Bauhaus Universität Weimar
Fakultät Medien

Dokumentation des Forschungsprojekts

Bahnberechnung von Gasblasen in zähen Flüssigkeiten

Uwe Hahne 992268
Martin Koske 20024

25. Februar 2005

Inhaltsverzeichnis

1 Einleitung

Dieses Projekt ist eine Zusammenarbeit mit Prof. Dr. L. Traversoni (UAM, Mexico City), der ein Forschungslabor betreibt, in dem die Reinigung von verschmutztem Öl mit Hilfe aufsteigender Gasblasen untersucht wird. Beim Aufsteigen der Gasblasen bleibt die Verunreinigung an der Hülle der Blase hängen und wird so an die Oberfläche transportiert. Hier greift das Prinzip der Oberflächenvergrößerung: viele kleine Blasen können mehr Schmutz aufnehmen als wenige große mit dem selben Gesamtvolumen. Wiederholt man den Vorgang sammelt sich der komplette Schmutz an der Oberfläche der Flüssigkeit. Jetzt kann die Schmutzschicht einfach abgeschöpft werden und man braucht keine teuren Ölfilter. Dafür ist es wichtig zu wissen, wieviele Blasen auf einer Fläche aufsteigen können, ohne dass sie sich zu nahe kommen. Wenn zwei Blasen sich annähern, werden sie zu einer Blase und die Reinigungswirkung wird verringert. Weiterhin wird untersucht wie groß eine Blase sein muss, damit sie die größte Menge an Verunreinigungen aufnehmen kann.
Für den Versuchsaufbau wird ein quaderförmiges Bassin mit 1 m Kantenlänge benutzt. Am Boden des Bassins befindet sich im Schnitt der Diagonalen eine kleine Öffnung, aus der eine Gasblase aufsteigt, die von zwei Kameras gefilmt wird. Die beiden Kameras sind jeweils wieder im Schnitt der Diagonalen auf zwei aneinandergrenzenden Seitenflächen positioniert. Wenn eine Blase in Z-Richtung aufsteigt, starten beide Kameras synchron die Aufnahme. Die Abbildung 1.1 verdeutlicht nochmal den Aufbau des Versuchs.

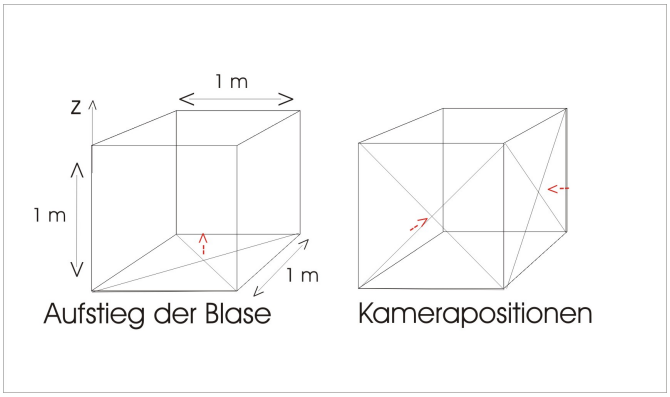


Abbildung 1.1. Versuchsaufbau.

2 Durchführung

2.1 Videoverarbeitung

Aus den vorliegenden Aufzeichnungen der beiden Kameras werden zuerst die einzelnen Bilder extrahiert. In den Bildern kann das Objekt lokalisiert und deren Position bestimmt werden. Anhand der Pixelkoordinaten ist eine Bestimmung der Position auf den Bildebenen der Kameras möglich. Die dazugehörigen Raum-Koordinaten können durch die bekannte Aufnahmesituation ausgerechnet werden. Die Abbildung 2.1 verdeutlicht die Vorgehensweise der Videobearbeitung.

Abbildung 2.1. Schematische Darstellung der Rekonstruktion der Raumkoordinaten

Die Blase ist auf dem Foto zwar eindeutig erkennbar (siehe Abbildung 2.2), jedoch läßt sich die Zugehörigkeit eines beliebigen Bildpunktes zum Objekt nicht eindeutig bestimmen. Die Ursache dafür liegt in der nur unwesentlichen Unterscheidung der Grauwerte der Blase und des Hintergrundes. Im Originalbild ist daher keine exakte Entscheidung hinsichtlich der Umrissbestimmung der Blase zu treffen.

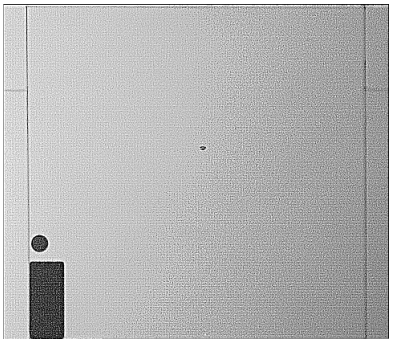
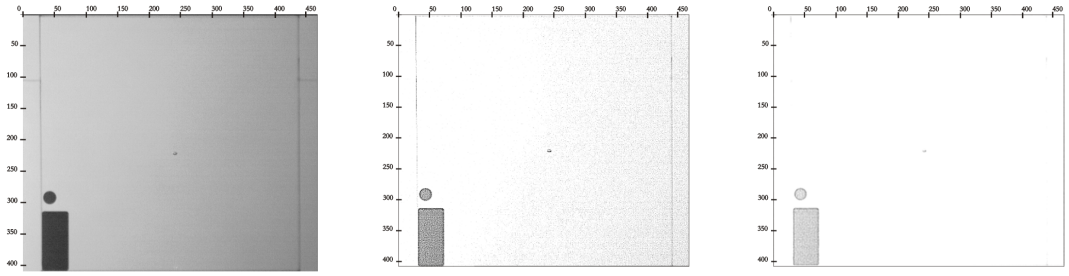


Abbildung 2.2. Ein extrahiertes Bild aus dem Video.

Das Ziel ist nun eine „Bildverbesserung“ dahingehend zu erreichen, dass eine klare Abgrenzung zwischen dem Umriss der Blase und dem Bildhintergrund möglich ist. Desweiteren soll die Bildmanipulation in Hinblick auf eine automatische Abarbeitung aller Frames des Videos einen einfarbigen Hintergrund liefern, so dass die Blase freigestellt werden kann. Dabei ist die Bildinformation so zu manipulieren, dass ein möglichst geringer Informationsverlust entsteht. Alle Bildpunkte der Blase sollen erhalten bleiben, aber dennoch muss eine klare Objektidentifizierung möglich sein.
Für den gesamten Prozess der Bildpunktbestimmung der Blase sind zwei Filteroperationen notwendig. Mit der ersten Operation wird der Kontrast im Bild verstärkt und die Anwendung der zweiten zieht eine Bildglättung nach sich. Die Zwischenschritte bei der Bearbeitung des Bildes sind in Abbildung 2.3 zusammengefasst. Auf das Ursprungbild wird ein Hochpassfilter angewandt und durch zusätzliche Addition eines Offsets von 105 aufgehellt 2.3 b. Anschließend erfolgt eine Tiefpassfilterung und nochmalige Addition von 55 Helligkeitsstufen 2.3 c.



(a) Originalbild (b) Hochpassfilterung (c) Tiefpassfilterung

Abbildung 2.3. Anwendung lokaler Bildoperationen.

2.2 Lokalisation der Blase

Bei der 3D-Rekonstruktion gehen wir von einem idealisierten Lochkameramodell aus, wie in Abbildung 2.4 zu sehen. Durch die bekannte geometrische Beziehung der beiden Kameras im Raum, ihren intrinsischen Parametern (Öffnungswinkel, Brennweite und Seitenverhältnisse) und die bekannten Bildpunkte ist es möglich die dreidimensionalen Koordinaten der Blase im Raum herzuleiten.

