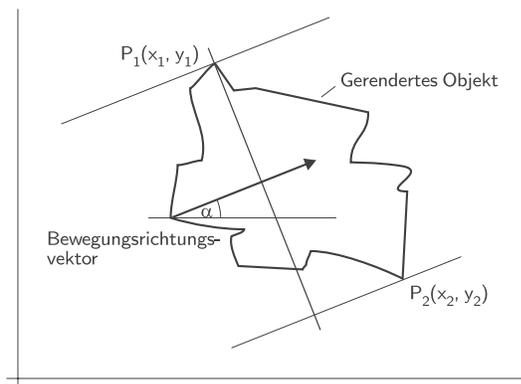
- Die in Relation zur Bewegungsrichtung oberste und unterste Bewegungslinie soll in der gleichen Höhe sein, wie die maximale Ausdehnung des dargestellten Objektes. (H1)
- Bewegungslinien sind möglichst an markanten Bereichen eines Objektes anzusetzen. (H2)
- Die Anzahl der Bewegungslinien muß kontrollierbar sein. (H3)
- Der Abstand der Bewegungslinien untereinander darf nicht zu gleichmäßig sein. (H4)
- Häufungen von Bewegungslinien an bestimmten Bereichen des Objektes müssen vermieden werden. (H5)
- Die Bewegungslinien dürfen erst nach dem Objekt beginnen und es nicht schneiden. (H6)
- Die Länge der Bewegungslinien muß konfigurierbar sein. (H7)

## 4.2.3 Bestimmung der Ausgangspunkte von Bewegungslinien

Da alle Objekte ausschließlich aus Dreiecksflächen zusammengesetzt werden, sind die Extrempunkte der Ausdehnung eines Objektes, egal aus welcher Richtung betrachtet, immer durch einen Eckpunkt definiert. Diese Eckpunkte sind im Resultat des Renderings mit Sicherheit sichtbar. Die Positionsverfolgung dieser Punkte über einen Abschnitt der vergangenen Animationszeit stellt also die maximalen Bewegungslinien dar, die laut aufgestellter Heuristik (H1) im Bild enthalten sein müssen. Die Größe dieses Abschnitt muß durch den Benutzer festgelegt werden. Dadurch wird gleichzeitig indirekt die Länge der Bewegungslinien angegeben und somit (H7) erfüllt.

## Berechnung der Ansatzpunkte für die äußersten Bewegungslinien

Abbildung 4.6 zeigt die Ausgangssituation für die Bestimmung der Ansatzpunkte der äußeren Bewegungslinien (im folgenden auch Extrempunkte genannt). Der Bewegungsrichtungsvektor wird durch die Position des Pivotpunktes des Objektes zu einem vergangenen und dem aktuellen Zeitpunkt bestimmt. Der Pivotpunkt wird durch Rotationen und Skalierungen des Objektes nicht beeinflusst und erscheint deshalb ideal für die Bestimmung des Richtungsvektors. Die Lage des Pivotpunktes zum Objekt ist dabei irrelevant, sie hat keine Auswirkungen auf die Funktionsweise des Algorithmus. Das Objekt, mit dem alle weiteren Handlungen durchgeführt werden, durchlief die Transformationen für den aktuellen Frame und das entsprechende Kamerasichtfeld. Jedem Eckpunkt kann die ursprüngliche Position in Weltkoordinaten zugeordnet werden.



**Abbildung 4.6:** Ausgangssituation für die Bestimmung der Extrempunkte des Objektes relativ zur Bewegungsrichtung

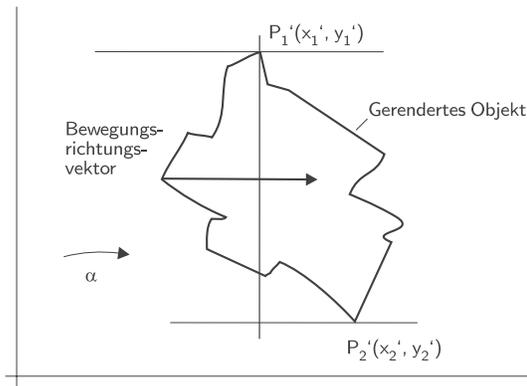
Die gesuchten Punkte haben die Eigenschaft, daß der Abstand der Geraden, die durch den jeweiligen Punkt und die Bewegungsrichtung eindeutig definiert sind, maximal ist. Die Punkte könnten also durch eine Extremwertberechnung, bei der jeder Punkt mit jedem anderen verglichen wird, ermittelt werden. Dies wäre jedoch ein viel zu hoher Aufwand.

Viel effektiver erscheint es, das gerenderte Objekt um den Winkel  $\alpha$  zwischen dem Bewegungsrichtungsvektor und der positiven x-Achse nach rechts zu drehen. Die Geraden durch die Punkte sind jetzt parallel zur x-Achse. Die Extrempunkte lassen sich ohne weiteren Rechenaufwand aus dem y-Wert der Koordinaten der Eckpunkte ablesen. Zur Verdeutlichung ist dies in Abbildung 4.7 dargestellt. Diese Methode zur Bestimmung der Extrempunkte ist deutlich effizienter als die oben angedachte.

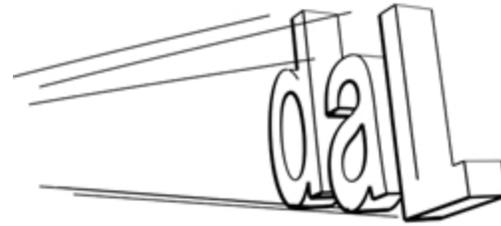
Abbildung 4.8 zeigt an einem Beispiel die Bewegungslinien, die an den ermittelten Extrempunkten angesetzt wurden.

## Berechnung weiterer Ansatzpunkte für Bewegungslinien

Nach der Ermittlung der Ansatzpunkte für die äußersten Bewegungslinien folgt die Berechnung der anderen Ansatzpunkte für die Linien, die sich zwischen den beiden eben gefundenen Eckpunkten befinden. Dazu werden, in etwa gleichverteilt auf den Abstand der Extrempunkte, Punkte an der der Bewegungsrichtung abgewandten Außenlinie des Objektes berechnet.

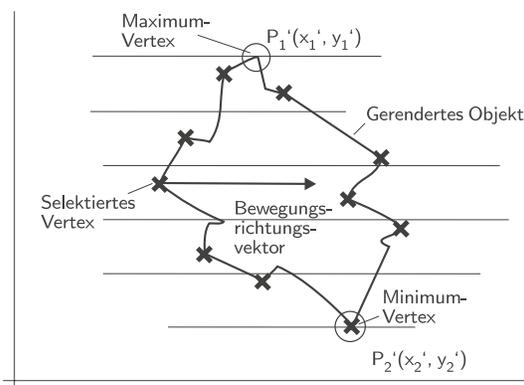


**Abbildung 4.7:** Nach der Rotation des Objektes um die z-Achse um den Winkel  $\alpha$  lassen sich die Extrempunkte einfach ablesen.



**Abbildung 4.8:** Für die Generierung der Bewegungslinien wurden nur die Extrempunkte genutzt. Jedes der 3 Objekte zieht jeweils 2 Bewegungslinien nach sich.

Um die Anzahl der Punkte und somit der Bewegungslinien kontrollieren zu können und zu gewährleisten, daß die Punkte einigermaßen gleichverteilt selektiert werden, um also die Heuristiken (H3) und (H5) umzusetzen, wird das Objekt in verschiedene gleichbreite Streifen eingeteilt. Ausgangssituation ist dabei das Resultat der Rotation von der Ermittlung der Extrempunkte. In jedem der Streifen wird genau der Eckpunkt selektiert, dessen x-Wert je nach Bewegungsrichtung minimal oder maximal ist. Damit sind diese Punkte Extrempunkte im jeweiligen Streifen und sie markieren fast immer einen markanten Teil eines Objektes. Ihre Wahl als Ausgangspunkt für eine Bewegungslinie entspricht also der Heuristik (H2). Gleichzeitig wird durch die Wahl der Extrempunkte sichergestellt, daß bei einer einfachen Bewegung keine Überschneidungen der Bewegungslinien mit dem eigentlichen Objekt auftreten. Somit ist grundsätzlich auch erst einmal die Heuristik (H6) erfüllt. In Abbildung 4.9 ist die Selektion der Punkte dargestellt.



**Abbildung 4.9:** Das Objekt wird in verschiedene gleichbreite Streifen eingeteilt und die Eckpunkte mit den Extremwerten werden selektiert. Die eingekreisten Eckpunkte sind die vorher ermittelten Extrempunkte.

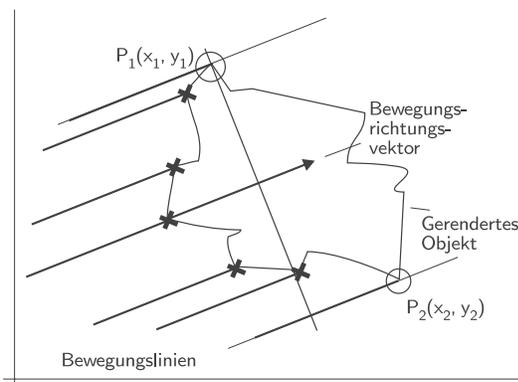
Der Vorteil dieser Methode zur Ermittlung der Ausgangspunkte ist, daß die Punkte nicht zu regelmäßig ausgewählt werden, wie in Heuristik (H4) gefordert. Dadurch, daß immer ein Punkt selektiert wird, der sich irgendwo im Streifen befindet, ist der Abstand zwischen diesen Punkten nicht konstant, die Speedlines wirken nicht zu künstlich und befinden sich immer an markanten Punkten, da der Extremwert

immer auf eine Änderung des weiteren Verlaufs der Linie hinweist. Solche Punkte werden auch bevorzugt von Künstlern gewählt.

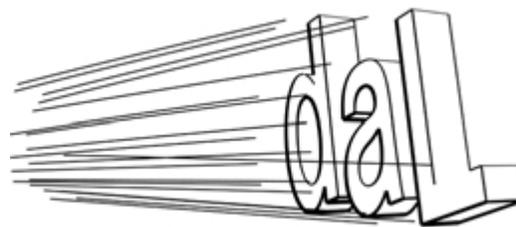
Als Nachteil erweist sich, daß beim vorgestellten Algorithmus davon ausgegangen wird, daß ein Vertex mit extremem x-Wert auch am richtigen Rand liegt. Es kann jedoch auch Situationen geben, in denen sich in dem Streifen nur ein Eckpunkt befindet, der sich nicht am Rand, im schlechtesten Fall aber sogar am gegenüberliegenden Rand des Objektes befinden kann. Dabei sind dann „Ausreißer“ bei den Speedlines zu erkennen. Sollte in einem Streifen kein Eckpunkt liegen, so ist das nicht weiter problematisch, da keine falsche Bewegungslinie erzeugt wird.

Um möglichst keine falschen Bewegungslinien zu erzeugen, kann ein Optimierungsalgorithmus angewandt werden. Dazu werden die Eckpunkte sowohl für die gewünschte, als auch für die der gewünschten Bewegungsrichtung entgegengesetzte Richtung nach oben beschriebener Methode ermittelt. Es müssen nach der Einteilung der Streifen lediglich auch die entgegengesetzten Extrempunkte selektiert werden. Befindet sich ein Vertex sowohl in der Menge der Punkte, die für die gewünschte Bewegungsrichtung ermittelt wurden, als auch in der Menge der Punkte für die entgegengesetzte Richtung, so kann ohne genaue Untersuchung nicht festgelegt werden, in welcher der beiden Mengen sich der Punkt zu Recht befinden würde. Aus diesem Grund wird der Eckpunkt sicherheitshalber aus beiden Mengen entfernt.

Von den ermittelten Maximalpunkten der gewünschten Bewegungsrichtung werden jetzt Rückschlüsse auf die Originalkoordinaten vorgenommen. Diese werden dann in die Positionen für die betreffenden bereits vergangenen Frames transformiert und anschließend in Kamerakoordinaten und Anzeigekoordinaten projiziert. Die entstandenen Punkte werden nach verschiedenen Interpolationsmethoden miteinander verbunden, im einfachsten Fall durch Linien. Das Ergebnis kann in Abbildung 4.10 begutachtet werden. Abbildung 4.11 zeigt das Ergebnis des Algorithmus am bereits vorher aufgeführten Beispiel.



**Abbildung 4.10:** Nach der Selektion der Eckpunkte werden diese zu ihrer Position in früheren Frames transformiert und entsprechend des aktuellen Frames projiziert und jede frühere Position eines Eckpunktes mit seinem Nachfolger verbunden.



**Abbildung 4.11:** Bei der zusätzlichen Selektion von 5 weiteren Eckpunkten zu den Extrempunkten entstehen schon fast zu viele Bewegungslinien. Die Bewegung an sich wirkt aber auf den Betrachter, als wäre sie schneller.

Schließlich soll der Ablauf des Algorithmus zur Selektion der Ausgangspunkte noch einmal als Pseudocode zusammengefaßt werden:

```

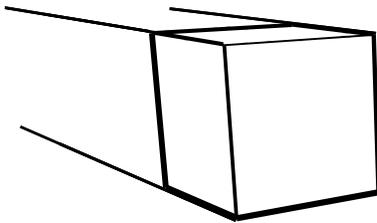
PROZEDUR Berechne Ansatzpunkte

Bewegungsvektor := Endosition Pivotpunkt - Startposition Pivotpunkt.
Alpha := BERECHNE Winkel zwischen x-Achse und Bewegungsvektor.
Rotierte Eckpunkte := ROTIERE sichtbare projizierte Eckpunkte UM DIE Z-ACHSE
    UM Alpha.
Sortierte Eckpunkte := SORTIERE Rotierte Eckpunkte NACH Y-WERT.
TEILE Sortierte Eckpunkte IN GLEICHBREITE STREIFEN AUF.
FÜR jeden Streifen FÜHRE AUS: {
    FÜGE Punkt mit minimalem X-Wert ZUM Linken Rand HINZU.
    FÜGE Punkt mit maximalem X-Wert ZUM Rechten Rand HINZU.
}
SELEKTIERE Linken/Rechten Rand IN ABHÄNGIGKEIT VOM Bewegungsvektor.

```

#### 4.2.4 Interpolation fehlender Eckpunkte

Wie oben bereits erwähnt, kann es vorkommen, daß in einem Streifen kein Eckpunkt liegt und folglich auch keine Bewegungslinie gezeichnet werden kann. Obwohl dies nicht unbedingt fehlerhaft wirkt, gibt es Fälle, speziell bei Objekten mit wenig Eckpunkten, in denen gewünscht wird, daß zusätzliche Eckpunkte erzeugt werden. Abbildung 4.12 zeigt dies am Beispiel eines Quaders.

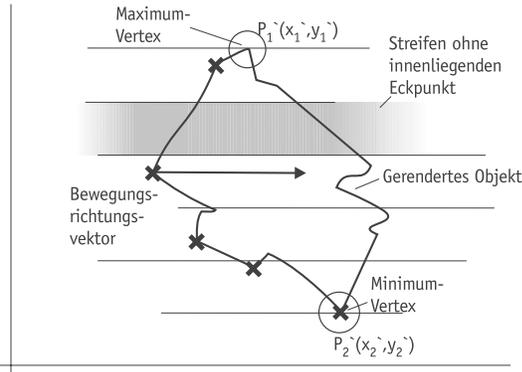


**Abbildung 4.12:** Darstellung der Bewegung eines Objektes mit nur wenigen Eckpunkten. Obwohl die Erstellung von 10 zusätzlichen Bewegungslinien vorgegeben wurde, konnten diese aufgrund wenigen Ansatzpunkte nicht berechnet werden.

Die Einteilung eines Objektes in verschiedene gleichbreite Streifen erwies sich als gute Methode zur sinnvollen Selektion von Vertices. Sowohl die Anzahl als auch die Anordnung läßt sich mit dieser Methode gut beeinflussen. Die neu zu erzeugenden Eckpunkte des Objektes sollten deshalb den gleichen Anforderungen standhalten. Dazu bietet es sich an, die Interpolation auch mit Hilfe der Einteilung des Objektes in Streifen umzusetzen.

Ausgangsbasis für die Interpolation neuer Punkte ist die Endsituation nach der Selektion der Eckpunkte, wie auch in Abbildung 4.13 dargestellt. Jeder Streifen soll genau ein Vertex enthalten. Für den ersten und den letzten Streifen trifft dies zu, da sie auch die jeweils maximalen Eckpunkte beinhalten.

Die verschiedenen Streifen werden der Reihenfolge nach auf das Vorhandensein eines Eckpunktes untersucht. Ist ein Punkt vorhanden, muß kein Punkt interpoliert werden und der nächste Streifen kann untersucht werden.



**Abbildung 4.13:** Ausgangsbasis für die Interpolation der Eckpunkte ist die Situation nach der Selektion der Vertices. Für jeden Streifen, der keinen Eckpunkt enthält, soll ein Punkt erzeugt werden.

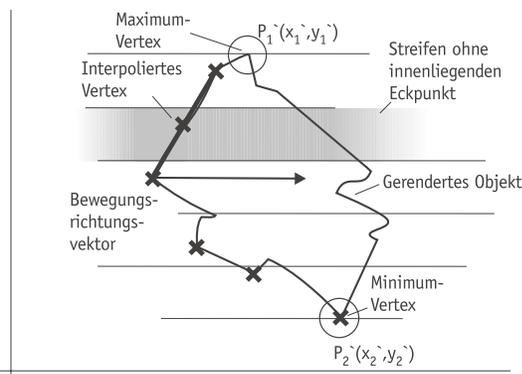
Sobald ein Streifen keinen Punkt enthält, werden die Originalkoordinaten des Punktes des vorhergehenden Streifens zur Interpolation eines neuen Randpunktes herangezogen. Dieser Punkt sei  $P_k$  und der gesuchte  $P_{k+1}$  genannt. Ein zweiter Punkt ist ebenfalls vonnöten. Dazu werden die nachfolgenden Streifen auf das Vorhandensein von Vertices untersucht. Sobald der nächste Eckpunkt  $P_{k+n}$  gefunden wurde, wird auch dieser, in seinen Originalkoordinaten, zur Interpolation eines neuen Punktes benutzt.  $n$  ist also die Anzahl der Übergänge zwischen den Streifen vom letzten vorhandenen bis zum ersten wieder vorhandenem Eckpunkt in einem Streifen. Mit Hilfe der gefundenen Punkte ist die Interpolation mittels folgender Formel eindeutig.

$$P_{k+1} = P_k + \frac{P_{k+n} - P_k}{n} \quad (4.1)$$

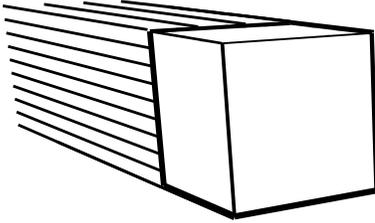
Sollten mehr als nur ein Punkt fehlen, können diese über Rekursion ebenfalls über die genannte Formel berechnet werden oder aber sofort nach Formel 4.2.

$$P_{k+x} = P_k + (P_{k+n} - P_k) \frac{x}{n} \quad x \in N, \quad 1 \leq x < n \quad (4.2)$$

Abbildung 4.14 zeigt die Interpolation des Punktes in den 2D-Koordinaten des aktuellen Frames. Da die geometrischen Verhältnisse des Objektes über die Animation erhalten bleiben, wird die eigentliche Interpolation jedoch in den Ausgangskordinaten vorgenommen und der erhaltene Punkt fortan als Punkt des Objektes behandelt. Dies ist notwendig, damit er ebenfalls in die Positionen der anderen Frames projiziert werden kann und dort keine weiteren Interpolationen notwendig sind.



