Messung der Glottisöffnungsfläche und deren Beziehung zum Rauschanteil der Stimme

Diplomarbeit

vorgelegt von

Sven Anderson

aus

Freiburg im Breisgau

angefertigt im

Dritten Physikalischen Institut

der Georg–August–Universität zu Göttingen

2003

Inhaltsverzeichnis

Abbildungsverzeichnis

Einleitung

Das menschliche Stimmsignal wird bei seiner Erzeugung aus verschiedenen Klangkomponenten zusammengesetzt, die jeweils einen unterschiedlichen Ursprung haben.

Zum einen erzeugen die durch die Luftströmung zur Schwingung angeregten Stimmlippen im Kehlkopf ein periodisches akustisches Signal. Dieses Signal wird der *stimmhafte Anteil* des Stimmsignals genannt und ist auch dafür verantwortlich, mit welcher *Grundfrequenz*, also in welcher Tonhöhe die Stimme wahrgenommen wird.

Ein anderer Teil wird durch turbulente Luftströmungen an Einengungen des Vokaltraktes erzeugt, die wiederum ein so genanntes turbulentes Rauschen erzeugen. Diesen Anteil nennt man *Rauschanteil* der Stimme. Es gibt verschiedene Bereiche im Vokaltrakt, die ein solches Rauschen erzeugen können. Bei dem Laut /s/ ist dies zum Beispiel die Einengung zwischen der Zunge und dem Gaumen, bei einem /f/ die zwischen den Zähnen und der Unterlippe.

Aber auch die *Glottis*, die Öffnung zwischen den Stimmlippen, ist eine solche Einengung, von der turbulentes Rauschen ausgeht. Sie ist somit gleichzeitig Quelle sowohl für den stimmhaften Anteil als auch für einen Rauschanteil.

Die Beobachtungen von Medizinern zeigen, dass mit einer heiseren und behauchten Stimme, also einer Stimme deren Rauschanteil hörbar höher ist als bei Normalstimmen, oft das Phänomen einhergeht, dass sich die Stimmlippen während einer Schwingungsperiode nie vollständig verschließen. Man spricht in diesem Fall von einem *unvollständigen glottalen Schluss*.

Die bisherigen Beobachtungsmethoden der schwingenden Stimmlippen, wie vor allem der Stroboskopie, lassen es noch nicht zu, die genauen Bewegungsabläufe

mit akustischen Parametern wie dem Rauschanteil in Beziehung zu setzen, um solche Phänomene genauer zu untersuchen. Speziell bei der Stroboskopie wird lediglich ein virtuelles Schwingungsbild rekonstruiert, indem aus dem akustischen Signal die Grundfrequenz der Schwingung ermittelt wird und die stroboskopische Lichtquelle mit dieser gleichgeschaltet wird. Durch leichte Abweichungen von dieser Grundfrequenz kann so mit einer herkömmlichen Videokamera der gesamte Schwingungsverlauf abgetastet werden. Dies kann aber zum einen strenggenommen nur bei völlig periodischen Schwingungen funktionieren, und zum anderen ist die Triggerung der Lichtquelle und somit die zeitliche Zuordnung des Schwingungsverlaufs viel zu ungenau.

Mit der sich nun etablierenden digitalen Hochgeschwindigkeitsglottographie hat sich diese Situation geändert. Hierbei wird der Schwingungsverlauf der Stimmlippen mit bis zu 4000 Bildern pro Sekunde aufgenommen. Damit wird jede Schwingungsperiode mit über 10 Einzelbildern dargestellt und kann mit dem simultan aufgenommenen akustischen Signal direkt in Beziehung gesetzt werden.

Diese Voraussetzung hat es ermöglicht, dass im Rahmen der vorliegenden Arbeit der Zusammenhang zwischen der *Glottisfläche* und dem Rauschanteil der Stimme untersucht werden konnte.

In Kapitel 1 werden zunächst die notwendigen Grundlagen hierfür erläutert, in Kapitel 2 wird dann beschrieben, wie und mit welchen Geräten die Daten erhoben wurden und woher diese stammen.

Wie aus den Bild- und Tondaten vergleichbare Parameter ermittelt werden und wie ihre Beziehung zueinander untersucht werden kann, wird in Kapitel 3 beschrieben.

Für den Rauschanteil der Stimme wurde der Parameter *Glottal to Noise Excitation Ratio (GNE)* gewählt, da er unabhängig von *Shimmer* und *Jitter* ist und so ein gutes Maß für das turbulente Rauschen darstellt.

Für die Bilddaten wurde ein Verfahren entwickelt, das aus den Bildern der Hochgeschwindigkeitskamera vollautomatisch das Gebiet der Glottis extrahiert. Zunächst wurde auf ein bekanntes Verfahren zurückgegriffen, welches dann aber zu einem völlig neuen Verfahren weiterentwickelt wurde. Ein Vorteil des neuen Verfahrens ist, dass es durch die dreidimensionale Bearbeitung der Bilddaten auch dann das vollständige Glottisgebiet findet, wenn dieses nicht nur aus einem einzigen zusam-

menhängenden Gebiet besteht, sondern zum Beispiel durch Schleimfäden in mehrere Gebiete unterteilt ist.

Aus dem ermittelten Verlauf der Glottisfläche können dann Parameter wie die *Restöffnungsfläche* und der *Closed-Quotient* bestimmt werden, die sich zum Vergleich mit dem GNE eignen.

Schließlich werden in den Kapiteln 4 und 5 die Ergebnisse dargestellt und diskutiert. Es stellt sich heraus, dass der GNE in der Tat mit dem Closed-Quotient und der Restöffnungsfläche korreliert beziehungsweise antikorreliert. Dies deckt sich mit den zuvor erwähnten medizinischen Beobachtungen und den qualitativen strömungsdynamischen Überlegungen aus Kapitel 1.

Kapitel 1

Grundlagen

In diesem Kapitel werden die notwendigen Grundlagen der Spracherzeugung vermittelt, die für das Verständnis der weiteren Untersuchungen notwendig sind. Außerdem wird kurz die Funktionsweise der Hochgeschwindigkeitsglottographie erläutert.

1.1 Glottis

Unter *Glottis* versteht man den aus beiden Stimmlippen bestehenden Stimmapparat oder nur die von ihnen gebildete Stimmritze [Pschyrembel 1982]. Im Folgenden ist mit »Glottis« allerdings lediglich die Stimmritze gemeint.

Um sie zu beobachten, wird eine *Laryngoskopie* durchgeführt. Dies geschieht gewöhnlich mit einem starren Endoskop, das in den Rachen eingeführt wird und so einen Einblick in den Kehlkopf erlaubt (siehe Abb. 1.1).

Man kann auf diese Weise gut die Glottis samt Stimmlippen und Umgebung betrachten, wie beispielsweise in Abbildung 1.2 zu sehen ist.

1.1.1 Funktion

Die Glottis befindet sich im Kehlkopf, also an der Stelle im Hals, an der die Luftröhre (Trachea) mit der Speiseröhre (Ösophagus) zusammentrifft.

Die primäre Funktion der Glottis ist daher auch die Ventilfunktion, da sie die Luftröhre verschließen kann, um so – vor allem beim Schluckvorgang – zu verhindern,

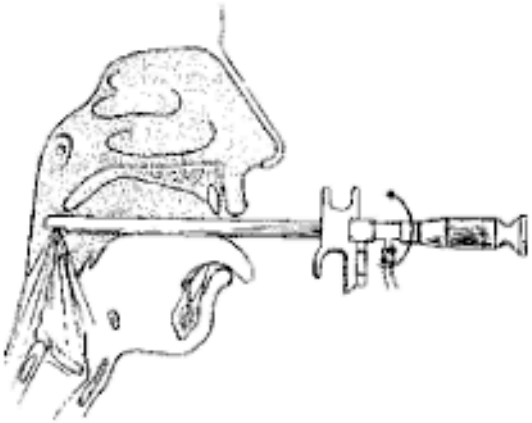


Abb. 1.1: Laryngoskopie mit starrem Endoskop

Abb. 1.2: Glottis aus Sicht des Endoskops

dass Fremdkörper in die Luftröhre und damit in die Lunge gelangen. Wenn dennoch ein Fremdkörper in die Luftröhre gelangt, kann sie sich außerdem verschließen, um einen Überdruck aufzubauen, mit dem der Fremdkörper herausgeschleudert wird, was genau beim Hustenreflex geschieht.

Während der Evolution hat sich später eine sekundäre Funktion entwickelt, die der Laut- und Stimmerzeugung dient [Wendler und Seidner 1987].

Hierbei wird die Glottis zunächst wie gewöhnlich geschlossen. Dann wird die Spannung der Stimmlippen etwas verringert und zugleich der *subglottale Druck*, also der Druck unterhalb der Glottis, erhöht. Hierdurch werden die Stimmlippen auseinandergedrängt und die Glottis öffnet sich. Durch diese Öffnung kann die Luft nun hindurchströmen, wodurch sich der subglottale Druck verringert. Zum einen durch die elastische Rückstellkraft der Stimmlippen und zum anderen durch den Unterdruck, der sich dem Bernoulli-Effekt zufolge an der durchströmten Öffnung einstellt, beginnt sich die Glottis wieder zu schließen. Ist dies geschehen, kann sich erneut ein subglottaler Druck aufbauen und der Kreislauf beginnt von vorne. Das glottale System beginnt zu schwingen. Dieser Vorgang ist sehr schön in Abbildung 1.3 in einem Frontalschnitt und in der Aufsicht zu sehen.

So entsteht eine pulsartig modulierte Luftströmung, die so genannte *glottale Anregung*, von der sich eine Schallwelle entlang des *Vokaltrakts* ausbreitet und aus Mund

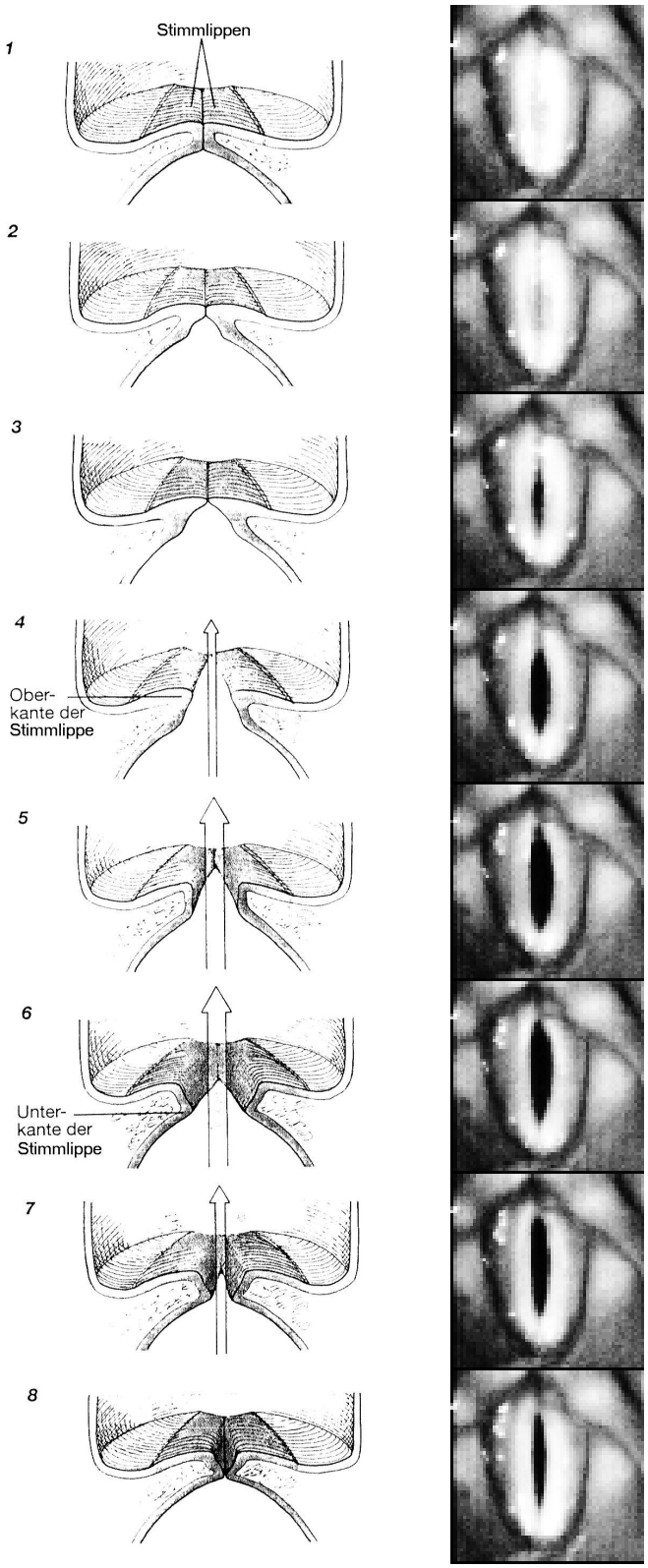


Abb. 1.3: Bewegungsablauf der Stimmlippenschwingung (links) nach Sataloff [1993] und korrespondierende Periode aus Sicht des Laryngoskops (rechts).

und/oder Nase austritt. Die Filtereigenschaften des Vokaltraktes geben dem Laut hierbei seinen charakteristischen Klang. Durch eine Veränderung der Form des Vokaltraktes werden die verschiedenen möglichen stimmhaften Laute erzeugt.

Die Frequenz der Glottisschwingung, auch *Grundfrequenz* genannt, hängt von vielen Parametern wie der Stimmlippenspannung, den Einstellungen und der Position des Kehlkopfes und dem subglottalen Druck ab. Die typische Frequenz liegt im Mittel bei 200 Hz bei Frauen und bei 125 Hz bei Männern [Titze 1994].

1.2 Rauschen der Stimme

Entstünde das Stimmsignal lediglich aus einer streng periodischen glottalen Anregung, wäre in ihm keinerlei Rauschen vorhanden. Tatsächlich gibt es aber drei verschiedene Quellen für Rauschen bei der Entstehung eines Stimmlautes:

1. Zufällige Variationen der Periodenlänge. Dieses Rauschen wird Frequenzmodulationsrauschen oder auch *Jitter* genannt.
2. Zufällige Variationen der Anregungsamplitude. Dieses Rauschen wird Amplitudenmodulationsrauschen oder auch *Shimmer* genannt.
3. Additives Rauschen. Hierbei handelt es sich um Rauschen, das durch turbulente Luftströmungen im Vokaltrakt entsteht.

1.2.1 Additives Rauschen

Das Rauschen, das in dieser Arbeit betrachtet werden soll, ist das additive Rauschen.

Damit eine Strömung turbulent wird, muss die *Reynolds-Zahl*

$$Re = \frac{d\varrho v}{\eta} \quad > 3000$$

sein, wobei d der Durchmesser des durchströmten Querschnittes, ϱ die Dichte, v die Strömungsgeschwindigkeit und η die Viskosität der Flüssigkeit oder des Gases ist [Tipler 1994].

Dieses ist vor allem an starken Einengungen des Vokaltraktes, wie zum Beispiel bei den Lauten /f/ oder /s/ an den Lippen und der Zunge der Fall. Aber auch die Glottis selbst stellt eine solche Einengung dar. Ob auch hier Turbulenzen entstehen können, kann leicht abgeschätzt werden:

Eine durchschnittliche Glottis ist 1 - 1,6 cm lang [Titze 1994], es sei also der Einfachheit halber $d \approx 0{,}01$ m. Wie durch Selbstversuch leicht festzustellen ist, liegt die Größenordnung des Volumenstroms bei normaler Phonation bei ca. 0,5 l/s. Bei einer Glottisfläche von ca. 0,5 cm^2 ergibt das eine Strömungsgeschwindigkeit von $v \approx 10$ m/s. Dies ergibt mit der Viskosität und Dichte von Luft eine Reynolds-Zahl von

$$Re \approx 7000.$$

Dies ist zwar nur eine sehr grobe Abschätzung, da sie sich aber auf Durchschnittswerte bezieht, zeigt sie dennoch, dass die Strömung an der Glottis zumindest in Spitzen durchaus in Bereiche kommt, in denen turbulente Strömungen zu erwarten sind.

Es ist also plausibel, dass additives Rauschen bei Vokalen, bei denen die Glottis die engste zu passierende Stelle der Luftströmung darstellt, durch genau diese Stelle entsteht.

Da in die Reynolds-Zahl zum einen die Größe der Glottisfläche in Form der Länge d mit eingeht und sie zum anderen ein Maß der Turbulenz und somit in gewisser Weise des turbulenten Rauschens ist, liegt die Vermutung nahe, dass es einen Zusammenhang zwischen der Glottisfläche und dem additiven Rauschen der Stimme geben könnte, was in dieser Arbeit untersucht wird.

Dies wird von der Erfahrung aus dem klinischen Alltag unterstützt, dass bei einer heiseren oder behauchten Stimme oftmals ein unvollständiger Glottisschluss zu beobachten ist.

Weitergehende Aussagen sind allerdings nur schwer zu machen. Es gibt so gut wie nichts im Bereich der Kehlkopfes, das eine feste Randbedingung darstellen würde, was das Ganze zu einem äußerst komplexen schwingenden System macht, bei dem sich alle Größen gegenseitig beeinflussen.

Daher ist es auch bisher nicht gelungen, ein theoretisches Modell für die Strömungen bei einer schwingenden Glottis zu erstellen. Lediglich die Modellierung der

Stimmlippenbewegung wird schon seit längerem versucht [Titze 1973] und ist gerade heutzutage noch ein aktives Forschungsgebiet [Alipour-Haghighi u. a. 2000]. Da die Stimmlippenbewegung und die Luftströmung stark von einander abhängig sind, wird letztendlich nur ein Modell, das beides hinreichend vereint, den Vorgang realistisch beschreiben können.

1.3 Digitale Hochgeschwindigkeitsglottographie

Das bisher gängige Verfahren, um die Schwingung der Glottis zu beobachten, ist das der Videostroboskopie. Bei ihr wird aus dem akustischen Signal die Grundfrequenz ermittelt und mit dieser wiederum eine stroboskopische Lichtquelle getriggert. So erhält man zunächst ein Standbild einer bestimmten Phase der Glottisschwingung. Variiert man nun leicht die Frequenz des Stroboskopes, kann man so alle Phasen der Schwingung abfahren und erhält so ein virtuelles Schwingungsbild. Dieses Verfahren setzt aber voraus, dass eine eindeutige Grundfrequenz vorhanden ist, also die Schwingung periodisch verläuft. Bei vielen funktionellen Stimmstörungen ist dies aber nicht der Fall, was das Verfahren in solchen Fällen stark einschränkt.