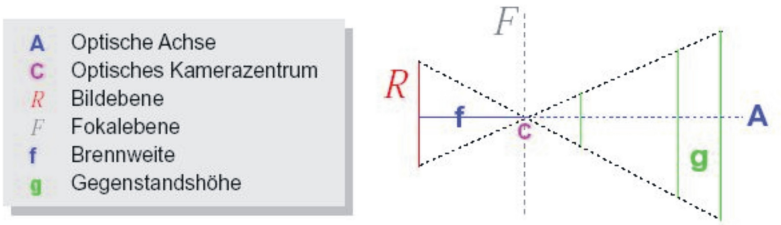


Abbildung 2.4. Lochkameramodell

Das Problem ist es anhand von Pixelkoordinaten eines bestimmten Punktes seine Position im Raum zu berechnen. Man legt zuerst ein Weltkoordinatensystem fest und ermittelt die Position und Ausrichtung der Kameras (extrinsische Parameter) in diesem. Der nächste Schritt ist die intrinsischen Parameter der Kamera herauszufinden, also die Brennweite und den Öffnungswinkel, um damit die Lage des aufgenommenen Kamerabildes im Raum zu bestimmen. Nun legt man jeweils eine Gerade von den Kamerazentren durch die entsprechenden Bildpunkte. Der Schnittpunkt dieser Geraden entspricht dem Raumpunkt der Blase. Im folgenden wird dieses Verfahren genauer beschrieben.
Die Bildebene und die Position der Blase kann über trigonometrische Beziehungen bestimmt werden. Wir legen jeweils eine Gerade durch Kamerazentrum und Bildpunkt und suchen deren Schnittpunkt.

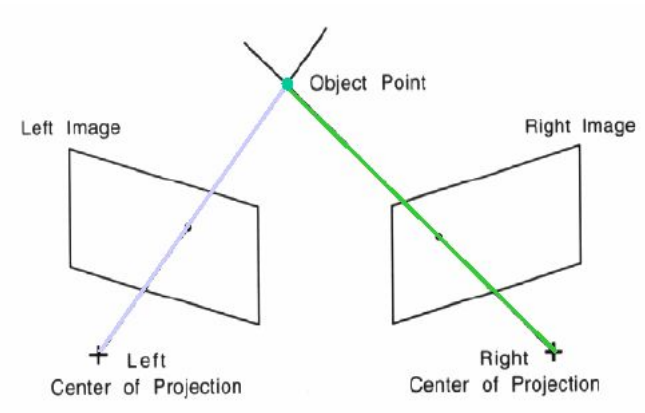


Abbildung 2.5. Schnittpunkt der Geraden.

Den Schnittpunkt erhält man durch Gleichsetzen der Geradengleichungen. Da sich die Geraden aufgrund von Ungenauigkeiten, die durch unser Schätzungen oder durch numerische Rundungen entstehen, manchmal nicht schneiden, suchen wir nach dem Ort des minimalen Abstands der Geraden. Die lineare Algebra liefert nur eine unbefriedigende Lösung, die den Abstand der Geraden, jedoch nicht die Punkte minimalen Abstands liefert. Also stellt man eine Gleichung auf, die den Abstand der Geraden im Quadrat in Abhängigkeit zweier Parameter bzw. zweier Punkte auf den Geraden liefert.

$$l^2 = (x_1 + pa_x - x_2 - qb_x)^2 + (y_1 + pa_y - y_2 - qb_y)^2 + (z_1 + pa_z - z_2 - qb_z)^2$$

Diese von zwei Parametern abhängige Gleichung wird nach jedem Parameter differenziert. Diese Ableitungen werden gleich null gesetzt.

$$\frac{dl^2(p,q)}{dp} = 0 \qquad \text{und} \qquad \frac{dl^2(p,q)}{dq} = 0$$

Daraus ergeben sich zwei Gleichungen mit zwei Unbekannten, die wir eindeutig lösen können. Die Lösungen der Gleichungen ergeben die Parameter, die die Punkte minimalen Abstands der beiden Geraden bestimmen. Als Raumpunkt wählen wir einfach den Punkt, der in der Mitte dieser beiden Punkte liegt.

2.3 Bézier-Spline

Pierre Bézier hat diese Art von Spline für die Firma Renault entwickelt, wo es bei der Entwicklung von Karosserien mit geringem Luftwiderstand zum Einsatz kam. Wir haben uns diese Art von Spline ausgesucht, weil die einzelnen Bézier-Segmente glatt zusammen gefügt werden können. Sie interpolieren aus allen unseren gegebenen Punkten eine zweimal stetig differenzierbare Kurve. Aus den gegebenen Gewichtspunkten können die Bézier-Punkte für jedes Intervall direkt bestimmt werden, womit die gesamte Approximationskurve festgelegt ist. Außerdem beeinflußt die Änderung oder das Hinzufügen von Gewichtspunkten nur wenige lokale Kurvensegmente. Man verhindert starkes Oszilieren der Kurve. Eine Polynomkurve

$$b(t) = \sum_{i=0}^{n} b_i * B_{i,n}(t) \tag{2.1}$$

heißt Bézierkurve vom Grad n über $\Delta = [s,t]$ mit dem allgemeinen Bernstein-Polynom $B_{i,n}(t)$. Für $n = 3$ berechnen sich die Bernstein-Polynome aus:

$$\begin{aligned} B_{0,3}(t) &= (1-t)^3 \\ B_{1,3}(t) &= 3t * (1-t)^2 \\ B_{2,3}(t) &= 3t^2 * (1-t) \\ B_{3,3}(t) &= t^3 \end{aligned} \tag{2.2}$$

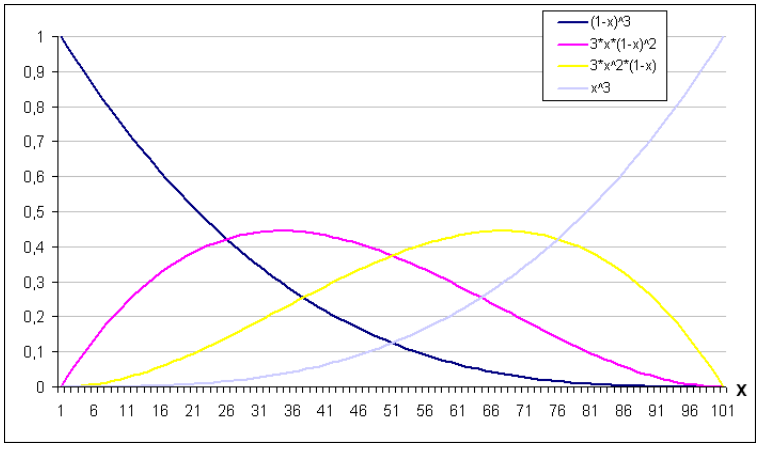


Abbildung 2.6. Bernsteinpolynome.

Die Punkte b_i heißen Kontroll- oder Bézierpunkte, das von den Kontrollpunkten gebildete Polygon heißt Kontrollpolygon. Die Bézierkurve wird vom Kontrollpolygon gesteuert und liegt in dessen konvexer Hülle.

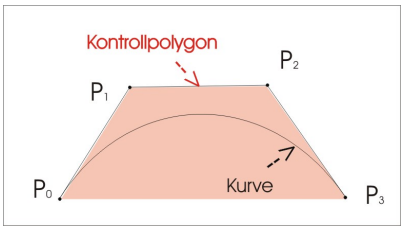


Abbildung 2.7. Kontrollpolygon.

Wir benutzen für unser Problem ein Polynom 3. Grades mit t ∈[0,1]:

$$b(t) = b_0 * (1-t)^3 + b_1 * 3t * (1-t)^2 + b_2 * 3t^2 * (1-t) + b_3 * t^3 \tag{2.3}$$

Die Punkte b_0 und b_3 werden interpoliert und die beiden inneren Punkte werden aproximiert. Die Vektoren $\overrightarrow{b_0 b_1}$ und $\overrightarrow{b_2 b_3}$ beschreiben die Tangente an der Kurve im Anfangs- bzw. Endpunkt.
Wenn $t = 0$ ist, erhält man als Ergebnis vom Polynom den Punkt b_0 und wenn $t = 1$ ist erhält man den Punkt b_3. Die Punkte b_0 und b_3 werden interpoliert. Das sind auch die Punkte die aus den zwei Videos extrahiert worden. Der Punkt b_3 des i-ten Polynoms ist der Punkt b_0 des $i+1$-ten Polynoms.

