



**Entwicklung und Programmierung einer
Applikation für Mac OS X zur Korrektur eines
Bild-Ton-Versatzes in QuickTime™
Filmdateien**

Diplomarbeit im Studiengang Ton

**Hochschule für Film und Fernsehen „Konrad Wolf“
Potsdam-Babelsberg**

Autor: Michael Gerbes

Betreuer: Dipl.-Ing. (FH) Robert Gräf

Eingereicht am: 4. Juli 2005

1. EINLEITUNG.....	5
2. TECHNOLOGISCHE GRUNDLAGEN	6
2.1 Ein Überblick über aktuelle Formate von Filmdateien	6
2.1.1 Containerformate	6
2.1.1.1 QuickTime	6
2.1.1.2 Audio Video Interleave (AVI).....	7
2.1.1.3 MP4.....	7
2.1.1.4 Ogg.....	7
2.1.1.5 Matroska.....	7
2.1.2 Video- und Tonformate (Codecs)	8
2.1.3 Streams	9
2.1.3.1 Real Media	9
2.1.3.2 Windows Media Player	9
2.2 QuickTime - Multimedia für Mac OS.....	10
2.2.1 Das Grundkonzept.....	10
2.2.2 Die QuickTime Architektur aus Sicht des Programmiers	11
2.2.3 Die Struktur von QuickTime Movies	14
2.2.3.1 Movies und Filmdateien	14
2.2.3.2 Spuren	16
2.2.3.2.1 Spuren und ihre Eigenschaften	17
2.2.3.2.2 Spuren und ihre Zeitskala.....	17
2.2.3.2.3 Spurdaten (Spur Media)	18
2.2.3.2.4 Schnittliste einer Spur	18
2.2.3.2.5 Länge einer Spur	18
2.2.3.3 Mediendaten (Samples)	19
2.2.3.4 Sampledauer und Bildrate	19
2.2.3.5 Zeitskalen	20
2.2.4 Einbindung in das Betriebssystem	21
2.2.5 Bemerkungen zu den QuickTime Termini	22
3. KONZEPTION DES PROGRAMMS „QT SYNC“	23
3.1 Asynchronitäten	23
3.2 Ursachen eines Bild-Ton-Versatzes	24

3.3	Analyse vorheriger Lösungen	27
3.3.1	QT Mutator	28
3.3.2	Synhole	30
3.3.3	QuickTime Player / QuickTime Player Pro	32
3.3.4	Fazit	34
3.4	Mögliche Einsatzgebiete von QT Sync.....	34
3.5	Wahl der Programmiersprache	35
3.6	Exkurs: Eine kurze Entstehungsgeschichte von QT Sync.....	37
3.7	Funktionsumfang	40
4.	DAS PROGRAMM „QT SYNC“.....	44
4.1	Referenz	44
4.1.1	Programmstart	44
4.1.2	Das Laden eines Films	45
4.1.3	Das Filmfenster	47
4.1.4	Die verschiedenen Betrachtungsmodi	49
4.1.5	Das Informationsfenster („Info Window“)	50
4.1.6	Das „Controller Window“	52
4.1.7	Erläuterungen zur Behandlung von Offsets innerhalb von QT Sync.....	53
4.1.8	Weitere Funktionen des „Controller Window“	56
4.1.9	Das Ändern des Seitenverhältnisses (Change Ratio...).....	58
4.1.10	Festlegung des sichtbaren Bildanteils („Cropping“).....	60
4.1.11	Aufheben des Croppings	62
4.1.12	Hinzufügen einer Spur.....	63
4.1.13	Entfernen einer Spur.....	64
4.1.14	Zusammenfügen mehrerer Filme.....	64
4.1.15	Hinzufügen einer TimeCode / Feet + Frames Einblendung.....	66
4.1.16	Das „About“ Fenster	70
4.1.17	Überprüfung der Verfügbarkeit einer neuen Programmversion.....	70
4.1.18	Schließen des Films	71
4.1.19	Speichern eines Films	71
4.1.20	Speichern eines Films in eine neue selbständige Filmdatei	72
4.1.21	Der Full Screen Modus	73
4.1.22	Der Full Screen Modus mit verstecktem Controller Fenster	74
4.1.23	Verlassen des Full Screen Modus	76