Bei der sich etablierenden digitalen Hochgeschwindigkeitsglottographie gibt es diese Einschränkungen nicht mehr. Hier wird mit bis zu 4000 Bildern pro Sekunde die Bewegung der Glottis genau aufgezeichnet. Da Grundfrequenzen von über 400 Hz praktisch nicht vorkommen, hat man so pro Periode mindestens zehn Einzelbilder, die den Verlauf der Schwingung genau darstellen. Die völlige Unabhängigkeit von der Grundfrequenz lässt es zu, dass auch bei stark gestörten Stimmen, ja sogar bei aphonen Stimmen die Stimmlippenbewegung genau betrachtet werden kann.

Erst mit Hilfe der Hochgeschwindigkeitsaufnahmen ist es nun möglich geworden, optische Parameter der Glottisschwingung auch bezüglich kurzer Zeitfenster in direkten Bezug zu akustischen Parametern zu setzen und eventuelle Zusammenhänge genauer zu untersuchen. In dieser Arbeit geschieht dies am Beispiel des optischen Parameters »Glottisfläche« und des akustischen Parameters »Rauschanteil«.

Kapitel 2

Datenaufnahme

Im Folgenden wird beschrieben, wie und mit welchen Geräten die Daten für diese Arbeit aufgenommen wurden und woher diese Daten stammen.

2.1 Hochgeschwindigkeitskamera

Zur Aufnahme der Glottisschwingung wurde die digitale Hochgeschwindigkeitskamera HS ENDOCAM 5560 der Firma Richard Wolf eingesetzt.

Sie besteht aus der Handkamera mit Endoskopoptik (Abb. 2.1) und einer Einheit, die die Daten zwischenspeichert, verarbeitet und auf einem Monitor darstellt (Abb. 2.2). Zusätzlich ist eine starke Lichtquelle notwendig (ebenfalls Abb. 2.2), da die Kamera notwendigerweise eine sehr kurze Belichtungszeit hat.

Nach erfolgter Aufnahme werden die Daten zur weiteren Verarbeitung, Analyse und Archivierung auf einen PC übertragen.

Die Bildauflösung beträgt 256×256 Bildpunkte, wobei allerdings in einer Schachbrettanordnung nur jedes zweite Pixel belegt ist. Effektiv sind also nur 32768 Bildpunkte genutzt.

Die zeitliche Auflösung kann 2000 oder 4000 Bilder pro Sekunde betragen, mit einer verringerten Lichtempfindlichkeit bei der höheren Bildrate. Bei den Aufnahmen, die in diese Arbeit eingeflossen sind, wurde allerdings ausschließlich eine Bildrate von 4000 bps verwendet.

Abb. 2.1: Hochgeschwindigkeitskamera

Abb. 2.2: Kamerasteuerung und Lichtquelle

Die Daten werden fortlaufend in einen Ringspeicher geschrieben und mittels eines so genannten *Posttriggers* eingefroren. Der Ringspeicher fasst genau 8192 Bilder, was demnach einer Dauer von ca. 2 bzw. 4 Sekunden entspricht.

Simultan wird über ein an der Kamera angebrachtes Mikrofon das akustische Signal im selben Ringspeicher aufgezeichnet. Die Abtastrate beträgt dabei 44100 Hz mit 8 Bit Genauigkeit. Wie sich herausstellte, ist diese Aufnahme allerdings mit Störungen versehen. Zum einen ist eine so genannte *Automatic-Gain-Control* vorgeschaltet, die versucht, die Amplitude auf ein festes Niveau anzuheben. Außerdem sind durch einen Fehler in der Kameraelektronik in unregelmäßigen Abständen fehlerhafte Bytes in den Audiodaten enthalten (zu hören als Knackgeräusch). Die direkte Verwendung dieser Daten ist somit nicht möglich.

2.2 Akustik

Zusätzlich zu der akustischen Aufnahme der Hochgeschwindigkeitskamera wurde eine weitere qualitativ höherwertige akustische Aufnahme erstellt, um erstere durch diese zu ersetzen. Dies geschah mittels eines PCs, der mit einer Soundkarte mit externem A/D-Wandler versehen ist (Terratec EWS88 MT). Hieran wurde ein Kondensatormikrofon angeschlossen. Bei diesem handelte es sich um ein so genanntes Headset-Mikrofon, so dass die relative Position zum Kopf während einer Aufnahme immer konstant war.

Auf einem zweiten Kanal wurde ein *Elektroglottogramm* aufgenommen [Baken 1992]. Hierbei wird durch das Anlegen von Elektroden an den Kehlkopf mittels einer hochfrequenten Wechselspannung der Widerstand von der einen zur anderen Seite des Kehlkopfes gemessen. Dieser Widerstand ist abhängig von der Kontaktfläche der beiden Stimmlippen und bildet so in gewisser Weise den Öffnungszustand der Glottis ab. Diese Daten wurden allerdings in dieser Arbeit nicht weiter verwendet.

Beide Kanäle wurden mit 48000 Hz in 16 Bit aufgenommen.

2.3 Datenmaterial

Um das notwendige Datenmaterial zu erstellen, wurden in der Abteilung für Phoniatrie und Pädaudiologie der Hals-Nasen-Ohren-Klinik des Universitäts-Klinikums Göttingen sowohl von Patienten mit verschiedenen Stimmstörungen als auch von Versuchspersonen mit Normalstimmen eine Hochgeschwindigkeitsglottographie und die zusätzliche zeitgleiche akustische Aufnahme angefertigt.

Hierzu wurde dem Patienten oder der Versuchsperson die Kamera in den Rachen eingeführt. Zumeist wurde zuvor der Rachen und der Kehlkopfbereich mit einem Oberflächenanästhetikum betäubt, um einen eventuellen Würgereiz zu unterbinden. Dann wurde die Person aufgefordert, ein paar Sekunden lang den Laut /i/ zu phonieren, da man bei diesem Laut einen möglichst guten Einblick in den Kehlkopf gewinnt. Der tatsächlich hörbare Laut ähnelt aber eher einem /ä/, da die in den Vokaltrakt eingeführte Kamera eine normale Phonation verhindert. Sowohl die Tonlage als auch die Lautstärke ist bei diesen Untersuchungsumständen gewöhnlich höher als bei gesprochener Sprache.

Obwohl insgesamt deutlich mehr Aufnahmen angefertigt wurden, schieden einige aus verschieden Ursachen aus. Zum einen waren manche Stimmlippen kaum einsehbar oder gar nicht vorhanden, z.B. bei Patienten, die einen operativen Eingriff hinter sich hatten. Manche Stimmen waren auch vollständig *aphon*, d.h. es war keinerlei Glottisschwingung vorhanden, was die im nächsten Kapitel beschriebene vollautomatische Auswertung unmöglich machte. Letzten Endes blieben 15 Aufnahmen übrig, die für die Auswertung verwendet werden konnten.

Kapitel 3

Verfahren

Im folgenden Kapitel werden die Verfahren beschrieben, die aus den akustischen und optischen Rohdaten, die mit Kamera und PC aufgenommen wurden, Parameter extrahieren, die sich für eine Untersuchung auf Korrelation eignen.

Die Entwicklung dieser Verfahren, speziell das der optischen Analyse, hat bei weitem den größten zeitlichen Teil dieser Arbeit eingenommen, da kaum auf vorhandene Verfahren zurückgegriffen werden konnte und für vieles spezielle Lösungen gefunden werden mussten. Daher wird auf die Erläuterung dieser Verfahren auch ein besonderer Schwerpunkt gelegt.

3.1 Akustische Analyse

Es wird nun beschrieben, wie die akustischen Daten des PCs mit denen der Kamera synchronisiert werden, um sie mit den Bilddaten in Zusammenhang zu bringen. Anschließend wird daraus der Rauschanteil des aufgenommen Lautes bezüglich eines gleitenden Zeitfensters bestimmt, so dass dieser mit einem korrespondierenden Zeitfenster der Hochgeschwindigkeitsaufnahme verglichen werden kann.

Es sei im folgenden S_k die akustische Aufnahme der Kamera und S_{pc} das separat mit einem PC aufgenommene akustische Signal.

3.1.1 Entstörung

Die Störungen in S_k können relativ leicht detektiert werden, da sich die Störung immer auf genau ein Sample beschränkt, es handelt sich somit um eine sehr scharfe Spitze. Die zweite Ableitung – wie sie im Diskreten üblicherweise definiert wird – einer Folge $\{0,0,1,0,0\}$, die einer solchen Spitze entspricht, ist $\{0,1,-2,1,0\}$. Es ist also zu erwarten, dass bei diesen Spitzen der Betrag der zweiten Ableitung sehr hoch ist und dieser bei den Nachbarwerten ebenfalls hoch ist, allerdings im Mittel ungefähr halb so hoch und mit umgekehrtem Vorzeichen.

Diese Überlegungen führen zu der Funktion

$$a(t) := S_k(t) - \frac{S_k(t-1) + S_k(t+1)}{2} = -\frac{S_k''(t)}{2},$$

mit der man folgendes Detektionskriterium definieren kann:

$$\begin{array}{ccccccc} a(t-1) < -\frac{s}{2} & \wedge & a(t) > s & \wedge & a(t+1) < -\frac{s}{2} \\ & & \vee & & \\ a(t-1) > \frac{s}{2} & \wedge & a(t) < -s & \wedge & a(t+1) > \frac{s}{2}. \end{array}$$

Der Schwellwert s wird experimentell ermittelt und dessen Größenordnung sollte ungefähr bei der doppelten mittleren Rauschamplitude liegen. Bei dieser Arbeit wurde er auf $s = 3{,}5$ gesetzt.

Für alle t, für die dieses Kriterium zutrifft, wird $S_k(t)$ durch das arithmetische Mittel der beiden Nachbarwerte $\frac{1}{2}(S_k(t-1) + S_k(t+1))$ ersetzt, also linear interpoliert.

3.1.2 Kreuzkorrelation

Um S_{pc} in zeitlichen Zusammenhang mit den Bilddaten zu bringen, muss es wie gesagt mit S_k synchronisiert werden, denn dieses entspricht in Zeitpunkt und zeitlicher Länge genau den zugehörigen Hochgeschwindigkeitsaufnahmen.

Es muss also der dem Segment S_k entsprechende Teilbereich innerhalb des längeren S_{pc} gefunden werden. Dazu wird zunächst S_k mit gefensterten sinc-Funktionen interpoliert und so die Abtastrate auf die von S_{pc} gewandelt. Nun werden die beiden

Signale wie üblich kreuzkorreliert:

$$c(\tau) := \sum_t S_{\mathsf{k}}(t) S_{\mathsf{pc}}(t+\tau)$$

An der Stelle der höchsten Korrelation, also bei

$$\delta = \operatorname{argmax}\big(c(\tau)\big)$$

beginnt in S_{pc} der S_{k} entsprechende Teil. Wenn l_{k} die Länge von S_{k} ist, erhält man also mit dem Bereich $S_{\mathsf{pc}}(\delta)$ bis $S_{\mathsf{pc}}(\delta + l_{\mathsf{k}} - 1)$ genau den Ausschnitt, der der Bildsequenz entspricht. Im Folgenden ist mit S_{pc} nur noch dieser Ausschnitt gemeint.

3.1.3 Eichung

Um die Abtastrate von S_{k} auf die von S_{pc} umzuwandeln, ist die genaue Kenntnis der ursprünglichen Abtastraten f_{pc} und f_{k} notwendig. Laut Werksangaben wird S_{k} mit $f_{\mathsf{k}} = 44100$ Hz und S_{pc} mit $f_{\mathsf{pc}} = 48000$ Hz Abtastrate aufgenommen. Es stellte sich aber heraus, dass diese Angaben für eine Kreuzkorrelation zu ungenau waren, so dass sich kein klares Maximum ausbildete. Es war daher notwendig, eine relative Eichung der beiden Samplingraten zueinander zu bestimmen. Hierbei wird davon ausgegangen, dass $f_{\mathsf{pc}} = 48000$ Hz ist, woran f_{k} geeicht wird.

Dazu werden mehrere monofrequente Sinustöne verschiedener Frequenzen und konstanter Amplitude mit beiden Systemen simultan aufgenommen. Diese Aufnahmen werden wie in Abschnitt 3.1.2 beschrieben kreuzkorreliert. Beim Interpolieren wird zunächst von $f_{\mathsf{k}} = 44100$ Hz ausgegangen und dieser Wert nach folgendem einfachen Algorithmus numerisch approximiert:

1. Mit einer Schrittweite f_Δ werden die Frequenzen von $f_k - 10f_\Delta$ bis $f_k + 10f_\Delta$ als Abtastrate für S_k angenommen und die zugehörigen maximalen Kreuzkorrelationswerte $c_{max} = \max\big(c(\tau)\big)$ bestimmt.
2. f_k wird auf die Frequenz mit dem größten c_{max} gesetzt und f_Δ durch 10 geteilt.
3. Bis zur gewünschten Genauigkeit von f_k wird nun von vorne begonnen.

Die so ermittelte Abtastfrequenz f_k ist auch die ursprüngliche Abtastfrequenz von S_k der eigentlichen Messdaten, die wie beschrieben in die von S_{pc} umgewandelt wird.

3.1.4 GNE

Um den Rauschanteil des akustischen Signals zu bestimmen, wird der Parameter *Glottal to Noise Excitation Ratio (GNE)* verwendet [Michaelis u. a. 1997]. Er stellt das Verhältnis von glottaler Stimmanregung zum Rauschanteil der Stimme dar. Um so niedriger er ist, desto höher ist der Rauschanteil.

Der GNE zeichnet sich dadurch aus, dass er unabhängig von den anderen Irregularitäten des Stimmsignals ist, wie dem *Jitter* (Schwankung der Grundfrequenz) und *Shimmer* (Schwankung der Amplitude), und auch bei stark pathologischen Stimmen, also Stimmen mit geringem oder keinem stimmhaften Anteil, sinnvolle Werte liefert. Er ist somit auch ein Maß für das turbulente Rauschen im Stimmsignal und daher bei diesen Untersuchungen besonders gut geeignet.

Über das gesamte Signal S_{pc} wird nun jeweils mit einer Fensterbreite von 100ms und einem Vorschub von 100ms der Verlauf des Rauschparameters $GNE(t)$ errechnet, wobei sich der Zeitindex t auf die zeitliche Mitte des 100ms-Fensters bezieht.

3.2 Optische Analyse

Im folgenden Abschnitt wird beschrieben, wie in den Bilddaten der Hochgeschwindigkeitskamera die Glottis vollautomatisch gefunden und ihr Flächeninhalt zu jedem Zeitpunkt, also die Flächenverlaufsfunktion ermittelt wird, um daraus dann geeignete Parameter zu extrahieren, die für eine Untersuchung auf Korrelation mit der Akustik geeignet sind.

Dies ist ein schwieriges Unterfangen, da es sich um eine organische Struktur handelt, deren Anatomie äußerst unterschiedlich aussehen kann. Dazu kommt das Problem, dass die Aufnahmen nicht immer von derselben Position mit demselben Abstand und derselben Beleuchtungsintensität vorgenommen werden können. In den Abbildungen 3.1 sind beispielhaft Bilder von verschiedenen Aufnahmen dargestellt.

3.2.1 Bekannte Verfahren

Da es sich bei dem Bildmaterial um sehr viele Einzelbilder handelt, kommt für die Ermittlung der Glottisfläche nur ein automatisches Verfahren in Frage. Da die digitale Hochgeschwindigkeitsphotographie im Allgemeinen und deren Anwendung auf die Glottographie im Speziellen noch relativ neu ist, wurde bisher noch kein ausgereiftes Verfahren veröffentlicht, das diese Aufgabe vollständig lösen könnte. Hinzu kommt, dass generell das Problem der automatischen Segmentierung in digitalen Bilddaten trotz dessen Bedeutsamkeit noch nicht vollständig gelöst ist und ein Feld der aktiven Forschung ist [Nikolaidis und Pitas 2001, S. 81-82].

Allerdings wurde von Wittenberg [1998] der Ansatz eines Verfahrens zur automatischen Glottissegmentierung beschrieben, welches im Groben in folgender Weise arbeitet:

Es wird davon ausgegangen, dass der dunkelste Punkt eines Bildes innerhalb der Glottis liegt. Von diesem dunkelsten Punkt als Startpunkt ausgehend wird dann die gesamte Fläche um diesen Punkt ausgefüllt, die dunkler ist als ein Schwellwert θ. Dieser Schwellwert wird bestimmt, indem ein *gewichtetes Histogramm* über alle Grauwerte des Films erstellt wird. Hierbei liegt die Beobachtung zu Grunde, dass die Glottis ein dunkles Gebiet und das übrige Gewebe ein im Verhältnis dazu

Abb. 3.1: Beispiele aus Hochgeschwindigkeitsglottographien

Abb. 3.2: Beispiel einer geteilten Glottisfläche

helles Gebiet darstellt. Es ist also zu erwarten, dass sich im Grauwerthistogramm zwei Häufungen abbilden, die sich durch ein Minimum voneinander trennen. Der Grauwert dieses Minimums sollte demnach genau dem Schwellwert θ entsprechen, der die beiden Gebiete voneinander trennt. Um dieses Minimum zu verstärken, wird das Histogramm zusätzlich gewichtet, das heißt Punkte in einer homogenen Umgebung erhalten mehr Gewicht als Punkte, die in einer inhomogenen Umgebung, also zum Beispiel an einer Kante liegen.

Es stellte sich aber heraus, dass die Voraussetzungen für das Funktionieren dieses Verfahrens nicht erfüllt waren.

Zum einen hat sich das erwartete Minimum im Histogramm oftmals nicht ausgebildet, da durch eine unzureichende Ausleuchtung Gebiete am Rand des Bildes so dunkel waren, dass sie dieses Maximum trotz Gewichtung verwaschen haben. Die Bestimmung des Schwellwertes θ war also oftmals nicht automatisch möglich. Außerdem führt dies dazu, dass der dunkelste Punkt eines Bildes nicht mehr unbedingt in der Glottis liegt.

Ein weiteres ganz grundsätzliches Problem bei diesem Verfahren ist, dass es voraussetzt, dass die Glottis sich als einzelnes zusammenhängendes Gebiet darstellt.

Dies ist aber durch Schleimfäden oder Ähnliches zu verschlussnahen Zeiten oft nicht der Fall, wie in Abbildung 3.2 zu sehen. Hier würde das Verfahren nur eines der beiden Gebiete erfassen.

Diese Punkte machen das Verfahren in dieser Form ungeeignet für den Einsatz in dieser Arbeit. Daher wurde dieses zunächst weiterentwickelt [Anderson u. a. 2002]. Die Gewichtung des Histogramms wurde verbessert und nur von Bildern erstellt, bei denen die Glottis geöffnet war. Außerdem wurde das Füllverfahren für die Glottisfläche auf das Dreidimensionale erweitert, so dass auch im Falle eines geteilten Glottisgebietes alle Teilgebiete erschlossen wurden.