2.3.1 Hilfspunkte P_1 und P_2

Für unser Problem ist es wichtig, dass alle Punkte der Blase interpoliert werden. Da bei kubischen Bézier-Splines nur die Punkte P_0 und P_3 interpoliert werden und die anderen approximiert, brauchen wir eine Lösung für die fehlenden Hilfspunkte die aproximiert werden. Eine Möglichkeit wäre die ersten vier von unseren Punkten der Blase zu nehmen, diese durch einen Bézier-Spline darzustellen und dann die nächsten vier Punkte. Wir möchten aber alle Punkte interpolieren also probieren wir es über die erste Ableitung bei $t = 0$ und $t = 1$.

Berechnung:
Gesucht: b_1 und b_2 Gegeben: b_0 und b_3

$$b(t) = b_0 * (1-t)^3 + b_1 * 3t * (1-t)^2 + b_2 * 3t^2 * (1-t) + b_3 * t^3 \tag{2.4}$$

1. Ableitung

$$b'(t) = b_0 * (-3*t^2 + 6*t - 3) + b_1 * (9*t^2 - 12*t + 3) + b_2 * (6*t - 9*t^2) + b_3 * (3*t^2) \tag{2.5}$$

Für die erste Ableitung 0 und 1 einsetzen, um den Anstieg an den jeweiligen Punkten zu berechnen.
$t = 0$

$$b'(0) = b_0 * (-3) + b_1 * (+3) = 3 * (b_1 - b_0) \tag{2.6}$$

$t = 1$

$$b'(1) = b_2 * (-3) + b_3 * (3) = 3 * (b_3 - b_2) \tag{2.7}$$

Beide Gleichungen nach den gegeben Punkten umstellen.

$$b_1 = \frac{1}{3} * b'(0) + b_0 \tag{2.8}$$

$$b_2 = b_0 - \frac{1}{3} * b'(1) \tag{2.9}$$

Für die Berechnung der ersten Ableitung benutzen wir den zentralen Differenzenquotienten, weil der Fehler beim vorwärtigen oder rückwertigen Differenzenquotienten größer ist. Allerdings müssen wir am Anfangs- und Endpunkt den vor- bzw. rückwärtigen verwenden, da es die Punkte x_{-1} und x_{n+1} nicht gibt. Mit h = Δt entsteht

$$f'(x) = h \overset{lim}{\rightarrow} 0 \frac{f(x_0 + h) - f(x_0 - h)}{2 * h} \tag{2.10}$$

So erhalten wir die fehlenden Hilfspunkte um den Bézier-Spline zu konstruieren.

2.3.2 Geometrischer Zugang

Eine geometrische Konstruktion für die Kurvenpunkte einer Kurve 3.Grades mit den Kontrollpunkten P_0, P_1, P_2 und P_3, entsteht durch Unterteilung der Strecken $\overrightarrow{P_0P_1}$, $\overrightarrow{P_1P_2}$, $\overrightarrow{P_2P_3}$ im Verhältnis $t : (1-t)t \in [0,1]$. Man erhält drei Hilfspunkte $H_{1,1}$, $H_{1,2}$ und $H_{1,3}$. Die sich ergebenden Strecken $\overrightarrow{H_{2,1}H_{2,2}}$ und $\overrightarrow{H_{2,2}H_{2,3}}$ werden wiederum im Verhältnis $t : (1-t)$ geteilt. Durch erneutes Unterteilen der Strecke im bekannten Verhältnis erhält man den Kurvenpunkt C_t. Der Vorgang lässt sich leicht anhand der nachfolgenden Abbildung erklären.

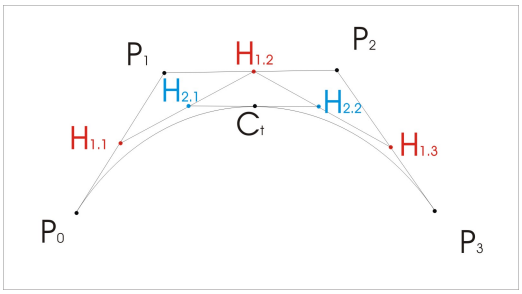


Abbildung 2.8. Unterteilung der Strecken.

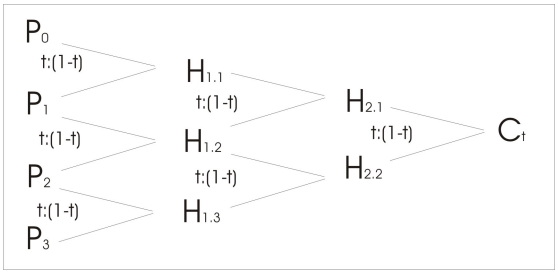


Abbildung 2.9. Der Algorithmus von de Casteljau.

Daraus lassen sich folgende Formeln ableiten.

$$\begin{aligned} H_{1,1} &= (1-t) * P_0 + t * P_1 \\ H_{1,2} &= (1-t) * P_1 + t * P_2 \\ H_{1,3} &= (1-t) * P_2 + t * P_3 \\ H_{2,1} &= (1-t) * H_{1,1} + t * H_{1,2} \\ H_{2,2} &= (1-t) * H_{1,2} + t * H_{1,3} \\ B_t &= (1-t) * H_{2,1} + t * H_{2,2} \end{aligned} \tag{2.11}$$

Wenn t das Intervall $[0, 1]$ durchläuft, erhalten wir die Werte für die Position eines Punktes auf $B(t)$.

2.3.3 Kopplung von Bézier Kurven

Da die Steigung eines Bézier-Splines am Anfangs- und Endpunkt bekannt ist (Steigung der Geraden vom ersten zum 2. Kontrollpunkt und Steigung der Geraden vom vorletzten zum letzten Kontrollpunkt), können zwei Bézier-Splines sehr einfach glatt (d.h. ohne Knick, erste Ableitungen sind gleich) zusammengefügt werden. Dazu muss der Anfangspunkt des $i+1$-ten Splines identisch mit dem Endpunkt des i-ten Splines sein. Außerdem muss der Kontrollpunkt P_1 des $i+1$-ten Splines mit dem Kontrollpunkt P_2 des i-ten Splines und dem Kopplungspunkt auf einer Geraden liegen.

2.4 Quaternionen

Der Ansatz von [LeTr03] sowie [WWW5] beschreibt die Bewegung der Blase als Überlagerungen von Translationen und Rotationen. Die Rotationen werden durch Quaternionen beschrieben. Jede Quaternion $q = (w, x, y, z)$ mit $|q| = 1$ beschreibt eine Rotation. Die Rotation um einen beliebigen normierten Vektor $\vec{v} = (x, y, z)^T$ wird durch die Quaternion $q = (cos(\frac{\varphi}{2}), xsin(\frac{\varphi}{2}), ysin(\frac{\varphi}{2}), zsin(\frac{\varphi}{2}))$ beschrieben.
Ein Punkt $P = (x_0, y_0, z_0)$ wird durch die Quaternion $p = (0, x_0, y_0, z_0)$ repräsentiert. Der rotierte Punkt $\tilde{p}$ ergibt sich aus

$$\tilde{p} = qpq' \tag{2.12}$$

wobei q' die konjugierte Quaternion darstellt.
In der Computergrafik werden Quaternionen eingesetzt um Rotationen auszudrücken. Sie haben nicht nur den Vorteil, dass es schneller zu berechnen ist als mit einer herkömmlichen Rotationsmatrix, sondern bei einer Interpolation des Rotationsweges, wie es bei Animationen häufig vorkommt, bleibt die Winkelgeschwindigkeit konstant. Man nennt dieses Verfahren Sphärische lineare Interpolation (SLERP). Im folgenden wird das Prinzip kurz erläutert.
Gegeben seien zwei Einheitsquaternionen q_1 und q_2 und ein Parameter $t \in [0, 1]$. Es gilt

$$\begin{aligned} slerp(q_1, q_2, t) &= (q_2 q_1^{-1})^t q_1 \\ &= q_1 \frac{sin((1-t)\varphi)}{sin\varphi} + q_2 \frac{sin(t\varphi)}{sin\varphi} \end{aligned}$$

mit $cos(\varphi) = q_1 \bullet q_2 = w_1 w_2 + x_1 x_2 + y_1 y_2 + z_1 z_2$
wobei $\bullet$ für das innere Produkt steht.
Die Funktion $slerp$ berechnet für $t \in [0, 1]$ die kürzeste Verbindung (Großkreis) auf der vierdimensionalen Einheitssphare zwischen q_1 und q_2. Dies ist für die Interpolation (Animation) von Orientierungen von Körpern ideal geeignet. Allerdings ist es nicht sehr gut für die Orientierung der Kamera geeignet, da sich der Up-Vektor der Kamera verändern kann. Dies würde zu einer Rotation entlang der Sichtachse führen, die in der Regel nicht erwünscht ist.
Für uns ergab sich der Ansatz die Rotationen zwischen den Stützstellen zu interpolieren, aber die Winkelgeschwindigkeiten sind nicht immer gleich, sondern springen, sind also nicht stetig. Dies könnte man mit Hilfe kubischer Splines umgehen, aber die sphärische Entsprechung ist notwendig und diese ist viel komplexer als herkömmliche Splines im euklidischen 3D-Raum. Dies führte zur Beendigung diesen Ansatz weiter zu verfolgen.

