Freeform Deformation

Karl Lueger, klueger@cosy.sbg.ac.at
Norbert Martinek, norbert.martinek@sbg.ac.at
Michael Noisternig, michael.noisternig@sbg.ac.at

18. Juni 2002

FFD - Freeform Deformation

Ein generelles geometrisches Verfahren zur **Simulation von Deformationsvorgängen** an festen Körpern bildet die sogenannte *Freeform Deformation* (FFD). Die FFD ist ein von der Repräsentationsart des Objektes unabhängig arbeitendes Verfahren zur Berechnung von Objektdeformation und wurde von T.W. Sederberg (USA) entwickelt. Diese Eigenschaft ermöglicht prinzipiell die Verwendung von Objekten die sowohl **parametrisch** als auch **polygonal** definiert werden.

Vorgehen bei der Freeform Deformation

Eine wesentliche Eigenschaft der FFD ist, dass nicht das Objekt selbst deformiert wird. Die Deformation des Objekts ergibt sich aus der **Deformation des das Objekt umgebenden Raumes**. Die Objekte werden dann analog zu dieser Deformation neu berechnet.

Beispiel (Sederberg):
Deformiert man einen Würfel aus verformbaren, durchsichtigen Plastik, in dem ein (ebenfalls verformbares) Objekt enthalten ist: Verformt man das Stück Plastik, verformt sich auch das eingeschlossene Objekt.

Die Grundidee ist also, das zu deformierende, flexible Objekt mit einem geometrisch einfach zu beschreibenden Volumen zu umhüllen, in dem jeder Objektpunkt eine eindeutige, parametrisch bestimmbare Position besitzt. Nach Deformation des umgebenden Volumens kann über die Kenntnis der Deformationsparameter sowie der initialen Parametrisierung der Objektpunkte auf deren neue Position im Raum geschlossen werden. Anstelle der direkten Verformung von Objekten erfolgt also die Verformung des Raumes, der diese Objekte umgibt.

Der von Sederberg vorgeschlagene **Hüllkörper** wird in allgemeiner Form durch einen Spat repräsentiert, wobei zur Vereinfachung stets ein Quader, als Spezialfall des Spats, verwendet wird. Durch diesen Quader lässt sich ein lokales kartesisches Koordinatensystem definieren, dessen Ursprung X_0 in einem Eckpunkt liegt und dessen Koordinatenachsen durch die anliegenden Quaderkanten gegeben sind (siehe Abbildung 1). Das Quadervolumen bildet den **Deformationsbereich** als sogenannten **FFD-Block**, über den sich Objektpunktkoordinaten leicht in lokale Koordinaten überführen lassen.

Ein FFD-Block wird nun in mehrere Teilvolumen (den sogenannten **Hyperpatches**) gegliedert, wobei diese zusätzliche Unterteilung die lokalen Ausmaße der Deformation auf die Objekte im FFD-Block vorgibt. Dabei ergibt sich, als wesentliche Komponente der Freeform Deformation, eine initiale **Gitterstruktur** (*lattice*) (siehe Abbildung 1). Ein solches *FFD-Gitter* besteht aus Gitterknoten, -kanten, -flächen und -zellen:

- **Gitterknoten:** Kreuzungspunkt von Gitterlinien
- **Gitterkante:** Kürzeste Verbindung zwischen zwei Gitterknoten
- **Gitterfläche:** Geschlossene Verbindung einer minimalen Anzahl von Gitterkanten

- **Gitterzelle:** Region im Gitter, die vollständig von Gitterflächen umgeben ist