**Abbildung 4.14:** Veranschaulichung des Interpolationsalgorithmus an einem einfachen Beispiel. Der neue Punkt liegt genau auf der Mitte des Vektors zwischen den beiden bekannten Eckpunkten.



**Abbildung 4.15:** Trotz im Modell nicht vorhandener Eckpunkte können zusätzliche Bewegungslinien gezeichnet werden. Die neuen Punkte wurden durch Interpolation erzeugt.

In Abbildung 4.15 ist das Ergebnis der Interpolation am bereits vorgestellten Beispiel ersichtlich. Dieses Beispiel ist ein Extremfall, denn nur sehr selten sind wirklich so wenige Eckpunkte in einem Objekt. Bei komplizierteren Beispielen wirken die Bewegungslinien an den interpolierten Punkten nicht so regelmäßig, sie passen sich ebenfalls der Objektform an.

Die Umsetzung in Pseudocode soll den Algorithmus zusammenfassen. Eine leichte Modifikation der Berechnung der Ansatzpunkte ist notwendig.

```
PROZEDUR Berechne Ansatzpunkte
```

```
Bewegungsvektor := Endosition Pivotpunkt - Startposition Pivotpunkt.
```

```
Alpha := BERECHNE Winkel zwischen x-Achse und Bewegungsvektor.
```

```
Rotierte Eckpunkte := ROTIERE sichtbare Projizierte Eckpunkte UM DIE Z-ACHSE  
UM Alpha.
```

```
Sortierte Eckpunkte := SORTIERE Rotierte Eckpunkte NACH Y-WERT.
```

```
TEILE Sortierte Eckpunkte IN GLEICHBREITE STREIFEN AUF.
```

```
FÜR jeden Streifen FÜHRE AUS: {
```

```
    FÜGE Punkt mit minimalem X-Wert ODER nil ZUM Linken Rand HINZU.
```

```
    FÜGE Punkt mit maximalem X-Wert ODER nil ZUM Rechten Rand HINZU.
```

```
}
```

```
SELEKTIERE Linken/Rechten Rand IN ABHÄNGIGKEIT VOM Bewegungsvektor.
```

```
PROZEDUR Interpoliere fehlende Punkte in den Streifen
```

```
InterpolierterRand := Interpoliere SelektierterRand (POSITION: 1 MIT LÜCKEN: 0)
```

```
FUNKTION Interpoliere SelektierterRand (POSITION: aktuellePosition  
MIT LÜCKEN: Lückenanzahl)
```

```
SelektierterRand[aktuellePosition] == nil ?
```

```
WENN WAHR FÜHRE AUS {
```

```
    SelektierterRand[aktuellePosition+1] == nil ?
```

```
    WENN WAHR FÜHRE AUS {
```

```
        SelektierterRand := Interpoliere SelektierterRand (POSITION:  
        aktuellePosition+1 MIT LÜCKEN: Lückenanzahl+1).
```

```
        StartPunkt := SelektierterRand[aktuellePosition-Lückenanzahl-1].
```

```
        EndPunkt := SelektierterRand[aktuellePosition+1].
```

```
        DifferenzVektor := ((EndPunkt-StartPunkt)/(Lückenanzahl+2))*  
        (Lückenanzahl+1).
```

```

    SelektierterRand[aktuellePosition+1] := StartPunkt+DifferenzVektor.
  }
  WENN FALSCH FÜHRE AUS {
    StartPunkt := SelektierterRand[aktuellePosition-Lückenanzahl-1].
    EndPunkt := SelektierterRand[aktuellePosition+1].
    DifferenzVektor := ((EndPunkt-StartPunkt)/(Lückenanzahl+2))*
      (Lückenanzahl+1).
    SelektierterRand[aktuellePosition+1] := StartPunkt+DifferenzVektor.
  }
}
(aktuellePosition+1) < Anzahl der Streifen ?
WENN WAHR FÜHRE AUS {
  SelektierterRand := Interpoliere SelektierterRand (POSITION:
    aktuellePosition+1 MIT LÜCKEN: 0).
}
GIB SelektierterRand ZURÜCK.

```

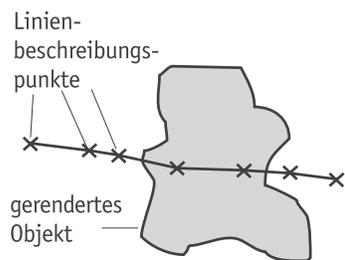
### 4.2.5 Umgehung von Überdeckungen

Manchmal kann es vorkommen, daß eine Bewegungslinie zwar am Ansatzpunkt sichtbar ist, es danach aber zu Überschneidungen mit dem Objekt kommt. Der im Folgenden beschriebene Algorithmus soll bewirken, daß die Bewegungslinien so in das Bild eingefügt werden, daß die Teile der Bewegungslinien, die nicht sichtbar wären, nicht gezeichnet werden und somit die Erfüllung der Heuristik (H6) garantiert wird (siehe Abbildung 4.16).

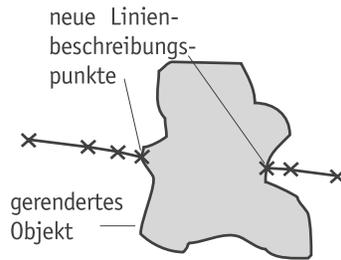
Als Lösung würde sich ein Standard-Hidden-Line-Removal Ansatz anbieten. Dieser errechnet für jede Linie Schnittpunkte mit dem Objekt und gibt die sichtbaren Teillinien zurück. Auf diese Weise würden die Linien jeweils bis an den Schnittpunkt heran sichtbar sein (siehe Abbildung 4.17). Das kann aber bei Liniengraphiken einen ungewollten Effekt haben. Die Teillinie endet genau in einer zum Objekt gehörigen Linie. In technischen Illustrationen oder auch in Comics würde die Linie aber kurz vorher aufhören. Um dies zu erreichen, gibt es zwei Möglichkeiten:

- Zum einen könnten die Endpunkte aller Teillinien verkürzt werden, was zusätzlichen Rechenaufwand und einen weiteren graphischen Algorithmus erfordern würde,
- zum anderen läßt sich dieser Effekt aber auch durch einen vereinfachten Sichtbarkeitsalgorithmus lösen, bei dem nur die die Linie beschreibenden Punkte untersucht werden. Ist ein Punkt verdeckt, wird die Linie in 2 Teillinien aufgeteilt, wobei es keine Verbindung mehr zu dem Verdeckungspunkt gibt (siehe Abbildung 4.18).

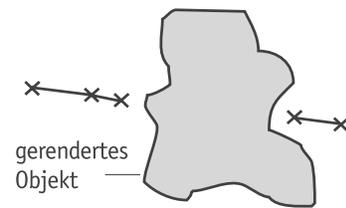
In dieser Arbeit wird die zweite Methode benutzt. Als Ausgangsbasis für den Algorithmus werden Zwischenergebnisse des Renderers benutzt, die nach dem eigentlichen Rendering noch vorhanden sind. Eines dieser Zwischenergebnisse enthält für



**Abbildung 4.16:** Ausgangssituation für den Sichtbarkeitsalgorithmus. Der graue Bereich soll nicht durch eine Linie überdeckt werden.



**Abbildung 4.17:** Ein normaler Hidden-Line-Removal Algorithmus schneidet die Linien so, daß sie direkt am Objekt aufhören.



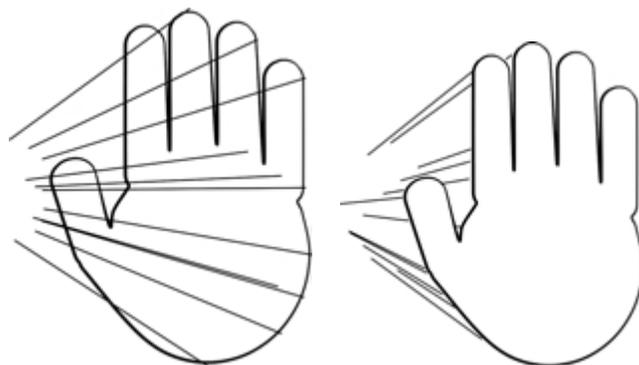
**Abbildung 4.18:** Der vereinfachte Algorithmus bestimmt die Sichtbarkeit der einzelnen Stützpunkte. Verdeckte Stützpunkte werden komplett entfernt.

jedes Objekt einer Szene die sichtbaren Linien und die sichtbaren Flächen.

Jeder Punkt der Bewegungslinien wird zuerst mit dem umschließenden Rechteck (*Bounding Box*) eines Objektes verglichen. Liegt der Punkt innerhalb der Bounding Box, so kann es eine Überschneidung geben und er wird mit jeder sichtbaren Fläche verglichen. Für den Fall, daß der Punkt in keiner der sichtbaren Flächen liegt, kann er gezeichnet werden, anderenfalls muß eine Sichtbarkeitsbestimmung anhand der Tiefeninformationen vorgenommen werden. Sobald der Punkt hinter einer Fläche liegt, wird er nicht gezeichnet und der Vergleich abgebrochen. Liegt der z-Wert des Punktes vor einer Fläche, muß weiter verglichen werden. Sobald alle Vergleiche positiv abgeschlossen sind, kann der Punkt gezeichnet werden. Ist auch nur ein Vergleich negativ, wird der Punkt nicht dargestellt.

Nachdem so für jeden Kontrollpunkt der Bewegungslinie die Sichtbarkeit entschieden wurde, werden die einzelnen sichtbaren Punkte miteinander verbunden. Dabei wird vorausgesetzt, daß mindestens zwei aufeinanderfolgende Punkte sichtbar sind. Ansonsten wird der Punkt nicht gezeichnet. Anhand der Abbildungen 4.19 und 4.20 ist das Resultat dieser Methode erkennbar.

Vorteil dieser Methode ist, daß die Bewegungslinien nicht sofort wieder hinter dem Objekt beginnen, sondern etwas Platz zwischen Objekt und der Fortsetzung der Bewegungslinie ist. Das wirkt deutlich natürlicher und Heuristik (H6) wird erfüllt.



**Abbildung 4.19:** Beispiel einer Handbewegung aus dem Hintergrund des Bildes auf die Kamera zu. Die verdeckten Linien wurden nicht entfernt. Es ist nicht unbedingt ersichtlich, daß die Hand aus dem Hintergrund kommt.

**Abbildung 4.20:** Nach der Entfernung der verdeckten Linien wirkt das Bild viel natürlicher. Der Betrachter weiß jetzt, daß die Hand aus dem Hintergrund des Bildes kommt.

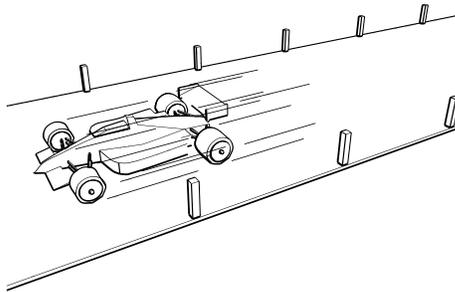
Doch gerade dieser Abstand kann auch ein Nachteil sein. Bei sehr schnellen Bewegungen kann er sehr groß werden und es besteht die Gefahr, daß die Bewegungslinie nicht mehr dem eigentlichen Objekt zugeordnet werden kann. In diesem Fall kann zwischen dem vorhergehenden nicht-sichtbaren Punkt und dem sichtbaren ein neuer, sichtbarer Punkt interpoliert werden.

Eine Einschränkung des Algorithmus ist, daß Objekte, die so klein sind, daß sie zwischen zwei Punkten liegen, vom Algorithmus nicht beachtet werden und sie von einer Linie übermalt werden.

### 4.2.6 Einfluß verschiedener Linienstile

Nach der Sichtbarkeitsbestimmung steht die Position der Linien jetzt endgültig fest. Mittels der in Kapitel 3.2.2 vorgestellten Linienstile kann nun das Aussehen der Bewegungslinien noch verändert werden, um bestimmte Wirkungen zu erreichen.

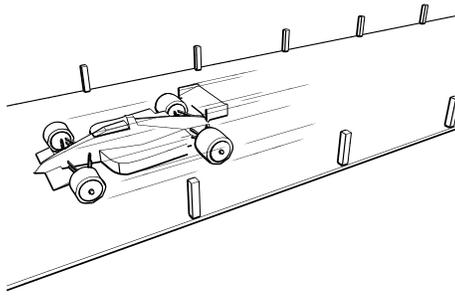
Abbildung 4.21 zeigt die Bewegungslinien ohne Beeinflussung durch einen Linienstil. Die Linien wirken alle gleich auf den Betrachter, die Bewegungslinien heben sich vom eigentlichen Objekt kaum ab. Durch die Linien ist das Vorhandensein einer Bewegung und die Bewegungsrichtung des Objektes deutlich erkennbar. Die Bewegungslinien machen auf den Betrachter trotzdem einen sehr künstlichen Eindruck, da sie immer konstant dick sind und an den Enden plötzlich aufhören.



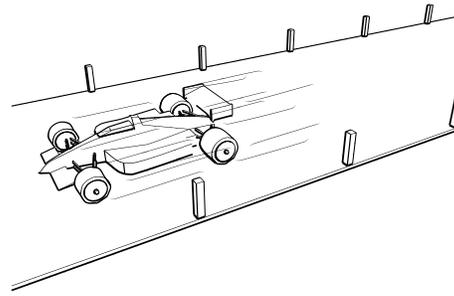
**Abbildung 4.21:** Darstellung von Bewegungslinien an einem fahrenden Rennwagen. Die Bewegungslinien wurden nicht durch einen Linienstil beeinflusst.

Um die Bewegungslinien ästhetischer wirken zu lassen, wurde ein Linienstil definiert, der die Linie am Anfang und am Ende jeweils auslaufen läßt. Die Linie wurde insgesamt dünner gezeichnet und hebt sich so besser vom Objekt ab, läßt das Objekt jedoch im Vordergrund. Das Ergebnis ist in Abbildung 4.22 sichtbar. In Abbildung 4.23 wurden die Kontrollpunkte der Linie zusätzlich noch zufällig verschoben. Das Ergebnis ist eine sichtbare Abweichung der Linien von der Originalposition, die die Linien insgesamt ungleichmäßiger wirken läßt. Mit dieser Kontrollpunktvariation muß aber vorsichtig umgegangen werden, da eine zu große Abweichung den Sinn der Bewegungslinien in Frage stellt.

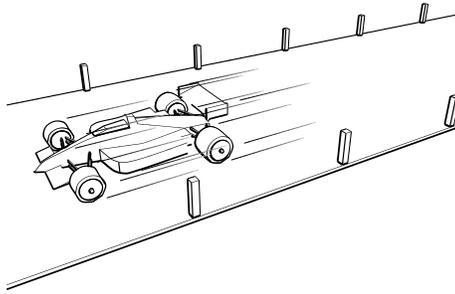
Um die beschränkte zeitliche Existenz der Linien besser zum Ausdruck zu bringen, können diese mittels eines Linienstils außerdem insgesamt in die Vergangenheit verjüngt oder verblaßt werden. Vom Verblässen der Linien ist aber eher abzuraten,



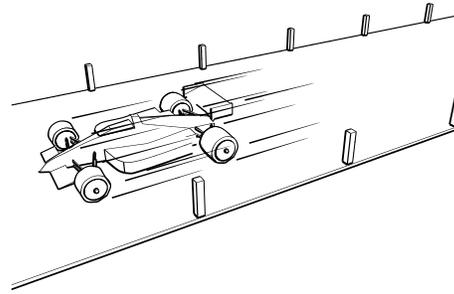
**Abbildung 4.22:** Anwendung eines anderen Linienstils auf die gleichen Bewegungslinien. Die Linien laufen am Ende aus und sind insgesamt dünner als im Ausgangsbild.



**Abbildung 4.23:** Zusätzlich zum Linienstil aus der links stehenden Abbildung wurden die Kontrollpunkte der Linien verschoben. Das Bild wirkt unruhiger.



**Abbildung 4.24:** Das Verblässen von Bewegungslinien in die Vergangenheit verstärkt den Eindruck der Bewegungslinien.



**Abbildung 4.25:** Verjüngungen in die Vergangenheit wirken ebenso auf den Betrachter wie Verblässungen.

wenn man Wert auf die Kopierbarkeit des Bildes legt. Farbverläufe wirken in Kopien immer störend, da sie nicht mehr als Farbverlauf erkennbar sind. In diesem Fall empfiehlt sich das Einsetzen von Verjüngungen. Durch die Erhaltung des Kontrastes sind keine Schwierigkeiten zu erwarten. Abbildungen 4.24 und 4.25 zeigen Beispiele für die angeführten Methoden.

All die oben angeführten Methoden orientieren sich an anerkannten und von Graphikern oder Künstlern benutzten Methoden. Doch darüber hinaus gibt es noch weitere Möglichkeiten der Anwendung der Linienstile, die bis jetzt noch nicht eingesetzt wurden.

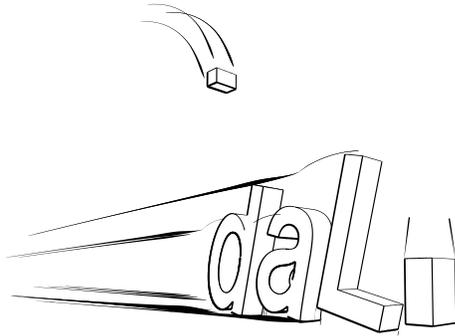
Eine solche stellt die Abbildung der relativen Geschwindigkeit auf die Liniendicke der Bewegungslinie dar. Dadurch läßt sich im Bild erkennen, wie schnell die Bewegung an bestimmten Positionen relativ zur Gesamtbewegung war. Die Stützpunkte der Bewegungslinien liegen zeitlich exakt gleich voneinander entfernt. Je weiter zwei aufeinanderfolgende Punkte auseinander sind, um so schneller hat sich das Objekt zwischen den beiden Punkten bewegt. Entsprechend stärker soll die Linie gezeichnet werden. Die Liniendicke  $D$  für eine Teillinie  $L_k$  der Bewegungslinie ergibt sich aus der Multiplikation einer Basisdicke  $B$  mit einem Modifikator  $M_k$  und der Modifikation durch den eigentlichen Linienstil  $S$ .

$$D(L_k) = B \cdot M_k \cdot S(L_k) \quad (4.3)$$

Der Modifikator  $M_k$  berechnet sich dabei nach der folgenden Formel:

$$M_k = \left( \frac{\overline{P_k P_{k+1}}}{\frac{\sum_{i=1}^{n-1} \overline{P_i P_{i+1}}}{n-1}} \right)^2 \quad (4.4)$$

Die Quadrierung des Modifikators dient der deutlicheren Unterscheidbarkeit da die Verdickung des Pfades nur zu 50% in die Berechnung der Gesamtdicke eingeht. Abbildung 4.26 zeigt das Ergebnis dieser Beeinflussung am Beispiel eines Bildes der *da!!*-Animation.



**Abbildung 4.26:** Einzelbild aus der *da!!*-Animation. Die Liniendicke lässt Rückschlüsse auf die Geschwindigkeit der einzelnen Objekte an den jeweiligen Positionen zu (übertriebene Darstellung).