2.5 Differentialgeometrie

Der nachfolgenden Einführung in die Theorie der Kurven liegen [BuHaWi90], [WWW2] und [Gr94] zugrunde.
Es gibt mehrere Möglichkeiten eine Kurve im Raum zu definieren. Eine Kurve läßt sich als Bahn eines sich stetig bewegenden Punktes vorstellen. Die Funktion

$$\vec{r}(t) = \begin{pmatrix} x(t) \\ y(t) \\ z(t) \end{pmatrix}$$

wird Parameterdarstellung der Kurve K

$$K = \{(x(t), y(t), z(t)) : t \in [a, b]\}$$

genannt. Sie ordnet jedem Zeitpunkt t einen Ortsvektor $\vec{r}(t)$ zu. Dabei heißt t mit $a \leq t \leq b$ Parameter und $[a, b]$ Parameterintervall. Wenn es ein Intervall $I = (a, b)$ gibt, auf dem folgende Bedingungen erfüllt sind:

1. $\vec{r}(t)$ is für alle $t \in I$ mindestens einmal stetig differenzierbar,
2. für alle $t \in I$ gilt $\left|\vec{\dot{r}}(t)\right| = \left|\frac{d\vec{r}}{dt}\right| \neq \vec{0}$,

wird von einer regulären Darstellung der Raumkurve gesprochen. Ist eine Kurve glatt, d. h. sie besitzt eine reguläre Parameterdarstellung, so bedeutet die zweite Bedingung, dass nicht alle Ableitungen $\dot{x}(t), \dot{y}(t), \dot{z}(t)$ gleichzeitig Null sind und immer

$$(\dot{x}(t))^2 + (\dot{y}(t))^2 + (\dot{z}(t))^2 > 0 \qquad \text{gilt.}$$

Nachdem eine allgemeine Definitionen für Kurven im Raum bestimmt ist werden im Anschluss charakteristische Eigenschaften von Raumkurven erläutert. Eine der wichtigsten geometrischen Größe einer Kurve ist deren Länge. Für die Bogenlänge gilt:

$$L = \int_a^b \sqrt{\dot{x}(t)^2 + \dot{y}(t)^2 + \dot{z}(t)^2} dt = \int_a^b \left|\vec{\dot{r}}(t)\right| dt. \tag{2.13}$$

Mit der Bogenlänge wird die Länge des Weges, den der Punkt $x(t)$ zurücklegt, wenn er von a nach b wandert, gemessen.
Für die weitere theoretische Betrachtung der geometrischen Gestalt von Raumkurven ist es nützlich, eine sogenannte natürliche Parameterdarstellung der Kurve zu betrachten. Diese gilt, falls für eine beliebig gegebene Parameterdarstellung $\vec{r} = \vec{r}(t)$ für alle

betrachteten t die Geschwindigkeit eins ist, also

$$\left|\frac{d\vec{r}}{dt}\right| = 1 \tag{2.14}$$

Dazu soll nun gezeigt werden, dass die Bogenlänge s stets als natürlicher Parameter der Kurve verwendet werden kann.
Wenn eine reguläre Kurvendarstellung $\vec{r} = \vec{r}(t)$ vorliegt, kann diese mittels einer Transformation $t = t(s)$ auf einen neuen Parameter bezogen werden, so dass

$$\vec{r} = \vec{r}(t) = \vec{r}(t(s)).$$

Diese Transformation liefert wieder eine reguläre Darstellung, wenn $t = t(s)$ stetig diffenzierbar ist und $\frac{dt}{ds} \neq 0$. Ist eine Kurve durch die reguläre Darstellung $\vec{r} = \vec{r}(t(s))$ definiert, liefert die Differentation nach s

$$\frac{d\vec{r}}{ds} = \frac{d\vec{r}}{dt}\frac{dt}{ds} = \frac{\frac{d\vec{r}}{dt}}{\left|\frac{d\vec{r}}{dt}\right|}, \qquad \text{also } \left|\frac{d\vec{r}}{ds}\right| = 1.$$

Dies erfüllt die Bedingung 2.14 und bedeutet somit, dass Kurven die nach der Bogenlänge parametrisiert sind, die Geschwindigkeit eins besitzen und s als natürlicher Parameter der Kurve verwendet werden kann. Die nachfolgende Definitionen für Eigenschaften raümlicher Kurven stammen aus [BuHaWi90] S. 81 ff.. Zunächst wird von einer glatten Kurve im $\mathbb{R}^3$

$$K : \vec{r} = \vec{r}(s), \qquad \vec{r} : [0, L] \to \mathbb{R}^3$$

ausgegangen, die nach der Bogenlänge parametrisiert ist und mindestens dreimal stetig differenzierbar ist. Die Ableitung von $\vec{r}$ nach der Bogenlänge

$$\vec{T}(s) = \frac{d\vec{r}}{ds} \tag{2.15}$$

wird als Tangenteneinheitsvektor bezeichnet. Ausgehend von einer konstanten Geschwindigkeit zwischen jedem Bogenelement, können Abweichungen der Kurve von einem geradlinigen Verlauf nur durch Richtungsänderungen zustande kommen. Wie schnell sich die Kurve von der Tangente wegdreht, wird durch die Krümmung charakterisiert, für die gilt:

$$\kappa = \left|\dot{\vec{T}}(s)\right|. \tag{2.16}$$

Wenn $\kappa \neq 0$ berechnet sich der Krümmungsradius aus dem Kehrwert der Krümmung zu

$$\rho = \frac{1}{\kappa}.$$

Im Falle $\kappa \neq 0$ wird

$$\vec{N}(s) = \frac{\dot{\vec{T}}(s)}{\left|\dot{\vec{T}}(s)\right|}, \qquad \text{mit } \dot{\vec{T}}(s) = \kappa \vec{N}(s). \tag{2.17}$$

als Hauptnormalenvektor bezeichnet. Dieser steht rechtwinklig auf dem Tangentialvektor und zeigt in Richtung der Kurvenkrümmung. Ein weiterer Einheitsvektor

$$\vec{B}(s) = \vec{T}(s) \times \vec{N}(s) \tag{2.18}$$

ist der Binormalenvektor, der zusammen mit $\vec{N}(s)$ und $\vec{T}(s)$ eine Orthonormalenbasis des $\mathbb{R}^3$ bildet. Diese drei Vektoren stehen paarweise senkrecht aufeinander und bilden das begleitende Dreibein der Kurve.
Die Windung oder Torsion gibt die Abweichung der Kurve von einer ebenen Kurve an und ergibt sich aus

$$\tau = -\dot{\vec{B}}(s)\vec{N}(s). \tag{2.19}$$

Ist die berechnete Windung verschieden von Null, so verwindet sich die Kurve schraubenartig. Im Falle einer positiven Windung liegt eine rechtsgewundene (Windungssinn entgegen Uhrzeigersinn) Kurve und bei $\tau < 0$ eine linksgewundene Kurve vor [BuHaWi90]. Der Windungsradius υ lässt sich aus dem reziproken Wert der Windung bestimmen:

$$\upsilon = \frac{1}{\tau}.$$

Obwohl jede reguläre Kurve nach der Bogenlänge parametrisiert werden kann, ist diese Darstellung meistens schwierig und kompliziert im Umgang. Trotzdem ist es für einige Betrachtungen im weiteren Verlauf notwendig, die gerade beschriebenen Eigenschaften für beliebig parametrisierte Kurvendarstellungen zu nennen. Dazu sei $\vec{r} = \vec{r}(t)$ die Darstellung einer regulären Kurve im $\mathbb{R}^3$ mit

$$K : \vec{r} = \vec{r}(t), \qquad \vec{r} : [a, b] \rightarrow \mathbb{R}^3, \qquad a \leq t \leq b.$$