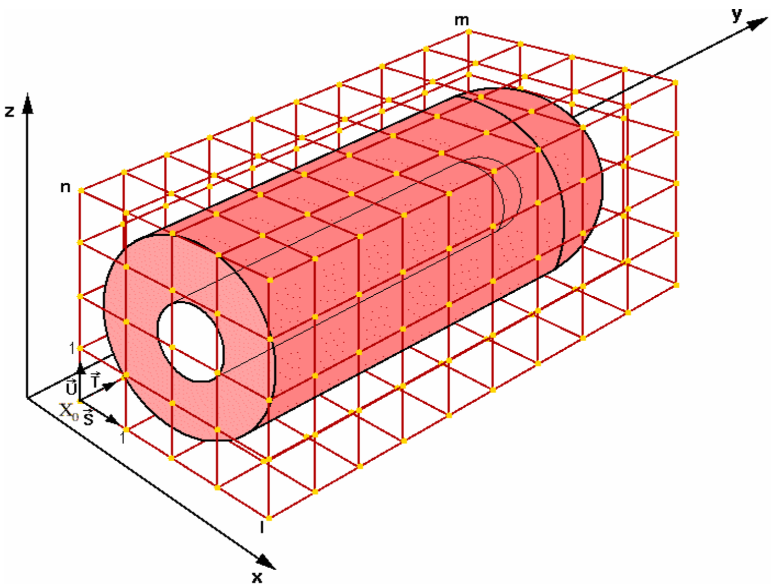

Abbildung 1: FFD-Block zu einem Objekt

Ein *kartesisches* FFD-Gitter besteht demzufolge aus sechsseitigen Gitterzellen, wobei jede Seite durch eine rechteckige Gitterfläche mit vier Gitterkanten und vier Gitterknoten beschrieben wird und jeder Gitterknoten maximal mit sechs Nachbarknoten verbunden ist. Eine Knotenverschiebung bewirkt eine Änderung aller Gitterkanten, die diesen Knoten gemeinsam haben. Die Größe der Gitterzellen bzw. der Diskretisierungsgrad bestimmen im Wesentlichen, wie genau eine Deformation modelliert werden kann.

Definition eines Hyperpatches

Die Definition eines Hyperpatches lässt sich aus dem 1-Dimensionalen anhand der Definition von **Bézier-Kurven** anschaulich ableiten.

Kurven werden im allgemeinen konstruiert, indem man einige Punkte der Kurve, die Kontrollpunkte, festlegt und die Zwischenwerte durch Interpolation ermittelt. Bei einfacher polynomischer Interpolation entstehen jedoch bei einer größeren Anzahl an Kontrollpunkten starke Oszillationen. Unterteilt man die Kontrollpunkte in Mengen von wenigeren so ist eine kontinuierliche Steigung an den Übergängen nicht mehr gegeben.

Bézier-Kurven interpolieren neben den Kontrollpunkten auch die Steigungen zwischen diesen, sodass die Kurve vollständig differenzierbar ist. Der erste und letzte Kontrollpunkt werden als Start- und Endpunkt der Kurve verwendet, die anderen Punkte steuern den Steigungsverlauf der Kurve. Dabei bewegt sich die Kurve nur annähernd auf diese Punkte zu und berührt sie im allgemeinen nicht. Bézier-Kurven bilden somit ein einfaches und intuitives Mittel zur Modellierung von Kurven.

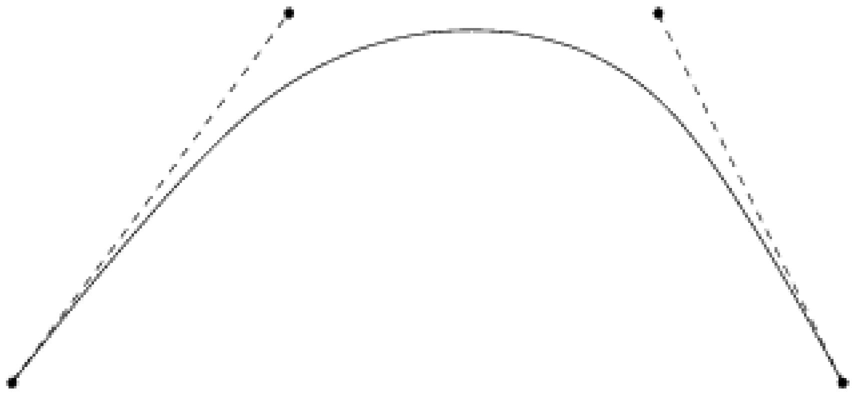

Abbildung 2: Bézierkurve mit 4 Kontrollpunkten und Tangenten

Bei $n+1$ Kontrollpunkten p_i ergibt sich eine Bézier-Kurve vom Grad n mit folgender Formel:

$$Q(s) = \sum_{i=0}^{n} p_i B_{i,n}(s); \qquad 0 \leq s \leq 1$$

Die $B_{i,n}(s)$ sind **Bernsteinpolynome**, welche die Form

$$B_{i,n} = \binom{n}{i} s^i (1-s)^{(n-i)}; \qquad 0 \leq s \leq 1$$

haben. Diese Polynome besitzen die Eigenschaft, dass sie immer zwischen den Werten 0 und 1 liegen und ihre Summe über alle i stets 1 ergibt. Durch diese Eigenschaft ergibt sich die weitere besondere Eigenschaft von Bézier-Kurven, dass diese sich immer innerhalb der konvexen Hülle der Kontrollpunkte befinden.

$$\sum_{i=0}^{n} B_{i,n}(s) = 1; \qquad 0 \leq B_{i,n} \leq 1$$

Diese Formel kann nun einfach für den 2-dimensionalen Raum auf folgende Art erweitert werden:

$$Q(s,t) = \sum_{i=0}^{n} \sum_{j=0}^{n} p_{ij} B_{i,n}(s) B_{j,n}(t); \qquad 0 \leq s,t \leq 1$$

Durch die Anordnung der Kontrollpunkte ergibt sich nun ein Gitter, das eine Fläche, den sogenannten **Patch**, beschreibt. Durch Versatz dieser Kontrollpunkte werden diese Patches verändert.

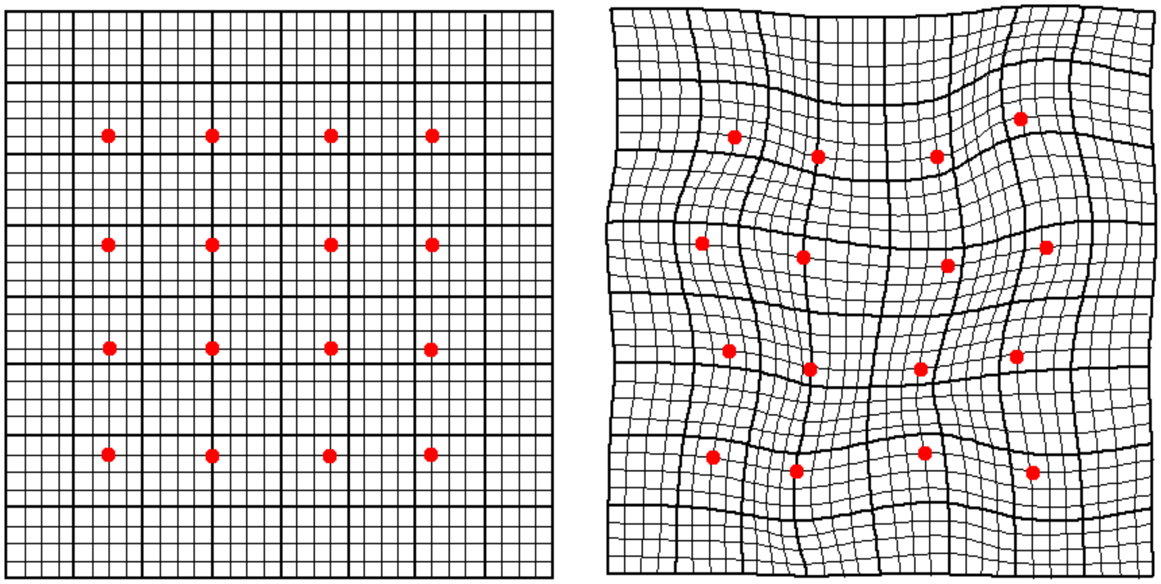

Abbildung 3: 2D-Patch mit Verformung

Durch die Berücksichtigung einer weiteren Dimension ergibt sich nun ein Hyperpatch:

$$Q(s,t,u) = \sum_{i=0}^{n}\sum_{j=0}^{n}\sum_{k=0}^{n} p_{ijk} B_{i,n}(s) B_{j,n}(t) B_{k,n}(u); \qquad 0 \leq s,t,u \leq 1$$

Deformation des Objekts

Nach der Freeform Deformation von Sederberg werden für jede Dimension des Hyperpatches 4 Kontrollpunkte verwendet. In den folgenden Erklärungen wird jedoch allgemeiner angenommen, dass jede Dimension $n+1$ Kontrollpunkte enthält.