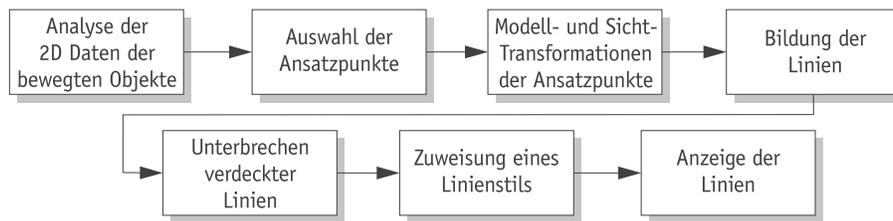
Durch die Abbildung der Geschwindigkeit der Objekte auf die Liniendicke der Bewegungslinien lassen sich jetzt auch positive und negative Beschleunigungen erkennen. Dies war bis jetzt lediglich durch unterschiedliche Abstände der Konturen bei Konturwiederholungen möglich.

Eine Abbildung der Geschwindigkeit der Objekte auf die Helligkeit der Bewegungslinie erscheint nicht sinnvoll, da der Farbbereich eingegrenzt ist. Damit würden bei einer gleichförmigen Bewegung alle Linien grau gezeichnet werden und der Grundkontrast der Bewegungslinien darunter leiden.

### 4.2.7 Zusammenfassung des Algorithmus zur Berechnung von Bewegungslinien

Abschließend soll der gesamte Prozeß der Generierung der Bewegungslinien zusammengefaßt werden. Die grundsätzliche Idee des vorgestellten Algorithmus ist, über eine Analyse der 2D-Koordinaten eines bewegten Objektes Vertices dieses Objektes zu selektieren. Durch einen Rückschluß auf die originalen 3D-Koordinaten ist es möglich, die Positionen der Vertices in jedem vorhergehenden Frame durch entsprechende Transformationen zu ermitteln und so perspektivische Darstellungen zu erzeugen. Wiederum in den jeweiligen 2D-Koordinaten wird ermittelt, ob es zu Überschneidungen mit anderen Objekten kommt. In diesem Fall werden die jeweils korrespondierenden 3D-Koordinaten zur Klärung der Sichtbarkeit zuhulfe genommen, da sie die Tiefeninformationen enthalten. Nach dem Entfernen aller verdeckten Eckpunkte der Bewegungslinien und der entsprechenden Aufspaltung der Linien wird

jeder Linie ein Linienstil zugewiesen, der das Aussehen modifiziert. Das Ergebnis wird anschließend in das 2D-Bild eingefügt. Abbildung 4.27 stellt den groben Ablauf graphisch dar.



**Abbildung 4.27:** Übersicht über die einzelnen Schritte zur Darstellung einer Bewegungslinie.

Eine genauere Übersicht bietet die Darstellung des Algorithmus als Pseudocode:

```

PROZEDUR Berechne Bewegungslinien
FÜR jedes bewegte Objekt FÜHRE AUS: {
  BERECHNE Ansatzpunkte.
  INTERPOLIERE fehlende Punkte in den Streifen.
  FÜGE Rotierte Eckpunkte mit minimalem und maximalem Y-Wert ZUM selektierten
  Rand HINZU.
  FÜR alle Punkte im Rand FÜHRE AUS: {
    FÜR alle Frames zwischen Anfang und Ende der Bewegungslinie FÜHRE AUS: {
      FÜGE projizierten Eckpunkt ZUM Pfad HINZU.
    }
    ENTFERNE verdeckte Eckpunkte aus dem Pfad.
    ZEICHNE Pfad MIT selektiertem Linienstil.
  }
}
  
```

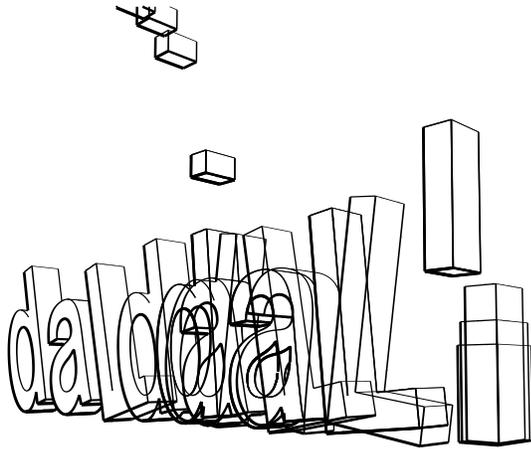
## 4.3 Konturwiederholungen

Eine weitere Möglichkeit, Bewegungen von Objekten in einem Bild darzustellen, ist die Wiederholung der Kontur. Auf diese Weise sind selbst kompliziertere Bewegungen vom Betrachter einfach zu erkennen. Konturwiederholungen haben gegenüber den Bewegungslinien aber noch weitere Vorteile. So kann man anhand von Konturwiederholungen Geschwindigkeitsänderungen erkennen. Dies ist möglich, weil eine Eigenschaft dieser Wiederholungen ist, daß die Konturen immer in zeitlich gleichen Abständen wiederholt werden. Liegen also plötzlich zwei Wiederholungen weiter auseinander, als andere, so war die Geschwindigkeit des Objektes zwischen diesen Wiederholungen höher.

Um die Objektkonturen in der Vergangenheit wirklich korrekt darzustellen, müssen die bewegten Objekte für jede vergangene Darstellung neu gerendert werden. Im Vergleich zu den Bewegungslinien ist der Aufwand für die Berechnung der Konturwiederholungen somit deutlich höher. Die Ergebnisse der verschiedenen Renderings

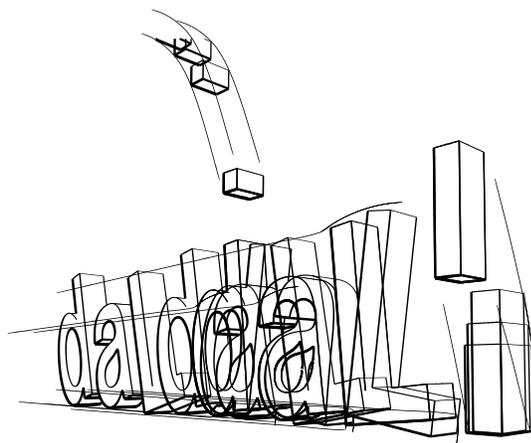
müssen anschließend in einem Bild vereint werden. Um Probleme falscher Überdeckungen zu umgehen, wird ein temporaler Painter's-Algorithmus angewandt, bei dem zuerst die zeitlich vorhergehenden Teile des Bildes gezeichnet werden. Dadurch befindet sich der aktuelle Teil des Bildes auch wirklich im Vordergrund.

Abbildung 4.28 zeigt ein Beispiel für die Wiederholung der Komplettkontur und weist auch gleich deutlich auf die auftretenden Probleme hin. Zwar kann der Betrachter eventuell eine Bewegung erkennen, doch er weiß nicht, welche der möglichen Richtungen die richtige ist. Dadurch, daß alle Konturwiederholungen im gleichen Linienstil gezeichnet wurden, wirkt das Bild unübersichtlich und es kann nicht erkannt werden, welche der Konturen die zeitlich vorhergehenden sind.



**Abbildung 4.28:** Dreifache Wiederholung der Komplettkontur im Abstand von jeweils 4 Frames.

Um zumindest die vergangenen Konturen erkennen zu können, wäre eine Kombination aus Konturwiederholungen und Bewegungslinien möglich, bei der die Bewegungslinien noch etwas weiter in die Vergangenheit gezeichnet werden, als die Konturen. Damit ist die aktuelle Kontur die einzige, die nur in eine Richtung Bewegungslinien nachzieht und somit als aktuelle Kontur erkennbar. Anhand der Abbildung 4.29 kann die Anwendung dieser Technik betrachtet werden.



**Abbildung 4.29:** Dreifache Wiederholung der Komplettkontur im Abstand von jeweils 4 Frames und zusätzlich Bewegungslinien über 20 Frames.

Trotzdem ist auch diese Darstellung für einen Großteil der Fälle nicht wirklich geeignet. Das Bild wirkt, jetzt noch mehr als vorher, unübersichtlich und überladen,

da der Betrachter nicht richtig zwischen den normalen Informationen des Bildes und den zusätzlichen Informationen über die Bewegung differenzieren kann. Vergangene Teile des Bildes müssen deutlicher als solche erkennbar sein, dürfen aber vom eigentlichen Bild nicht zu sehr ablenken.

Als Lösung für dieses Problem sind mehrere Methoden denkbar. Die naheliegendste von ihnen ist die Anwendung eines Hidden-Line-Removal-Algorithmus. Dabei würde aber unter Umständen die Kontur des Objektes völlig verloren gehen und eine Zuordnung der verbliebenen Fragmente zum Originalobjekt wäre nicht möglich. Die Erhaltung der Kontur oder zumindest eines markanten Teils der Kontur ist eine zwingende Voraussetzung, die durch die beiden im folgenden angeführten Methoden erfüllt wird. Die erste von ihnen versucht die Linien soweit zu deakzentuieren, daß sie nicht mehr als störend empfunden werden und die zweite erzeugt nur Teile der Kontur. Dadurch wird die Anzahl der Linien reduziert, es ist aber immer noch eine Zuordnung zum Originalobjekt möglich.

### 4.3.1 Abgeschwächte Konturen

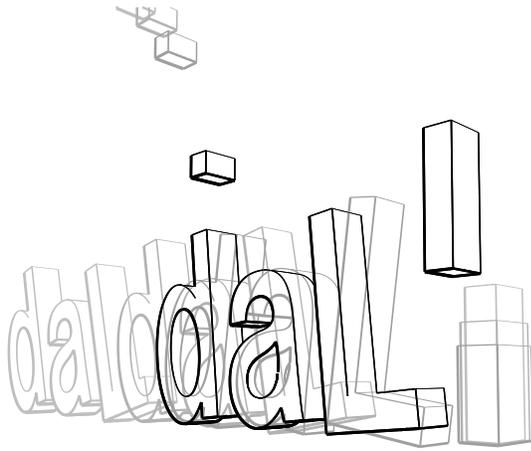
Zwei Möglichkeiten bieten sich zur Lösung des Problems an. Zum einen kann die Dicke der Linien verändert werden, zum anderen deren Helligkeit. Die Variation der Liniendicke erscheint eher ungünstig, da auf diese Weise keine genügende Unterscheidung der Zusatzinformation von den anderen dargestellten Objekten möglich ist. Viel mehr könnte die Konturwiederholung wie ein deakzentuiertes Objekt des Bildes erscheinen, da der Betrachter ja nicht wissen kann, welche Objekte sich in der Szene befinden und welche nicht. Außerdem ist der Kontrast bei solchen Wiederholungen immer noch zu hoch.

Im Gegensatz dazu ist ein Objekt, das in einer anderen Helligkeit dargestellt wird, vom Betrachter leichter als „nicht zum eigentlichen Bild gehörig“ einzuordnen. Es erscheint außerdem leichter, mehrere Grautöne voneinander zu unterscheiden, als minimal variierte Liniendicken. Der Kontrast der Wiederholungen ist deutlich geringer als der des Originalobjektes und das Bild wirkt somit übersichtlicher.

Nach mehreren Untersuchungen zur günstigen Beeinflussung der Helligkeit der Konturwiederholungen hat es sich als anwendbar erwiesen, die Helligkeit der Linie über die Sättigung  $S$  zu regeln. Je geringer die Sättigung um so heller erscheint die Linie. Die Sättigung  $S_k$  für die  $k$ -te zeitliche Wiederholung von insgesamt  $n$  Wiederholungen berechnet sich mit Hilfe der Basissättigung  $S_B$  nach folgender Formel:

$$S_k = \frac{S_B}{1 + \frac{n-k+1}{n}} \quad n, k \in N, 1 \leq k \leq n \quad (4.5)$$

Dabei ist die Wiederholung um so älter je kleiner  $k$  ist und somit wird sie auch um so heller gezeichnet. Durch die Formel wurde die minimale Sättigung auf 50% der Basissättigung festgelegt. Damit sollte auf jeden Fall gewährleistet sein, daß die



**Abbildung 4.30:** Dreifache Wiederholung der Komplettkontur im Abstand von jeweils 4 Frames wobei die Konturwiederholungen heller gezeichnet wurden.

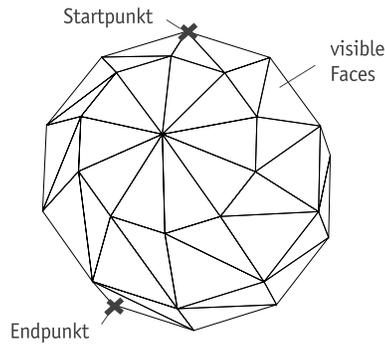
letzte Wiederholung nicht zu hell ist. Natürlich kann es passieren, daß bei einer höheren Anzahl an Wiederholungen einzelne Graustufen nicht durch das menschliche Auge unterscheidbar sind. Doch auf jeden Fall ist die zeitlich am weitesten zurückliegende Wiederholung immer heller als eine nicht ganz so weit zurückliegende. Alle Wiederholungen unterscheiden sich damit ganz deutlich vom Originalobjekt, wie in Abbildung 4.30 gezeigt.

### 4.3.2 Bildung von Teilkonturen

Obwohl die erwähnte Methode recht übersichtliche Bilder erzeugt, ist es manchmal nicht nötig, die komplette Kontur in der Wiederholung zu zeichnen. Vielmehr genügt ein optisch markanter Teil der Kontur. Für einen Künstler ist dies eine Erleichterung der Arbeit, da er nicht die komplette Kontur zeichnen muß und an der Teilkontur auch noch Vereinfachungen vornehmen kann. Für einen Betrachter ergibt sich der Vorteil, daß er trotzdem noch ein übersichtliches Bild hat. Für die rechentechnische Umsetzung stellen sich die Verhältnisse umgekehrt dar. Jede Vereinfachung bedeutet eine Abweichung vom Original und somit zusätzliche Rechenarbeit. Im folgenden soll ein einfacher Algorithmus beschrieben werden, der nur einen Teil der Außenkontur zeichnet. Eine Erkennung optisch markanter Merkmale eines Objektes und eine natürlich wirkende Vereinfachung der Objektkontur würde den Rahmen dieser Arbeit sprengen, stellt aber eine interessante Herausforderung dar, die es Wert wäre, weiterverfolgt zu werden.

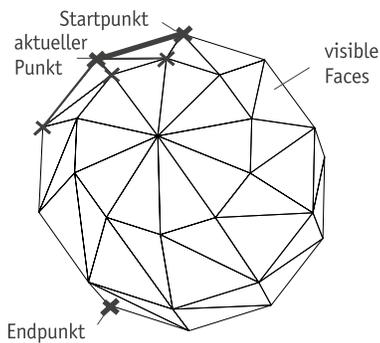
Ausgangsbasis für den Algorithmus seien zwei sichtbare Vertices an der Außenlinie des Objektes. Zwischen diesen beiden Punkten soll die Außenlinie gezeichnet werden. Abbildung 4.31 soll dies verdeutlichen.

Der Algorithmus sucht zuerst alle Dreiecksflächen heraus, die den aktuellen Punkt beinhalten (siehe Abbildung 4.32). Jeder dieser Eckpunkte wird mit dem aktuellen Punkt zu einer Linie verbunden. Ist der Winkel, der vom aktuellen Punkt in Blickrichtung des vorigen Punktes rechts, zwischen der zu testenden Linie und der Linie, die aus dem aktuellen und dem vorherigen Punkt gebildet wird, maximal,

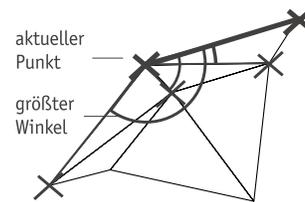


**Abbildung 4.31:** Ausgangssituation für die Bestimmung der Teilkontur zwischen zwei gegebenen Punkten.

so ist der entsprechende Punkt ein neuer Punkt an der Außenlinie des Objektes. Dies funktioniert auch bei konkaven Objekten. Voraussetzung ist lediglich, daß keine Selbstüberschneidungen auftreten. Abbildung 4.33 verdeutlicht dies an einer Vergrößerung des Beispiels.



**Abbildung 4.32:** Typische Situation bei der Teilkonturbildung.



**Abbildung 4.33:** Ermittlung der Kante, deren Winkel nach rechts relativ zur vorhergehenden Kante am größten ist.

Da zu Beginn noch keine zwei Punkte bekannt sind, anhand derer ein Winkelvergleich vorgenommen werden kann, wird zuerst der Punkt der angrenzenden Dreiecksflächen selektiert, der in Blickrichtung des Endpunktes am weitesten rechts liegt.

Sobald mittels dieser Methode der Endpunkt erreicht wurde, ist die Suche nach der Teilkontur erfolgreich abgeschlossen. Abbildung 4.34 zeigt die gefundene Teilkontur des Beispiels, Abbildung 4.35 die Resultate des Programmes für ein ähnliches Objekt.

Für die Bildung der Teilkontur eines Objektes zwischen zwei gegebenen Objekten soll zusammenfassend der Algorithmus in Pseudocode aufgeführt werden:

```

PROZEDUR Teilkontur
FÜR jede Konturwiederholung FÜHRE AUS: {
  Teilkontur := BERECHNE Teilkontur (OBJEKT: Objekt STARTPUNKT: StartPunkt
    ENDPUNKT: EndPunkt WIEDERHOLUNG: Wiederholungsnummer)
  ZEICHNE Teilkontur
}

```

```

FUNKTION BERECHNE Teilkontur (OBJEKT: Objekt STARTPUNKT: spunkt
  ENDPUNKT: epunkt WIEDERHOLUNG: rep)

```

```

BERECHNE Transformationen FÜR Objekt FÜR Frame der Wiederholung rep.
FÜGE spunkt ZUR outline HINZU.
relativpunkt := spunkt + (spunkt - epunkt).
Erfolgsflag := BERECHNE nächsten Punkt (VON: spunkt RELATIV ZU:
  relativpunkt).
Erfolgsflag ?
WENN FALSCH FÜHRE AUS: {
  FEHLER: Algorithmus versagt.
  Berechnung abbrechen
}
GIB outline ZURÜCK

```

```

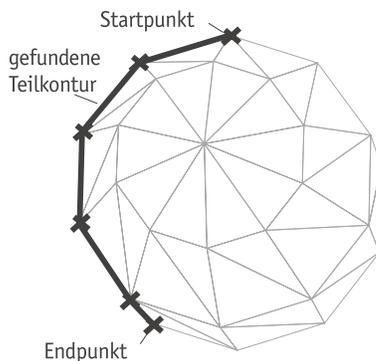
FUNKTION BERECHNE nächsten Punkt (VON: spunkt RELATIV ZU: relativpunkt).

```

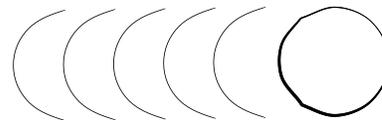
```

Liste := ERMITTLE alle Eckpunkte der an spunkt angrenzenden Dreiecksflächen.
BERECHNE Winkel zwischen allen neuen Kanten und der Bezugskante.
selPunkt := SELEKTIERE den Eckpunkt AUS Liste für den der Winkel maximal ist.
selPunkt = endPoint ?
WENN WAHR FÜHRE AUS {
  FÜGE selPunkt ZUR outline HINZU.
  GIB WAHR ZURÜCK.
}
WENN FALSCH FÜHRE AUS {
  selPunkt bereits in outline vorhanden ?
  WENN WAHR FÜHRE AUS { FEHLER: Rekursion entdeckt. GIB FALSCH ZURÜCK }
  FÜGE selPunkt ZUR outline HINZU.
  Erfolgsflag := BERECHNE NÄCHSTEN PUNKT (VON: spunkt RELATIV ZU:
    relativpunkt).
  Erfolgsflag ?
  WENN FALSCH FÜHRE AUS { GIB FALSCH ZURÜCK. }
}
GIB WAHR ZURÜCK.