Dennoch war das Ergebnis noch nicht zufriedenstellend. Denn es wird nur ein einziger Schwellwert für das ganze Bild, ja sogar die ganze Aufnahme bestimmt. Das setzt aber voraus, dass die Beleuchtung sowohl über die Zeit der Aufnahme hinweg als auch örtlich im Bereich der Glottis konstant ist. Dies ist aber oft nicht der Fall. Zum einen kommt es gelegentlich zu einem Pulsieren der Helligkeit, vermutlich durch Interferenzen zwischen Lichtquellen- und Bildfrequenz. Zum anderen ist die Beleuchtungsstärke in vertikaler Bildrichtung nicht konstant, so dass es in Extremfällen dazu kommt, dass die im Hintergrund reflektierenden Ringknorpel des Kehlkopfes innerhalb der Glottisfläche am unteren Rand der Glottis heller sind als das Stimmlippengewebe am oberen Rand der Glottis. Unter solchen Umständen ist es natürlich grundsätzlich unmöglich, einen Schwellwert θ zu bestimmen, der für das ganze Bild oder gar den ganzen Film die Glottisfläche vom Rest des Bildes trennt.

Das im Folgenden beschriebene Verfahren arbeitet nicht mehr mit einer Grauwertschwelle, ist also unabhängig von einem Grauwerthistogramm und setzt auch keine gleichmäßige Beleuchtung mehr voraus. Es enthält kaum noch etwas vom ursprünglichen Verfahren, kann also als ein völlig neues Verfahren betrachtet werden.

3.2.2 Interpolation

Da die Bilddaten nach einem Schachbrettmuster nur jedes zweite Pixel beschreiben, die meisten Bildalgorithmen aber eine lückenlose kartesische Darstellung der Bilddaten voraussetzen, müssen die fehlenden Pixel interpoliert werden.

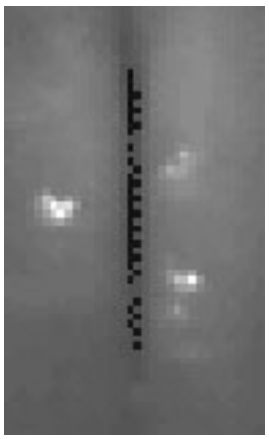


Abb. 3.3: Interpoliert mit Mittelwert (Kontrast verstärkt)

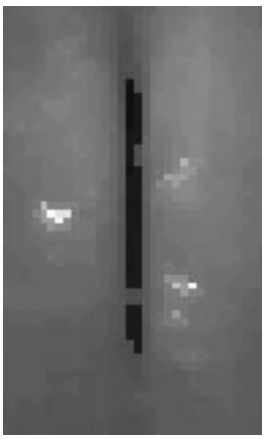


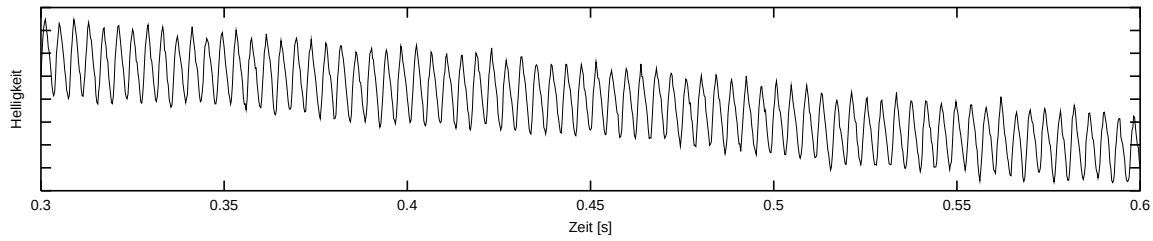
Abb. 3.4: Interpoliert mit zweitdunkelstem Wert (Kontrast verstärkt)

Interpoliert man linear zwischen den vier Nachbarpixeln, kommt es bei Pixeln, die an Kanten liegen, zu einem Kammeffekt (Abb. 3.3), das heißt, die Pixel entlang der Kante sind alternierend heller und dunkler.

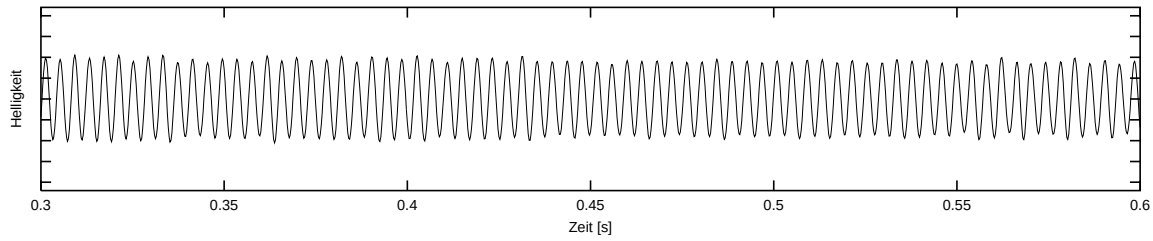
Eine andere Möglichkeit wäre, den Median zu verwenden. Da es sich hier aber um eine gerade Anzahl, nämlich vier Nachbarpixel handelt, müsste man die beiden bzgl. Helligkeit mittleren Werte mitteln oder sich nach einem weiteren Kriterium für einen der beiden entscheiden.

Es hat sich gezeigt, dass es im Fall der Glottisabbildungen am sinnvollsten ist, den dunkleren der beiden mittleren Werte zu verwenden. Das verhindert den Kammeffekt auch bei dünnen dunklen Linien, als welche sich die Glottis kurz vor oder nach dem Verschluss abbilden kann (Abb. 3.4). Dafür wird der Kammeffekt bei dünnen hellen Linien verstärkt, aber solche sind, sofern sie vorkommen, nicht im Fokus der Auswertung.

Bei Randpixeln, bei denen nur drei Nachbarpixel existieren, wird der mittlere Wert, also der gewöhnliche Median verwendet. Bei den beiden zu berechnenden Eckpixeln mit jeweils nur zwei Nachbarpixeln wird der dunklere der beiden Werte übernommen.



(a) unverändert



(b) bandpassgefiltert

Abb. 3.5: Verlauf der Gesamthelligkeit

3.2.3 Glottiszustand

Es ist hilfreich zu wissen, in welchem Zustand die Glottis zu einem bestimmten Zeitpunkt ist, also ob sie geöffnet oder geschlossen ist. Da sich die Glottis immer als ein dunkleres Gebiet darstellt als die Stimmlippen, ist die Gesamthelligkeit

$$B(t) := \sum_{x,y} p_t(x,y)$$

eines Bildes p_t zu einem Zeitpunkt t, an dem die Glottis geöffnet ist, kleiner als zu den nächstgelegenen Zeitpunkten, zu denen die Glottis geschlossen ist. Dies gilt nicht beim Vergleich zeitlich weit auseinanderliegender Glottiszustände, da eine Änderung der Beleuchtungssituation oder der Kehlkopfstellung längerfristig die Gesamthelligkeit ebenfalls beeinflusst. Dies ist beispielhaft in Abbildung 3.5(a) zu sehen.

Daher wird $B(t)$ für die zu erwartenden Grundfrequenzen von 100 - 440 Hz bandpassgefiltert, was vor allem die langsamen Helligkeitsveränderungen entfernt

(Abb. 3.5(b)).

Mit dem Schwellenwert

$$s_B := \frac{\sqrt{\frac{\sum_t B(t)^2}{N}}}{2} \qquad (N\text{: Anzahl der Zeitpunkte } t)$$

wird nun bestimmt, in welchem Zustand sich die Glottis befindet:

$$B(t) > s_B \Rightarrow \text{Glottis ist geschlossen} \qquad B(t) < -s_B \Rightarrow \text{Glottis ist geöffnet}$$

In den übrigen Fällen ist der Zustand der Glottis unbestimmt.

Dieser Zustand wird nun für jeden Zeitpunkt t gespeichert.

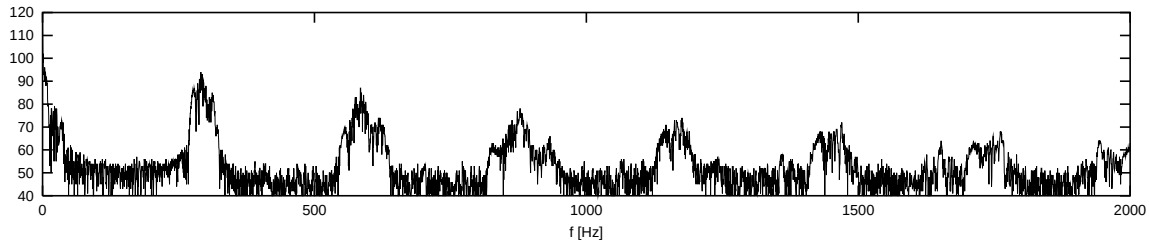
3.2.4 Region Of Interest

Bei den Hochgeschwindigkeitsaufnahmen geht es um für heutige Verhältnisse noch große Datenmengen. Ein interpolierter Film hat $256 \times 256 \times 8192$ Byte $= 512$ MB. Um bei der Verarbeitung Ressourcen zu sparen, ist es hilfreich zu wissen, in welchem Bereich sich die Glottis befindet, so dass man die Verarbeitung auf dieses Gebiet, die so genannte *Region Of Interest (ROI)* beschränken kann.

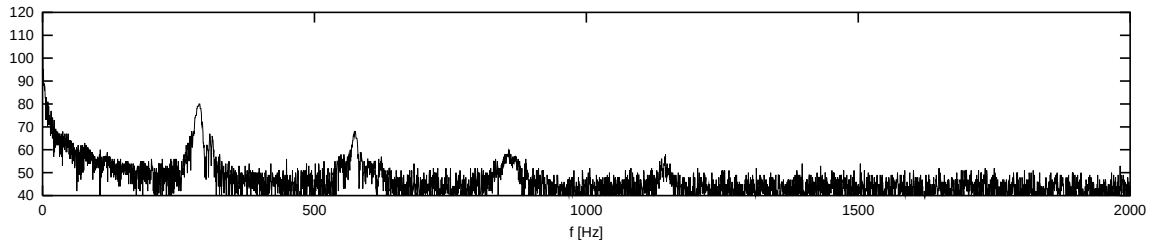
Ein wichtiges Merkmal der Glottis ist ihre Bewegung. Eine Möglichkeit diese Bewegung zu analysieren ist, die Fouriertransformierte der Zeitentwicklung eines jeden Pixels $\tilde{P}_{x,y}(f)$ zu betrachten [Granqvist und Lindestad 2001].

Bei Pixeln, über die sich während der Schwingung die Glottiskante bewegt, ist die Grundfrequenz der Schwingung im Spektrum enthalten. Allerdings trifft das nicht nur für solche Pixel zu. Es gibt auch andere Stellen im Bild, die mit derselben Frequenz schwingen, allen voran Reflexionen der Lichtquelle. Betrachtet man aber den spektralen Abfall des Spektrums eines Pixels im Gebiet der Glottis, so sieht man, dass auch noch hohe Vielfache der Grundfrequenz einen großen Anteil haben (Abb. 3.6(a)). Das ist umso mehr der Fall, je schärfer die Kante der Glottis abgebildet wird, denn dann ähnelt der Zeitverlauf eines Pixels umso mehr einer Rechteckfunktion, die viele hohe Frequenzen beinhaltet.

Im Gegensatz dazu fällt das Spektrum bei Pixeln auf pulsierenden Reflexionen sehr viel schneller ab, wie in Abbildung 3.6(b) zu sehen ist. Man kann so also zwischen



(a) Glottis



(b) pulsierende Reflexion

Abb. 3.6: Spektrum des Intensitätsverlaufs eines Pixels

diesen beiden *aktiven* Gebieten unterscheiden, indem man nur den oberen Teil des Spektrums betrachtet. So kann man die *Aktivität*

$$A(x,y) := \frac{\sum_{f=f_{\mathsf{min}}}^{f_{\mathsf{max}}} 20 \cdot \log\left(|\tilde{P}_{x,y}(f)|\right)}{f_{\mathsf{max}} - f_{\mathsf{min}} + 1} \text{ dB}$$

definieren, wobei f_{max} der halben Bildrate, also 2000 Hz entspricht, und f_{min} so gewählt wird, dass Gebiete mit pulsierenden Reflexionen niedrige Werte ergeben, das Gebiet der Glottis aber noch deutlich zum Vorschein kommt. Ein Wert von f_{min}, der einer Frequenz von 1200 Hz entspricht, hat sich hier als geeignet herausgestellt. Ein typisches *Aktivitätsbild* ist in Abbildung 3.7 zu sehen.

In diesem Aktivitätsbild gilt es nun, das größte aktive Gebiet zu finden. Dazu wird zunächst $A(x,y)$ in Bytewerte quantisiert und in einem Histogramm $H(A)$ mit 256 Stufen aufsummiert. Wie man beispielhaft in Abbildung 3.9 sieht, bildet die »Grundaktivität« in diesem Histogramm ein prominentes Maximum. So wird der

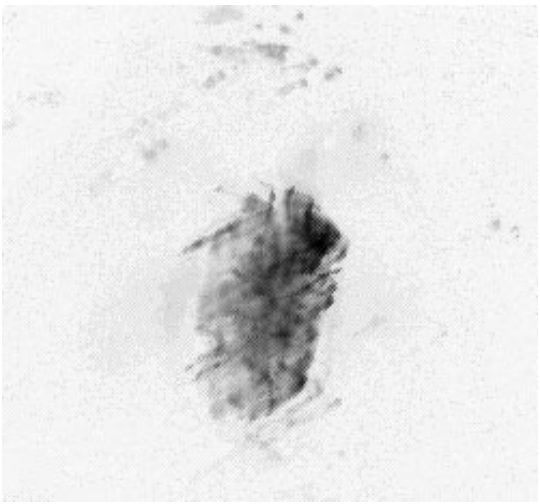


Abb. 3.7: Aktivität

Abb. 3.8: Aktivitätsmaske

Schwellwert

$$s_A := \text{argmax}(H) + \Delta_A$$

definiert, der bei geeignetem Δ_A die aktiven Gebiete vom Hintergrund trennt. Bei den hier vorliegenden Aufnahmen wurde durchgehend $\Delta_A = 2\text{dB}$ verwendet.

Bei jedem Punkt $A(x, y) > s_A$ wird ein *Flood-Fill* im Aktivitätsbild durchgeführt, der von dort aus (dem sog. *Samenpunkt*) ein Gebiet so lange wachsen lässt, bis an selbiges kein Punkt $A(x, y) > s_A$ mehr grenzt. Hierbei bezieht sich »grenzen an« auf die so genannte *4er-Umgebung*, also nur die vertikalen und horizontalen Nachbarn, das heißt diagonale Wege werden nicht beschritten. Das genaue Verfahren des Flood-Fills funktioniert analog zu dem dreidimensionalen Verfahren, das in Abschnitt 3.2.6.3 genau beschrieben wird.

Das größte so gefundene Gebiet wird als *Aktivitätsmaske* gespeichert (Abb. 3.8), und um dieses wird ein Rechteck mit einem gewissen Abstand gesetzt, das dann die ROI bildet. In dieser Arbeit wurde ein Abstand der halben Höhe bzw. Breite in der jeweiligen Richtung verwendet.

Alle weiteren Schritte werden nur noch auf die ROI angewendet.

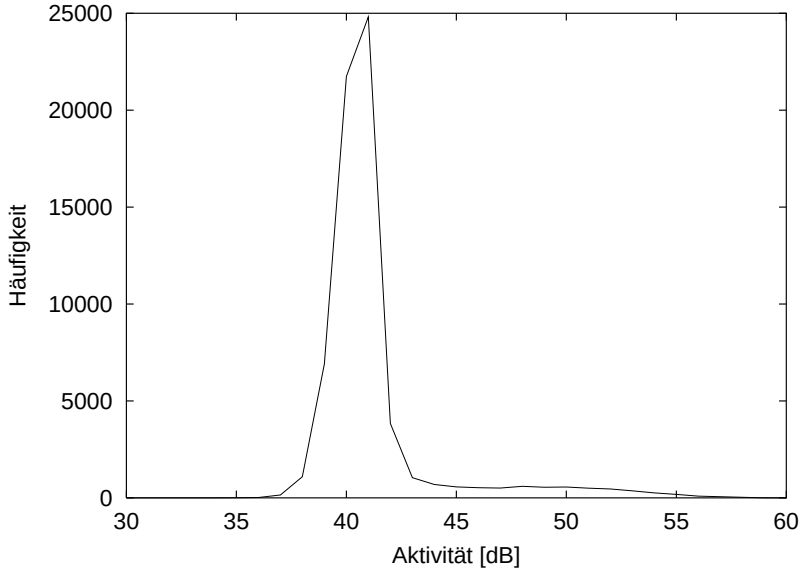


Abb. 3.9: Histogramm der Aktivität

3.2.5 Gammakorrektur

Je nach Aufnahmequalität ist es hilfreich, die Verteilung der Graustufen zu verändern. Diese Verzerrung der Helligkeitsverteilung nennt man auch *Gammakorrektur*. Mit der mathematisch definierten Gammafunktion hat dies allerdings nichts zu tun.

Bei dieser Arbeit wurde bei fast allen Aufnahmen die in Abbildung 3.10 dargestellte Funktion

$$y(x) = 255 \cdot \left(\frac{x}{255}\right)^{\frac{1}{\gamma}} \qquad \text{mit } \gamma = 2$$

verwendet, die den Kontrast – und damit auch die Kanten – in dunklen Gebieten verstärkt, zu Lasten des Kontrastes in hellen Gebieten. Da gelegentlich die Glottiskante durch Schattenwurf verdunkelt erscheint, führt dies in den meisten Fällen zu einer Verbesserung des Ergebnisses.

3.2.6 Segmentierung

Betrachtet man die Hochgeschwindigkeitsaufnahme als dreidimensionalen Raum, in dem die dritte Dimension von der Zeit gebildet wird, ergibt sich ein typisches Segmentierungsproblem, wie es zum Beispiel bei der Visualisierung der Daten von Kernspintomographie auftritt.