Es wird vorrausgesetzt, dass die dritte Ableitung existiert, d.h. $\vec{r}(t)$ ist für alle t mindestens dreimal stetig differenzierbar. Nach den Formel 7.13 bis 7.17 aus [Gr94] S. 121

ist

$$\vec{T}(t) = \frac{\dot{\vec{r}}(t)}{\left|\dot{\vec{r}}(t)\right|} \tag{2.20}$$

der Tangenteneinheitsvektor im Kurvenpunkt $(x(t), y(t), z(t))$. Für den Binormalenvektor gilt

$$\vec{B}(t) = \frac{\dot{\vec{r}}(t) \times \ddot{\vec{r}}(t)}{\left|\dot{\vec{r}}(t) \times \ddot{\vec{r}}(t)\right|} \tag{2.21}$$

und $\vec{N}(t) = \vec{T}(t) \times \vec{B}(t)$ ist der Normalenvektor. Die Krümmung κ der Kurve mit dem Krümmungsradius ρ ergibt sich aus

$$\kappa = \frac{\left|\dot{\vec{r}}(t) \times \ddot{\vec{r}}(t)\right|}{\left|\dot{\vec{r}}(t)\right|^3}, \qquad \rho = \frac{1}{\kappa} \qquad \text{falls } \kappa \neq 0. \tag{2.22}$$

Die Berechnung der Windung erfolgt durch

$$\tau = \frac{det\left(\dot{\vec{r}}(t), \ddot{\vec{r}}(t), \dddot{\vec{r}}\,(t)\right)}{\left|\dot{\vec{r}}(t) \times \ddot{\vec{r}}(t)\right|^2}. \tag{2.23}$$

Der Fundamentalsatz für Raumkurven besagt, dass zwei Kurven mit gleicher Krümmung und gleicher Torsion übereinstimmen. Alle Eigenschaften der Kurve sind in den Beziehungen zwischen ihrer Länge, Krümmung und Torsion enthalten. Nur die Lage der Kurve im Raum muss bestimmt werden. Die Lage ist durch das begleitende Dreibein am Startpunkt und durch den selbigen bestimmt.

Diese Grundlagen ermöglichen es uns die komplette Bahnkurve nur durch die Parameter der Krümmung, Windung und der Startausrichtung zu beschreiben. Dies führte uns zum Schraublinienansatz.

2.5.1 Bahnkurve - Schraubenlinie

Die Schraubung ist ein räumlicher Bewegungsvorgang, der sich zusammensetzt aus einer Drehung um eine Achse $\vec{a}$ und einer Schiebung in Richtung $\vec{a}$. Der Drehwinkel ω beschreibt die Frequenz und bestimmt somit die Anzahl der Drehungen pro Zeitabschnitt. Der Weg, den ein Punkt während einer Schraubung durchläuft, wird als Schraublinie bezeichnet. Die Ganghöhe h ist die zu einer vollen Umdrehung (2π) gehörige Schubstrecke.

$$\begin{aligned} x(t) &= R\,cos(\omega t) \\ y(t) &= R\,sin(\omega t) \\ z(t) &= ht \end{aligned} \tag{2.24}$$

Die Schraubline hat sowohl eine konstante Krümmung und als auch eine konstante Windung. Das Aufsteigverhalten unserer Blase zeigt eine sehr deutliche Ähnlichkeit zu einer Schraublinie. Auch Beobachtung von Personen die den Versuchsaufbau beobachten konnten, bestätigten diese Schraubbewegung bei allen aufsteigenden Blasen. Allerdings reichte eine gerade, symmetrische Schraublinie nicht zur Beschreibung aus. Um die Form einer Schraublinie zu verändern muss man die Parameter variabel machen:

$$\begin{aligned} x(t) &= R(t)\,cos(\omega t) \\ y(t) &= R(t)\,sin(\omega t) \\ z(t) &= h(t)t \end{aligned} \tag{2.25}$$

Wir haben diese Verallgemeinerung an unsere Bedürfnisse angepasst:

$$\begin{aligned} x(t) &= x_1 cos(x_2 t) + x_3 \\ y(t) &= y_1 sin(y_2 t) + y_3 \\ z(t) &= z_1 t + z_2 \end{aligned} \tag{2.26}$$

Die Parameter x_1 und y_1 beschreiben den Radius, x_2 und y_2 die Frequenzen und x_3, y_3 und z_2 die Translation im Raum. z_1 beschreibt die Steiggeschwindigkeit der Blase. Die erste Idee war es diese Parameter mit Hilfe eines Minimierungsverfahrens so anzupassen, dass die Schraublinie möglichst der Bahn der Blase entspricht. Wir nahmen an, dass die drei Gleichungen für $x(t)$, $y(t)$ und $z(t)$ voneinander unabhängig sind und sie

deshalb einzeln minimiert werden können.

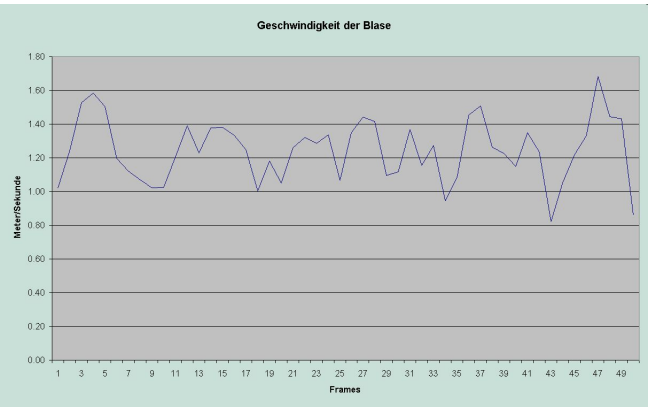


Abbildung 2.10. Bahngeschwindigkeit der Blase

Zuerst musste jedoch die Bahngeschwindigkeit untersucht werden, denn wie erwähnt ist die Krümmung und Torsion nur für Kurven mit konstanter Geschwindigkeit so einfach zu berechnen. Eine Untersuchung auf dieselbe ergab keine erkennbare Konstanz, jedoch konnte man auch nicht erkennen, ob die Schwankungen aus zufälligen Messfehlern oder aus einer ungleichmäßigen Bewegeung der Blase herrührten (siehe Abbildung 2.10). Deshalb entschieden wir uns dafür, zwei Wege der Rekonstruktion zu untersuchen und zu implementieren. Es wurde eine Fehlerabschätzung der beiden Wege verglichen und die Ergebnisse der Implementation bestätigten daraufhin diese. Da wir allerdings nicht ausschließen können, dass wir mit anderen Meßreihen möglicherweise bessere Ergebnisse über den Weg des Bogenmaßes erreichen könnten,haben wir uns für eine Implementation beider Wege entschieden. Für den ersten Weg, eine Minimierung der Funktion die von der Zeit t abhängt, können die Daten, die wir aus dem Video erhalten, direkt verwendet werden. Für die „first guess“ Abschätzung benötigen wir allerdings die erste Ableitung.
Um die Kurve in eine Kurve in Abhängigkeit von der Bogenlänge s zu erhalten, mußten wir folgendermaßen vorgehen. Zuerst wird die Gesamtlänge und die Länge der einzelnen Teilabschnitte zwischen den Messpunkten errechnet. Die Gesamtlänge beträgt wie erwähnt

$$L = \int_{t_0}^{t_n} \left| \dot{\vec{r}}(t) \right| dt \tag{2.27}$$

Es ist also sowohl eine Differentation als auch ein Integral zu lösen. Da wir von diskreten Werten ausgehen, muss dies numerisch geschehen. Zur Differentation benutzen wir den

zentralen Differenzenquotienten und für den ersten und letzten Wert helfen wir mit dem vorwärtigen bzw. mit dem rückwärtigen Differenzenquotienten aus. Mit $h = \Delta t$ entsteht für

$$f(x) = \{x_0, x_1, x_2, ..., x_n\} \tag{2.28}$$

$$f'(x) \approx \left\{ \frac{x_1 - x_0}{h}, \frac{x_2 - x_0}{2h}, \frac{x_3 - x_1}{2h}, \ldots, \frac{x_n - x_{n-2}}{2h}, \frac{x_n - x_{n-1}}{h} \right\} \tag{2.29}$$