```



**Abbildung 4.34:** Gefundene Teilkontur des Objektes zwischen den gegebenen Punkten.



**Abbildung 4.35:** Ein fliegender Ball als Beispiel für die Anwendung von Teilkonturen.

### 4.3.3 Zusammenfassung des Algorithmus zur Generierung von Konturwiederholungen

Zum besseren Verständnis des Gesamt Ablaufs der Berechnung der Konturwiederholungen soll der Algorithmus nochmals zusammengefaßt werden. Nachdem die normalen Berechnungen der Linien abgeschlossen sind, erfolgt die Berechnung der Teilkonturen. Dazu werden zuerst die bewegten Objekte selektiert und für jeden vergangenen Frame transformiert, gegen den Sichtkörper geclippt, die verdeckten Kanten und Flächen entfernt und die selektierten Linien in Kurven umgewandelt. Alles so wie ein exklusives Rendering der bewegten Objekte. Daraufhin wird die Helligkeit für die Darstellung der Linien des jeweiligen Zeitpunktes bestimmt. Ist die Berechnung von Teilkonturen nicht gewünscht, wird das Resultat angezeigt. Anderenfalls werden aus den berechneten Konturen Teilkonturen gebildet, die abzubildenden Linien selektiert und in Kurven umgewandelt. Daraufhin können sowohl die Konturwiederholungen als auch die Teilkonturen angezeigt werden. Erst danach werden die Kurven aus dem aktuellen Frame angezeigt. Abbildung 4.36 stellt die Zusammenhänge noch einmal graphisch dar.

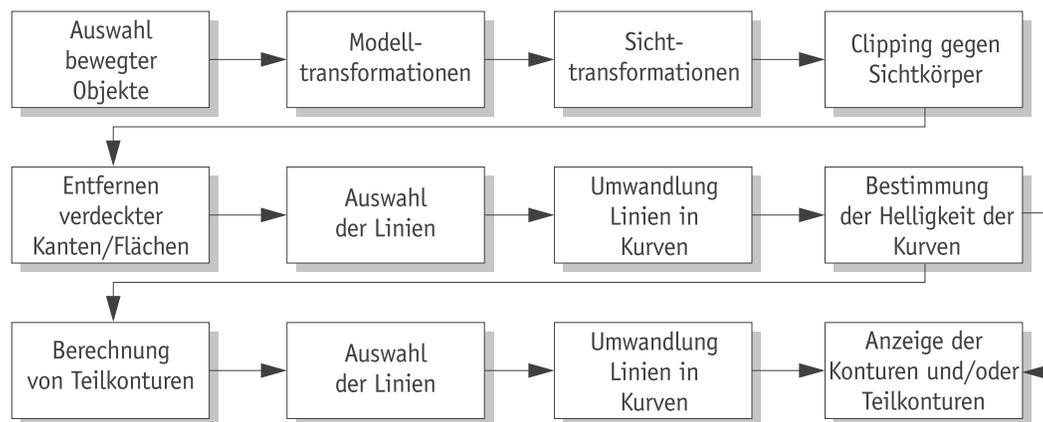


Abbildung 4.36: Schritte zur Darstellung von Konturwiederholungen.

## 4.4 Bewegungspfeile

Nachdem alle vorgestellten Techniken nur Bewegungen der Vergangenheit darstellen, befaßt sich dieser Abschnitt mit zukünftigen Bewegungen. Als einziges geeignetes Mittel für die Darstellung dieser Bewegungen erscheinen Pfeile, da in allen untersuchten Bildern nie ein anderes Stilmittel für eine derartige Aufgabe genutzt wurde.

Pfeile sind prinzipiell nichts anderes als in die Zukunft weisende Bewegungslinien, denen am Ende noch eine Pfeilspitze angehängt wurde. Genau so sollen sie auch berechnet werden. Entsprechend sind fast alle Algorithmen und Operationen genau die gleichen, wie in Kapitel 4.2 beschrieben. Jedoch sind hier die beiden Extrempunkte nicht so bedeutend wie bei den Bewegungslinien. In vielen Fällen reicht bereits

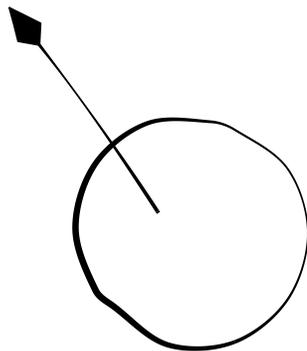
ein Pfeil aus, um die Bewegung anzudeuten. Dieser Pfeil müßte dann etwa mittig zwischen den Extrempunkten liegen. Denkbar wäre auch die Selektion des absoluten Maximums der x-Koordinaten oder des Pivot-Punktes als Ausgangspunkt für die Pfeildarstellung. Je nach Anwendungsfall muß die dazu passende Selektion gefunden werden.

Pfeile sind im normalen Alltag häufig gebraucht und eigentlich jedem bekannt. Demnach sind entsprechende Anforderungen an das Aussehen von Pfeilen zu beachten. Die gebräuchlichsten Pfeile bestehen aus einer Linie, die an allen Punkten gleich stark ist, und einer gefüllten Pfeilspitze in Form eines schlanken Dreiecks am Ende des Pfeiles.

Es bieten sich zwei Möglichkeiten für die Umsetzung dieser Anforderungen an. Zum einen kann mit Hilfe der Linienstile einer Linie die Form eines Pfeiles gegeben werden, zum anderen kann an die Position des letzten Punktes tangential eine Pfeilspitze nachträglich gezeichnet werden.

#### 4.4.1 Pfeildarstellung durch Linienstile

Um mittels eines Linienstils einer Linie die Form eines Pfeils zu geben, muß der Linienstil exklusiv die Dicke der Linie beeinflussen. Die Liniendicke läßt sich über den Parameter *Pressure*<sup>3</sup> steuern. Durch die Definition eines Linienstiles, der zu etwa 90% gleichmäßig dünn verläuft und dann plötzlich die 20-fache Breite zugewiesen bekommt, um am Ende wieder normal breit zu sein, wird die Form des Pfeiles bestimmt. Abbildung 4.37 zeigt ein Beispiel für die Anwendung von auf diese Weise erzeugten Pfeilen.



**Abbildung 4.37:** Darstellung der Bewegung einer Kugel mittels eines durch Linienstile entstandenen Pfeiles

Der Vorteil dieser Methode zur Pfeilgenerierung ist die automatische Platzierung der Pfeilspitze am Ende des Pfeiles. Der Linienstil sorgt außerdem dafür, daß die Pfeilspitze auch wirklich tangential anliegt, da entsprechende Berechnungen durch ihn automatisch erfolgen. Die Pfeilspitzen passen sich in ihrer Größe an die Länge der Pfeile an, da der gesamte Linienstil von der Länge des Pfades beeinflusst wird. Dies muß jedoch nicht unbedingt vorteilhaft wirken.

<sup>3</sup> Eine Übersicht über die vorhandenen Parameter kann [Sch97b] entnommen werden.

Ein entscheidender Nachteil dieser Methode ist, daß die Pfeilspitzen trotz exakter Linienstildefinition nicht das geforderte Aussehen annehmen. Die Pfeilspitze hat kaum Ähnlichkeiten mit einem Dreieck. Außerdem gibt es Fälle, in denen die Pfeilspitze völlig entartet und so kaum noch Assoziationen mit einem Pfeil möglich sind. Dadurch, daß die Pfeile sich an die Länge der Linien anpassen, können in einem Bild so viele unterschiedengroße Pfeilspitzen auftreten, daß insbesondere Techniker diese Art der Pfeildarstellung ablehnen werden, bei denen alle Pfeilspitzen zumindest ähnlich groß sein müssen.

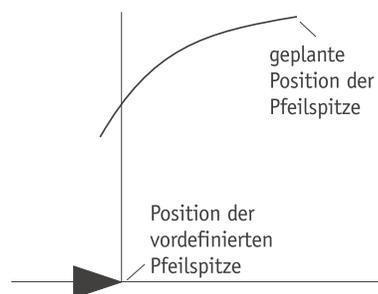
#### 4.4.2 Pfeildarstellung durch separates Zufügen einer Pfeilspitze

Für die andere Möglichkeit der Darstellung von Pfeilen wird von einer vordefinierten Pfeilspitzengröße ausgegangen, die später nicht änderbar ist. Dadurch sind alle Pfeilspitzen im Bild gleich groß, was besonders technischen Illustrationen entgegenkommt.

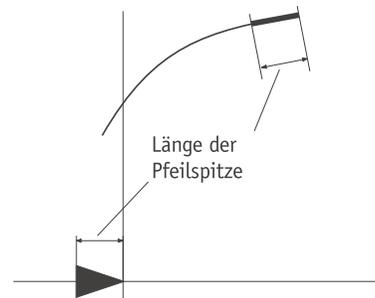
Nachdem die Position und Länge des Pfeiles im zweidimensionalen Bild bekannt ist, wird die Pfeilspitze auf den Pfeil eingepaßt. Dazu werden die Position und die Länge des Pfeiles im 2D-Bild verwendet. Ausgangsbasis ist die vordefinierte Pfeilspitze, die so positioniert wurde, daß der breite Teil auf dem negativen Teil der x-Achse und die Spitze des Dreiecks im Koordinatenursprung liegt (siehe Abbildung 4.38).

Zuerst wird die Länge des Pfeiles auf die Kurve abgebildet, die später einen Pfeil darstellen soll. Auf diese Weise erhält man zwei Punkte. Der Winkel  $\alpha$  zwischen der Linie zwischen den beiden Punkten und der positiven x-Achse ist der Winkel, um den die Pfeilspitze gedreht werden muß. Um diesen Winkel wird die Pfeilspitze dann um  $z$  gedreht. Abschließend wird die Pfeilspitze mittels Translation an die Position des Endes des Pfeiles gebracht, wie in Abbildung 4.41 zu sehen.

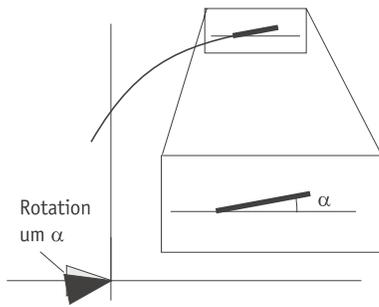
Die Vorteile dieser Darstellung sind relativ deutlich. Die Pfeile wirken gleichmäßig und die Pfeilspitzen gewohnt. Alle Pfeilspitzen sind gleich groß. Es kann nicht zu entarteten Pfeilen kommen. Sollte ein Pfeil so kurz sein, daß die Pfeilspitze länger



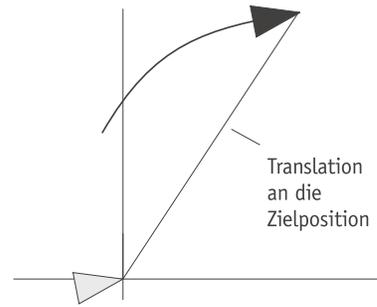
**Abbildung 4.38:** Ausgangssituation zur Anpassung einer Pfeilspitze an eine Linie.



**Abbildung 4.39:** Zuerst wird die Länge der Pfeilspitze auf die Linie projiziert.



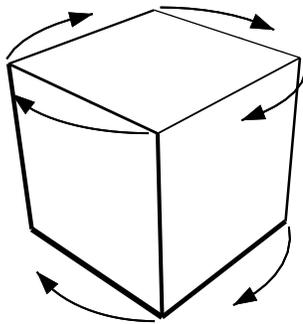
**Abbildung 4.40:** Ermittlung des Drehwinkels  $\alpha$  um den die Pfeilspitze um die z-Achse rotiert wird.



**Abbildung 4.41:** Abschließend wird die gedrehte Pfeilspitze an die Endposition verschoben.

wäre, kann die Pfeilspitze skaliert werden. Es empfiehlt sich aber in diesem Fall den kompletten Pfeil nicht darzustellen, da die Bewegung, die er beschreibt, zu unerheblich ist. Der Rechenaufwand ist für diese Methode zwar höher, als bei der Pfeildarstellung mittels Linienstils, fällt aber nicht weiter ins Gewicht.

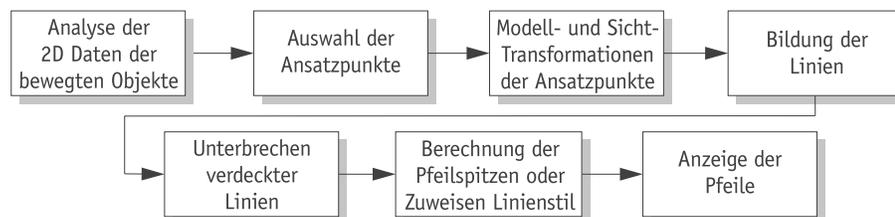
Diese Art der Darstellung hat so viele Vorteile, daß der Einsatz von Linienstilen zur Darstellung von Pfeilen eigentlich nicht zu empfehlen ist. Abbildung 4.42 zeigt das Ergebnis der nachträglichen Ergänzung der Pfeilspitze.



**Abbildung 4.42:** Darstellung der Rotation eines Quaders mit Pfeilen.

### 4.4.3 Zusammenfassung des Algorithmus zur Berechnung von Pfeilen

Da sich die Berechnung der Pfeile nur unwesentlich von der Berechnung der Bewegungslinien unterscheidet, wird lediglich eine graphische Übersicht gegeben, die die einzelnen Phasen der Erzeugung eines Pfeiles zusammenfaßt (siehe Abbildung 4.43).



**Abbildung 4.43:** Schritte zur Darstellung eines Pfeiles.

Nachdem die Pfeile berechnet und angezeigt wurden, ist die letzte Phase der Bewegungsdarstellung, wie in Abbildung 3.3 in Abschnitt 3.2.3 dargestellt, abgeschlossen. Alle Linien können auf das Endmedium übertragen und angezeigt werden.

Das nächste Kapitel enthält Bilder, die durch die entwickelten Algorithmen berechnet wurden. Alle Bilder sind auflösungsunabhängig und frei skalierbar, die Qualität ist somit nur durch die Möglichkeiten des Ausgabegerätes begrenzt.



## Beispiele

---

In diesem Kapitel sollen einige automatisch generierte Bilder die Leistungsfähigkeit der entwickelten Algorithmen verdeutlichen.

### 5.1 Translationen graphischer Primitive

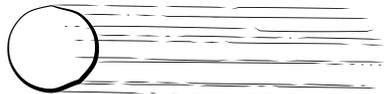
Ein fliegender Ball ist ein in Comics oft zu sehendes Beispiel für die Anwendung von Speedlines. Insbesondere Baseballs werden aufgrund ihrer hohen Fluggeschwindigkeit und großen Beliebtheit in Nordamerika gerne verwendet. So auch von MCCLOUD in [McC93]. Dieses Bild dient als erstes Referenzbild einer einfachen Translation.



**Abbildung 5.1:** Referenzbild eines fliegenden Baseballs aus einem Comic.



**Abbildung 5.2:** Als erste Darstellung eines fliegenden Balls wurde die Kontur des Balls dreimal wiederholt.



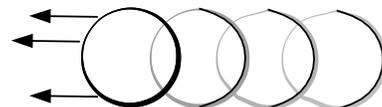
**Abbildung 5.3:** Die Bewegung des Balls durch Bewegungslinien dargestellt. Den Linien wurde der typische Liniestil zugewiesen (zum Ende hin dünner werdend) und die Linien wurden zusätzlich zufällig unterbrochen.



**Abbildung 5.4:** Die Kombination aus beiden Methoden verdeutlicht die Bewegung noch stärker als jede für sich.



**Abbildung 5.5:** Statt der kompletten Kontur beschränkt sich die Referenz auf Teilkonturen zur Ergänzung der Bewegungslinien. Die gerenderten Teilkonturen sind nicht ganz so diffizil.



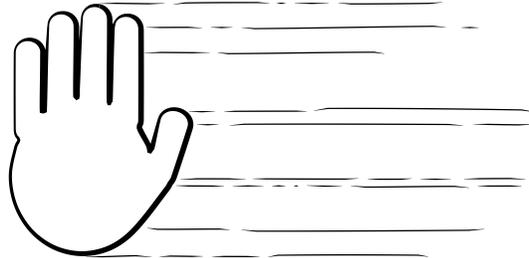
**Abbildung 5.6:** Eher ungewöhnlich ist die Kombination aus der Nutzung von Pfeilen, Konturen und Teilkonturen, trotzdem kommt die Bewegung zur Geltung.

## 5.2 Einfache Bewegungen einfacher Objekte

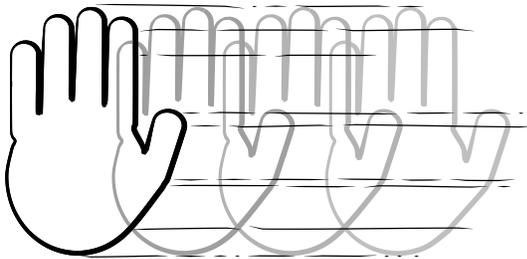
Am Beispiel einer Hand werden verschiedene einfache Bewegungen dargestellt. Das Modell besteht nicht mehr nur aus einem graphischen Primitiv sondern aus einer Kombination mehrerer Primitive. Ein Referenzbild ist wiederum [McC93] von McCLOUD entnommen.



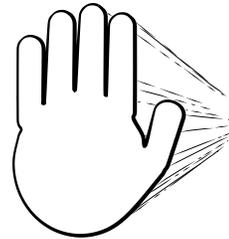
**Abbildung 5.7:** Beispiel aus [McC93] zur Darstellung von Bewegungen in Comics. Hauptaugenmerk liegt auf der Hand.



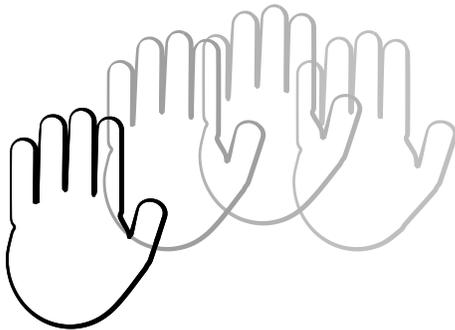
**Abbildung 5.8:** Einfache Translationsdarstellung der Hand mittels zufällig unterbrochener Bewegungslinien.



**Abbildung 5.9:** Als Ergänzung der Bewegungslinien wurden, ähnlich wie im Referenzbild, die Konturen der Hand zu früheren Zeiten wiederholt.



**Abbildung 5.10:** Eine andere Bewegungsrichtung verändert auch die Bewegungslinien. In diesem Fall bewegt sich die Hand aus dem Hintergrund auf den Betrachter zu.