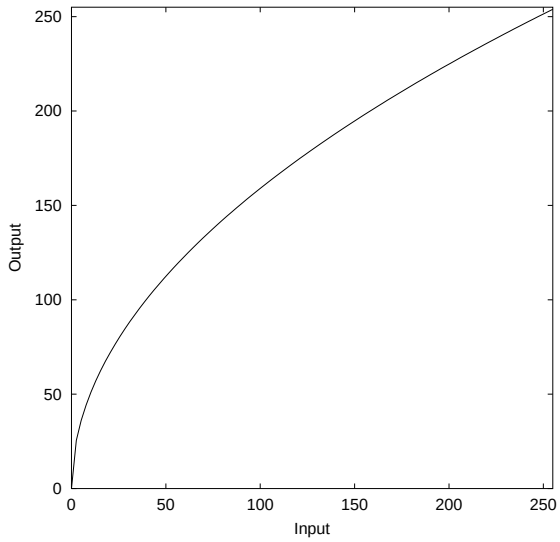


Abb. 3.10: Funktion für Gammakorrektur

Es muss klar sein, dass es nicht um die dreidimensionale Darstellung des Atemweges geht, sondern um die dreidimensionale Auftragung der zeitlich aufeinander folgenden zweidimensionalen Projektionen der Glottisebene.

Die Glottis bildet in diesem Raum einen oder mehrere »Körper«, die sich als pulsierender Schlauch oder als Blasen darstellen, je nachdem ob die Glottis sich in der Verschlussphase vollständig schließt oder nicht. Diesen Körper gilt es nun möglichst gut vom restlichen Gebiet, das vom Gewebe um die Glottis herum gebildet wird, abzugrenzen.

Um dies zu bewerkstelligen, wurde eine Variante eines konkurrierenden *Gebietswachstumsverfahren* verwendet [Nikolaidis und Pitas 2001, S. 83ff]. Hierbei werden in verschiedenen voneinander zu segmentierenden Gebieten Startpunkte, so genannte *Samenpunkte* gewählt, von denen sich gleichzeitig die Gebiete ausbreiten. Dabei wachsen die Gebiete immer an der Stelle, an der bezüglich eines noch zu definierenden Homogenitätskriteriums der geringste Widerstand herrscht. So ist sichergestellt, dass sich die unterschiedlichen Gebiete immer an der Stelle der geringsten Homogenität treffen.

3.2.6.1 Setzen der Samenpunkte

Damit das Verfahren erfolgreich sein kann, muss in jedes abgeschlossene Gebiet mindestens ein Samenpunkt gesetzt werden. Das heißt im Falle eines kompletten Glottisschlusses, dass in jeder Schwingungsperiode mindestens ein Samenpunkt zum Zeitpunkt der geöffneten Phase in das Gebiet der Glottis gesetzt werden muss. Die Information, ob die Glottis geöffnet ist, wurde in Abschnitt 3.2.3 ermittelt. So wird nun zu allen Zeitpunkten, zu denen die Glottis geöffnet ist, der Punkt innerhalb der Aktivitätsmaske aus 3.2.4 gewählt, dessen Umgebungshelligkeit

$$\overline{b}_t(x,y) = \text{Summe der Helligkeit von } p_t(x,y) \text{ und aller 8 Nachbarpixel}$$

minimal ist. Da die Glottis immer dunkler ist als die umgrenzenden Stimmlippen, kann man davon ausgehen, dass dieser Punkt immer innerhalb der Glottis liegt, was er bei den Aufnahmen dieser Arbeit auch stets tat.

Für das konkurrierende Außengebiet werden alle Randpunkte der ROI als Samenpunkte gesetzt, sofern sie außerhalb der Aktivitätsmaske liegen. Diese Einschränkung ist notwendig, da gelegentlich die Aktivitätsmaske den Rand des Bildes und somit auch den Rand der ROI berührt, der dann den vorgegebenen Abstand zu selbiger nicht einhält.

3.2.6.2 Homogenitätskriterium

Wie schon erwähnt muss ein geeignetes Homogenitätskriterium festgelegt werden, das die beiden Gebiete möglichst gut voneinander trennt. Da die Grenze zwischen Glottis und Stimmlippen eine mehr oder weniger scharfe Kante darstellt, liegt es nahe, die L_2-Norm des Gradienten der Intensität,

$$g(x,y,t) := \|\nabla p(x,y,t)\| = \sqrt{\left(\frac{\partial p}{\partial x}\right)^2 + \left(\frac{\partial p}{\partial y}\right)^2 + \left(\frac{\partial p}{\partial t}\right)^2}$$

zu verwenden. Für die partielle Ableitung wird der *dreidimensionale Sobel-Operator* verwendet [Nikolaidis und Pitas 2001, S. 95ff]. In zur Ableitungsrichtung orthogo-

nal angeordneten Matrizen hat dieser die Darstellung:

$$\begin{bmatrix} -1 & -2 & -1 \\ -2 & -3 & -2 \\ -1 & -2 & -1 \end{bmatrix}, \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}, \begin{bmatrix} 1 & 2 & 1 \\ 2 & 3 & 2 \\ 1 & 2 & 1 \end{bmatrix}.$$

Je kleiner $g(x, y, t)$ ist, umso homogener ist die 26er-Umgebung des *Voxels* $p(x, y, t)$.

3.2.6.3 Konkurrierendes dreidimensionales Gebietswachstum

Ausgehend von den festgelegten Samenpunkten sollen nun zwei Gebiete in allen drei Dimensionen heranwachsen, bis sie entweder den Rand der ROI oder das jeweils andere Gebiet erreichen. Die Grenze der beiden segmentierten Gebiete sollte dann dort liegen, wo $g(x, y, t)$ möglichst groß ist. Daher muss das Wachstumsverfahren so arbeiten, dass sich die Gebiete immer an den Stellen ausbreiten, bei denen $g(x, y, t)$ möglichst klein ist, ungeachtet um welches der beiden Gebiete es sich handelt.

Hierzu wird eine *Priority-Queue* eingerichtet, in der Voxelpositionen samt deren Gebietzugehörigkeit abgespeichert werden können. Die Priority-Queue hat hierbei die Eigenschaft, dass alle in ihr abgelegten Objekte nach ihrer Priorität sortiert vorliegen.

Als Priorität eines Voxels $p(x, y, t)$ wird der Wert $-g(x, y, t)$ verwendet, so dass das nächste zu verarbeitende Voxel in der Queue stets das mit dem geringsten Gradientenbetrag $g(x, y, t)$ ist, unabhängig davon, zu welchem der beiden Gebiete es gehört.

Das Wachstum eines Voxels geschieht hier nur in Richtung der *6er-Umgebung*, das heißt nur zu einem der sechs Nachbarn hin, die man durch Veränderung von nur einer der drei Koordinaten erreicht. Das Wachstum findet also nicht in diagonaler Richtung statt.

Unter Verwendung dieser Priority-Queue (PQ) wurde folgender Algorithmus entwickelt, der das eigentliche Gebietswachstum realisiert:

1. jedes Voxel, das einen Samenpunkt darstellt, seinem Gebiet zuordnen und in der PQ abgelegen
2. nächstes Voxel v aus der PQ holen
3. freies Voxel w der 6er-Umgebung von v auswählen
4. w dem Gebiet von v zuordnen
5. wenn es ein freies Voxel in der 6er-Umgebung von w gibt, w in der PQ ablegen
6. wenn es ein freies Voxel in der 6er-Umgebung von v gibt, v in der PQ ablegen
7. ist die PQ nicht leer, dann weiter mit 2
8. Ende

Nach Ablauf des Algorithmus ist jedes Voxel entweder dem Gebiet innerhalb oder dem außerhalb der Glottis zugeordnet.

3.2.6.4 Erosion

Die visuelle Kontrolle zeigt, dass die so gefundene Glottis im Mittel etwa um eine Pixelbreite zu groß ist (Abb. 3.11). Das lässt sich damit erklären, dass die beiden konkurrierenden Gebiete sich auf der Glottiskante, also der Stelle des stärksten Gradientenbetrags, treffen und diese Kante ungefähr die mittlere Helligkeit zwischen den beiden angrenzenden Gebieten hat. Die Glottis als »Loch« wird aber nur dort angenommen, wo lediglich eine Hintergrundhelligkeit vorhanden ist, es also sehr dunkel ist. Insofern können die Kantenpixel noch den Stimmlippen zugerechnet werden.

Daher wird eine dreidimensionale *Erosion* am Gebiet der Glottis durchgeführt [Gonzalez und Woods 1993]. Hierbei wird jedes Voxel, das zum inneren Gebiet der Glottis gehört, überprüft, ob eines seiner 26 Nachbarvoxel zum äußeren Gebiet gehört, und wenn ja, wird dieses Voxel nicht mehr als Glottisinneres, sondern als

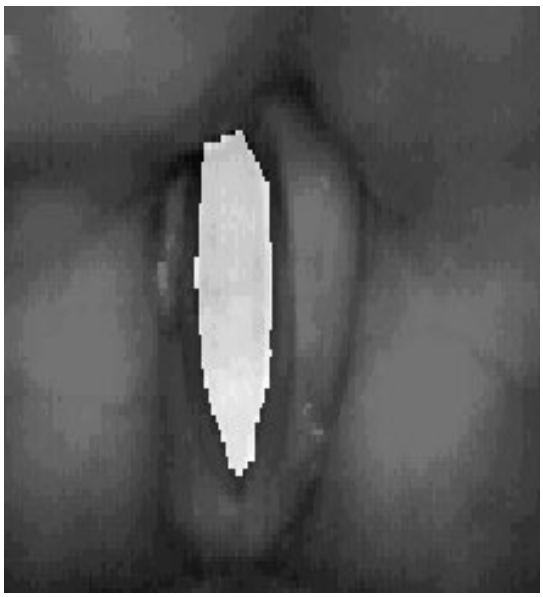


Abb. 3.11: vor Erosion

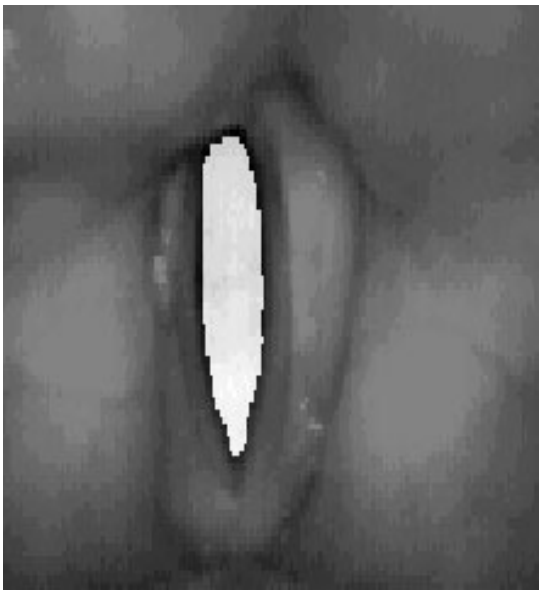


Abb. 3.12: nach Erosion

Oberfläche markiert. Dadurch wird das eigentliche Glottisgebiet etwas verkleinert und entspricht so besser dem visuellen Eindruck (Abb. 3.12).

3.2.6.5 Segmentierte Glottis

Man erhält durch das beschriebene Verfahren also das vom übrigen Bild segmentierte Glottisgebiet. In Abbildung 3.13 ist der während der Erosion ermittelte Rand des Glottisgebietes hervorgehoben. Man sieht bei diesem Beispiel sehr deutlich den Vorteil dieses Verfahrens. Durch die dreidimensionale Segmentierung werden auch in mehrere Teilgebiete unterteilte Glottisflächen vollständig erfasst.

3.2.7 Flächenverlauf

Da man nun nach der Segmentierung das Gebiet der Glottis zu jedem Zeitpunkt kennt, kann man den Flächeninhalt der Glottis zu jedem Zeitpunkt leicht errechnen, indem man alle Pixel einer Zeitebene, also eines Einzelbildes, aufaddiert. Dies führt zum *Flächenverlauf* der Glottis:

$$F_{\mathsf{gl}}(t) := \sum_{x,y} b_{\mathsf{gl}}(x, y, t)$$

Abb. 3.13: Rand der Segmentierung bei geteilter Glottisfläche

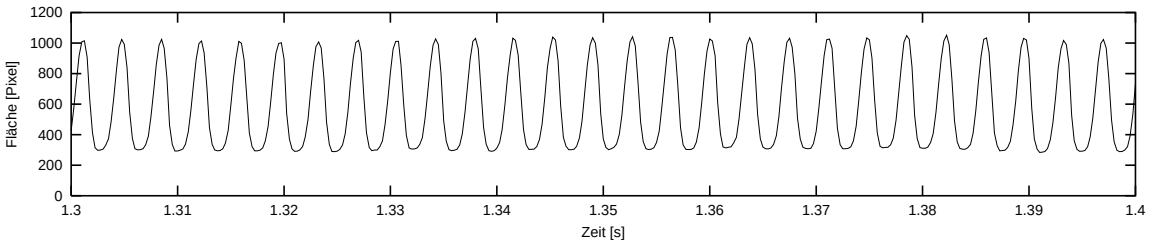
mit dem binarisierten Glottisabbild

$$b_{\mathrm{gl}}(x,y,t) := \begin{cases} 1 & \text{wenn } p(x,y,t) \text{ in der Glottis liegt,} \\ 0 & \text{sonst.} \end{cases}$$

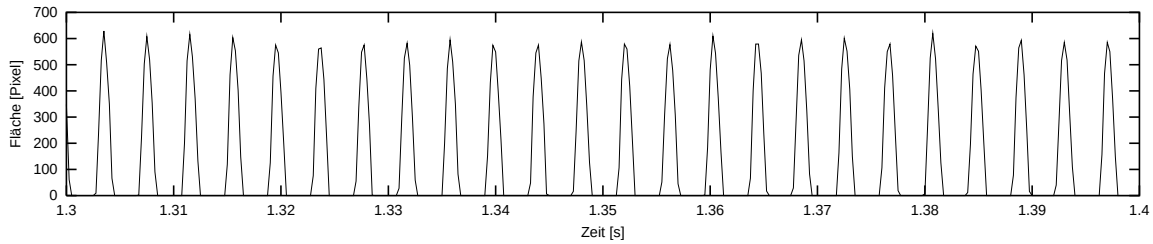
Wie man in Abbildung 3.14 sieht, ist dies eine gute Darstellung, um zu erkennen, ob die Glottis vollständig schließt. Dann nämlich sieht man regelmäßige »Nullstellenplateaus« (Abb. 3.14(b)).

Aus dem Flächenverlauf werden nun weitere hilfreiche Parameter extrahiert. Diese werden in der Weise erstellt, dass sich ihre Werte auf dasselbe Zeitfenster beziehen, auf das sich der korrespondierende GNE-Wert aus Abschnitt 3.1.4 bezieht und somit ein direkter Vergleich möglich ist.

Um diese Gleichzeitigkeit der Parameter zu erreichen, müsste hierbei grundsätzlich die akustische Laufzeit von der Glottis zu dem aufnehmenden Mikrofon beachtet werden. Bei einer durchschnittlichen Vokaltraktlänge von 17 cm wäre diese ca. 0,6 ms. Da sich die Parameter aber jeweils auf 100 ms-Fenster beziehen, kann die in Relation dazu kurze Laufzeit vernachlässigt werden.



(a) ohne Glottisschluss



(b) mit Glottisschluss

Abb. 3.14: Flächenverlauf an zwei Beispielen

3.2.7.1 Closed-Quotient

In der Medizin ist der Begriff *Closed-Quotient* geläufig. Er ist ein Maß dafür, wie lange während einer Schwingungsperiode die Glottis im Verhältnis zur Periodendauer vollständig geschlossen ist.

Die Berechnung erfolgt für jedes zu bearbeitende 100ms-Fenster in folgender Weise:

Zunächst muss eine ganze Anzahl Perioden in dem Zeitfenster gefunden werden. Dazu werden alle Zeitpunkte t_n gesucht, für die gilt:

$$F_{\mathrm{gl}}(t) \leq \overline{F}_{\mathrm{gl}} \quad \wedge \quad F_{\mathrm{gl}}(t-1) > \overline{F}_{\mathrm{gl}},$$

wobei $\overline{F}_{\mathrm{gl}}$ der gleitende Mittelwert von F_{gl} innerhalb des Fensters ist.

Ist die Anzahl N der gefunden Punkte t_n groß genug, in diesem Fall größer als 4, ergibt sich der Closed-Quotient aus

$$CQ = \frac{\text{Anzahl der } F_{\mathrm{gl}}(t) \leq s_A \text{ in } [t_0, t_{N-1}]}{t_N - t_0},$$

mit einem Schwellwert s_A. Dieser Schwellwert ist hilfreich bei dem selten beobachteten Effekt, dass trotz geschlossener Glottis in der Falte zwischen den Stimmlippen, zum Beispiel durch Schattenbildung, wenige Pixel erkannter Glottisfläche bestehen bleiben. In dieser Arbeit wurde $s_A = 5$ gewählt.

Ist die Anzahl N nicht groß genug, wird $CQ = 0$ gesetzt, da dann die Schwingung zu instabil und kein sinnvoller Wert zu erwarten ist.

So wird zu jedem $GNE(t)$ ein korrespondierender $CQ(t)$ errechnet, der sich jeweils auf dasselbe 100ms-Fenster bezieht.

3.2.7.2 Flächenminima und -maxima

Ebenfalls für jedes 100ms-Fenster wird der Mittelwert der Periodenmaxima und -minima bestimmt.

Dazu werden alle Zeitpunkte t_n gesucht, für die gilt:

$$F_{\mathsf{gl}}(t) \leq \overline{F}_{\mathsf{gl}} \quad \equiv \quad F_{\mathsf{gl}}(t-1) > \overline{F}_{\mathsf{gl}} \qquad (\equiv\text{: logische Äquivalenz}),$$

also jeweils der erste Punkt nach den Schnittpunkten von $F_{\mathsf{gl}}(t)$ mit $\overline{F}_{\mathsf{gl}}$. So wird F_{gl} in die Bereiche segmentiert, die über oder unter $\overline{F}_{\mathsf{gl}}$ liegen. Entsprechend wird in jedem Intervall $[t_n, t_{n+1}]$ nun das Maximum bzw. Minimum bestimmt und anschließend von allen Maxima und allen Minima jeweils der Mittelwert gebildet.

Diese werden nun ebenfalls zu mit $GNE(t)$ korrespondierenden Werten $MIN(t)$ und $MAX(t)$ zusammengefügt.

3.2.8 Datenbereinigung

Alle zuvor beschriebenen Verfahren machen nur Sinn bei einer schwingenden, phonierenden Glottis. Dies war nicht immer über die ganze Aufnahme hinweg der Fall, zum Beispiel wenn der Einschwingvorgang mit aufgenommen wurde. Daher wurden diese Daten bereinigt, indem alle Datensätze entfernt wurden, bei denen $MAX - MIN \leq 150$ war. Der Wert 150 ist hierbei nach Betrachtung des Datenmaterials abgeschätzt worden.