Durch folgende drei Schritte lassen sich nun Objekte im FFD-Block lokalisieren und verformen:

1. **Bestimmung der Position der Stützpunkte des Objektes im FFD-Block**
   Es wird ein lokales Koordinatensystem für den FFD-Block definiert. Dieses Koordinatensystem wird i.A. durch drei achsenparallele Vektoren $\vec{S}, \vec{T}, \vec{U}$ aufgespannt (siehe Abbildung 4). Die Menge aller Punkte des FFD-Blocks lassen sich definieren

   $$\vec{X}(s,t,u) = \vec{X}_0 + s\vec{S} + t\vec{T} + u\vec{U}; \qquad s,t,u \in [0,1]$$

   Die Koordinaten der Kontrollpunkte (im Objektkoordinatensystem) berechnen sich

   $$p_{ijk} = \vec{X}_0 + \frac{i}{n}\vec{S} + \frac{j}{n}\vec{T} + \frac{k}{n}\vec{U}; \qquad n+1 \text{ ... Anzahl der Kontrollpunkte}$$

   Die Berechnung der Koordinaten der Objektpunkte im FFD-Block erfolgt durch eine Translation um $-\vec{X}_0$ und eine Skalierung, beschrieben durch F:

$$F = \begin{pmatrix} \frac{1}{||\vec{S}||} & 0 & 0 \\ 0 & \frac{1}{||\vec{T}||} & 0 \\ 0 & 0 & \frac{1}{||\vec{U}||} \end{pmatrix}$$

$$\begin{pmatrix} s \\ t \\ u \end{pmatrix} = \left[\begin{pmatrix} x \\ y \\ z \end{pmatrix} - \vec{X}_0 \right] F$$

Vorraussetzung ist dabei, dass das zu deformierende Objekt vollständig in den FFD-Block paßt; $s, t, u \in [0, 1]$. Damit sind die Koordinaten der Objektpunkte im FFD-Block bis auf Translation um $-\vec{X}_0$ und Skalierung um F den Objektkoordinaten gleich. Diese Vereinfachung gilt aber nur, wenn $\vec{S}, \vec{T}$ und $\vec{U}$ achsenparallel und paarweise orthogonal sind. Bei der *Extended Freefrom Deformation* sind diese Voraussetzungen nicht gegeben (siehe später).

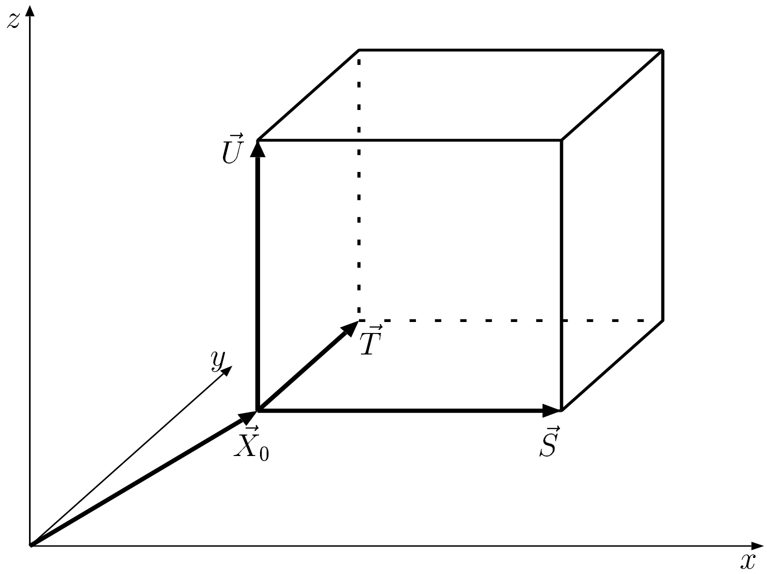

Abbildung 4: Lattice-space im Objektspace

2. **Verformung des FDD-Blocks**
   Durch die Verschiebung der Kontrollpunkte p_{ijk} lässt sich der Raum verformen. Nach der Deformation unterscheiden sich der Objektraum und der FFD-Block. Daher müssen die Objektpunkte nun vom FFD-Block in den Objektraum über die Gleichung

$$Q(s,t,u) = \sum_{i=0}^{n} \sum_{j=0}^{n} \sum_{k=0}^{n} p_{ijk} B_i(s) B_j(t) B_k(u) \tag{1}$$

umgerechnet werden, um die Deformation am Objekt im Objektraum sichtbar zu machen. Entscheidend dabei ist, dass sich die FFD-Block-Koordinaten (s, t, u) eines Punktes bei einer Deformation nicht ändern, die Objektkoordinaten aber entsprechend der p_{ijk} verändert werden müssen.