Wir erhalten dadurch genauso viele Werte für die Ableitung wie wir Stützstellen haben. Der dabei entstehende Fehler ist $O(h^2)$. Er hängt also nur von der Anzahl der Stützstellen bzw. der Abtastrate ab. Je schneller abgetastet wird, desto kleiner der Fehler durch die Differenzenbildung. Eine schnellere Abtastung ist nur mit speziellen Kameras möglich die im Gegensatz zu den herkömmlichen 30 über 400 Bilder pro Sekunde aufnehmen können. Solch eine Ausrüstung ist aber wie erwähnt im Labor in Mexiko vorhanden.
Der Betrag dieses Vektors muss nun integriert werden. Hierzu benutzen wir die einfache Trapezformel. Die Trapezsumme hat die Fehlerordnung 2 und ist durch

$$\frac{(x_n - x_0)h^2}{12} M_2$$

beschränkt. M_2 ist dabei die obere Schranke von $|f''(x)|$ über den gesamten Bereich. Da dies die zweite Ableitung der Geschwindigkeit ist, also die Ableitung der Beschleunigung der Gasblase, kann man hier davon ausgehen, dass eine Quadraturformel der Ordnung 2 ausreicht, um das Integral genau genug abzuschätzen.

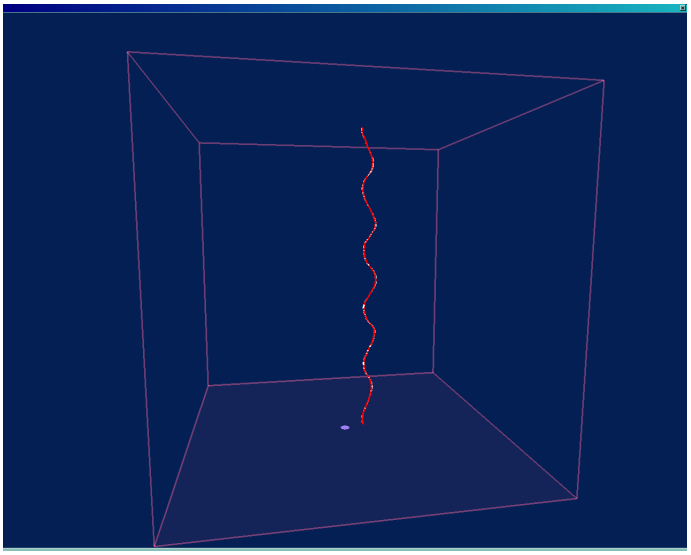


Abbildung 2.11. Überlagerung $P(s)$ und $P(t)$

Nun wird die Kurve in gleich lange Stücke unterteilt. Die Anzahl dieser Teilstücke wird vom Benutzer festgelegt, als Standardwert ist 100 gesetzt. Diese Unterteilung ist notwendig, um eine Kurve mit konstantem Parameter zu erzeugen. Doch jetzt haben wir neue Stützstellen deren zugehörigen Punkt auf der Bahnkurve wir aber nicht kennen. Hier kommt die Bezier-Interpolation ins Spiel. Über das Verhältnis zwischen der Anzahl der von der Bogenlänge abhängigen, gewählten Stützstellen und der durch das Video festgelegten Anzahl der von t abhängigen Stützstellen kann für jede neue Stützstelle ein Faktor x ermittelt werden. Dieser wird als eine Gleitkommazahl $x = a + b$ interpretiert. Das a ist die größte kleinere natürliche Zahl zu x und drückt aus, zwischen welchen Stützstellen der gesuchte Wert liegen wird. Das b ist dadurch als $b = x - a$ festgelegt und beschreibt den Wert den die Laufvariable der Bezierinterpolation annehmen muss, um an der gewünschten Stelle den gesuchten Wert zu ermitteln. Wie klein der Fehler ist, der dabei entsteht, erkennt man auch an Abbildung 2.11. Hier sieht man die Überlagerung der Originalwerte (rot) mit den neuen von s abhängigen Werten (weiß), jeweils mit geraden Linien verbunden.

2.5.2 Minimierung

Hier führen beide Wege wieder zusammen und die Optimierung der Parameter ist bis auf die Anzahl der Stützstellen für beide Wege gleich. Im folgenden wird unsere Vorgehensweise die zur finalen Implementation führte, Schritt für Schritt beschrieben.
Ausgangspunkt war die Interpretation der Kurve als Schraublinie. Daraus resultierte der Ansatz, die einzelnen Parameter der Schraublinie variabel zu machen und diese über ein Fehlerminimierungsverfahren zu optimieren. Hier nochmal der Ansatz:

$$\begin{aligned} x(t) &= x_1 cos(x_2 t) + x_3 \\ y(t) &= y_1 sin(y_2 t) + y_3 \\ z(t) &= z_1 t + z_2 \end{aligned} \tag{2.30}$$

Zuerst nutzten wir das Programm „Maple“. Doch die `stats[fit, leastsquare]`-Methode unterstützt nicht die Varianz der Parameter x_2 und y_2. Aber auch Kurven, die mit festen vorgegenbenen Parametern für x_2 und y_2 berechnet wurden, sahen der echten Kurve einigermaßen ähnlich (siehe Abbildung 2.12).

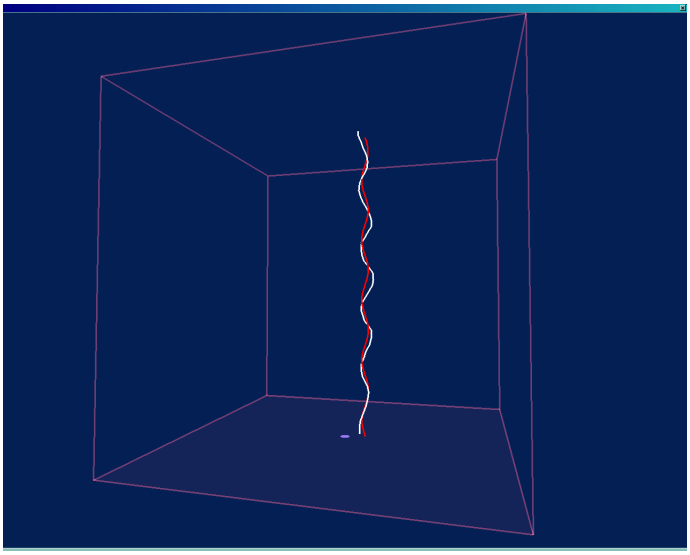


Abbildung 2.12. Überlagerung Maple (rot) und original (weiß)

Dies ermutigte uns diesen Ansatz zu vertiefen. Wir suchten nach einer Möglichkeit die Parameter x_2 und y_2 zu berechnen, bzw. sie möglichst gut aus den Daten abzuschätzen. Die Parameter der Kurve x und y entsprechen den Frequenzen der Schwingungen in den einzelnen Gleichungen. Diese Frequenz schätzten wir durch zählen der Nulldurchläufe der Ableitung ab. Als Frequenzen erhielten wir somit:

$$\begin{aligned} x_2 &= \frac{numzeros(x'(t))\pi}{numnodes} \\ y_2 &= \frac{numzeros(y'(t))\pi}{numnodes} \end{aligned} \tag{2.31}$$

Diese Abschätzung erwies sich als gut, aber wir hielten es für besser ein Minimierungsverfahren zu suchen, welches auch diesen Parameter optimieren kann. Wir wurden in Matlab fündig. Die Methode `lsqcurvefit` erfüllte alle unsere Bedürfnisse. Sie basiert auf der kleinsten Quadrate Methode nach folgendem Prinzip. Die Gleichung

$$\frac{1}{2}\sum_{i=0}^{n}(f(x, xdata) - y)^2$$

soll miniert werden. Wir unterteilten die Daten in x, y und z und stellten die Gleichung für jede Datenreihe auf, um die Paramter mit `lsqcurvefit` zu optimieren. Die Funktion `lsqcurvefit` benötigt allerdings einen sogenannten „first guess“, also eine erste grobe Abschätzung der Parameter, um einen Bereich vorzugeben in dessen Nähe nach einem Minimum gesucht werden soll. Für x_2, y_2 kannten wir die Abschätzung schon aus dem Ansatz mit Maple. Den „first guess“ der Radiusparameter x_1, y_1 schätzen wir durch:

$$x_1 = \frac{1}{2}(\max(x(t)) - \min(x(t))) \qquad \text{und} \qquad y_1 = \frac{1}{2}(\max(y(t)) - \min(y(t)))$$

ab. Die Translationen x_3, y_3 und z_2 wurden einfach durch die Mittelwerte abgeschätzt. Den letzten verbliebenen Parameter z_1 belegen wir mit der Differenz aus dem ersten und dem letzten Wert der Kurve, um so eine durchschnittliche Steigungsgeschwindigkeit zu approximieren. Die Ergebnisse aus Matlab sind in Abbildung 2.13 zu begutachten.