**Abbildung 5.11:** Darstellung einer Handbewegung entlang einer Kurve mittels Konturwiederholungen.

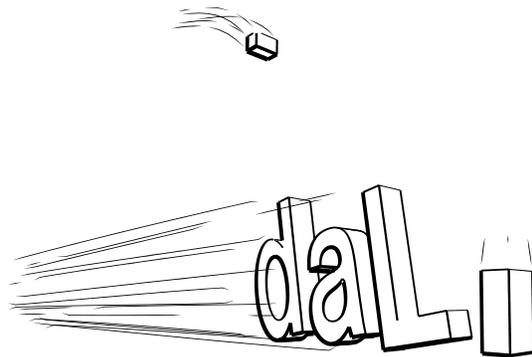


**Abbildung 5.12:** Durch Bewegungslinien wird eine Kombination aus Translation und Rotation dargestellt.

## 5.3 Bewegungen komplexerer Objekte

### 5.3.1 daL!-Animation

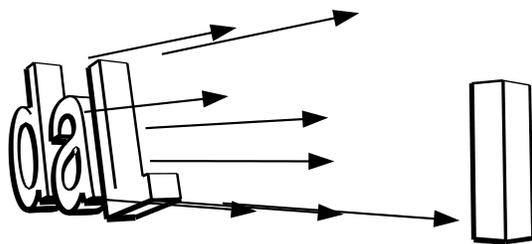
Die daL!-Animation entstand als Anwendungsbeispiel zur Veranschaulichung der Leistungsfähigkeit des daL!-Systems, daß von SCHÖNWÄLDER, SCHLECHTWEG und MASUCH entwickelt wurde (nähere Informationen in [MSS97]). Die Objekte sind hierarchisch aufgebaut und die Bewegung nicht mehr trivial.



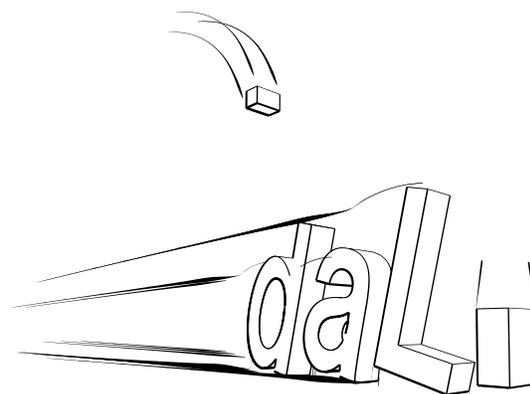
**Abbildung 5.13:** Einzelbild aus der daL!-Animation. Die Buchstabengruppe daL kommt schräg aus dem Hintergrund in das Bild gerast. Die Bewegungslinien zeigen die Geschwindigkeit.



**Abbildung 5.14:** Die Bewegung der Buchstabengruppe wird zusätzlich durch verblassende Konturwiederholungen beschrieben. Die Bewegung ist dadurch für das Auge leichter zu erfassen.



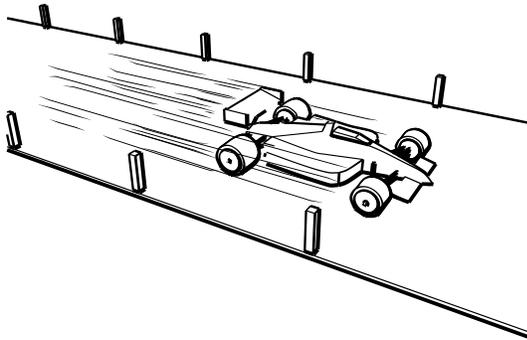
**Abbildung 5.15:** Durch die Pfeile wird die zukünftige Bewegung der Buchstabengruppe beschrieben. Jeder Buchstabe außer dem L bindet 3 Pfeile an sich. Durch die Unterbrechungen bei Überdeckungen lassen sich die Pfeile jedoch nicht mehr so schnell dem Objekt zuordnen.



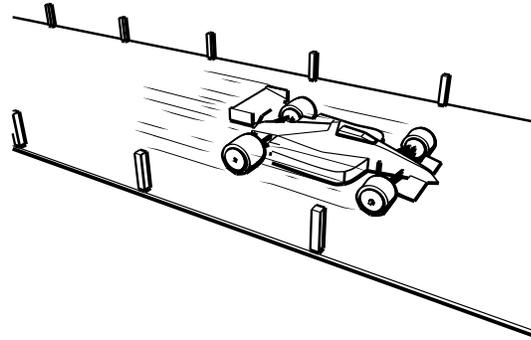
**Abbildung 5.16:** Durch die Abbildung der relativen Geschwindigkeit auf die Liniendicke lassen sich Beschleunigungen einfach erkennen. Je dicker die Linie um so schneller war das Objekt. (Liniendicke im Beispielbild zur besseren Erkennbarkeit übertrieben)

### 5.3.2 Animation eines Rennwagens

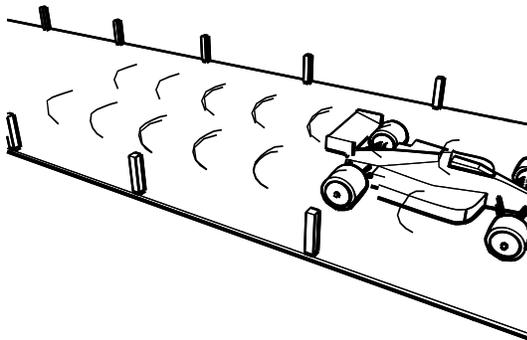
Die klassische Anwendung für Bewegungslinien ist die Darstellung extrem schneller Bewegungen wie sie durch schnelle Fortbewegungsmittel wie z. B. Rennwagen ausgeführt werden. In Abbildung A.25 im Anhang ist die Anwendung von Speedlines auf einem Plakat einer Ankündigung zu einem Flugzeugrennen zu erkennen. Da dieses eine häufige Anwendung ist, wird ein Rennwagenmodell als Beispiel eingesetzt. Das Modell wurde ursprünglich erschaffen, um durch Animation der Linienstile den Eindruck von Bewegung zu erwecken, ohne daß das Objekt selbst bewegt wird [MSS98].



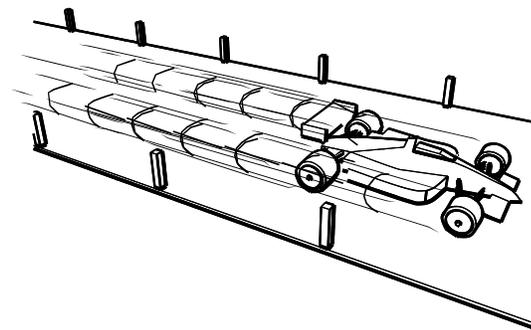
**Abbildung 5.17:** Lange Bewegungslinien deuten auf eine hohe Geschwindigkeit hin. Bei solch langen Linien ist eine zufällige Unterbrechung empfehlenswert. Außerdem läßt die zufällige Variation der Kontrollpunkte der Bewegungslinien das Bild natürlicher erscheinen.



**Abbildung 5.18:** Kürzere Bewegungslinien lassen die Bewegung für das menschliche Auge als nicht ganz so schnell erscheinen wie es lange Linien tun. Bei kurzen, durch wenig Kontrollpunkte definierten Bewegungslinien sollten zufällige Unterbrechungen vermieden werden.



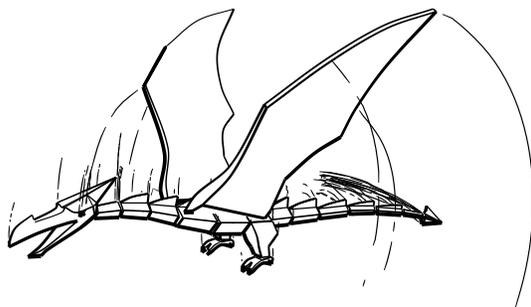
**Abbildung 5.19:** Bei schnellen Bewegungen werden häufig markante Bereiche des bewegten Objektes wiederholt. In diesem Fall wurden die 4 Reifen durch den Benutzer selektiert und für die Generierung von Teilkonturen genutzt.



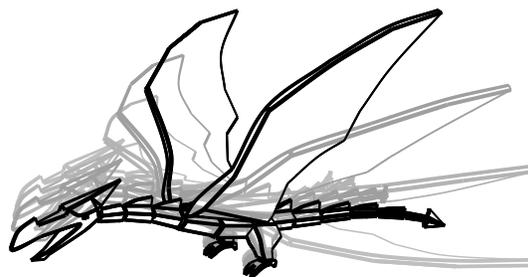
**Abbildung 5.20:** Die Wirkung der Teilkonturen entfaltet sich erst richtig in Kombination mit Bewegungslinien. Die Bewegung erscheint dem Betrachter extrem schnell und somit ist die Absicht des Erzeugers des Bildes erkannt worden.

### 5.3.3 Animation eines fliegenden Drachens

Das Besondere an der Animation des auf der Stelle fliegenden Drachens ist die Vielzahl der bewegten Objekte. Jedes Glied der Schwanzspitze ist ein einzelnes Objekt. Bei der Darstellung aller bewegten Objekte mittels Bewegungslinien in Abbildung 5.21 zeigt sich, daß eine Häufung von Linien am Schwanzende des Drachens entsteht. Günstiger wäre die Zusammenfassung der Schwanzglieder zu einem Objekt. Dadurch würde eine gleichmäßigere Verteilung erreicht werden. Alternativ können aber auch nur ausgewählte Objekte für die Bewegungsberechnungen angegeben werden. Dieses Problem tritt bei der Wahl von Konturwiederholungen als Bewegungsdarstellungsmethode nicht auf, wie Abbildung 5.22 zu entnehmen ist.



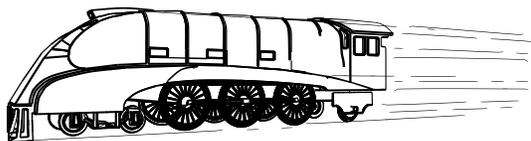
**Abbildung 5.21:** Bei zu vielen kleinen animierten Objekten treten Häufungen von Bewegungslinien auf. Gleiches gilt bei der Verwendung von Pfeilen.



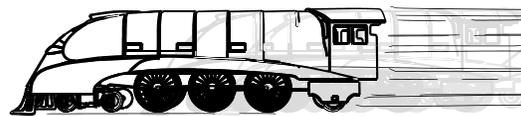
**Abbildung 5.22:** Bei der Nutzung von Konturwiederholungen zur Bewegungsdarstellung spielt die Anzahl der animierten Objekte keine Rolle.

### 5.3.4 Animation eines Zuges

Im Gegensatz zum eben erwähnten Drachen wurden beim Zug-Modell alle Objekte zu einem Objekt zusammengefaßt. Das Objekt selbst besteht aufgrund der fein ausgearbeiteten Modellierung aus etwa 20.000 Dreiecken und ist damit relativ groß. Trotz der Größe entstehen bei der Berechnung der Bewegungsdarstellungen keine Probleme. Abbildungen 5.23 und 5.24 zeigen den Einsatz verschiedener Methoden an diesem Modell.



**Abbildung 5.23:** Alle Objekte des Zuges wurden zu einem Objekt zusammengefaßt. Häufungen von Bewegungslinien entstehen dadurch nicht mehr.

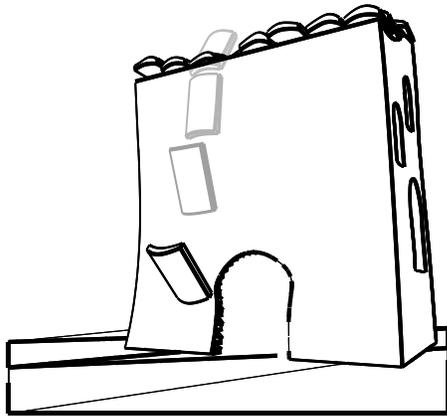


**Abbildung 5.24:** Die Zusammenfassung mehrerer Objekte zu einem großen wirkt sich nicht negativ auf die Verwendung von Konturwiederholungen aus.

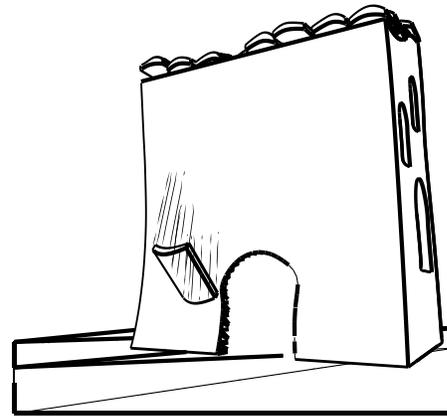
## 5.4 Animation von Teilen eines komplexeren Modells

Bei vielen Animationen bewegen sich nicht alle Objekte. Um zu untersuchen, wie sich die Bewegungsdarstellungen in die Szene einfügen wurden verschiedene Modelle erstellt.

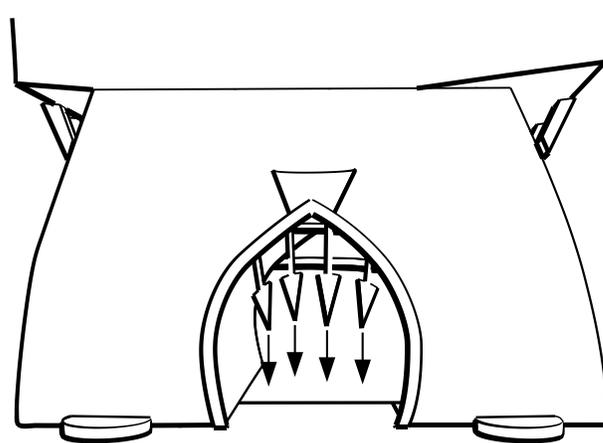
Anhand eines fallenden Dachziegels wird die Wirkung einer Kombination aus Rotation und Translation vor einem nicht animierten Teil der Szene gezeigt. Ein Fallgitter eines Durchganges verdeutlicht eine problematische Bewegungen.



**Abbildung 5.25:** Darstellung des fallenden Ziegels durch Konturwiederholungen die mit zunehmendem Alter heller werden. Da zuerst die älteren bewegten Objekte berechnet und gezeichnet werden ist die Kontur des Hauses an allen Stellen gleichmäßig und nicht durch hellere Linien überlagert.

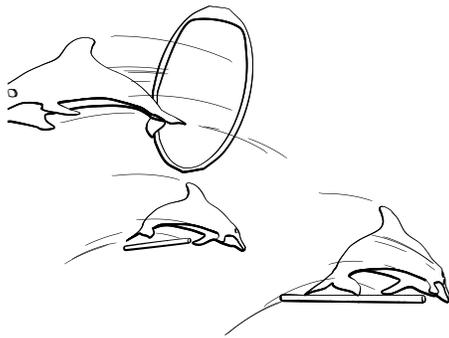


**Abbildung 5.26:** Darstellung des fallenden Ziegels durch Bewegungslinien. Auf das Entfernen verdeckter Linien wurde testweise verzichtet. Die Bewegung des Ziegels ist erkennbar, jedoch wirken die nicht entfernten Linien störend auf das Gesamtbild. Die Sichtbarkeitsbestimmung ist also empfehlenswert.

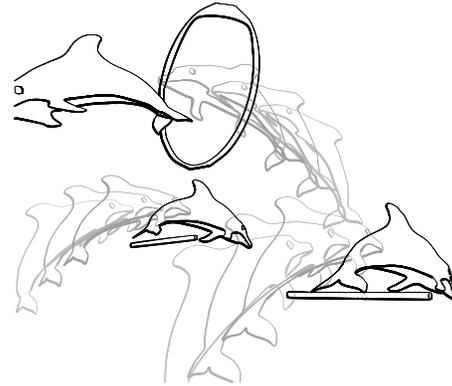


**Abbildung 5.27:** Die Bewegung eines Fallgitters in einem Tor ist eine problematische Situation. Bewegungslinien nach Definition dieser Arbeit beginnen erst nach dem Objekt und sind nicht mehr sichtbar, da sie vom Gebäude verdeckt werden. Eine Nutzung von Pfeilen erscheint in diesem Fall als günstigere Methode.

Bei einigen Animationen bewegen sich Objekte durch andere Objekte hindurch. Die Abbildungen 5.28 und 5.29 zeigen die Ergebnisse der Algorithmen für diesen Fall am Beispiel eines durch einen Reifen springenden Delphins.

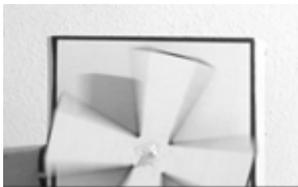


**Abbildung 5.28:** Darstellung dreier springender Delphine. Der mittlere Delphin springt durch ein anderes Objekt, den Reifen, hindurch.

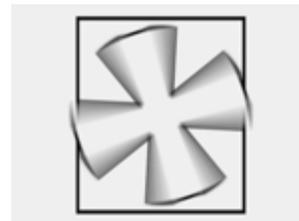


**Abbildung 5.29:** Darstellung der gleichen Szene wie links durch Konturwiederholungen.

Die Darstellung einer reinen Rotationsbewegung ist problematisch, da der Ansatz zur Selektion der Vertices, die der Ausgang für eine Bewegungslinie oder einen Pfeil sind, nicht unbedingt greift. Er geht davon aus, daß alle Bewegungslinien an der gleichen Seite ansetzen, was bei Rotationen nicht stimmt. Trotzdem ist das Resultat unter Nutzung dieses Ansatzes eine interessante Darstellung, die ihre Aufgabe, die Darstellung einer Bewegung, erfüllt. Abbildung 5.32 zeigt das Ergebnis des normalen Ansatzes am Beispiel rotierender Mühlenflügel. Vorher sollen jedoch photorealistische Ergebnisse einer ähnlichen Rotation als Vergleichsmöglichkeit in den Abbildungen 5.30 und 5.31 dargestellt werden.

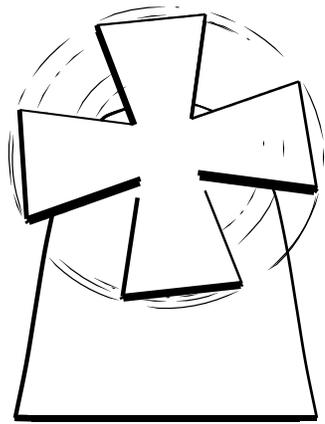


**Abbildung 5.30:** Aufnahme einer schnellen Rotationsbewegung eines Mühlenmodells mit einer digitalen Kamera.

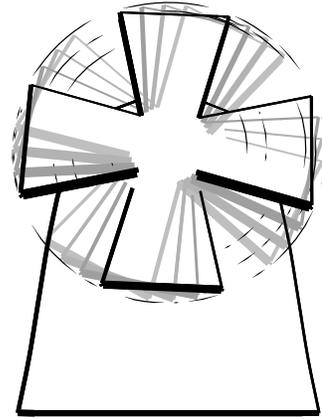


**Abbildung 5.31:** Berechnung von Bewegungsunschärfe in einem vergleichbaren Modell durch das Autodesk 3D-Studio Max 2.0.

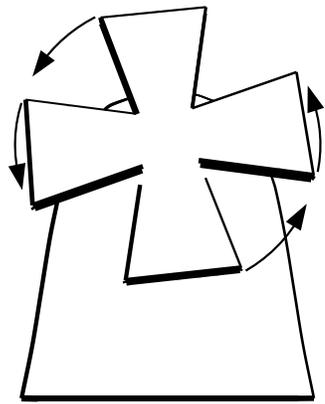
Bei der Nutzung von Pfeilen zur Bewegungsdarstellung liefert der Ansatz ebenfalls ein brauchbares Resultat (siehe Abbildung 5.34). Das Ergebnis ist allerdings unregelmäßig. Teilweise wird jedoch eine regelmäßige Darstellung gewünscht. Um dies zu erreichen, wird der Algorithmus mehrmals durchlaufen und bei jedem neuen Durchlauf der Winkel, um den das Objekt zur Selektion der Vertices gedreht wird, um jeweils  $45^\circ$  verändert. Das Resultat kann in 5.35 begutachtet werden.