3.3 Statistische Analyse

Da es sich bei den stimmgebenden Elementen des Kehlkopfes, allen voran den Stimmlippen, um sehr komplexe, anatomisch heterogene Gebilde handelt, ist es bisher nicht möglich gewesen, ein geschlossenes mathematisches Modell zu entwickeln, das einen Zusammenhang zwischen Glottisfläche und akustischem Rauschanteil beschreiben könnte und welches man nun mit den Daten überprüfen könnte.

Außerdem sind die Werte von $MIN(t)$ und $MAX(t)$ abhängig vom Abstand zwischen Glottis und Kameraobjektiv, welcher je nach Kehlkopfstellung und Anatomie des Patienten von Aufnahme zu Aufnahme unterschiedlich ausfällt. Vergleicht man also einzelne Werte von unterschiedlichen Aufnahmen, ist mit einem von diesem Abstand abhängigen Fehler zu rechnen.

Es ist also am sinnvollsten, die Daten lediglich mit den üblichen statistischen Hilfsmitteln auf Zusammenhänge hin zu untersuchen.

3.3.1 Lineare Korrelation

Bei Datenpaaren $(x_i, y_i), i = 1, ..., N$, ist der *Korrelationskoeffizient*

$$r = \frac{\sum_i (x_i - \overline{x})(y_i - \overline{y})}{\sqrt{\sum_i (x_i - \overline{x})^2} \sqrt{\sum_i (y_i - \overline{y})^2}}$$

gegen leichte Nichtlinearitäten recht robust. Es kann hiermit also auch eine generelle Korrelation untersucht werden. Um abzuschätzen, wie verlässlich der Wert r ist, kann bei großem N das Signifikanzniveau mit dem Ausdruck

$$p_r = \text{erfc}\left(\frac{|r|\sqrt{N}}{\sqrt{2}}\right)$$

genähert werden. Es gibt an, wie wahrscheinlich es ist, dass unkorrelierte Daten einen größeren Wert $|r|$ als den beobachteten ergeben.

Ein Problem ist allerdings, dass die Signifikanzabschätzung von der eigentlichen Verteilung der Daten abhängig und daher nicht sehr verlässlich ist [Press u. a. 1992, S. 636ff].

3.3.2 Spearman-Rangkorrelation

Wenn eine eventuelle Abhängigkeit der Daten stark nichtlinear ist oder eine verlässlichere Signifikanz notwendig ist, kann eine nichtparametrische statistische Analyse bessere Ergebnisse liefern.

Bei der *Spearman-Rangkorrelation* werden nicht die eigentlichen Rohdaten kreuzkorreliert, sondern deren *Ränge*. Es werden also zunächst die Werte jedes Parameters durch ihren Rang ersetzt, wobei mehreren gleichen Werten der Mittelwert der von ihnen belegten Ränge zugewiesen wird (»corrected for ties«). Mit diesen Rängen wird dann der Korrelationskoeffizient r_s berechnet. Insofern überprüft dieser nicht einen linearen Zusammenhang, sondern lediglich, inwieweit dieser monoton ist.

Das hat den Vorteil, dass das Signifikanzniveau p_{r_s} hierbei nicht von der ursprünglichen Verteilung abhängig und immer gleich verlässlich ist. Mit

$$\nu = N - 2 \qquad \text{und} \qquad t = r_s \sqrt{\frac{\nu}{1 - r_s}}$$

ergibt sich dieses aus

$$p_{r_s} = I_{\frac{\nu}{\nu + t^2}} \left(\frac{\nu}{2}, \frac{1}{2} \right).$$

Die Funktion $I_x(a, b)$ ist hierbei die *unvollständige Betafunktion*, wie sie in Numerical Recipes von Press u. a. [1992, S. 226ff] definiert wird.

Kapitel 4

Ergebnis

Aus den 15 verwertbaren Aufnahmen wurden wie beschrieben je 20 Datensätze erstellt, die aus dem GNE, der mittleren minimalen Öffnungsfläche, der mittleren maximalen Öffnungsfläche und dem Closed-Quotient bestehen. Diese beziehen sich jeweils auf ein 100ms-Fenster, das von Datensatz zu Datensatz um 100ms weitergeschoben wird.

Nun wurden die Daten wie zuvor beschrieben bereinigt, d.h. es wurden die Datensätze entfernt, bei denen keine Schwingung der Glottis vorhanden war. Hiernach blieben 263 Datensätze für die statistische Auswertung übrig.

4.1 GNE $\Leftrightarrow$ MIN

In Abbildung 4.1 ist der GNE gegen die mittlere minimale Glottisfläche aufgetragen. Der Korrelationskoeffizient liegt bei

$$r = \mathbf{-0{,}464}$$

mit einem Signifikanzniveau von

$$p_r = \mathbf{5{,}06 \cdot 10^{-14}}.$$

Die eingezeichnete Gerade entspricht – wie auch in den folgenden Auftragungen – der Hauptachse der Streuellipse der Datenverteilung, wobei für deren Bestim-

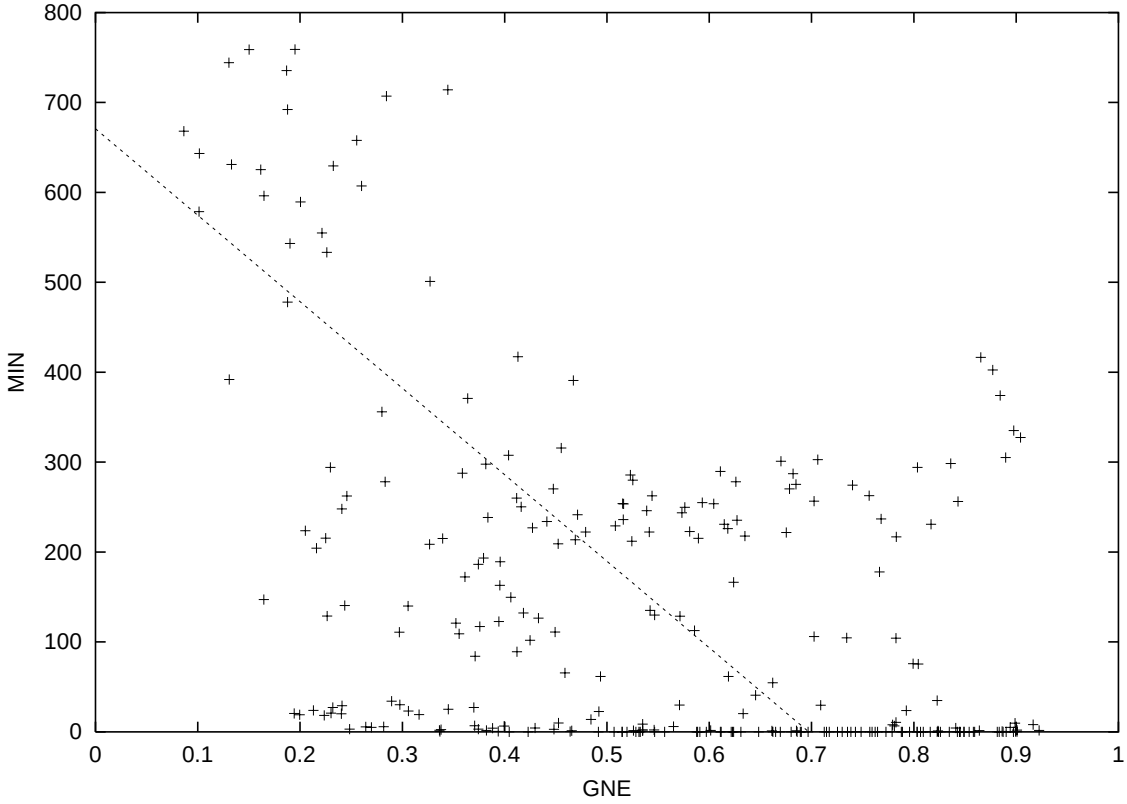


Abb. 4.1: Auftragung *GNE* gegen *MIN*

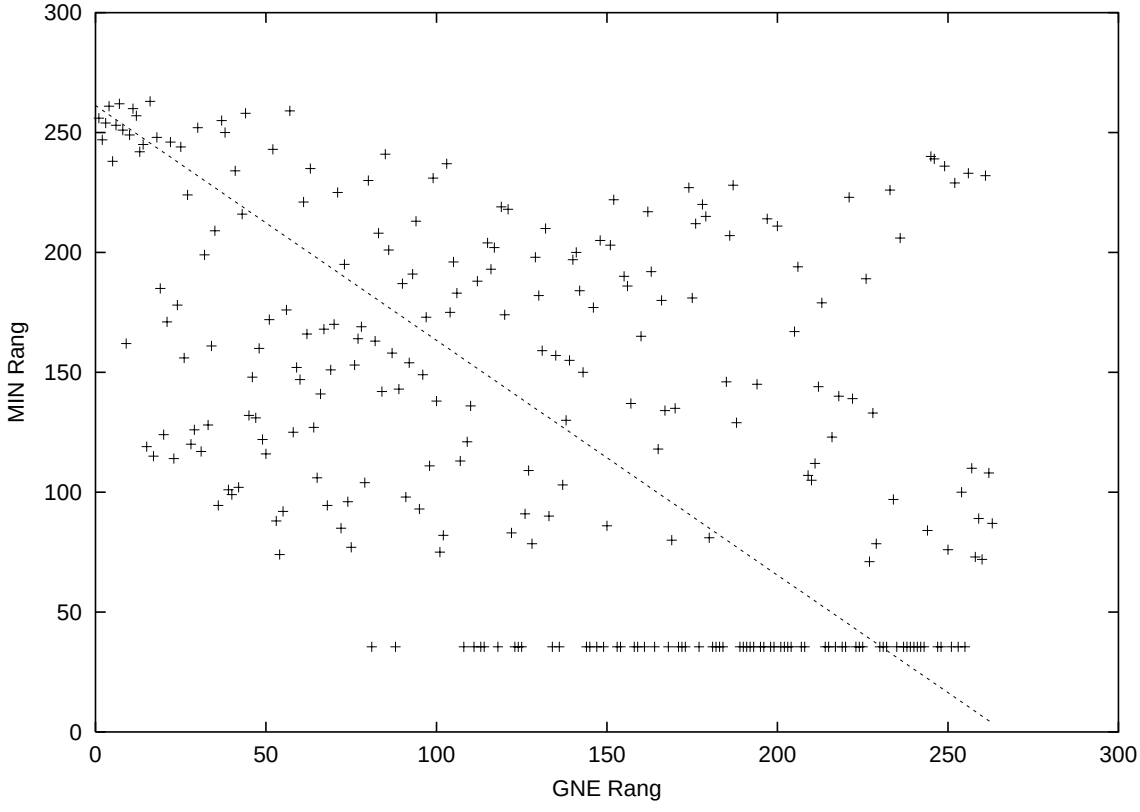


Abb. 4.2: Auftragung der Ränge von *GNE* und *MIN*

mung in diesem ersten Fall MIN mit $\max(MIN)$ normiert wurde, um die Achsen ungefähr gleich zu gewichten.

Auf Grund der unbestimmten Aussagekraft der Signifikanz bei der linearen Korrelation sind zur Kontrolle in Abbildung 4.1 die jeweiligen Ränge aufgetragen. Der Spearman-Rangkorrelationskoeffizient liegt hier bei

$$r_s = \mathbf{-0{,}459}$$

mit einem Signifikanzniveau von

$$p_{r_s} = \mathbf{3{,}94 \cdot 10^{-15}}.$$

Die vielen abgesetzten Punkte parallel zur Abszisse sind die Punkte mit $MIN = 0$, denen wie zuvor beschrieben als Rang der Mittelwert der ansonsten von ihnen belegten Ränge zugewiesen wird.

4.2 GNE ⇔ MIN/MAX

Da es sich bei MIN um eine unnormierte Größe handelt, ist damit zu rechnen, dass sie mit einem mehr oder weniger großen Fehler behaftet ist (siehe Abschnitt 3.3). Um zu überprüfen, ob die mittlere Maximalfläche der Glottis MAX eine geeignete Normierungsgröße ist, um so den Fehler zu verkleinern, ist in Abbildung 4.3 der GNE gegen das Verhältnis von mittlerer minimaler zu mittlerer maximaler Glottisfläche, also $\frac{MIN}{MAX}$ aufgetragen.

Hier liegt der Korrelationskoeffizient bei

$$r = \mathbf{-0{,}441}$$

mit einem Signifikanzniveau von

$$p_r = \mathbf{8{,}50 \cdot 10^{-13}}.$$

Auch hier werden die in Abbildung 4.4 aufgetragenen Ränge korreliert. Das ergibt

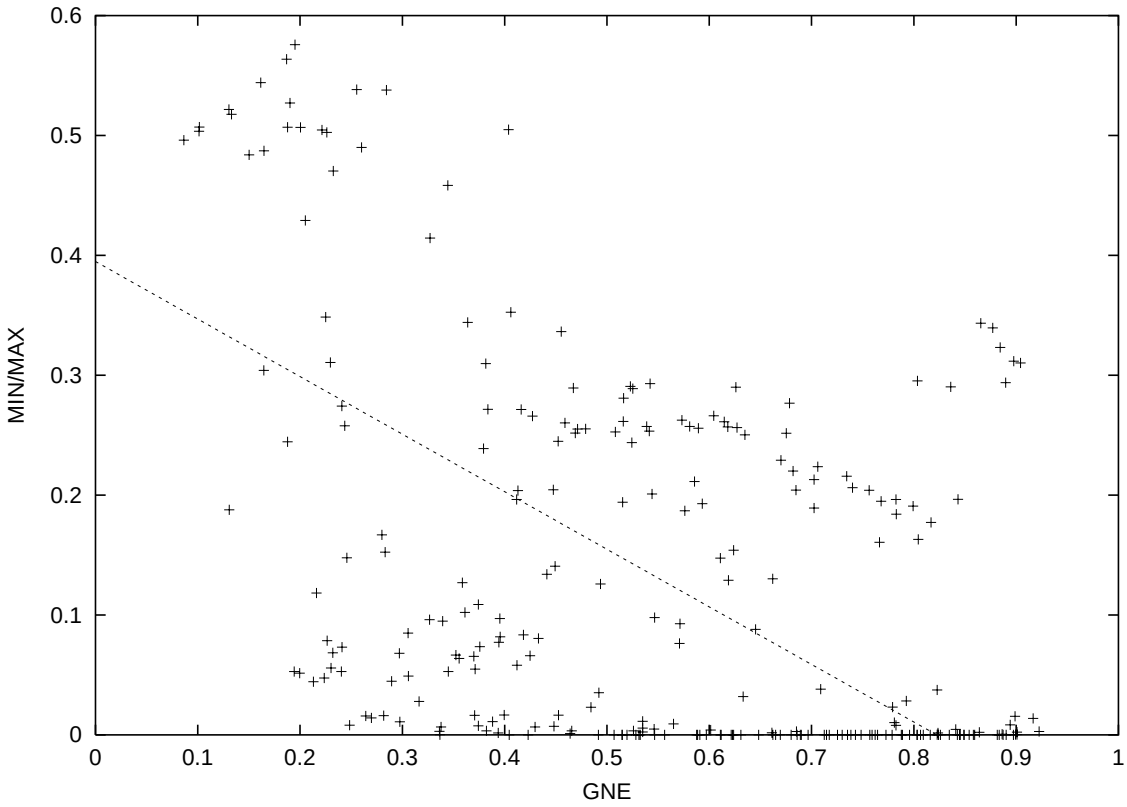


Abb. 4.3: Auftragung GNE gegen MIN/MAX

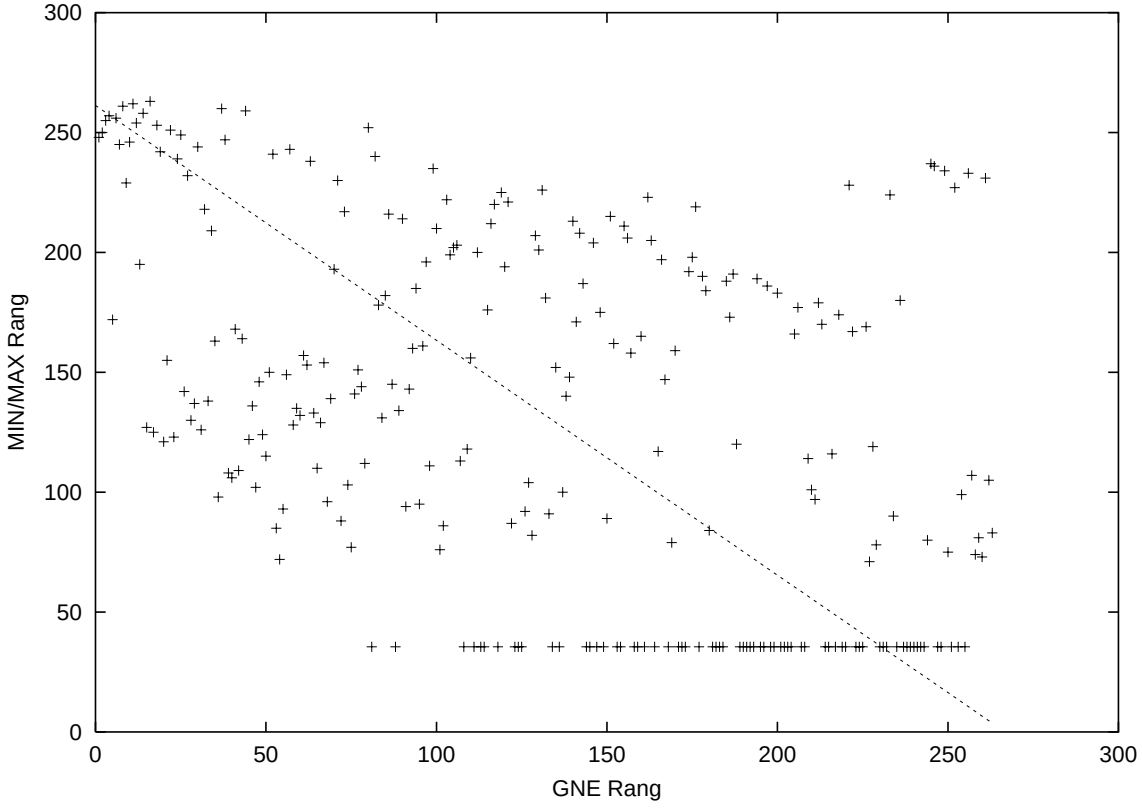


Abb. 4.4: Auftragung der Ränge von GNE und MIN/MAX

einen Korrelationskoeffizienten von

$$r_s = \mathbf{-0{,}460}$$

mit einem Signifikanzniveau von

$$p_{r_s} = \mathbf{3{,}83 \cdot 10^{-15}}.$$

4.3 GNE ⇔ CQ

In Abbildung 4.5 ist schließlich der GNE gegen den Closed-Quotient aufgetragen. Der Korrelationskoeffizient liegt bei