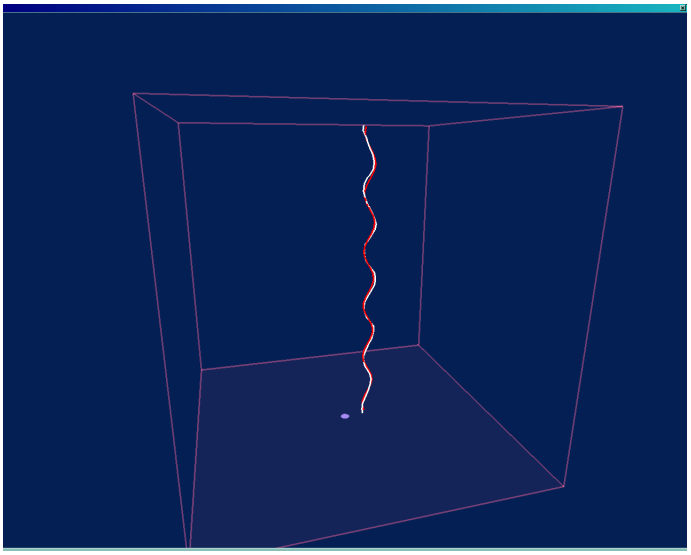


Abbildung 2.13. Überlagerung MATLAB (rot) und original (weiß)

Die Implementation in MATLAB erwies sich als sehr einfach, da nur eine vorgefertigte Funktion angewendet werden musste. Allerdings schränkte uns dies sehr stark ein, denn damit lief das Programm auch nur noch auf Rechner, die über eine Matlab-Lizenz verfügten. Dies war für uns das Kriterium eine Portation in C++ zu schreiben.
Aufgrund positiver Erfahrungen aus früheren Projekten entschieden wir uns für die Verwendung der „Powell's direction set"-Methode. Sie ist ein multidimensionales Minimierungsverfahren, für beliebige Funktionen. Als Funktion wählten wir wieder eine kleinste Fehlerquadrate Funktion. Dieser Fehler wird vom Minimierungsverfahren über die gegebene Anzahl an Parametern minimiert. Das Verfahren erfordert einen recht guten „first guess"-Wert. Die Abschätzung eines solchen wurde direkt übernommen und die Berechnungsverfahren in C++ implemetiert. Zunächst möchte ich die „Powell's direction set"-Methode etwas genauer erläutern.
Die zu minimierende Funktion f ist von n (in unserem Fall 5 bzw. 2 für die z-Werte) Parametern abhängig. Sie werden als ein n-dimensionaler Vektor $\vec{P}$ angesehen, der einen Punkt im n-dimensionalen Raum beschreibt. Nun gilt es diesen Punkt so zu verschieben, bis er die Funktion f minimiert hat. Dies wird erreicht, indem der Punkt entlang seiner Basisvektoren nacheinander jeweils so lange verschoben, bis er ein Minimum der Funktion f erreicht. Da die erreichten Minima auch lokal sein können und es in der Regel auch sind, werden in unserer Implementation die einzelnen Dimensionen stückweise auf mehrere lokale Minima hin untersucht. Wenn eine Genauigkeit von ϵ, oder die maximale Anzahl an Iterationen erreicht ist, bricht der Algorithmus ab. Die Genauigkeit ist in

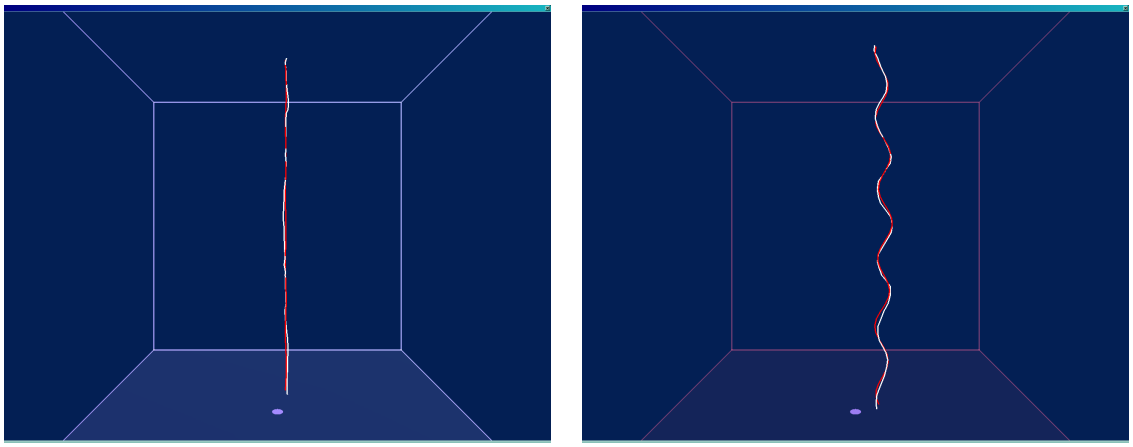
unserer Implementation auf $\epsilon = 10^{-4}$ festglegt. Die maximale Anzahl der Iterationen ist vom Benutzer frei wählbar, 1024 ist voreingestellt. Aus rechnertechnischen Gründen ist eine Potenz von 2 besonders geeignet. Desweiteren sind die Anzahl und die Breite der Fenster, die zur Unterteilung der einzelnen Dimensionen gebildet werden, einstellbar; allerdings nur im Quellcode und erfordern somit eine neue Kompilierung.
Nach zahlreichen Durchläufen fanden wir günstige Parameter, die zu Ergebnissen führten, die uns fast zufrieden stellten. Wir waren noch unzufrieden mit dem Radius der Rotation. Die Ergebnisse von [LeTr03], sowie die Beobachtungen von ihm und Prof. Gürlebeck führten zu der Annahme, dass es außer der großen Rotation noch eine zweite deutlich schwächere Rotation geben muss. Die läßt sich als eine Eigenrotation der Blase deuten, die in einer kleinen Abweichung von der Schraubenkurve führen kann. Wir approximierten diese kleine Rotation auch als Rotation um die Senkrechte. Diese Idee hielt die Formel einfach und Untersuchungen von Rotationen um andere Achsen kaum voneinander zu unterscheiden waren. Also stellten wir eine neue Formel auf:

$$\begin{aligned} x(t) &= x_1 cos(x_2 t) + x_3 + x_4 cos(x_5 t) \\ y(t) &= y_1 sin(y_2 t) + y_3 + y_4 sin(y_5 t) \\ z(t) &= z_1 t + z_2 \end{aligned} \tag{2.32}$$