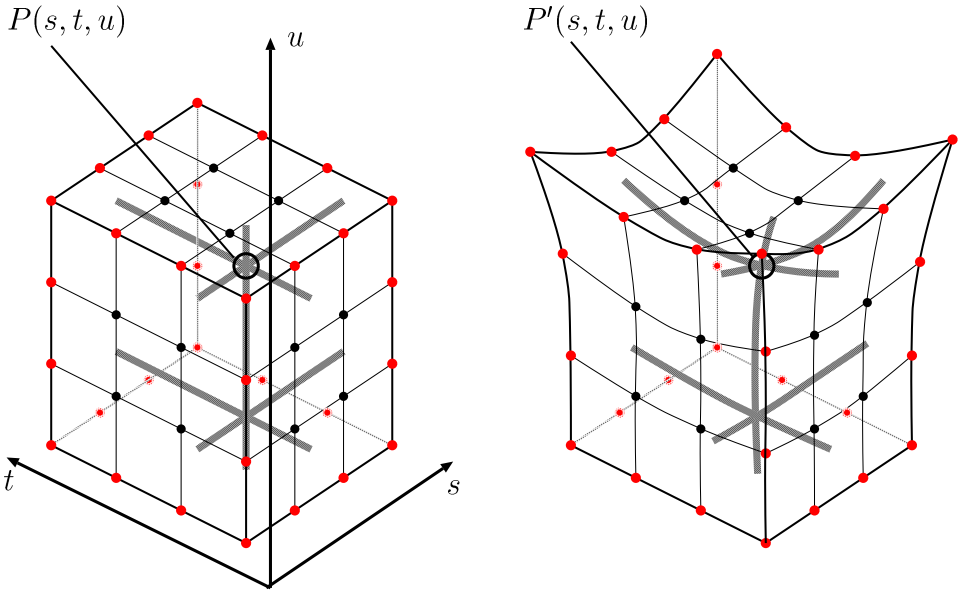

Abbildung 5: Versatz der Kontrollpunkte im 3D

3. **Rücktransformation der veränderten Objektpunkte**
   Zuletzt müssen die veränderten Objektpunkte noch in das ursprüngliche Koordinatensystem rücktransformiert werden. D.h. die Translation um $\vec{X}_0$ und die Skalierung F müssen umgekehrt werden.

Vor- und Nachteile der Freeform Deformation

Vorteile: Der Vorteil dieses Verfahrens ist die leichte Berechenbarkeit der Deformation. Daraus resultiert eine hohe Berechnungsgeschwindigkeit, wenn eine polygonale Objektrepräsentation gewählt wurde.

Nachteile: Der Nachteil dieses Verfahrens ist, dass es sich nicht um eine Berechnung der Deformation nach physikalischen Gesichtspunkten handelt, sondern um eine mathematische Modellierung, die spezifische Objekteigenschaften nicht berücksichtigt.

Weitere Nachteile der FFD sind:

- Die exakte Form ist schwer zu erreichen.
- Eine exakte Verformung einzelner Objektpunkte ist schwer zu erreichen
- Probleme, sich Hyperpatches vorzustellen und gezielt zu manipulieren.