$$r = \mathbf{0{,}536}$$

mit einem Signifikanzniveau von

$$p_r = \mathbf{3{,}32 \cdot 10^{-18}}.$$

Die Rangkorrelation aus Abbildung 4.6 ergibt schließlich

$$r_s = \mathbf{0{,}510}$$

und

$$p_{r_s} = \mathbf{7{,}71 \cdot 10^{-19}}.$$

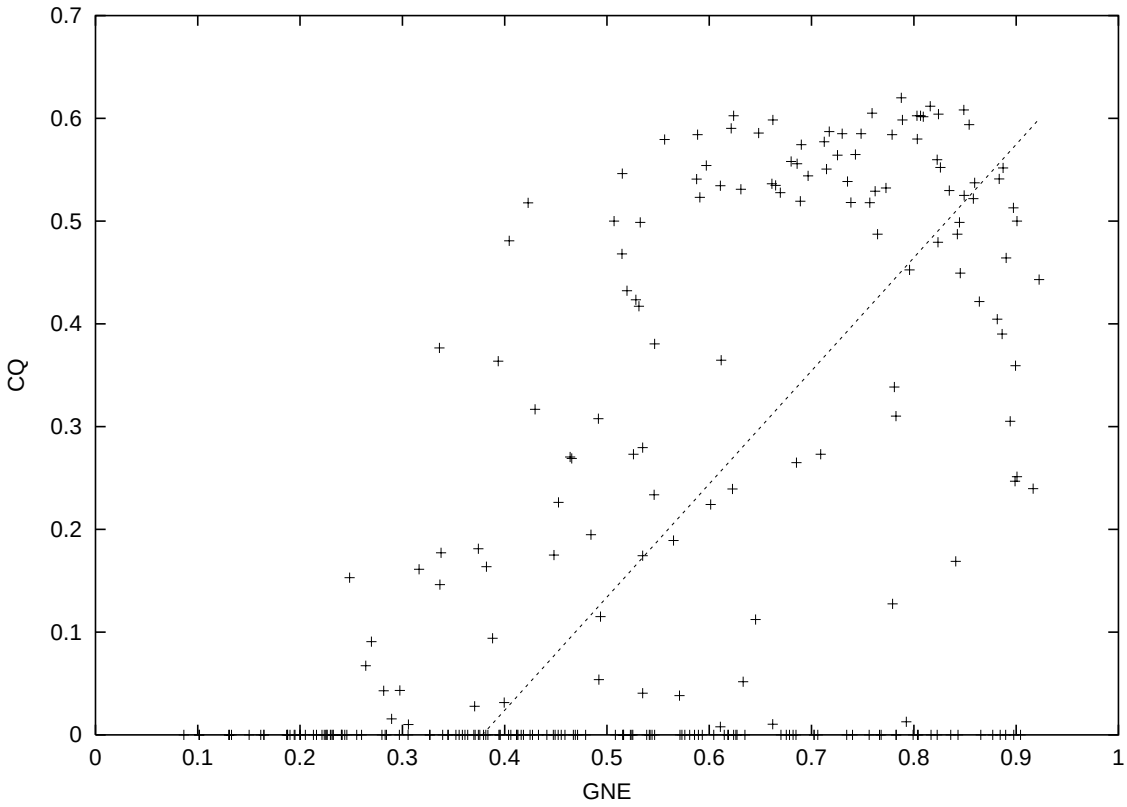


Abb. 4.5: Auftragung GNE gegen CQ

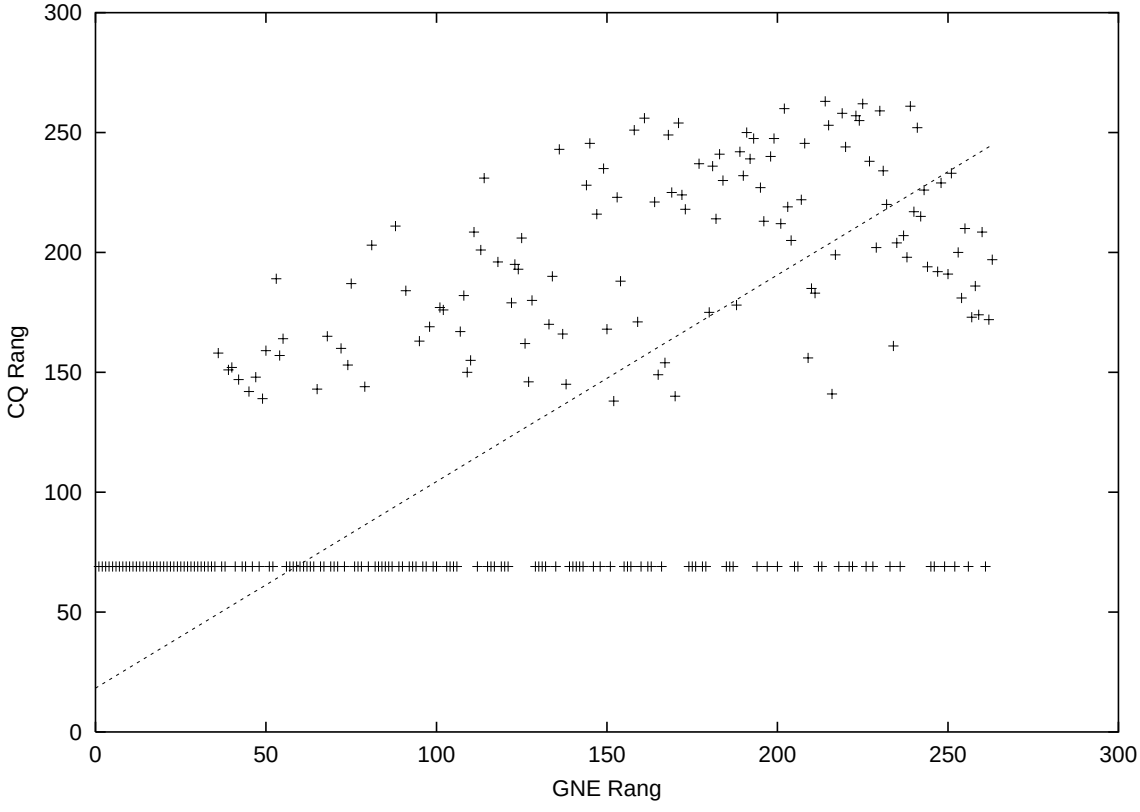


Abb. 4.6: Auftragung der Ränge von GNE und CQ

Kapitel 5

Diskussion

Die Auswertung der Daten ergibt, dass bei allen untersuchten Beziehungen eine deutliche Korrelation vorhanden ist. Vor allem die äußerst niedrigen Signifikanzniveaus der Spearman-Rangkorrelationen belegen dies.

Nachdem man weiß, dass die Daten grundsätzlich korrelieren, zeigen die ebenfalls sehr niedrigen Signifikanzniveaus bei der linearen Korrelation, dass eine gewisse Linearität in der Datenstruktur durchaus vorhanden ist. Allerdings sieht man an den Graphen, dass die Streuung doch so stark ist, dass hier andere Faktoren noch eine erhebliche Rolle spielen müssen.

5.1 GNE $\Leftrightarrow$ MIN

Man sieht auf Abbildung 4.1 deutlich, dass man von einer außergewöhnlich großen Restöffnungsfläche auf einen geringen GNE, also einen hohen Rauschanteil schließen kann. Umgekehrt gibt es aber bei einer Restöffnungsfläche von <300 Pixel ein breites Spektrum an GNE-Werten. Erst bei vollständigem Glottisschluss ($MIN = 0$) erkennt man eine Häufung bei höheren GNE-Werten, kann man also auf einen geringeren Rauschanteil schließen.

Durch die fehlende Normierung von MIN auf ein absolutes Längenmaß könnte hinter dieser Messung aber auch ein Zusammenhang zwischen GNE und dem Abstand Objektiv↔Glottis stecken, denn je größer dieser ist, desto kleiner erscheint eine Fläche gleichen Ausmaßes auf dem Bild.

5.2 GNE ⇔ MIN/MAX

Zwischen dem GNE und $\frac{MIN}{MAX}$ ist der Betrag des Korrelationskoeffizienten $|r|$ leicht geringer als bei MIN ohne Normierung. Die starke Streuung kann so also nicht vermindert werden und hat andere Ursachen. Da der Korrelationskoeffizient aber kaum abweicht und die Verteilung der Daten nahezu unverändert bleibt, wie man bei Vergleich von Abbildung 4.1 mit 4.3 sieht, heißt das auch, dass die Korrelation nicht vom Abstand zwischen Glottis und Objektiv kommen kann, denn von diesem hängt ebenso MIN wie MAX ab, dieser Effekt würde sich hier also aufheben.

Dies ist ein starkes Indiz dafür, dass die Korrelation tatsächlich auch mit der absoluten Fläche vorhanden ist.

5.3 GNE ⇔ CQ

Die Korrelation zwischen GNE und Closed-Quotient ist die deutlich stärkste unter den untersuchten. Da der CQ lediglich ein Zeitverhältnis ist, hängt er nicht vom Abstand zwischen Glottis und Objektiv ab, und es ist diesbezüglich kein Fehler enthalten.

Man erkennt deutlich in Abbildung 4.5, dass man von einem hohen CQ auf einen hohen GNE, also niedrigen Rauschanteil schließen kann. Ebenfalls kann man von einem niedrigen GNE, also hohen Rauschanteil darauf schließen, dass die Glottis nur kurz oder überhaupt nicht schließt.

Bei vorgegebenem hohen GNE oder niedrigen CQ ist die Streuung des jeweils anderen Parameters allerdings recht hoch und lässt keine Aussage über jenen zu, auch wenn der Mittelwert dieser Streuung jeweils im Sinne des linearen Zusammenhangs, das heißt in der Nähe der Streuachse liegt.

Kapitel 6

Ausblick

Die Hochgeschwindigkeitsglottographie wird die klassische stroboskopische Laryngoskopie ablösen, das zumindest ist die Meinung vieler Ärzte und Wissenschaftler. Hierin zeigt sich eine Entwicklung, wie sie in sehr vielen Bereichen vergleichbar abläuft. Diese führt zu dem allgemeinen Problem, dass die ungleich größeren Datenmengen, die bei solchen neueren Instrumenten anfallen, nicht mehr in Rohform begutachtet werden können, denn dies würde jeden Begutachter überfordern.

Daher ist es umso wichtiger, Verfahren zu entwickeln, die aus den zusätzlichen Daten und Informationen, die man durch ein solches neues Messinstrument hinzugewinnt, die Informationen herausziehen, die für die jeweilige Anwendung von Bedeutung sind und an Umfang den Betrachter nicht überfordern.

Ein Beispiel eines solchen Verfahrens zeigt diese Arbeit. Aus dem Hochgeschwindigkeitsfilm wird ein Teilaspekt – die Fläche der Glottis – vollautomatisch extrahiert. Dies allein ist schon eine massive Reduktion der Daten auf einen bestimmten Aspekt, der von Bedeutung ist. Dieser wird wiederum auf seine zeitliche Entwicklung hin untersucht, um daraus auf objektive Weise Parameter zu bestimmen, die zuvor mit den klassischen Untersuchungsmethoden höchstens subjektiv von einem Arzt abgeschätzt werden konnten.

So weist diese Arbeit nach, dass der Behauchtheitsgrad einer Stimme, welcher durch den GNE abgebildet wird, tatsächlich mit dem Verschlussgrad der schwingenden Glottis – also dem Closed-Quotient bei vollständig schließender Glottis und der Restöffnungsfläche bei nicht vollständigem Glottisverschluss – in Zusammenhang steht, wenn auch mit einer recht großen Streuung. Ein Sachverhalt, der zwar

schon lange beobachtet wurde, aber nie wirklich gemessen werden konnte.

Das hier vorgestellte Verfahren der Glottisfindung kann hierbei aber nur als erster Schritt gesehen werden. Es ist in vielerlei Hinsicht verbesserungswürdig.

Mit einem für heutige Verhältnisse sehr gut ausgerüsteten Computer dauert die Berechnung einer einzigen Aufnahme um die 30 Minuten. Das ist für die wissenschaftliche Datenanalyse wie bei dieser Arbeit durchaus vertretbar. Aber das sind Zeiten, die im klinischen Alltag in der Phoniatrie inakzeptabel sind, da der Informationsgewinn diese nicht rechtfertigen würde. Bei der rasanten Entwicklung der Rechenleistung ist dies aber nur eine Frage der Zeit.

Das Verfahren bestimmt die Glottisfläche für die Anwendung in dieser Arbeit hinreichend gut. Allerdings wird in keiner Weise die Form des Glottisrandes überprüft. Hier kann durch Ansätze wie dem der *aktiven Konturen* [Kass u. a. 1987] eine genauere Entsprechung des Glottisrandes gefunden werden [Marendic 2001], was bei anderen zu ermittelnden Parametern als dem Flächeninhalt hilfreich und wichtig sein kann.

Ebenfalls wünschenswert wäre eine Extraktion weiterer Parameter, wie sie von Wittenberg [1998] bereits in Grundzügen beschrieben wurde. Dazu gehört u.a. eine sinnvolle Achsenfindung der Glottis, die getrennte Betrachtung der beiden Stimmlippen und die Konturenbestimmung der Stimmlippen in Abgrenzung zum benachbarten Gewebe. Hiermit können dann weitergehende Analysen durchgeführt werden, wie z.B. die Ermittlung der *empirischen orthogonalen Funktionen* der Stimmlippenkanten [Neubauer u. a. 2001], mit denen funktionelle Stimmstörungen untersucht werden können.

Und schlussendlich ist eine geeignete Visualisierung der gewonnen Daten unerlässlich. Nach dem Motto »Ein Bild sagt mehr als tausend Worte« kann eine geschickte Visualisierung von Daten gerade bei komplexen Sachverhalten eine äußerst effiziente Methode zur Übermittlung an den Menschen sein. Diese kann letzten Endes aber nur in Zusammenarbeit mit den Ärzten und der Medizin geschehen, mit denen neue Visualisierungs- und Diagnosemethoden entwickelt werden können.

Anhang A

Quellcode

Zu Dokumentationszwecken ist hier der Quellcode der Implementationen der in dieser Arbeit beschrieben Verfahren abgedruckt. Dieser ist in ISO-C++ geschrieben. Er basiert allerdings auf der Signalverarbeitungs-Bibliothek Leaf, die in der Arbeitsgruppe entwickelt wurde, in der auch diese Arbeit entstanden ist. Insofern ist der Code nicht direkt verwendbar. Aber er stellt dennoch dar, wie die verschiedenen Algorithmen realisiert wurden, und sollte leicht auf andere Signalklassen portierbar sein.