Die Optimierung der erweiterten Formel wurde durch eine weitere Minimierung durchgeführt. Das bedeutet, wir optimieren genau wie vorher zuerst die Parameter x_1, x_2 und x_3 sowie y_1, y_2 und y_3. Die Ergebnisse nutzen wir als „first guess“ für die Optimierung nach fünf Parametern. Die „first guess“ Werte der beiden neuen Parameter x_4 und x_5 sowie y_4 und y_5 wurden etwas aufwendiger abgeschätzt. x_4 und y_4 ergeben sich aus dem Absolutwert der Differenz zwischen dem optimierten großen Radius und dem zuerst abgeschätzten großen Radius (also $|new(x_1 - old(x_1)|$). x_5 und y_5 stellen wieder eine Frequenz dar, diesmal allerdings die der kleinen Rotation. Auch sie wurde über das Zählen der Nulldurchläufe eines Parameters abgeschätzt, nämlich die der Winkelgeschwindigkeit. Die kleine Rotation verändert die Winkelgeschwindigkeit und den Radius der großen Rotation. Ausgedrückt in einer Formel heißt das:

$$\begin{aligned} x_5 &= \frac{numzeros(Winkelgeschwindigkeit(P(t))\pi}{numnodes} \\ y_5 &= \frac{numzeros(Winkelgeschwindigkeit(P(t))\pi}{numnodes} \end{aligned} \tag{2.33}$$

Die Minimierung läuft dann genauso ab wie die nach drei Parametern nur dass andere Formeln minimiert werden. In Abbildung
Die Ergebnisse sind in den Abbildungen 2.14 und 2.15 zu sehen. Allerdings muss man beachten, dass erst bei der Drehung der Sicht eine vernünftige Beobachtung gemacht werden kann. Dazu gibt es das Visualisierungsprogramm von Marcel Schlönvoigt, welches im Anschluss noch kurz vorgestellt wird.



(a) aus Sicht von Kamera 1 (b) aus Sicht von Kamera 2

Abbildung 2.14. Ergebnisse aus Kameraperspektive

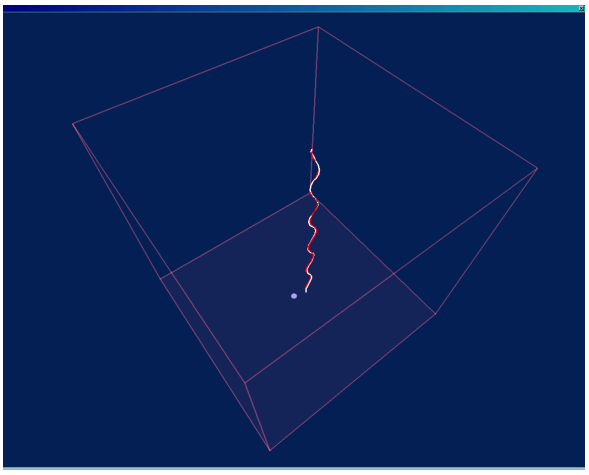


Abbildung 2.15. Ergebnis mit Sicht von schräg oben

3 Fazit

Am Ende unserer Betrachtungen muss angeführt werden, dass die Software von Marcel Schlönvoigt mit entwickelt worden ist, der sich allerdings schwerpunktmäßig mit der Visualisierung beschäftigt hat. Aus dieser Software stammen die Ergebnis-Screenshots. Die Qualität der Approximation der Bahnkurve läßt sich nicht aus einfachen Bildern herauslesen, sondern es erfordert die Möglichkeit sich die Kurve von allen Seiten anzuschauen. Genau das ermöglicht diese Software. Sie soll dem Benutzer die Möglichkeit geben, den Weg der Blase nachzuvollziehen und verschiedene Kurven vergleichen zu können. Das Ziel dieses Projektes war es die Bahnkurve mathematisch zu beschreiben. Diese von uns gefundene Beschreibung der Kurve ist nicht so genau, wie es eine stückweise Spline-Interpolation wäre (die wir auch implementiert haben), jedoch wird sie durch nur 12 Parameter vollständig beschrieben. Der Spline besteht im Gegensatz dazu aus 50 Polynomen 3. Grades. In folgenden Projekten soll untersucht werden, ob sich der Weg der Blase aus bestimmten Parametern voraussehen lässt. Mögliche Kriterien für den Weg wären der Verschmutzungsgrad der Flüssigkeit, der Winkel des Gaseinlassventils am Boden des Bassins, das Volumen, der Druck oder die Geschwindigkeit der Blase beim Einlass. Wir hoffen, dass mit Hilfe unserer Software in Zukunft die Bahnen zukünftiger Blasen genauer und besser untersucht werden können, um somit das Herleiten von solchen Voraussagen zu ermöglichen. Eine genaue Vorhersage wird unserer Meinung nach nicht möglich sein, eine zweckerfüllende Vorhersage bezüglich der Einschränkung der Bahnkurve auf einen bestimmten Bereich jedoch schon.
Das Problem, dass wir nur eine einzige Blase als Referenzbahn hatten, ist jedenfalls nicht zu überbewerten, denn das bearbeitete Referenzvideo zeigt die Bahn einer Gasblase in sauberem, strömungsfreiem Wasser. Dadurch dass ohne störende Einflüsse trotzdem eine so markante Drehbewegung entstanden ist, lässt darauf schließen, dass sich auch andere Blasen unter anderen Bedingungen so verhalten werden, dass sie mit unserer Software erfasst werden können.
Die Implementation zweier unterschiedlicher Parametrisierungsmethoden, nach der Zeit und nach der Bogenlänge, erhöht die Flexibilität zusätzlich. Dem Benutzer stehen verschiedene Methoden zur Verfügung die Optimierung zu beeinflussen. Das Arbeiten mit der Software ist sehr einfach und erfordert kaum Einarbeitungszeit. Der Quellcode wird selbstverständlich zur Verfügung gestellt, so dass zukünftige Entwickler auch die Mög-

lichkeit haben, das Programm zu erweitern. Hier sehen wir zahlreiche Möglichkeiten die Software weiter zu entwickeln. Zum Beispiel ist es denkbar, die Optimierungsfunktionen auszutauschen um eine Alternative zu unserer Schraublinie zu bieten. Dies ist relative einfach einzubauen, jedoch schien es für uns ausreichend, aufgrund fehlender Datenreihen, an der Schraublinie festzuhalten. Wir denken mit unserer Software eine gute Basis für weitere Forschungen an den Gasblasen erschaffen zu haben. Für die Bestätigung unserer Ergebnisse wäre es interessant gewesen, sie mit den Resultaten von anderen Messwerten vergleichen zu können. Eine Gegenüberstellung mit weiteren experimentell ermittelten Blasenbewegungen ist deshalb wünschenswert und kann mithilfe der Software jederzeit durchgeführt werden.
Neben dieser Dokumentation ist der Quelltext ausführlich kommentiert und erklärt. Wir wünschen allen viel Erfolg bei der Arbeit mit unserm Programm.

Literaturverzeichnis

[Wa02] Watt, Allan; 3D-Computergrafik; Addison Wesley 2002.

[Ku99] Kuipers, Jack B.; Quaternions and Rotation Sequences - a primer with applications to orbits, aerospace, and virtual reality; ISBN: 0-691-05872-5; Princeton University Press; Princeton, NJ; 1999.

[BuHaWi90] Burdick, Howard E.; Höhere Mathematik für Ingenieure, Bd. 4, Vektoranalysis und Funktionentheorie; ISBN: 3-519-02958-8; Teubner; Stuttgart; 1990.

[Gr94] Gray, Alfred; Differentialgeometrie : klassische Theorie in moderner Darstellung; ISBN: 3-86025-141-4; Spektrum Akademischer Verlag; Heidelberg, u.a.; 1994.

[ScKö04] Schwarz, Hans-Rudolf, Köckler, Norbert; Numerische Mathematik; ISBN: 3-519-42960-8; Teubner; Stuttgart u.a.; 2004.

[Bock98] Sebastian Bock BUW; Mathematisch Grundlagen für Bèzier-Kurven im CAD; Weimar 1998.

[LeTr03] Traversoni, Leonardo; Bubble Motion and Evolution described with Quaternions; Mexico

Weblinks

[WWW1] http://geometrie.tuwien.ac.at/Splines

[WWW2] http://www.math.uni-wuppertal.de/~heilmann/math3_0405/skript/teil3.pdf (01.02.2005)

[WWW3] http://www.msl.uni-bonn.de/vorlesung/vermessung/skripte/dg1.pdf (05.02.2005)

[WWW4] http://marvin.sn.schule.de/~mathe/dateien/ma_os/windsch.pdf (11.02.2005)

[WWW5] http://www.uni-weimar.de/~schloenv/gasblasen/Rapport_stage2004.pdf (25.02.2005)

Abbildungsverzeichnis