Matrixschreibweise

Angenommen ein Objekt habe k Objektpunkte und der FFD-Block n Kontrollpunkte. Bestimmt man für jeden Objektpunkt seine FFD-Block-Koordinaten, erhält man k Vektoren (s_i, t_i, u_i) mit $i = 0...k-1$. Für die Objektkoordinaten gilt

$$\begin{pmatrix} x_i \\ y_i \\ z_i \end{pmatrix} = \vec{p}_{000} B_0(s_i) B_0(t_i) B_0(u_i) + ... + \vec{p}_{n-1,n-1,n-1} B_{n-1}(s_i) B_{n-1}(t_i) B_{n-1}(u_i)$$

oder kürzer

$$\vec{q_i} = \begin{pmatrix} x_i \\ y_i \\ z_i \end{pmatrix} = \vec{b_i} P$$

wobei hier P die n^3 x 3 Matrix mit den Zeilen $\vec{p}_{000}$ bis $\vec{p}_{n-1,n-1,n-1}$ und $\vec{b_i}$ die 1 x n^3 Matrix mit den Elementen $[B_0(s_i)B_0(t_i)B_0(u_i)]$ bis $[B_{n-1}(s_i)B_{n-1}(t_i)B_{n-1}(u_i)]$ ist. Über alle k Objektpunkte ergibt sich somit eine k x n^3 Matrix B derart

$$B = \begin{pmatrix} \vec{b}_0 \\ \vec{b}_1 \\ ... \\ \vec{b}_{k-1} \end{pmatrix}$$

$$= \begin{pmatrix} B_0(s_0)B_0(t_0)B_0(u_0) & \cdots & B_{n-1}(s_0)B_{n-1}(t_0)B_{n-1}(u_0) \\ \vdots & \ddots & \vdots \\ B_0(s_{k-1})B_0(t_{k-1})B_0(u_{k-1}) & \cdots & B_{n-1}(s_{k-1})B_{n-1}(t_{k-1})B_{n-1}(u_{k-1}) \end{pmatrix}$$

Nun kann man die Koordinaten aller k Objektpunkte in einer k x 3 Matrix Q mit den Zeilen $\vec{q}_0$ bis $\vec{q}_{k-1}$ berechnen über

$$Q = BP \tag{2}$$

Aus der Repräsentation der FFD in Matrixschreibweise wird folgendes ersichtlich:

- die Elemente der Matrix B ändern sich bei der Berechnung nicht und können deshalb im Voraus berechnet werden.
- durch Invertierung der Matrix B kann aus gegebenen versetzten Objektpunkten auf die nötigen Kontrollpunkte geschlossen werden.

Matrixinversion

Angenommen die Kontrollpunkte P verschieben sich um ΔP. Es gilt nach Formel (2):

$$\Delta Q = B\Delta P \tag{3}$$

Mit Formel (3) kann man bei gegebenem Versatz der Kontrollpunkte des FFD-Blocks mit einer einfachen Matrixmultiplikation den Versatz der Objektpunkte berechnen. Ein Ansatz von John Hughes und Henry Kaufman behandelt die Umkehrung dieser Berechnung. Die Pseudoinverse unter Berücksichtigung der Beziehung $B^{-1}B = E$ folgt aus Formel (3) sehr schnell

$$\begin{aligned} B^{-1}\Delta Q &= B^{-1}B\Delta P \\ B^{-1}\Delta Q &= E\Delta P \\ \Delta P &= B^{-1}\Delta Q \end{aligned} \tag{4}$$

Mit Gleichung (4) kann bei gegebenem Versatz der Objektpunkte ΔQ der Versatz der Kontrollpunkte - und somit die neuen Kontrollpunktkoordinaten - berechnet werden. Leider ist i.A. die Matrix B^{-1} nicht eindeutig, da das System unterdeterminiert ist. Nun hat die Pseudoinverse B^+ der Matrix B die Eigenschaft, dass $\Delta P_0 = B^+\Delta Q$ eine minimale fehlerhafte Lösung $(B\Delta P_0 - \Delta Q)$ für sehr kleine ΔP_0 ist. Wählt man also anstelle von B^{-1} die Pseudoinverse B^+ so zeigen Hughes und Kaufman, dass der so berechnete Kontrollpunktversatzvektor das erwünschte Ergebnis mit minimalen Kontrollpunktversatz löst. Mit Hilfe dieses Ansatzes ist eine Translation der Objektpunkte möglich, die sich dann deformierend auf den gesamten Körper auswirkt. Der Nachteil der komplizierten Manipulation des Objektes über die Kontrollpunktkoordinaten kann mit dieser Methode beseitigt werden. Desweiteren kann die Pseudoinverse B^+ ebenso wie B vorberechnet werden.