A.1 Glottissegmentierung

```
#include "LHGGProcessing.h"
#include "LSignalfile.h"
#include "LRealFFTW.h"
#include "LInverseRealFFTW.h"
#include "LChangeSamplingrate.h"
#include <qprogressdialog.h>
#include <vector>
#include <queue>
#include <deque>
#include <iostream>
#include <iomanip>


// 3D-Sobel-Matrizen

const int sobelT[] = {
 -1,-2,-1,
 -2,-3,-2,
 -1,-2,-1,

  0, 0, 0,
  0, 0, 0,
  0, 0, 0,

  1, 2, 1,
  2, 3, 2,
  1, 2, 1
};

const int sobelX[] = {
 -1,-2,-1,
  0, 0, 0,
  1, 2, 1,

 -2,-3,-2,
  0, 0, 0,
  2, 3, 2,

 -1,-2,-1,
  0, 0, 0,
  1, 2, 1
};

const int sobelY[] = {
 -1, 0, 1,
 -2, 0, 2,
 -1, 0, 1,

 -2, 0, 2,
 -3, 0, 3,
 -2, 0, 2,

 -1, 0, 1,
 -2, 0, 2,
 -1, 0, 1
};


class voxel
{
public:
  unsigned short t;
  unsigned short x;
  unsigned short y;
  float grad;
  direction gradDir;
  segmentType st;
```

```
// Konstruktor
voxel (
     unsigned short _t,
     unsigned short _x,
     unsigned short _y,
     segmentType _st
     ):
 t(_t),
 x(_x),
 y(_y),
 st(_st)
{}


bool operator< (const voxel& v) const { return grad > v.grad; }

inline void calcGrad ( const LSignal<unsigned char>& si )
{
 unsigned int ix=0;
 short gradX=0;
 short gradY=0;
 short gradT=0;

 for (int dt=-1; dt<2; dt++)
  for(int dx=-1; dx < 2; dx++)
   for(int dy=-1; dy < 2; dy++)
    {
     if (sobelX[ix])
      gradX += si(t+dt,x+dx,y+dy)*sobelX[ix];
     if (sobelY[ix])
      gradY += si(t+dt,x+dx,y+dy)*sobelY[ix];
     if (sobelT[ix])
      gradT += si(t+dt,x+dx,y+dy)*sobelT[ix];
     ix++;
    }

 grad = sqrt(LSquare(gradX)+LSquare(gradY)+LSquare(gradT));
 //grad = abs(gradX)+abs(gradY)+abs(gradT);
 //grad = LMaximum(abs(gradX),LMaximum(abs(gradY),abs(gradT)));

 return;
}

inline bool calcGDir ( const LSignal<unsigned char>& gl )
{
 if (gl(t,x+1,y) == NONE)
  gradDir = XPLUS;
 else if (gl(t,x-1,y) == NONE)
  gradDir = XMINUS;
 else if (gl(t,x,y+1) == NONE)
  gradDir = YPLUS;
 else if (gl(t,x,y-1) == NONE)
  gradDir = YMINUS;
 else if (gl(t+1,x,y) == NONE)
  gradDir = TPLUS;
 else if (gl(t-1,x,y) == NONE)
  gradDir = TMINUS;
 else
  return (false);
 return (true);
}
};

struct point
{
 short x;
 short y;
};


LSignal<unsigned char> LGlottisDetect(
                            const LSignal<unsigned char> &si,
                            unsigned char randtyp = GLOTTIS,
                            unsigned int f_g1 = 100,
                            unsigned int f_g2 = 440,
                            float     baselevel = 0,
                            bool      debug   = false
                            )
{


 /*********************************
  * Definitionen
  *********************************/

 const float act_f = 0.6;     // ab welcher Frequenz Aktivität messen
 const float act_l = 2;       // Schwellwert für ROI in dB
 const unsigned int cycles = 10; // Wieviele Zyklen pro Segmentierung
 const unsigned int t_diff_max = 1000; // Maximale Segment-Länge
 const float ROI_w = .5;      // Breite des Randes der ROI


 // 6er-Umgebung

 const bool neighbour6[] = {
  0,0,0,
  0,1,0,
  0,0,0,

  0,1,0,
  1,0,1,
  0,1,0,

  0,0,0,
  0,1,0,
  0,0,0
 };


 // 26er-Umgebung

 const bool neighbour26[] = {
  1,1,1,
  1,1,1,
  1,1,1,

  1,1,1,
  1,0,1,
  1,1,1,
```

```
  1,1,1,
  1,1,1,
  1,1,1
};


vector<unsigned char> gStat(si.length(0));

// Helligkeitsverlauf

LSignalDimensions sd1(1);
sd1.signalDimension(0)=( si.signalDimension(0));
sd1.setY( "Helligkeit", "AU", 1.0);
LSignal<float> brightness(sd1);
for ( unsigned long t=0; t<si.length(0); t++) brightness(t)=0;


// Flaechenverlauf

sd1.signalDimension(0)=( si.signalDimension(0));
sd1.setY( "Fläche", "Pixel", 1.0);
LSignal<float> area(sd1);
for ( unsigned long t=0; t<si.length(0); t++) area(t)=0;


// Histogramm

sd1.setX( 0, 256, 1, 0, "Helligkeit", "AU");
sd1.setY( "Gewicht", "AU", 1.0);
LSignal<long> histo(sd1);
for ( int i=0; i<256; i++) histo(i)=0;

float brightnessmean=0;

// Aktivitaetsbild

LSignalDimensions sd2(2);
sd2.signalDimension(0)=(si.signalDimension(1));
sd2.signalDimension(1)=(si.signalDimension(2));
sd2.setY( "Aktivität", "dB", 1.0);
LSignal<float> actpic(sd2);


for (unsigned long x=0;x<si.length(1);x++)
  for (unsigned long y=0;y<si.length(2);y++)
    actpic(x,y)=0;

// Glottis-Film
LSignal<unsigned char> region(si.signalDimensions());
for (unsigned long t=0;t<si.length(0);t++)
  for (unsigned long x=0;x<si.length(1);x++)
    for (unsigned long y=0;y<si.length(2);y++)
      region(t,x,y) = NONE;

// Maske
LSignal<bool> mask(si.reduce( LIndex3( 0, -1, -1)).signalDimensions());


unsigned int acthistomax;


/*******************************************
 * Begin
 *******************************************/


if (si.dimensions() != 3)
  {
    cerr << "Falsche Dimension! (kein Film?)\n";
    return region;
  }

//
// Brightness berechnen
//

if (debug)
  cerr << endl << "Brightness... Frame:     ";

for (unsigned long t=0;t<si.length(0);t++)
  {
    if (debug)
      cerr << "\b\b\b\b" << setw(4) << t << flush;
    for (unsigned short x=0;x<si.length(1);x++)
      for (unsigned short y=0;y<si.length(2);y++)
        brightness(t) += si(t,x,y);
  }

if (debug)
  brightness.writeFile("bri_orig.sig");

{
  unsigned long g1start, g1stop, g2start, g2stop;
  LRealFFTW fft(brightness.length(0),true);
  LInverseRealFFTW ifft(brightness.length(0),true);
  LSignal<LComplex<float> > tmp = fft(brightness.toReal());
  g1start = int((f_g1-25)*tmp.length(0)*2/brightness.samplingRate());
  g1stop = int((f_g1+25)*tmp.length(0)*2/brightness.samplingRate());
  g2start = int((f_g2-25)*tmp.length(0)*2/brightness.samplingRate());
  g2stop = int((f_g2+25)*tmp.length(0)*2/brightness.samplingRate());

  for (unsigned long i=0; i<g1start; i++)
    {
      tmp(i) *= baselevel;
    }
  for (unsigned long i=g1start; i<g1stop; i++)
    {
      tmp(i) *= 1-(1+cos((i-g1start)*M_PI/(g1stop-g1start-1)))/2*(1-baselevel);
    }
  for (unsigned long i=g2start; i<g2stop; i++)
    {
      tmp(i) *= 1-(1-cos((i-g2start)*M_PI/(g2stop-g2start-1)))/2*(1-baselevel);
    }
  for (unsigned long i=g2stop; i<tmp.length(0); i++)
    {
      tmp(i) *= baselevel;
    }

  brightness = ifft(tmp);
  if (debug)
    brightness.writeFile("bri_mod.sig");
```

```
315   }


      //
      // Zeitpunkte der offenen/geschlossenen Glottis bestimmen
320   //


      if (debug)
        cerr << endl << "Offene␣Glottis␣finden...";
325
      float stddev=0;

      for (unsigned long t=0; t<si.length(0);t++)
        {
330       stddev += LSquare(brightness(t));
        }

      stddev = sqrt(stddev/(si.length(0)))/2;

335
      if (debug)
        cerr << endl << "Standardabweichung:␣" << stddev;

      //
340   // offene Glottis finden
      //

      for (unsigned long t=0; t<si.length(0);t++)
        {
345       if ( brightness(t) < -stddev )
          {
            // Glottis offen!
            gStat[t]|=GOPEN;
          }
350
          if ( brightness(t) > stddev )
          {
            // Glottis geschlossen!
            gStat[t]|=GCLOSED;
355       }
        }

      brightness.writeFile("bri.sig");

360   //
      // Aktivitaet
      //

      if (debug)
365     cerr << endl << "Aktivitaet...␣␣␣␣␣";

      {
        LRealFFTW fft(si.length(0),true);
        LSignal<LComplex<float> > tmp( fft.dimensions( si.reduce( LIndex3( -1, 0, 0))\
                                      .toReal()));
370
        for ( int i=0; i<256; i++) histo(i)=0;

        for (unsigned long x=0;x<si.length(1); x++)
          {
375         for (unsigned long y=0;y<si.length(2); y++)
              {
                fft.transform(si.reduce( LIndex3(-1, x, y)).toReal(), tmp);
                for (unsigned long f=(unsigned long)(tmp.length(0)*act_f); f<tmp.length\
                                                  (0)-1;f++)
                  actpic(x, y) += LdB(LAbsolute(tmp(f)), 0);
380
                actpic(x,y) /= tmp.length(0)-1-(unsigned long)(tmp.length(0)*act_f);
                histo((unsigned char)(actpic(x,y)))++;
              }
            if (debug)
385           cerr << "\b\b\b\b" << setw(3) << (x+1)*100/si.length(1) << "%" << flush;
          }
        if (debug)
          histo.writeFile("acthist.sig");

390     if (debug)
          actpic.writeFile("actpic.sig");

        acthistomax = histo.argmax()(0);
        if (debug)
395       cerr << endl << "Acthistomax:␣" << acthistomax;
      }


      //
400   // ROI
      //


      if (debug)
405     cerr << endl << "ROI:␣";

      unsigned short x_max=0, y_max=0, x_min=0, y_min=0;
      {

410     for (unsigned long X=0;X<mask.length(0); X++)
          for (unsigned long Y=0;Y<mask.length(1); Y++)
            mask(X,Y) = false;

        deque<point> activePt;
415
        unsigned long max=0;
        for (unsigned long x=0;x<si.length(1); x++)
          for (unsigned long y=0;y<si.length(2); y++)
            if (actpic(x, y) > signed(acthistomax + act_l) && !mask(x,y))
420           {
                unsigned long sum=1;


                point startPt = { x, y };

425             activePt.push_back(startPt);

                mask(x, y) = true;

                while(! activePt.empty())
430               {
                    unsigned int ix=9;
                    for(short dx=-1; dx < 2; dx++)
                      for(short dy=-1; dy < 2; dy++)
```

```
                {
                  if( ! neighbour6[ix++] )
                    continue;

                  point p = activePt.front();
                  p.x += dx;
                  p.y += dy;

                  if( p.x >= 0 && p.y >= 0 &&
                     p.x < short(si.length(1)) &&
                     p.y < short(si.length(2)) &&
                     !mask(p.x, p.y) &&
                     actpic(p.x, p.y) > acthistomax + act_l
                     )
                    {
                      mask(p.x, p.y) = true;
                      sum++;
                      activePt.push_back(p);
                    }
                }
              activePt.pop_front();
            }

          if ( max < sum )
            {
              x_max = x;
              y_max = y;
              max = sum;
            }
        }

for (unsigned long X=0;X<mask.length(0); X++)
  for (unsigned long Y=0;Y<mask.length(1); Y++)
    mask(X,Y) = false;

x_min = x_max;
y_min = y_max;

point startPt = { x_max, y_max};

activePt.push_back(startPt);

mask(x_max, y_max) = true;

while(! activePt.empty())
  {
    unsigned int ix=9;
    for(short dx=-1; dx < 2; dx++)
      for(short dy=-1; dy < 2; dy++)
        {
          if( ! neighbour6[ix++] )
            continue;

          point p = activePt.front();
          p.x += dx;
          p.y += dy;

          if( p.x >= 0 && p.y >= 0 &&
             p.x < short(si.length(1)) &&
             p.y < short(si.length(2)) &&
             !mask(p.x, p.y) &&
             actpic(p.x, p.y) > acthistomax + act_l
             )
            {
              mask(p.x, p.y) = true;
              x_max = LMaximum(x_max, p.x);
              x_min = LMinimum(x_min, p.x);
              y_max = LMaximum(y_max, p.y);
              y_min = LMinimum(y_min, p.y);
              activePt.push_back(p);
            }
        }
    activePt.pop_front();
  }

 if (debug)
  mask.writeFile("mask.sig");

 int xd = int((x_max - x_min)*ROI_w);
 int yd = int((y_max - y_min)*ROI_w);
 x_max = LMinimum(x_max+xd, (unsigned short)(si.length(1)-2));
 x_min = LMaximum(x_min-xd, (unsigned short)(1));
 y_max = LMinimum(y_max+yd, (unsigned short)(si.length(2)-2));
 y_min = LMaximum(y_min-yd, (unsigned short)(1));

 if (debug)
  cerr << "x: " << x_min << "-" << x_max << " y: " << y_min << "-" << y_max;

}


//
// Kontrast-Verstärkung
//

if (debug)
  cerr << endl << "Gammakorrektur...";

for (unsigned short t=0; t<si.length(0); t++)
  for (unsigned short x=x_min-1; x<x_max+2; x++)
    for (unsigned short y=y_min-1; y<y_max+2; y++)
      si(t,x,y) = (unsigned char)(15.9*sqrt(si(t,x,y)));


//
// Füll-Schleife
//

unsigned long t_min,t_max;

t_max=2; // Erstes Bild ist kaputt.

do
  {
    t_min=t_max;

    for (unsigned int z=0; z < cycles; z++)
      {
        while( t_max < si.length(0)-2 )
          if (t_max - t_min > t_diff_max - 2 ||
             gStat[++t_max] & GOPEN && ~gStat[t_max+1] & GOPEN )
            break;
```

```
  }

if (debug)
  cerr << endl << "Abschnitt t=" << t_min << "-" << t_max;

priority_queue<voxel> av;

for (unsigned short x=x_min; x<x_max+1; x++)
  for (unsigned short y=y_min; y<y_max+1; y++)
    {
      region(t_min+1,x,y)=NONE;
      if (segmentType st=segmentType(region(t_min,x,y) & (GLOTTIS|OUTER)))
        {
          voxel v(t_min, x, y, st);
          v.calcGrad(si);
          if (v.calcGDir(region))
            av.push(v);
        }
    }


unsigned short gSeedX=0, gSeedY=0, gSeedMin=2550;
unsigned short lSeedX=0, lSeedY=0, lSeedMax=0;
unsigned short rSeedX=0, rSeedY=0, rSeedMax=0;

for (unsigned long t=t_min; t<t_max+1; t++)
  {
    if (gStat[t] & GOPEN)
      {
        gSeedX=gSeedY=0;
        gSeedMin=2550;
      }

    if (gStat[t] & GCLOSED)
      {
        lSeedX = lSeedY = lSeedMax = rSeedX = rSeedY = rSeedMax = 0;
      }

    // Seeds am Rand der ROI und im innern der Glottis

    for (unsigned short x=x_min; x<x_max+1; x++)
      for (unsigned short y=y_min; y<y_max+1; y++)
        {
          // Nur die Eckpunkte der ROI
          if (!mask(x,y) && (x==x_min || x==x_max) && (y==y_min || y==y_max))
          // Alle Randpunkte der ROI
          //if (!mask(x,y) && (x==x_min || x==x_max || y==y_min || y==y_max))
            {
              region(t, x, y) |= OUTER|SEED;

              voxel v(t, x, y, OUTER);
              v.calcGrad(si);
              if (v.calcGDir(region))
                av.push(v);
            }

          if (gStat[t] & GOPEN && mask(x,y))
            {
              unsigned long sum=0;
              for(int dx=-1; dx < 2; dx++)
                for(int dy=-1; dy < 2; dy++)
                  sum += si(t,x+dx,y+dy);

              if (sum < gSeedMin)
                {
                  gSeedMin = sum;
                  gSeedX = x;
                  gSeedY = y;
                }
            }
        }

    // Samenpunkt Glottis
    if (gStat[t] & GOPEN && gSeedMin < 2550)
      {
        region(t, gSeedX, gSeedY) |= GLOTTIS|SEED;

        voxel v(t, gSeedX, gSeedY, GLOTTIS);
        v.calcGrad(si);
        if (v.calcGDir(region))
          av.push(v);
      }
  }


// Segmentierung

while(! av.empty())
  {
    voxel v = av.top();
    av.pop();
    unsigned short x=v.x;
    unsigned short y=v.y;
    unsigned short t=v.t;
    switch(v.gradDir)
      {
      case XPLUS:
        x++;
        break;
      case XMINUS:
        x--;
        break;
      case YPLUS:
        y++;
        break;
      case YMINUS:
        y--;
        break;
      case TPLUS:
        t++;
        break;
      case TMINUS:
        t--;
        break;
      default:
        cerr << "OOPS!" << endl;
      }
    if (region(t, x, y) == NONE)
      {
        region(t, x, y) |= v.st;
```

```
            if ( t != t_min-1 && t != t_max+1 &&
                 x != x_min-1 && x != x_max+1 &&
                 y != y_min-1 && y != y_max+1 )
              {
                voxel w(t, x, y, v.st);
                w.calcGrad(si);
                if (w.calcGDir(region))
                  av.push(w);
              }
          }
        if (v.calcGDir(region))
          av.push(v);
      }
    } while (t_max!=si.length(0)-2);


  if (debug)
    cerr << endl << "Erosion...";

  for (unsigned long t=1; t<si.length(0)-1; t++)
    for (unsigned short x=x_min;x<x_max+1; x++)
      for (unsigned short y=y_min;y<y_max+1; y++)
        if (region(t,x,y) & GLOTTIS)
          for (int dt=-1; dt<2; dt++)
            for(int dx=-1; dx < 2; dx++)
              for(int dy=-1; dy < 2; dy++)
                if (region(t+dt,x+dx,y+dy) & OUTER)
                  {
                    region(t,x,y) &= ~GLOTTIS;
                    region(t,x,y) |= GSURFACE;
                    if (dt==0)
                      region(t,x,y) |= GBORDER;
                  }


  if (debug)
    cerr << endl << "Flaecheninhalt...";

  // Flaecheninhalt auslesen
  for (unsigned long t=0; t<si.length(0); t++)
    for (unsigned short x=x_min;x<x_max+1; x++)
      for (unsigned short y=y_min;y<y_max+1; y++)
        if( region(t,x,y) & GLOTTIS )
          area(t)+=1;

  if (randtyp)
    for (unsigned long t=0; t<si.length(0); t++)
      for (unsigned short x=x_min-1;x<x_max+2; x++)
        for (unsigned short y=y_min-1;y<y_max+2; y++)
          region(t,x,y) &= randtyp;

  // Offene Glottis markieren
  for (unsigned long t=0; t<si.length(0); t++)
    if (gStat[t] & GOPEN)
      region(t,1,1)=SEED;

  if (debug)
    {
      LSignal<float> area2=LChangeSamplingrate( area, 48000, 51);

      LSignalfile outfile6(area2);
      outfile6.writeFile ("area2.sig");

      LSignalfile outfile2(area);
      outfile2.writeFile ("area.sig");
    }

  if (debug)
    cerr << endl;

  return region;
}
```