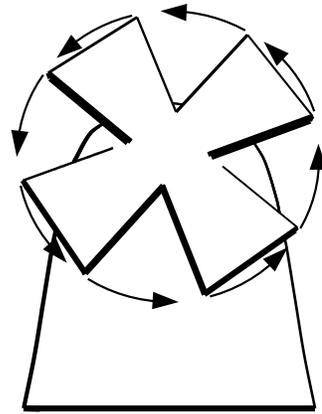
**Abbildung 5.32:** Resultat des normalen Ansatzes für die Bestimmung der Ausgangsvertices von Bewegungslinien bei der Rotationsbewegung von Flügeln einer Mühle. Die Linien sind nicht zu gleichmäßig verteilt, die Bewegung erkennbar.



**Abbildung 5.33:** Wird die Bewegungskdarstellung der rotierenden Mühlenflügel mittels Speedlines durch die Nutzung von verblassenden Konturwiederholungen unterstützt, ist die Bewegung an sich und die Bewegungsrichtung eindeutig zu erkennen.



**Abbildung 5.34:** Resultat des normalen Selektionsalgorithmus zur Bestimmung der Ausgangspunkte der Pfeile.



**Abbildung 5.35:** Resultat der Pfeildarstellung nach mehrfacher Ausführung der Pfeildarstellungsprozedur.

## 5.5 Verbesserung der Ergebnisse

Während der Erstellung der Beispielbilder zeigten sich einige Eigenschaften des erweiterten Renderers, die von herkömmlichen photorealistischen Renderern abweichen. Das Ergebnis des Renderings kann unter Umständen anders ausfallen als erwartet. Dieser Abschnitt gibt Hinweise zur Modellierung und Handhabung des Programms, deren Beachtung sich positiv auf die Qualität der Ergebnisse auswirkt.

### 5.5.1 Hinweise zur Modellierung

Zur Verbesserung der Rendering-Resultate empfiehlt es sich, das Modell an die Anforderungen des erweiterten Liniengraphik-Renderers anzupassen. Die folgenden

Hinweise sollten beachtet werden, wenn ein Modell speziell für die Nutzung des vorgestellten Programmes erstellt wird:

- Die Modelle sollten nicht zu fein modelliert werden. Der Liniengraphik-Renderer wurde dahingehend optimiert, daß er eine hohe Darstellungsqualität schon bei groben Modellen erreicht. Die Einsparung von Polygonen resultiert in einer deutlich schnelleren Bearbeitung des Modells bei gleicher Qualität des Ergebnisses.
- Bei zusammengesetzten animierten Objekte sollte darauf geachtet werden, sie so zu gestalten, daß eine einfache Selektion eines Hauptobjektes die Hauptbewegung erkennen läßt und zur Generierung von Bewegungslinien ausreicht.
- Die Kamerapositionen sollten so gewählt werden, daß eine sinnvolle Bewegungsdarstellung möglich ist. Ein Objekt das sich direkt aus dem Hintergrund auf die Kamera zubewegt, ist unbedingt zu vermeiden, da die Darstellung einer solchen Bewegung in einem Bild schon theoretisch nicht möglich ist.
- Es empfiehlt sich, die Animation über möglichst viele Frames zu erstellen. Da die kleinste zeitliche Auflösung, die durch das Programm ermöglicht wird, ein Frame ist, kann durch die Nutzung mehrerer Frames das Ergebnis besser kontrolliert werden.
- Da Texturen durch den Liniengraphik-Renderer nicht interpretiert werden, kann auf diese verzichtet werden.
- Auf alle Flächen sollten die Methoden *Autosmooth* und *Unify Normals* angewandt werden. Dies bewirkt, daß nicht-konturgebende Kanten im Modell nicht dargestellt werden. Abbildung 5.36 zeigt den Unterschied in der Darstellung ohne und mit Anwendung dieser Methoden.
- Um bei der Nutzung von Bewegungslinien und Pfeilen bessere Ergebnisse zu erzielen, empfiehlt es sich, Objekte, die die gleiche Bewegung ausführen, zu einem Objekt zusammenzufassen.



**Abbildung 5.36:** Durch Optimierungen des Modells kann das Ergebnis des Renderings stark beeinflusst werden. Zu sehen sind die Renderergebnisse ohne (links) und mit Anwendung der Methoden *autosmooth* und *unify normals* (rechts) auf das Modell.

Die Beachtung der meisten dieser Hinweise wirkt sich vorwiegend auf die Darstellung durch den Linienrenderer aus und somit auch auf Konturwiederholungen. Die Qualität der anderen Bewegungsdarstellungen läßt sich durch verschiedene Programmeinstellungen verbessern, die im folgenden angeführt werden.

## 5.5.2 Hinweise zur Anwendung des Programmes

Der erweiterte Liniengraphik-Renderer ist vielseitig konfigurierbar und erlaubt so eine Kontrolle über die verschiedenen Bewegungsdarstellungsmöglichkeiten. Das Aussehen der Bewegungslinien kann durch die Definition eines eigenen Linienstils beliebig angepaßt werden, die Anzahl der Konturwiederholungen und der Abstand zueinander wie auch die Anzahl der möglichst zu verwendenden Bewegungslinien und Pfeile sowie indirekt der Abstand der eingesetzten Mittel zum Ausgangsobjekt sind einstellbar. Es können auch einige programmspezifische Alternativen ausgewählt werden, auf die hier aber nicht näher eingegangen werden soll.

- Die Darstellung der Einzelbilder ist völlig unabhängig von der eigentlichen Animationszeit. Um eine schnellere Bewegung darzustellen kann unter Nutzung der gleichen Anzahl von Frames die Objektgeschwindigkeit im Modell geändert werden. Es empfiehlt sich aber, einfach nur eine höhere Anzahl von Frames zur Generierung von Bewegungen zu nutzen da diese Methode keine Änderungen am Modell erfordert und deutlich schneller umzusetzen ist. Die Bewegungslinien sind dadurch kontrollierbarer.
- Das Programm erlaubt das Auslassen einer beliebigen Anzahl von Frames am Ansatzpunkt der Bewegungslinien und Pfeile. Dadurch beginnt die entsprechende Linie erst mit etwas Abstand vom Objekt, was natürlicher wirkt. Bei konventionellen Animationen<sup>1</sup> erscheint ein Frame als geeignete Anzahl. Bei den meisten Modellen ist eine sinnvolle Wahl dieses Abstands von der Geschwindigkeit des animierten Objektes abhängig.
- Die Anzahl der Bewegungslinien und Pfeile ist beim vorgestellten Ansatz frei konfigurierbar. Sinnvoll ist beim Einsatz von Bewegungslinien eine Anzahl von 3–15 Linien pro Objekt, darüber hinaus wirkt das Bild normalerweise unübersichtlich.
- Bei sehr schnellen Bewegungen, die durch viele Frames definiert sind, und bei denen die Bewegungslinien ziemlich lang werden, empfiehlt es sich, diese zufällig unterbrechen zu lassen. Bei kurzen Bewegungslinien oder Bewegungen, die nur durch wenige Frames definiert sind, sollte der Einsatz der zufälligen Unterbrechung allerdings unterbunden werden.
- Bei Darstellung einer nicht-trivialen Bewegung, bei der sich über die im Frame dargestellte Zeit die Bewegungsrichtung ändert, sollte die Option *short angle detection* aktiviert werden, bei der der Winkel, der für die Erkennung der Bewegungsrichtung zur Selektion der Vertices benötigt wird, nicht mittels des ältesten Frames sondern des jüngsten generiert wird.

---

<sup>1</sup> optimiert für die Darstellung von 25 Frames pro Sekunde

# Diskussion

---

Es wurden Verfahren entwickelt und implementiert, durch die eine Darstellung von Bewegungen in einem einzigen liniengraphischen Bild möglich ist. Als Grundlage dient ein 3D-Studio-Modell, das sowohl Geometriedaten der einzelnen Objekte als auch Animationsdaten enthält. Die Bewegungsdarstellungen orientieren sich an traditionellen Graphiken aus verschiedenen Bereichen (Comics, technische Illustrationen, etc.).

Es wurde ein Konzept entwickelt, um einen Liniengraphik-Renderer nach Definition von SCHLECHTWEG und RAAB in [Str98] um die Fähigkeit verschiedener Bewegungsdarstellungen in einem Bild zu erweitern. Dabei werden teilweise verschiedene Phasen des Renderers mehrmals durchlaufen, um zurückliegende Konturen zu berechnen und die Darstellungsmöglichkeiten dieser Konturen erweitert, aber auch neue Phasen hinzugefügt, in denen Bewegungslinien und Pfeile berechnet werden, die, genau wie die Liniengraphik selbst, mit verschiedenen Parametern konfigurierbar sind.

Es können die gleichen Modelle wie für photorealistische Renderer benutzt werden. Um jedoch bessere Ergebnisse zu erreichen, empfiehlt es sich, das Modell auf die speziellen Eigenschaften der vorgestellten Methoden zu optimieren. Hinweise zur Verbesserung der Ergebnisse des vorgestellten Systems werden im Abschnitt 5.5 gegeben.

Vorteil der entwickelten Methoden gegenüber der im Photorealismus benutzten Bewegungsunschärfe ist, daß die Bewegungen direkt und deutlich genauer visualisiert werden können, d. h. die Konturen bleiben erhalten. Dadurch bieten sich deutlich mehr Anwendungen an, die vom Gebrauch der entwickelten Methoden profitieren können.

Im Laufe der Arbeit wurden aber auch Schwächen des gewählten Ansatzes deutlich, die im Folgenden dargelegt werden und für die Lösungsansätze genannt werden. Es schließen sich Ausführungen zu Ideen an, die bei der Bearbeitung der Aufgabenstellung auftraten und die eine Herausforderung für die Zukunft darstellen.

## 6.1 Kritik

Die in dieser Arbeit entwickelten Verfahren zur Darstellung von Bewegungen erheben weder den Anspruch, physikalisch korrekt zu sein, noch sollen sie Ergebnisse liefern, die einen Zeichner ersetzen können. Vielmehr sollen sie die Bewegung *an sich* visualisieren und die Ergebnisse sich dabei nicht zu sehr von üblichen Darstellungsmethoden abheben.

Fast jede Bewegung kann durch eine der vorgestellten Methoden dargestellt werden. Dabei hat jede Methode Vor- und Nachteile, die im folgenden herausgestellt werden. Die Auswahl, welche der Methoden benutzt werden soll, ist einzig dem Benutzer überlassen. Das Programm kann keine derartige Analyse durchführen da die Bildaussage nicht erkennbar ist. Die Wahl der Bewegungsdarstellung ist außerdem eher eine Frage der persönlichen Präferenzen.

### 6.1.1 Bewegungslinien

Bewegungslinien eignen sich am besten für die Darstellung einfacher Bewegungen wie Translationen. Der Algorithmus zur Selektion der Ausgangspunkte dieser Linien geht ganz allgemein von einer Translation aus, liefert aber auch bei anderen Bewegungen brauchbare Ergebnisse. Trotzdem gibt es Bewegungen, die durch Bewegungslinien nicht dargestellt werden können. Dazu gehören vorwiegend solche, bei denen die Linien hinter dem animierten Objekt verdeckt sind. Dies ist aber ein generelles Problem denn manche Bewegungen sind so kompliziert, daß sie sich nicht in nur einem Bild darstellen lassen. Entsprechend gerät auch das Programm in Schwierigkeiten.

Der implementierte Algorithmus versucht die Linien so am Objekt zu plazieren, daß sie das Objekt selbst möglichst nicht schneiden. Das wird erreicht, indem die Linien erst nach<sup>1</sup> dem Objekt beginnen. Bei Anwendung des Sichtbarkeitsalgorithmus werden alle verdeckten Linien entfernt. Das Resultat ist, daß keine einzige Bewegungslinie gezeichnet wird und eine Bewegung im Bild nicht mehr zu erkennen ist.

Ein Zeichner würde solch eine Szene sicher umgehen. Notfalls könnte er markante Teile des bewegten Objektes suchen, die vom verdeckenden Objekt deutlich entfernt sind und diese als Ansatz für die Bewegungslinien nutzen.

Eine ähnliche Rückfallposition bietet das Programm an. Es ist möglich Ansatzpunkte im vorderen Bereich des Objektes zu berechnen. Dabei muß dann allerdings vom Benutzer in Kauf genommen werden, daß alle Bewegungslinien das eigentliche Objekt schneiden. Dadurch ist die Bewegungsdarstellung gesichert.

Alternativ bietet es sich an, verdeckte Linien unterscheidbar anders als normale Bewegungslinien darzustellen. Dazu muß aber eine geeignete Darstellungsmethode

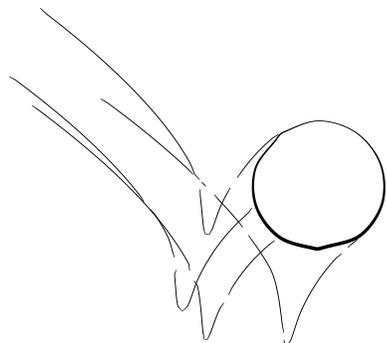
---

<sup>1</sup> relativ zur Bewegungsrichtung

gefunden werden, bei dem auch ein anderer Sichtbarkeitsalgorithmus notwendig ist, der die genauen Schnittpunkte mit anderen Objekten ermittelt.

Die beste Lösung eines solchen Problems ist aber die Wahl eines anderen Zeitpunktes für die Darstellung, bei dem das Objekt etwas weiter vom überdeckenden Objekt entfernt ist und der Einsatz von Bewegungslinien unproblematisch erscheint.

Ein weiteres Problem bei der Nutzung von Bewegungslinien sind komplexe Bewegungen, bei denen sich die Richtung extrem ändert. Dabei kommt es zur Überschneidung der Bewegungslinien untereinander, was einen Verlust der Übersichtlichkeit nach sich zieht. Abbildung 6.1 zeigt dies am Beispiel eines springenden Balls.



**Abbildung 6.1:** Bei einer plötzlichen Richtungsänderung kommt es zu Überschneidungen der Bewegungslinien. Die Übersichtlichkeit geht verloren.

Eine einfache Umgehung dieses Problems ist nicht möglich. Vielversprechend erscheint jedoch eine Aufteilung in zwei verschiedene Bewegungen und die Generierung separater Bewegungslinien für jede der Teilbewegungen. Als Trennstelle bietet sich der Frame an, in dem die Änderung der Bewegungsrichtung beginnt. Durch diese Methode entstehen zwar getrennte Linien, jedoch würden auch für die erste Teilbewegung sinnvolle Ansatzpunkte für die Bewegungslinien gefunden werden und somit ein harmonischerer Gesamteindruck entstehen. Die Überschneidungen würden nicht verschwinden, aber nicht mehr so störend wirken.

Ein bereits erwähntes Problem bei den Bewegungslinien ist der gewählte Sichtbarkeitsalgorithmus. Dadurch daß lediglich die Eckpunkte auf Sichtbarkeit kontrolliert werden, kommt es vor, daß kleinere Objekte fälschlich durch Linien überdeckt werden. Bei den getesteten Modellen tauchte dieser Fehler nur einmal bei einem durch einen Ring springenden Delphin auf. Da die Liniengraphiken aber nicht unbedingt Anspruch auf Exaktheit in der Darstellung erheben und die Fehler kaum auffallen, auf keinen Fall aber das Gesamtbild stören, ist ein spezieller Test auf diese Fehler und eine entsprechende Umgehung nicht unbedingt nötig. Trotzdem sei eine Lösung des Problems angeführt.

Bei Verwendung eines normalen Hidden-Line-Removal-Algorithmus können Schnittpunkte einer Teilstrecke zwischen zwei Punkten, die bereits als sichtbar bestimmt wurden, mit anderen Objekten erkannt werden. Existieren Schnittpunkte, werden die beiden Punkte nicht miteinander verbunden. Auf diese Weise bleibt auch wieder genügend Abstand zwischen den Bewegungslinien und den überdeckenden Objekten gewahrt.

### 6.1.2 Pfeile

Da der Algorithmus zur Generierung von Pfeilen auf dem der Bewegungslinien basiert, entstehen hier ähnliche Probleme, die nicht noch einmal aufgeführt werden sollen. Zusätzlich dazu gibt es jedoch noch weitere Auffälligkeiten.

Die Platzierung der Pfeile differiert von der, die ein Zeichner vornehmen würde. Der Algorithmus setzt Pfeile immer an einen zu dem Objekt gehörigen Eckpunkt. Ein Zeichner würde den Pfeil teilweise aber an einen vollkommen anderen Platz setzen, der nicht unbedingt mit dem Objekt selbst in Verbindung steht. Pfeile sind ein hoch stilisiertes graphisches Mittel und die Nutzung dem Betrachter sehr vertraut. Durch die vielen Vorschriften für technische Illustrationen ist der Betrachter an eine gewisse Anwendungsweise von Pfeilen gewöhnt. Der in dieser Arbeit entwickelte neue Ansatz zur Verwendung von Pfeilen wirkt daher teilweise ungewohnt. Trotzdem erfüllen die generierten Pfeile ihre Aufgabe vollständig.

Da es bei Rotationsbewegungen mit dem vorgestellten Algorithmus nicht möglich ist, die Pfeile gleichmäßig an das Objekt anzupassen, wurde zusätzlich die Möglichkeit geschaffen, den Algorithmus mehrfach mit verschiedenen Startwinkeln auszuführen. Dadurch werden mehr Pfeile gezeichnet und gleichmäßiger am Objekt platziert.

Für Translationen, die durch einen einzigen Pfeil dargestellt werden sollen, bietet sich alternativ die Möglichkeit an, den Pivotpunkt als Ausgangspunkt für Pfeildarstellungen zu nutzen. Dieser erscheint dazu teilweise geeigneter als ein Eckpunkt des Objektes.

### 6.1.3 Konturwiederholungen

Auch die Konturwiederholungen sind ein wichtiger Beitrag zur Visualisierung von Bewegungen. Prinzipiell kann der Einsatz von Konturwiederholungen bei Abschwächung älterer Konturen immer eingesetzt werden. Sie sind die allgemeinste Methode zur Bewegungsdarstellung, ihre Berechnung dauert aber auch deutlich länger als die von Bewegungslinien und Pfeilen. Dafür gibt es aber auch so gut wie keine Probleme bei ihrer Berechnung und Darstellung.

Einziges Manko ist der Algorithmus zur Generierung von Teilkonturen. Bei komplizierteren Objekten versagt der einfache Ansatz. Das hängt damit zusammen, daß nicht immer eine durchgehende Verbindung zwischen zwei Punkten über aneinandergesetzte Dreiecksflächen gegeben ist. Dies geschieht häufig bei aus mehreren Teilen zusammengesetzten Objekten, die sich teilweise durchdringen. In diesem Fall kann der Algorithmus die Abbruchbedingung nicht erfüllen und läuft in eine Endlosschleife. Wird durch den Algorithmus eine solche Rekursion erkannt<sup>2</sup>, bricht er die Teilkonturbildung ab. Somit eignet sich die Darstellung von Teilkonturen nur für einfache Objekte.

---

<sup>2</sup> Eine Rekursion tritt ein, sobald ein Eckpunkt ein zweites mal zur Teilkontur hinzugefügt wird.

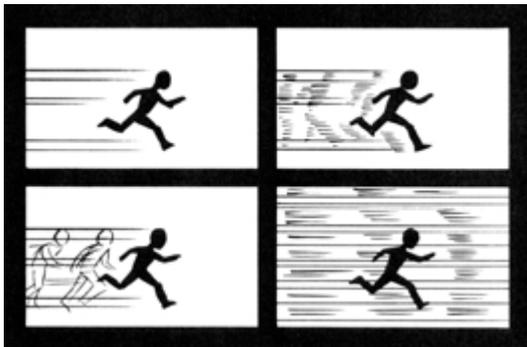
Um auch für kompliziertere Objekte den Algorithmus durchführen zu können, muß das entsprechende Objekt erst auf Durchdringungen durchsucht und die Faces neu berechnet werden, so daß sie den Voraussetzungen des Algorithmus entsprechen.