EFFD - Extended Freeform Deformation

Bei der **Extended Freeform Deformation** (EFFD) (nach Coquillart) wird der quaderförmige Deformationsbereich, welcher Einschränkungen der Deformation mit sich bringt, durch andere Gitterstrukturen verfeinert. Die EFFD erlaubt die Verwendung unterschiedlicher Gittertopologien wie z.B. prismatische oder zylindrische Hüllobjekte. Durch diese Erweiterung lässt sich das Verfahren besser an beabsichtigte Verfahren anpassen und ermöglicht die Modellierung von leichten Oberflächendeformationen, wobei es gegenüber der herkömmlichen FFD-Methode allerdings an Flexibilität und Stabilität verliert.

Die EFFD arbeitet nach dem selben Prinzip wie die FFD, mit dem einzigen Unterschied, dass ein Deformationsbereich in Form von Hüllobjekten beschrieben werden kann, der der beabsichtigten Deformation am ehesten entspricht. Mehrere seperat erzeugte Hüllobjekte können miteinander kombiniert werden.

Die Zuordnung von Kontrollpunkten zu einem Hyperpatch des EFFD-Gitters erfordert ein numerisches Iterationsverfahren, das zum einen zeitaufwendiger ist und zum anderen nicht unbedingt konvergiert. Die Bestimmung aller Kontrollpunkte in der zugrunde liegenden Gitterstruktur muss dabei jedoch nur einmal erfolgen.

Sowohl bei der FFD-Methode als auch bei der EFFD-Methode gestaltet sich die gezielte Deformation eines Objektes über die Kontrollpunkte recht schwierig. Für eine intuitive Deformation eignen sich diese Verfahren somit schlecht. Es gibt Erweiterungen von Hughes und Kaufmann welche eine direkte Manipulation von Objekten erlauben. Diese Verfeinerungen sind jedoch nicht mehr Gegenstand dieser Arbeit und werden aus diesem Grund nicht näher erläutert.

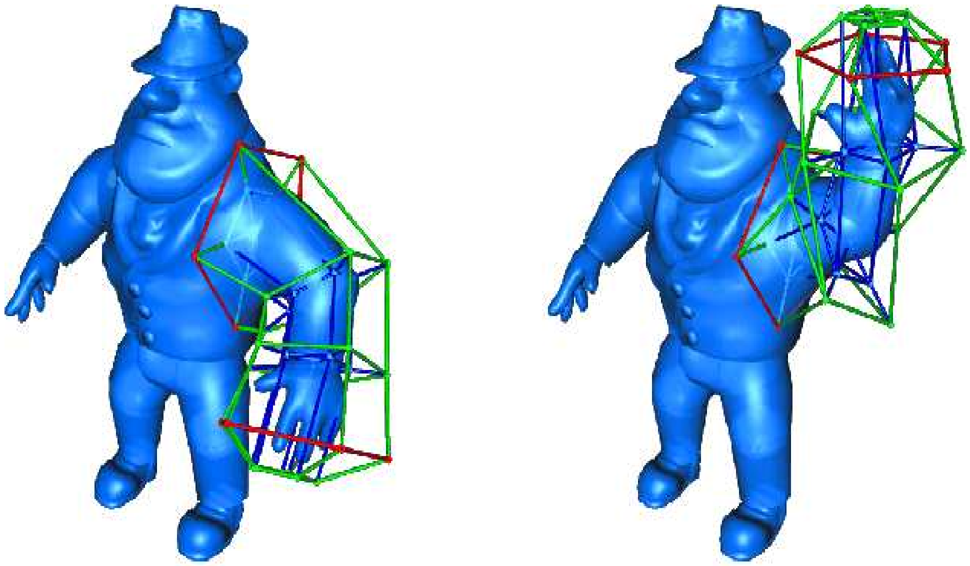

Abbildung 6: EFFD-Beispiel aus der Animation

Quellenangabe

- Ralf Herbich, Eric Heymann. *Animation von flexiblen Objekten*, 1996.
- Stefan Zachow. *Simulation von Deformation und Destruktion*, http://www.tfh-berlin.de/~stevie/mod+sim/index.html.