A.2 Audioeichung

```
#include "LSignalfile.h"
#include "LSignal.h"
#include "LComplex.h"
#include "LSignalsAdjust.h"
#include "LSignalCommandline.h"
#include "LChangeSamplingrate.h"
#include "LRealFFTW.h"
#include "LInverseRealFFTW.h"
#include <qlist.h>

int main( int argc, char** argv)
{
  QApplication app( argc, argv);

  LSignalCommandline cl(argc, (const char**)argv);
  cl.usage() = "AudioEich␣wavfile1␣wavfile2␣-␣Eicht␣Samplingfrequenz";
  cl.description() = "Eicht␣die␣Samplingfrequenz␣von␣wavfile1␣an"
    "␣der␣von␣wavfile2";
  if( !cl.check(2) )return false;

  LSignalfile wav1(cl.argument(0),cl);
  LSignalfile wav2(cl.argument(1),cl);

  if (wav1.good() && wav2.good())
    {
      LSignal<float> Micro=wav1.toReal().reduce(LIndex2(-1,0)).toDCFree(); // \
                                    Micro
      Micro.writeFile("micro_terratec.sig");
      LSignal<float> EGG=wav1.toReal().reduce(LIndex2(-1,1)).toDCFree(); // EGG
      LSignal<float> HGGAudio=wav2.toReal().toDCFree();

      LSignal<float> errors(HGGAudio.signalDimensions());

      //
      // Entknacksen
      //

      float s=3.5; // Experimentell ermittelt

      for (unsigned long i=1; i<HGGAudio.length(0)-1; i++)
        errors(i)=HGGAudio(i)-(HGGAudio(i-1)+HGGAudio(i+1))/2;

      for (unsigned long i=1; i<HGGAudio.length(0)-1; i++)
        if ((errors(i-1)<-s/2 && errors(i)>s && errors(i+1)<-s/2) ||
            (errors(i-1)>s/2 && errors(i)<-s && errors(i+1)>s/2))
          {
            HGGAudio(i)= (HGGAudio(i-1)+HGGAudio(i+1))/2;
          }


      LSignalDimensions tmps_d( HGGAudio.signalDimensions());

      //
      // Samplingrate Eichen
      //

      tmps_d.samplingRate(0)=44100;

      LSignal<float> tmps( tmps_d );
      tmps.copy( HGGAudio );
      LSignal<float> HGGAudioHS=LChangeSamplingrate( tmps, Micro.samplingRate()\
                                    , 51);

      LSignalDimensions sd( Micro.signalDimensions());
      sd.length(0)=LNextFastFFTW( Micro.length(0)+HGGAudioHS.length(0));

      LSignal<float> corr1( sd);
      LSignal<float> corr2( sd);

      //
      // Anfangswerte für Intervallschachtelung
      //
      double step=10;
      double freq=47800;
      double lastmax = 0;
      double lastmax_f = 0;
      cerr.precision(12);
      int j = 0;

      //
      // Iteration-Schleife
      //
      while(true) {
        cerr << "Samplingrate:␣" << freq+j*step << "␣step:␣" << step << endl;
      HGGAudioHS=LChangeSamplingrate( wav2.toReal(), freq+j*step, 51);

      corr1.sub (LIndex1(0),     LIndex1(Micro.length(0)-1) ).copy( Micro);
      corr1.sub( LIndex1(Micro.length(0)), LIndex1( corr1.length(0)-1))=0.0;

      corr2.sub( LIndex1(0),    LIndex1(HGGAudioHS.length(0)-1) ).copy( \
                                    HGGAudioHS);
      corr2.sub( LIndex1(HGGAudioHS.length(0)), LIndex1( corr2.length(0)-1))=0.0;

      LRealFFTW     fft( corr1.length(0), true);
      LInverseRealFFTW ifft( corr1.length(0), true);

      //
      // Kreuzkorrelation
      //
      LSignal<LComplex<float> > f1=fft( corr1);
      LSignal<LComplex<float> > f2=fft( corr2);
      LSignal<LComplex<float> > f3=( f1.signalDimensions());

      for( unsigned long i=0; i<f3.length(0); i++)
        {
          LComplex<float> c=f2[i];
          c.imaginary()*=-1;
          f3[i]=f1[i]*c;
        }

      LSignal<float> corr=ifft( f3);

      LIndex1 ix=corr.argmax();
      cerr << "Maximum␣der␣Korrelation:␣" << corr(ix(0)) << "␣bei␣" << ix(0) << \
                                    endl << endl;

      //
      // Aktuelle Frequenz besser als letzte?
      //
```

```
115     if ( corr(ix(0)) > lastmax )
          {
            lastmax_f = j;
            lastmax = corr(ix(0));
          }
120
        //
        // Am Ende des aktuellen Intervalls?
        //
        if (j == 20)
          {
125         //
            // Wenn Ergebnis genau genug
            //
            if (step <= 0.001)
              {
130             //
                // Frequenz ausgeben und Iteration abbrechen
                //
                cerr << "Max:␣" << freq + (lastmax_f)*step << endl;
                exit(0);
              }

            // freq auf Anfang von neuem Intervall setzten
            freq = freq + (lastmax_f-1)*step;
140         j=0;

            // step verkleinern
            step /= 10;

145         // lastmax initialisieren
            lastmax = 0;
            lastmax_f = 0;
          }
        else
150       j++;
      }
    }
  }
```

A.3 Audiosynchronisation

```
#include "LSignalfile.h"
#include "LSignal.h"
#include "LComplex.h"
#include "LSignalsAdjust.h"
#include "LSignalCommandline.h"
#include "LChangeSamplingrate.h"
#include "LRealFFTW.h"
#include "LInverseRealFFTW.h"
#include <qlist.h>

int main( int argc, char** argv)
{
 QApplication app( argc, argv);

 LSignalCommandline cl(argc, (const char**)argv);
 cl.usage() = "AudioSync␣wavfile1␣wavfile2␣-␣Syncronisiert␣zwei␣WAVs";
 cl.description() = "Versucht␣die␣Wav-Datei␣mit␣hoher␣Qualitaet␣(wavfile1)"
  "␣mit␣der␣WAV-Datei␣von␣der␣HGG-Kamera␣(wavfile2)␣zu␣synchronisieren";
 if( !cl.check(2) )return false;

 LSignalfile wav1(cl.argument(0),cl);
 LSignalfile wav2(cl.argument(1),cl);

 if (wav1.good() && wav2.good())
  {
   LSignal<float> Micro=wav1.toReal().reduce(LIndex2(-1,0)).toDCFree(); // \
                                  Micro
   Micro.writeFile("micro_terratec.sig");
   LSignal<float> EGG=wav1.toReal().reduce(LIndex2(-1,1)).toDCFree(); // EGG
   LSignal<float> HGGAudio=wav2.toReal().toDCFree();

   LSignal<float> errors(HGGAudio.signalDimensions());

   //
   // Entknacksen
   //

   float s=3.5; // Experimentell ermittelt

   for (unsigned long i=1; i<HGGAudio.length(0)-1; i++)
    errors(i)=HGGAudio(i)-(HGGAudio(i-1)+HGGAudio(i+1))/2;

   for (unsigned long i=1; i<HGGAudio.length(0)-1; i++)
    if ((errors(i-1)<-s/2 && errors(i)>s && errors(i+1)<-s/2) ||
       (errors(i-1)>s/2 && errors(i)<-s && errors(i+1)>s/2))
     {
      HGGAudio(i)= (HGGAudio(i-1)+HGGAudio(i+1))/2;
     }


   LSignalDimensions tmps_d( HGGAudio.signalDimensions());

   //
   // Samplingrate korrigieren
   //

   tmps_d.samplingRate(0)=44166.894;

   LSignal<float> tmps( tmps_d );
   tmps.copy( HGGAudio );
   LSignal<float> HGGAudioHS=LChangeSamplingrate( tmps, Micro.samplingRate()\
                                  , 51);

   LSignalDimensions sd( Micro.signalDimensions());
   sd.length(0)=LNextFastFFTW( Micro.length(0)+HGGAudioHS.length(0));

   LSignal<float> corr1( sd);
   LSignal<float> corr2( sd);

   corr1.sub (LIndex1(0), LIndex1(Micro.length(0)-1) ).copy( Micro);
   corr1.sub( LIndex1(Micro.length(0)), LIndex1( corr1.length(0)-1))=0.0;

   corr2.sub( LIndex1(0), LIndex1(HGGAudioHS.length(0)-1) ).copy( HGGAudioHS);
   corr2.sub( LIndex1(HGGAudioHS.length(0)), LIndex1( corr2.length(0)-1))=0.0;

   //
   // Kreuzkorrelation
   //

   LRealFFTW    fft( corr1.length(0), true);
   LInverseRealFFTW ifft( corr1.length(0), true);

   LSignal<LComplex<float> > f1=fft( corr1);
   LSignal<LComplex<float> > f2=fft( corr2);
   LSignal<LComplex<float> > f3=( f1.signalDimensions());

   for( unsigned long i=0; i<f3.length(0); i++)
    {
     LComplex<float> c=f2[i];
     c.imaginary()*=-1;
     f3[i]=f1[i]*c;
    }

   LSignal<float> corr=ifft( f3);


   //
   // Korrelationsmaximum
   //

   LIndex1 ix=corr.toAbsolute().argmax();
   cerr << "Maximum␣der␣Korrelation:␣" << corr(ix(0)) << "␣bei␣" << ix(0) << \
                                  endl << endl;


   LSignalDimensions sd2( HGGAudioHS.signalDimensions());
   sd2.xFirst(0)+=ix(0);
   LSignal<float> HGGAudioHS_Versetzt( sd2);
   HGGAudioHS_Versetzt.copy( HGGAudioHS );

   LSignal<float> corr2_Versetzt( sd2);
   corr2_Versetzt.copy( corr2.sub( LIndex1(0), LIndex1(HGGAudioHS.length(0)-1)\
                                  ));

   LSignalDimensions MicroCut_D( Micro.signalDimensions() );
   MicroCut_D.length(0) = LMinimum( HGGAudioHS.length(0), Micro.length(0)-ix\
                                  (0));
   LSignal<float> MicroCut( MicroCut_D );
```

```
        MicroCut.copy( Micro.sub( LIndex1(ix(0)), LIndex1(ix(0)+MicroCut.length(0)\
                                        -1)));

115
        LSignalDimensions EGGCut_D( EGG.signalDimensions() );
        EGGCut_D.length(0) = LMinimum( HGGAudioHS.length(0), EGG.length(0)-ix(0));
        LSignal<float> EGGCut( EGGCut_D );
        EGGCut.copy( EGG.sub( LIndex1(ix(0)), LIndex1(ix(0)+EGGCut.length(0)-1)));
120
        MicroCut.writeFile("micro.sig");
        EGGCut.writeFile("egg.sig");


125     // Signale anzeigen

        QList<LSignalAtom> slist;
        slist.append( &Micro);
        slist.append( &HGGAudioHS_Versetzt);
130     slist.append( &corr);

        LSignalsAdjust sa( slist);
        sa.show();
        app.setMainWidget( &sa);
135     return app.exec();
      }
    }
```

A.4 Flächenparameter

```
#include <cmath>
#include "LSignal.h"
#include "LSignalfile.h"
#include "LHeader.h"

int main( int argc, char** argv)
{
  const unsigned int c_thr=5;

  LSignalCommandline cl(argc, (const char**)argv);
  cl.usage() = "AreaCalc";
  cl.description() = "Extrahiert␣Parameter␣aus␣dem␣Flächenverlauf";
  cl.defineOption("wlen", ".1", "Fensterbreite␣in␣s");

  cl.check();

  if (cl.isHelp())
    {
      cl.showHelp();
      return false;
    }

  float wlen = LToDouble(cl["wlen"]);

  LSignalfile gnefile( "gne.sig", cl );
  if( !gnefile.good() )
    {
      cerr << "gne.sig␣fehlt!\n";
      exit(1);
    }

  LSignalfile areafile( "area.sig", cl );
  if( !areafile.good() )
    {
      cerr << "area.sig␣fehlt!\n";
      exit(1);
    }

  LSignal<float> gne = gnefile.toReal();
  LSignal<float> area = areafile.toReal();

  LSignal<float> area_min(gne.signalDimensions());
  LSignal<float> area_max(gne.signalDimensions());
  LSignal<float> cquot(gne.signalDimensions());

  if ( (gne.xFirst() / gne.samplingRate() - wlen/2) * area.samplingRate() - area\
                                  .xFirst() < 1 )
    {
      cerr << "Fensterbreite␣zu␣gross!\n";
      exit(1);
    }


  //
  // GNE Signal durchlaufen
  //
  for (unsigned long i=0; i < gne.length(0); i++)
  {
    //
    // aktuelle Zeit bestimmen
    //
    float time = (i + gne.xFirst()) / gne.samplingRate();

    //
    // Fenstergrenzen bezüglich area bestimmen
    //
    unsigned long w_start = (unsigned long)((time - wlen/2) * area.samplingRate\
                                    () - area.xFirst());

    unsigned long w_end = (unsigned long)((time + wlen/2) * area.samplingRate()\
                                     - area.xFirst());

    float max=0, min=0, mean=0, tmp=-1;
    unsigned int maxcount=0, mincount=0, closed=0, c_first=0, c_last=0, periods\
                                    =0;

    //
    // Mittelwert im Fenster von area bestimmen
    //
    for (unsigned int j=w_start; j < w_end+1; j++)
      mean += area(j);

    mean /= w_end - w_start + 1;

    //
    // area Fenster durchsuchen
    //
    for (unsigned int j=w_start; j < w_end+1; j++)
      {
        //
        // Wurde mean gekreuzt?
        //
        if ((area(j-1) < mean) != (area(j) < mean))
          {
            //
            // nicht das erste mal?
            //
            if (tmp != -1)
              if (tmp >= mean)
                {
                  //
                  // Maximum speichern
                  //
                  max += tmp;
                  maxcount++;
                }
              else
                {
                  //
                  // Minimum speichern
                  //
                  min += tmp;
                  mincount++;
                }

            //
            // tmp neu initialisieren
            //
```

```
115           tmp = area(j);
            }
          else
            //
            // tmp aktualisieren
120         //
            if ( tmp != -1 && abs(area(j) - mean) > abs(tmp - mean))
              tmp = area(j);

          //
125       // Ganze Anzahl Perioden bestimmen
          //
          if ((area(j-1) < mean) == (area(j) >= mean))
            {
              if (!periods)
130             c_first=j;
              periods++;
              c_last=j;
            }
        }
135


      closed=0;

      //
140   // Closed-Quotien bestimmen
      //
      for (unsigned int j=c_first; j < c_last; j++)
        if ( area(j) <= c_thr )
          closed++;
145


        //
        // Mittelwert der Extrema bestimmen
        //
        if (maxcount)
150       area_max(i) = max / maxcount;
        else
          area_max(i) = 0;

        if (mincount)
155       area_min(i) = min / mincount;
        else
          area_min(i) = 0;

        //
160     // Genügend Perioden für CQ?
        //
        if (periods > 4)
          cquot(i) = closed / float(c_last - c_first);
        else
165       cquot(i) = 0;
      }

    //
    // Abspeichern
170 //
    area_max.writeFile("amax.sig");
    area_min.writeFile("amin.sig");
    cquot.writeFile("cquot.sig");

175 }
```

Literaturverzeichnis

Alipour-Haghighi u. a. 2000 ALIPOUR-HAGHIGHI, F. ; BERRY, D. A. ; TITZE, I. R.: A finite element model of vocal fold vibration. In: *J. Acoust. Soc. Am.* 108 (2000), S. 3003–3012

Anderson u. a. 2002 ANDERSON, Sven ; MICHAELIS, Dirk ; STRUBE, H.W.: Vollautomatische Glottisdetektion bei Hochgeschwindigkeitsaufnahmen. In: JEKOSCH, U. (Hrsg.): *Fortschritte der Akustik - DAGA 02 [Bochum]*. Oldenburg : DEGA, 2002, S. 624–625

Baken 1992 BAKEN, Ronald J.: Electroglottography. In: *Journal of Voice* 6 (1992), Nr. 2, S. 98–110

Gonzalez und Woods 1993 GONZALEZ, Rafael C. ; WOODS, Richard E.: *Digital Image Processing*. Reading, MA, USA : Addison Wesley Publishing Company, 1993

Granqvist und Lindestad 2001 GRANQVIST, Svante ; LINDESTAD, Per-Åke: A method of applying Fourier analysis to high-speed laryngoscopy. In: *J. Acoust. Soc. Am.* 110 (2001), Dezember, Nr. 6, S. 3193–3197

Kass u. a. 1987 KASS, M. ; WITKIN, A. ; TERZOPOULOS, D.: Snakes: Active contour models. In: *Int. J. of Computer Vision* 1 (1987), S. 321–331

Marendic 2001 MARENDIC, Boris: *Analysis of high-speed digital recordings of the larynx*. Chicago, Illinois, Illinois Institute of Technology, Diplomarbeit, Mai 2001

Michaelis u. a. 1997 MICHAELIS, Dirk ; GRAMSS, Tino ; STRUBE, Hans W.: Glottal-to-noise excitation ratio - a new measure for describing pathological voices. In: *Acustica / acta acustica* 83 (1997), S. 700–706

Neubauer u. a. 2001 NEUBAUER, Jürgen ; MERGELL, Patrick ; EYSHOLDT, Ulrich ; HERZEL, Hanspeter: Spatio-temporal analysis of irregular vocal fold oscillations: Biphonation due to desynchronization of spatial modes. In: *J. Acoust. Soc. Am.* 110 (2001), Dezember, Nr. 6, S. 3179–3192

Nikolaidis und Pitas 2001 NIKOLAIDIS, Nikos ; PITAS, Ioannis: *3-D Image Processing Algorithms*. 605 Third Avenue, New York, N.Y. 10158-0012 : John Wiley & Sons, Inc., 2001. – ISBN 0-471-37736-8

Press u. a. 1992 PRESS, W.H. ; TEUKOLSKY, S.A. ; VETTERLING, W.T. ; FLANNERY, B.P.: *Numerical Recipes in C : The Art of Scientific Computing*. Cambridge University Press, 1992. – ISBN 0-521-43108-5

Pschyrembel 1982 PSCHYREMBEL, W. (Hrsg.): *Klinisches Wörterbuch*. Berlin ; New York : de Gruyter, 1982. – ISBN 3-11-007187-8

Sataloff 1993 SATALOFF, Robert T.: Die menschliche Stimme. In: *Spektrum der Wissenschaft* (1993), S. 74–81

Tipler 1994 TIPLER, Paul A.: *Physik*. Heidelberg ; Berlin ; Oxford : Spektrum Akad. Verl., 1994. – ISBN 3-86025-122-8

Titze 1973 TITZE, I. R.: The human vocal cords: A mathematical model I. In: *Phonetica* 28 (1973), S. 129–170

Titze 1994 TITZE, Ingo R.: *Principles of Voice Production*. Englewood Cliffs, NJ : Prentice Hall, 1994

Wendler und Seidner 1987 WENDLER, Jürgen ; SEIDNER, Wolfram: *Lehrbuch der Phoniatrie*. 2. Auflage. Leipzig : VEB Georg Thime, 1987

Wittenberg 1998 WITTENBERG, Thomas ; EYSHOLDT, U. (Hrsg.): *Kommunikationsstörungen - Berichte aus Phoniatrie und Pädaudiologie*. Bd. 4: *Wissensbasierte Bewegungsanalyse von Stimmlippenschwingungen anhand digitaler Hochgeschwindigkeitsaufnahmen*. Aachen : Shaker Verlag, 1998. – ISBN 3-8265-4143-X

Danksagung

Ich danke allen sehr, die diese Arbeit ermöglicht und zu ihrem Gelingen beigetragen haben. Insbesondere sind das:

- Prof. Dr. Ronneberger für das Ermöglichen dieser interdisziplinären Diplomarbeit,
- Dr. Hans Werner Strube für die Aufnahme in die Arbeitsgruppe *Sprache und Neuronale Netze* und die sehr kompetente Beratung bei allen fachlichen Fragen,
- Dr. Dirk Michaelis für die intensive und freundschaftliche Betreuung der Arbeit und die vielen lehrreichen und interessanten Diskussionen,
- Prof. Dr. E. Kruse, Dr. Arno Olthoff, Dr. Rolf Schiel und alle anderen Mitarbeiter der *Abteilung Phoniatrie und Pädaudiologie* für die ständige Bereitschaft, diese Arbeit zu unterstützen und für die Durchführung der Hochgeschwindigkeitsaufnahmen,
- die gesamte *Forschungsgruppe Stimme und Sinnesentwicklung* für die fachliche Unterstützung und die sehr angenehme Arbeitsatmosphäre,
- die Patienten und Versuchspersonen, ohne die es kein Datenmaterial gäbe,
- meine Eltern für die bedingungslose Finanzierung meines Studiums und
- Chen-Yu, Paul und all meine anderen Freunde für die seelische Unterstützung und die notwendige Regeneration zwischendurch.