## 6.2 Weiterführende Betrachtungen

In diesem Abschnitt werden Ideen formuliert, die während der Arbeit entstanden sind, aber aus verschiedenen Gründen keine Berücksichtigung finden konnten. Sie sollen als Anregung für weiterführende Arbeiten gelten, die Funktionsvielfalt des Programms noch zu erweitern.

### 6.2.1 Darstellung von relativen Bewegungen

In dieser Arbeit wurden drei Methoden zur Darstellung von Bewegungen untersucht und computertechnisch umgesetzt. Eine weitere Methode wird von McCLOUD in [McC93] vorgestellt. Dabei wird der Betrachter in die Lage des bewegten Objektes versetzt. Der Bezugspunkt für die Bewegung ist nicht mehr ein feststehender Punkt im Weltkoordinatensystem sondern das bewegte Objekt an sich. Der Zeichner simuliert das Verschwimmen der Objekte indem er versucht sie aus andersartigen Bewegungslinien zusammensetzen. Abbildung 6.2 zeigt diese Methode rechts unten im Bild im Vergleich zu anderen Methoden, in Comics Bewegungen darzustellen.



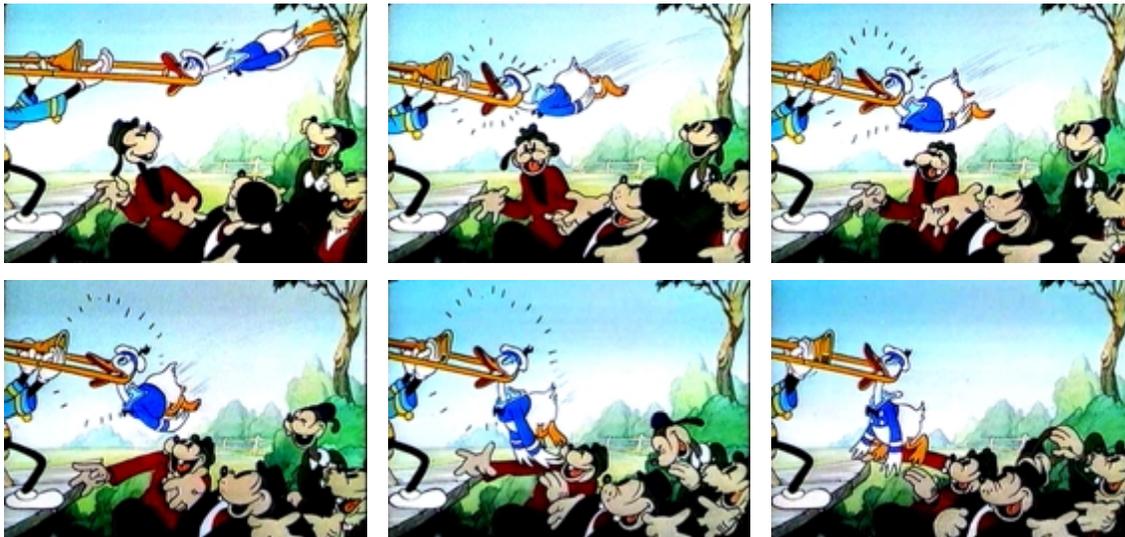
**Abbildung 6.2:** Beispiel für verschiedene Arten in Comics Bewegungen zu visualisieren. In: McCLOUD, Understanding Comics

Zur Umsetzung ist ein völlig neuer Algorithmus notwendig. Es wäre erforderlich, die Objektbewegung auf die nicht bewegten Objekte zu übertragen bzw. einen neuen Algorithmus zu entwickeln, der mit Relativbewegungen umgehen kann. Dadurch wären auch Darstellungen möglich bei denen Bewegungslinien für ein bewegtes Objekt relativ zur Kamerabewegung erzeugt werden. Nach der Entwicklung eines solchen Algorithmus' kann die Unschärfe der sich relativ bewegenden Objekte durch Darstellung der Objektkontur aus Bewegungslinien erreicht werden. Durch die Vorgabe der Liniendicke und der Abstände zwischen den Linien kann die Anzahl der pro Objekt benötigten Linien ermittelt werden. Für die Berechnung der jeweiligen Ansatzpunkte würde sich wieder eine Streifeneinteilung anbieten, wie sie in Abschnitt 4.2.3 vorgestellt wird.

Die Umsetzung dieser Methode stellt eine Alternative zu den vorgestellten Methoden dar. Sie wird sich wahrscheinlich nicht mit den anderen Methoden kombinieren lassen, da die Darstellung von Bewegungen die zu verschiedenen Relativpunkten Bezug nehmen in einem Bild nicht sinnvoll erscheint.

## 6.2.2 Nutzung der entwickelten Methoden in Animationen

Obwohl die vorgestellten Methoden für den Einsatz in Einzelbildern ausgelegt sind, ist ihre Anwendung für Animationen denkbar. In Zeichentrickfilmen ist es durchaus üblich, Bewegungslinien anzuwenden. Dies geschieht insbesondere bei sehr schnellen Bewegungen eines Objektes, das einen relativ großen Platz auf dem Bild einnimmt. Die Bewegungslinien helfen dabei dem Auge, die genaue Art der Bewegung zu erfassen, da sie aus den Einzelbildern selbst in manchen Fällen nicht ersichtlich ist. Als Beispiel sei ein Ausschnitt eines Trickfilmes von Walt Disney in Abbildung 6.3 angeführt. In sechs Bildern wird Donald Duck vom rechten Bildrand zur Bildmitte gezogen. In vier Bildern wird dabei die Zeichnung durch Bewegungslinien ergänzt.



**Abbildung 6.3:** Sechs aufeinanderfolgende Frames in einem Disney-Cartoon. Die Bewegungslinien treten nur bei schnellen Bewegungen der Hauptobjekte auf.

Bewegungslinien werden in Animationen demnach nur sehr gezielt eingesetzt. Um die erarbeiteten Methoden automatisch berechnen zu können, müssen die Situationen, in denen Zeichner eines Trickfilms auf Bewegungslinien zurückgreifen, genau analysiert werden, damit solche Situationen anhand der Animationsdaten erkannt und entsprechend der Einsatz von Bewegungslinien initiiert werden kann.

Da die Bewegungslinien in Trickfilmen nur über wenige aufeinanderfolgende Frames gezeichnet werden, wird die Berechnung eines Kontrollpunktes der Bewegungslinien pro Frame nicht ausreichend sein. Demnach muß der Algorithmus zur Kontrollpunkt-Berechnung verfeinert werden indem zusätzliche Kontrollpunkte zu Zeitpunkten zwischen zwei Frames ermittelt werden.

## 6.3 Fazit

Die in dieser Arbeit vorgestellten Methoden ermöglichen erstmalig für Liniengraphiken eine Bewegungsdarstellung in einem einzigen Bild. Diese Darstellungen werden durch computergraphische Mittel erzeugt. Ihr Erscheinungsbild ist durch den Benutzer konfigurierbar.

Die entwickelten Methoden stellen eine Basis für weiterführende Arbeiten dar. Der Einsatz der Methoden in weiteren Arbeiten am ISG ist bei Übergabe der erforderlichen Parameter und leichten Programm-Modifikationen möglich.

Ein Einsatz dieser Methoden in gängigen Renderingsystemen ist zwar möglich, jedoch kurzfristig nicht zu erwarten. Zu sehr werden photorealistische Darstellungsmethoden in diesen Systemen favorisiert.

Sollte die Erzeugung nicht-photorealistischer Graphiken einmal den gleichen Stellenwert wie die Erzeugung photorealistischer Graphiken erlangen, wird man um die Darstellung von Bewegungen auch in Einzelbildern nicht mehr umhinkommen. Bis dahin ist es aber noch ein langer Weg und ich bin stolz, mit dieser Arbeit den Grundstein dafür gelegt zu haben.



# Literaturverzeichnis

- [Bö87] BÖCKE, DOROTHEE: *Graphische Hilfsmittel beim Wissenserwerb*. Diplomarbeit, Universität Stuttgart, 1987.
- [Bru93] BRUGGER, RALF: *3D-Computergraphik und -animation. Mit einer Beschreibung des Autodesk 3D Studio*. Addison-Wesley, 1993.
- [FvDFH90] FOLEY, JAMES D., ANDRIES VAN DAM, STEVEN K. FEINER und JOHN F. HUGHES: *Computer Graphics. Principle and Practice*. Addison-Wesley, Reading, MA, 2. Auflage, 1990.
- [GSW96] GODENSCHWEGER, FRANK, THOMAS STROTHOTTE und HUBERT WAGENER: *Presentation of Freeform Surfaces as Line Drawings*. In: GIROD, B., H. NIEMANN und H.-P. SEIDEL (Herausgeber): *3D Image Analysis and Synthesis '96*, Seiten 87–93, Erlangen, Januar 1996. infix.
- [Hop98] HOPPE, AXEL: *Validierung und Nachbearbeitung von gerenderten Bildern*. Doktorarbeit, Otto-von-Guericke-Universität Magdeburg, 1998.
- [Imm95] IMMLER, CHRISTIAN: *3D Studio Version 4*. Data Becker, Paderborn, 1995.
- [KB83] KOREIN, JONATHAN und NORMAN BADLER: *Temporal Anti-Aliasing in Computer Generated Animation*. In: *Computer Graphics (SIGGRAPH '83 Proceedings)*, Band 17, Seiten 377–388, Juli 1983.
- [Koc84] KOCHANEK, D. H. U.: *Interpolating Splines with Local Tension, Continuity and Bias Control*. In: *Computer Graphics (SIGGRAPH '84 Proceedings)*, Seiten 33–41. ACM SIGGRAPH, 1984.
- [LD95] LASSETER, JOHN und STEVE DALY: *Toy Story: The Art and Making of the Animated Film*. Hyperion, New York, 1995.
- [Lex95] *Meyers kleines Lexikon*. Teil der Microsoft Lexirom, 1995.
- [LS95] LANSDOWN, JOHN und SIMON SCHOFIELD: *Expressive Rendering: A Review of Nonphotorealistic Techniques*. IEEE Computer Graphics and Applications, 15(3):29–37, Mai 1995.
- [McC93] MCCLOUD, SCOTT: *Understanding Comics - The Invisible Art*. HarperCollins Publishers, Inc. New York, 1993.

- [Mea92] MEALING, STUART: *The Art and Science of Computer Animation*. Intellect, Oxford, 1992.
- [MSS97] MASUCH, MAIC, STEFAN SCHLECHTWEG und BERT SCHÖNWÄLDER: *daLi! – Drawing Animated Lines!* In: OLIVER DEUSSEN, PETER LORENZ (Herausgeber): *Proceedings of Simulation und Animation '97*, Seiten 87–96, Delft, 1997. SCS Europe.
- [MSS98] MASUCH, MAIC, LARS SCHUMANN und STEFAN SCHLECHTWEG: *Animating Frame-to-Frame Consistent Line Drawings for Illustrative Purposes*. In: LORENZ, PETER und BERNHARD PREIM (Herausgeber): *Simulation und Animation '98*, Seiten 101–112, Erlangen, Ghent, 1998. SCS Europe.
- [Pat88] PATURI, FELIX R.: *Chronik der Technik*. Chronikverlag Dortmund, 1988.
- [PC83] POTMESIL, MICHAEL und INDRANIL CHAKRAVARTY: *Modeling Motion Blur in Computer-Generated Images*. In: *Computer Graphics (SIGGRAPH '83 Proceedings)*, Band 17, Seiten 389–399, Juli 1983.
- [Sch94] SCHNEIDER, STEVE: *That's all Folks*. Aurum Press, London, 1994.
- [Sch97a] SCHÖNWÄLDER, BERT: *Generierung charakteristischer Linien aus 3D Modellen*. Diplomarbeit, Otto-von-Guericke-Universität Magdeburg, 1997.
- [Sch97b] SCHUMANN, LARS: *Ein parametrisierbares Modell zur Darstellung von Linien*. Diplomarbeit, Otto-von-Guericke-Universität Magdeburg, 1997.
- [Sch98] SCHULZ, RONNY: *Untersuchungen zur Bewegungsunschärfe in hybriden Animationen*, 1998. Laborpraktikumsbericht, Otto-von-Guericke-Universität Magdeburg, ISG.
- [SPR<sup>+</sup>94] STROTHOTTE, THOMAS, BERNHARD PREIM, ANDREAS RAAB, JUTTA SCHUMANN und DAVID ROBERT FORSEY: *How to Render Frames and Influence People*. In: *Computer Graphics Forum*, Nummer 3, Seiten 455–466, Oxford, UK, 1994. Blackwell.
- [SS97] STROTHOTTE, CHRISTINE und THOMAS STROTHOTTE: *Seeing Between the Pixels: Pictures in Interactive Systems*. Springer-Verlag, Berlin, 1997.
- [SSSS98] SCHLECHTWEG, STEFAN, BERT SCHÖNWÄLDER, LARS SCHUMANN und THOMAS STROTHOTTE: *Surfaces to Lines: Generating Rich Line Drawings*. In: *Proceedings of the 6th International Conference on Computer Graphics and Visualization '98*, Band 2, Seiten 354–361, Plzeň, Februar 1998.

- [ST90] SAITO, TAKAFUMI und TOKIICHIRO TAKAHASHI: *Comprehensible Rendering of 3-D Shapes*. In: *Computer Graphics (SIGGRAPH '90 Proceedings)*, Seiten 197–206. ACM SIGGRAPH, 1990.
- [Str98] STROTHOTTE, THOMAS: *Computational Visualization: Graphics, Abstraction, and Interactivity*. Springer-Verlag Berlin, 1998.
- [TJ81] THOMAS, FRANK und OLLI JOHNSTON: *The Illusion of Life: Disney Animation*. Hyperion, New York, 1981.
- [vB95] VAN OVERVELD, KEES und BART BARENBRUG: *All You Need Is Force: a constraint-based approach for rigid body dynamics in computer animation*. In: *Computer Animation and Simulation '95*, Seiten 80–94, September 1995.
- [WA77] WEILER, K. und K. ATHERTON: *Hidden surface removal using polygon area sorting*. In: *Computer Graphics (SIGGRAPH '77 Proceedings)*, Computer Graphics Proceedings, Annual Conference Series, Seiten 214–222. ACM SIGGRAPH, 1977.
- [Web90] WEBER, ERNST A.: *Sehen, Gestalten und Fotografieren*. Birkhäuser Basel, Boston, Berlin, 1990.



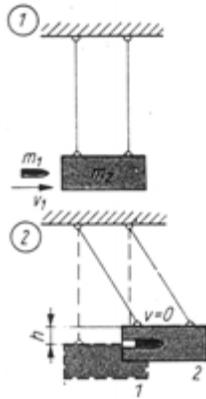
# A Bildersammlung

## A.1 Photographie



**Abbildung A.1:** Photo eines Artisten, In: Ernst A. WEBER, Sehen, Gestalten und Fotografieren, Basel, Boston, Berlin, Brinkhäuser, 1990

## A.2 Konturwiederholungen



**Abbildung A.2:** Anschauungsbilder zur Berechnung von Kräften, die bei dem Einschlag eines Projektils in einen hängenden Körper auftreten, In: Übungen zur Physik, VEB Fachbuchverlag Leipzig, 1986



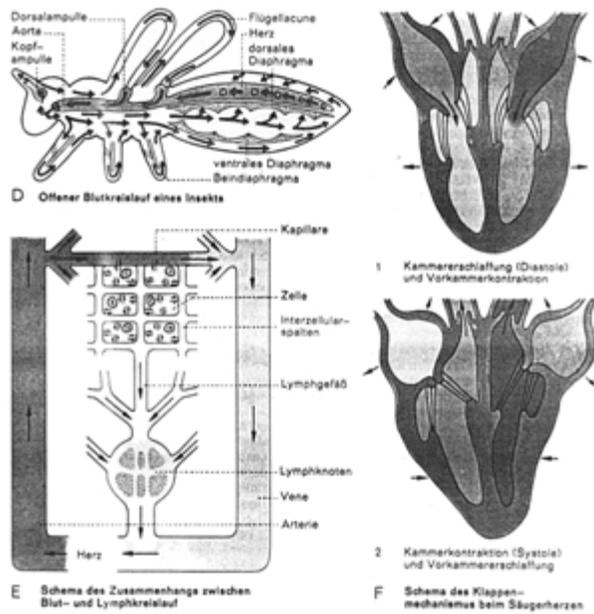
**Abbildung A.3:** Wild E. Coyote beim Rennen. In: That's all Folks, Steve SCHNEIDER, Aurum Press, 1994



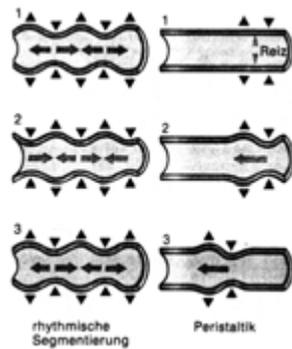
**Abbildung A.4:** Animationsstudie von Goofy. In: The Illusion of Life: Disney Animation, Hyperion, 1981

## A.3 Pfeile

### A.3.1 Wissenschaftliche Beispiele



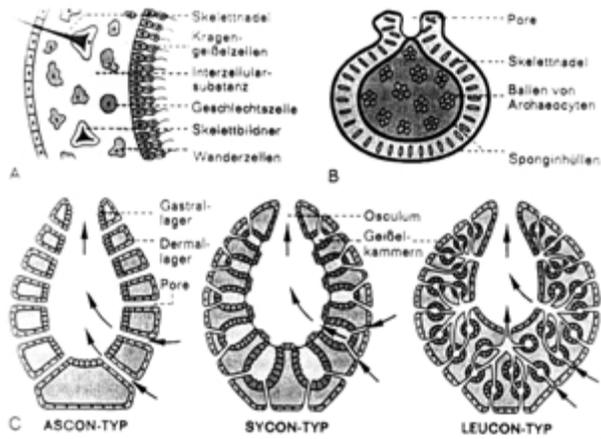
**Abbildung A.5:** Darstellung biologischer Abläufe, In: Günter VOGEL, Hartmut Angermann, Atlas zur Biologie, Deutscher Taschenbuchverlag GmbH & Co. KG, 1967, Seite 292



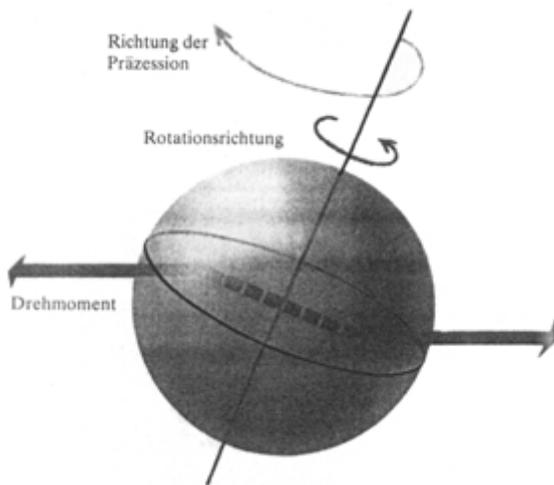
**Abbildung A.6:** Darstellung biologischer Abläufe, In: Günter VOGEL, Hartmut Angermann, Atlas zur Biologie, Deutscher Taschenbuchverlag GmbH & Co. KG, 1967, Seite 392



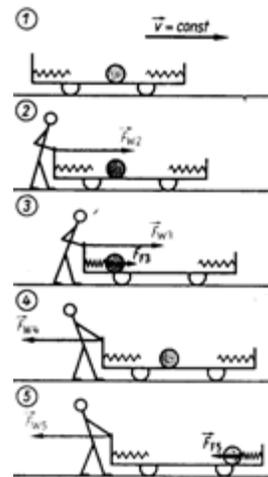
**Abbildung A.7:** Darstellung biologischer Abläufe, In: Günter VOGEL, Hartmut Angermann, Atlas zur Biologie, Deutscher Taschenbuchverlag GmbH & Co. KG, 1967, Seite 392



**Abbildung A.8:** Darstellung biologischer Abläufe, In: Günter VOGEL, Hartmut Angermann, Atlas zur Biologie, Deutscher Taschenbuchverlag GmbH & Co. KG, 1967, Seite 74



**Abbildung A.9:** Rotationsbewegung der Erde, In: Johannes F. GELLERT und Autoren, Die Erde – Sphären, Zonen, Regionen und Territorium, Urania Verlag, Leipzig, Jena, Berlin, 1982



**Abbildung A.10:** Darstellung der Bewegung zur näheren Erklärung einer physikalischen Übungsaufgabe, In: Physik – Fundament der Technik, VEB Fachbuchverlag Leipzig, 1987

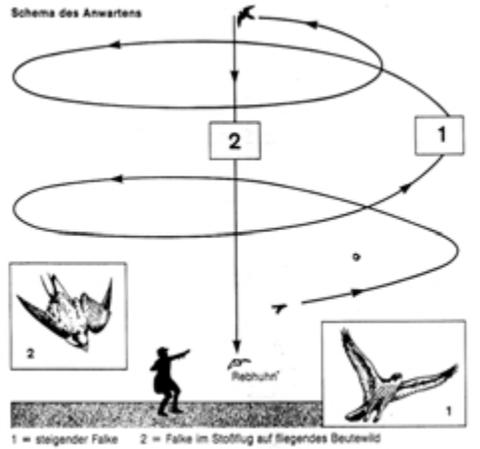


Abbildung A.11: Anwarten eines Falken, In: Karl BERRENS et al., Jagdlexikon, BLV Verlagsgesellschaft mbH München, 1994

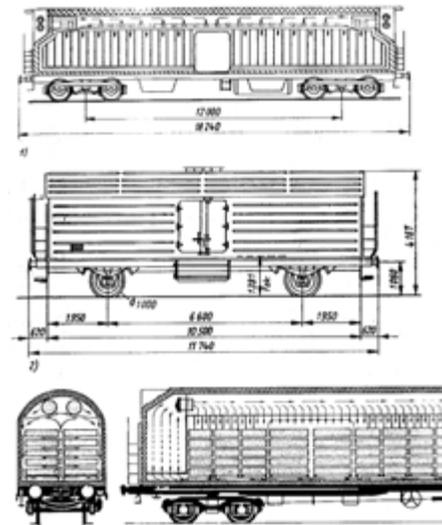


Abbildung A.12: Luftzirkulation in einem Kühlwagen, In: Gerhard ADLER et al., Lexikon der Eisenbahn, Stuttgart, Motorbuchverlag, 1990

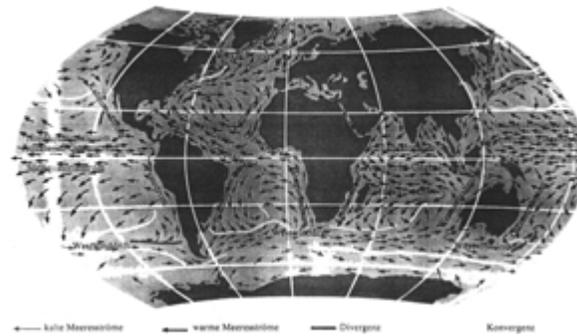


Abbildung A.13: Meeresströmungen der Erde, In: Johannes F. GELLERT und Autoren, Die Erdsphären, Zonen, Regionen und Territorium, Urania Verlag, Leipzig, Jena, Berlin, 1982

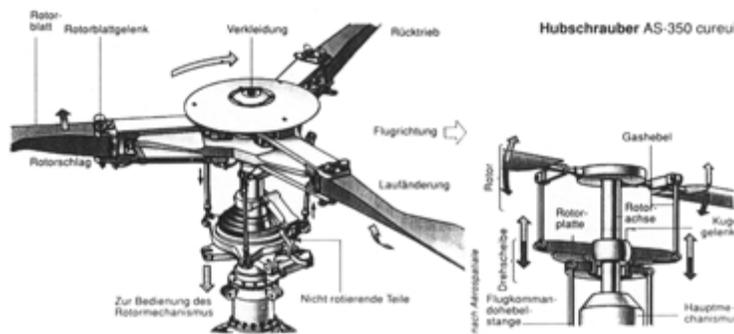
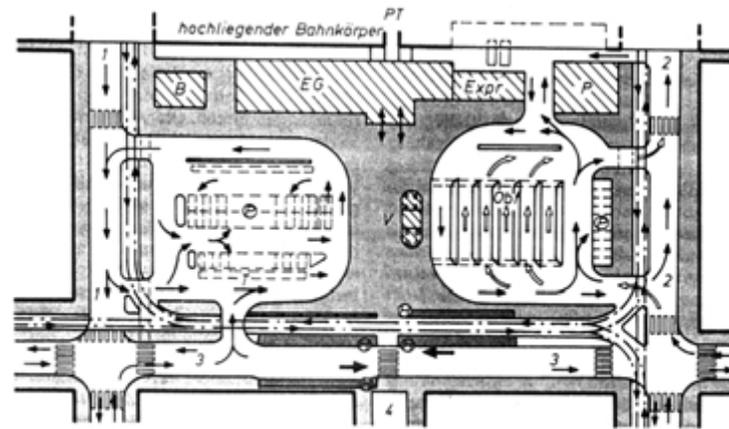


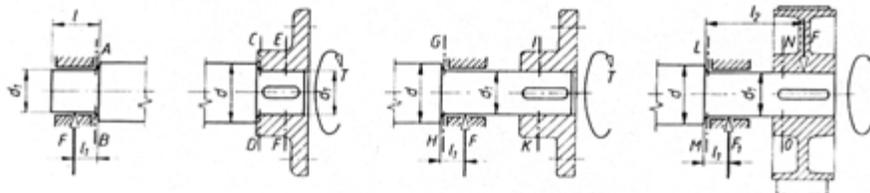
Abbildung A.14: Aufbau eines Rotors, In: Bilderlexikon der Technik · Instrumente · Apparate · Bauwerke, Bechtermünz Verlag GmbH, Eltville am Rhein, 1992



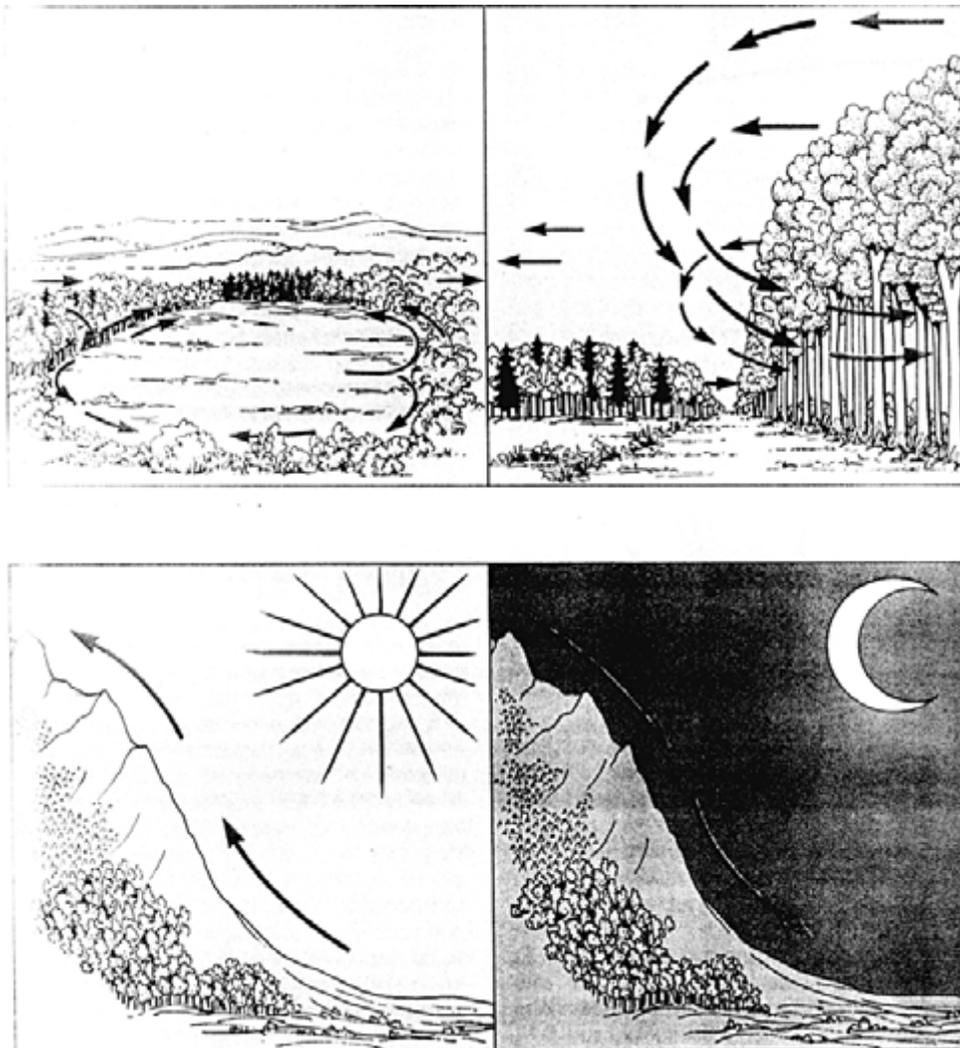


Fahrwege:  
 - - - Straßenbahn  
 → PKW, Taxen Exprefgüt - Kfz.  
 → Innerstädtische Omnibusse  
 → Omnibusse des Fernverkehrs

**Abbildung A.19:** Verkehrsanbindung eines Bahnhofes, In: Gerhard ADLER et al., Lexikon der Eisenbahn, Stuttgart, Motorbuchverlag, 1990

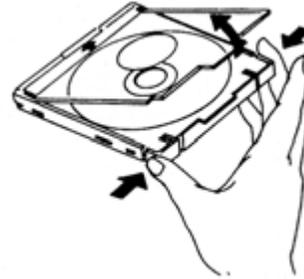
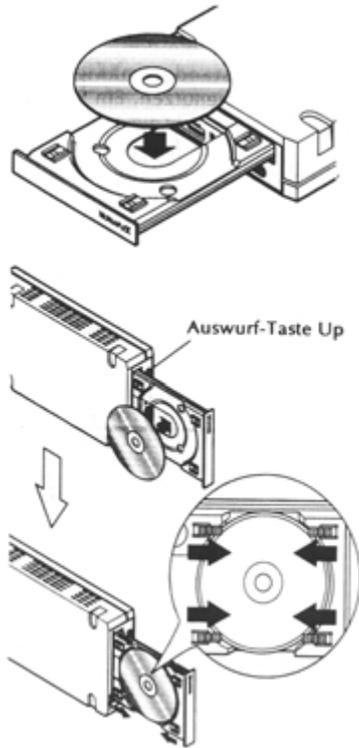


**Abbildung A.20:** technische Darstellung einer Welle, In: Physik-Fundament der Technik, VEB Fachbuchverlag Leipzig, 1987



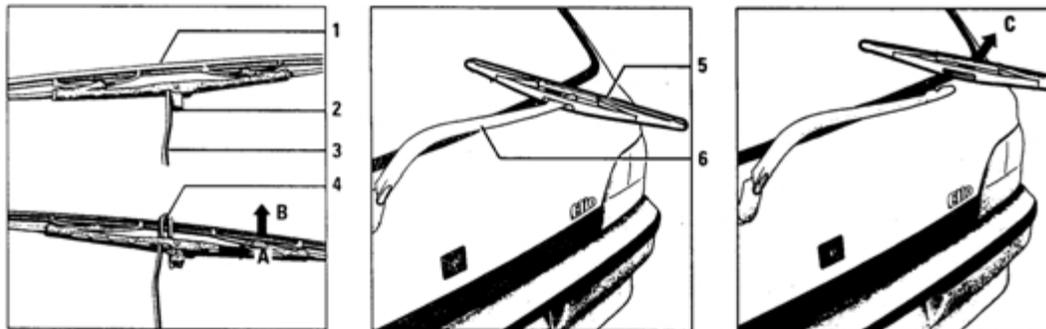
**Abbildung A.21:** Verschiedene Winde, In: Karl BERRENS et al, Jagdlexikon, BLV Verlagsgesellschaft mbH München, 1994

### A.3.2 Praktische Beispiele



**Abbildung A.22:** Handhabung eines Caddy, In: Plextor Ultraplex Bedienungsanleitung

**Abbildung A.23:** Handhabung des Plextor UltraPlex CD-ROM Laufwerks, In: Plextor Ultraplex Bedienungsanleitung

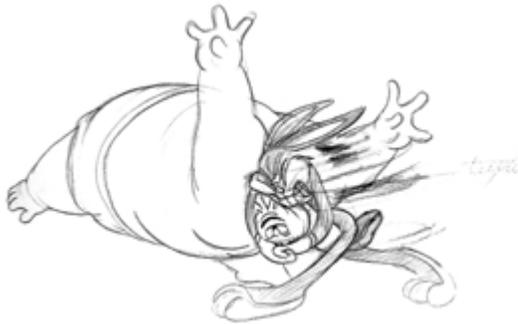


**Abbildung A.24:** Anleitung zum Wechseln von Scheibenwischern am Renault Clio, In: Bedienungsanleitung Renault Clio

## A.4 Speedlines



Abbildung A.25: Plakat aus dem Jahre 1930, In: Paolo MATRICARDI, Bilderlexikon der Flugzeuge – Geschichte der Luftfahrt, Südwest Verlag München, 1986



**Abbildung A.26:** Bugs Bunny im Kampf mit einem dicken Chinesen. In: That's All Folks, Aurum Press, 1994



**Abbildung A.27:** Bugs Bunny im Kampf mit einem dicken Chinesen. In: That's All Folks, Aurum Press, 1994



**Abbildung A.28:** Dumbo der fliegende Elephant. In: The Illusion Of Life: Disney Animation, Hyperion, 1981

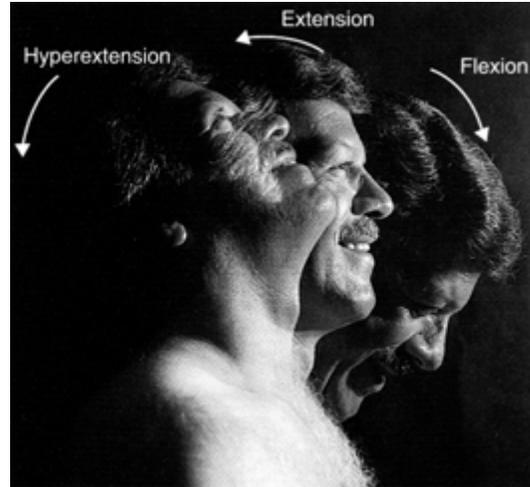


**Abbildung A.29:** Motorradrennen in einem japanischen Comic. In: Scott McCloud, Understanding Comics

## A.5 Kombinationen



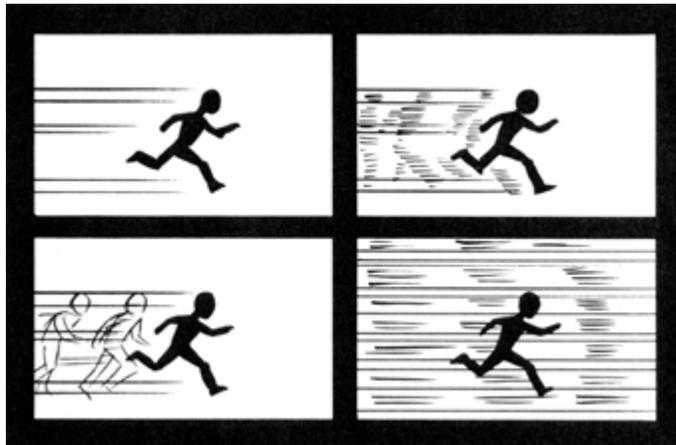
**Abbildung A.30:** Donald Duck fährt Auto. In: *The Illusion Of Life: Disney Animation*, Hyperion, 1981



**Abbildung A.31:** Bewegungsmöglichkeiten des Kopfes. In: *Introduction to the Human Body. The essentials of anatomy and physiology*, Addison Wesley, Longman, 1996

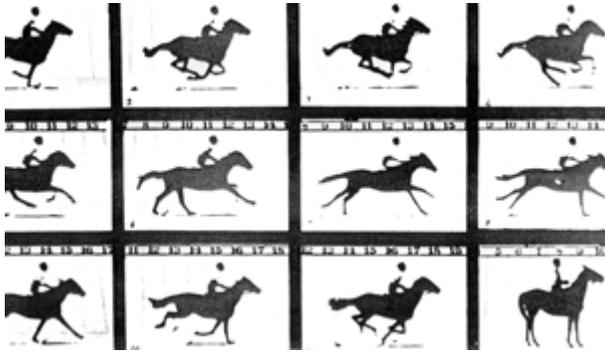


**Abbildung A.32:** Karrikatur von Larson In: *Larson Calendar* 1998



**Abbildung A.33:** Beispiel für verschiedene Arten Bewegungen zu visualisieren. In: *McCLOUD, Understanding Comics*

## A.6 Andere Methoden



**Abbildung A.34:** Bewegungsstudien eines galoppierenden Pferdes. Die Bewegung wird durch mehrere zeitlich aufeinanderfolgende Bilder dargestellt. In: Felix R. PATURI, Chronik der Technik, Chronikverlag, Dortmund, 1988



**Abbildung A.35:** Meteorologische Darstellung von Wetterfronten. Durch Spezialsymbole werden die Bewegungen der Wetterfronten dargestellt. In: Johannes F. Gellert und Autoren, Die Erde – Sphären, Zonen, Regionen und Territorium, Urania Verlag, Leipzig, Jena, Berlin, 1982



**Abbildung A.36:** Unterteilung eines Bildes in 3 Teile. Jeder Teil stellt einen anderen Zeitpunkt dar. In: XENA Comic 3, Dino Comics, November 1998.



# Thesen zur Diplomarbeit

- Die Analyse von Bewegungsdarstellungen in traditionell erstellten Bildern liefert stilistische Mittel, die in Liniengraphiken einsetzbar sind.
- Bewegungsdarstellungen erweitern das Bild um zusätzliche Informationen. Sie vermitteln einen dynamischen Kontext.
- Bewegungsdarstellungen sind in computergenerierte Liniengraphiken bisher nicht integriert worden.
- Die Einbeziehung von einem Bild zugrundeliegenden Geometrie- und Animationsdaten stellt eine gute Ausgangsbasis zur Berechnung dieser stilistischen Mittel dar.
- Bewegungslinien und Konturwiederholungen eignen sich zur Darstellung vergangener Bewegungen während Bewegungspfeile zukünftige Bewegungen andeuten.
- Der kombinierte Einsatz verschiedener Stilmittel kann den Eindruck von Bewegung beim Betrachter verstärken.
- Es ist möglich, die Attribute einiger der verwendeten Stilmittel zur Abbildung weiterer Informationen zu nutzen.