4.1.24	Voreinstellungen	76
4.1.24.1	Kategorie „General“ (Allgemeines).....	78
4.1.24.2	Kategorie „Controls“ (Vorgaben für das Controller Fenster)	78
4.1.24.3	Kategorie „Full Screen“	78
4.1.24.4	Kategorie „Tools“	79
4.1.25	Das Beenden des Programms	79
5.	PROGRAMMIERUNG.....	80
5.1	Programmierbeispiele.....	81
5.1.1	Vorbemerkungen.....	82
5.1.2	Öffnen von Filmen	82
5.1.3	Das Berechnen der Bildrate für verschiedene Filmformate	89
5.1.4	Das Ändern des Seitenverhältnisses (Change Ratio...)	93
6.	FAZIT	101
7.	ANHANG	105
7.1	Literatur- und Quellenverzeichnis.....	105
7.2	Abbildungsverzeichnis	106
7.3	Danksagung	107
7.4	Selbständigkeitserklärung.....	108

1. Einleitung

Diese Arbeit beschreibt die Entwicklung und Programmierung der Mac OS X Applikation „QT Sync“.

„QT Sync“ ist primär ein Abspielprogramm für QuickTime Filme, welches während des Abspielens die Möglichkeit bietet, Asynchronitäten innerhalb der Filmdatei zu korrigieren.

Zunächst soll ein Überblick über die zugrunde liegenden Technologien (insbesondere QuickTime) gegeben werden.

Im Weiteren werden die Gedanken zur Konzeption des Programms diskutiert. Der Autor wird Ursachen einer Asynchronität vorstellen, vorherige Lösungen anderer Entwickler diskutieren, einen Aufgabenkatalog für sein Programm präsentieren und Gedanken zur praktischen Umsetzung darlegen.

Ferner widmet sich die Arbeit dem Programm selbst und beschreibt die Funktionen, die innerhalb der Applikation vorhanden sind. Dieser Abschnitt stellt eine Referenz zur Bedienung des Programms dar.

Abschließend sollen ausgewählte Codebeispiele und die Programmierung in der Programmierumgebung REALbasic vorgestellt werden.

Im Folgenden werden mehrere Themenbereiche berührt, deren Komplexität nur ihre eingeschränkte Darstellung im Umfang dieser Arbeit zulässt. Dies betrifft beispielsweise die Diskussion über verschiedene Programmiersprachen, eine Diskussion aller Möglichkeiten von QuickTime oder auch die Beschreibung der Programmierumgebung REALbasic.

Ebenso würde eine vollständige Wiedergabe des Quellcodes von „QT Sync“ den Rahmen dieser Arbeit sprengen.

Die Diskussion von Themen, welche sich mit Programmierung oder Computer-
verwandten Sachverhalten befassen, beinhalten naturgemäß viele englische Fachbegriffe, die teilweise auch schon in den alltäglichen Sprachgebrauch eingegangen sind. Der Autor wagt die Gratwanderung zwischen der unveränderten Übernahme einiger Begriffe und der sinnvollen Übersetzung anderer. Bei nicht alltäglichen englischen Begriffen soll durch Fußnoten eine Erklärung bzw. Verdeutlichung gegeben werden.

2. Technologische Grundlagen

2.1 Ein Überblick über aktuelle Formate von Filmdateien

Die absolute Anzahl von verschiedenen Standards für Filmdateien ist groß. Im Laufe der Jahre sind viele Formate entwickelt worden, von denen sich nur wenige etabliert haben und der schnelllebigen Entwicklung standhalten konnten.

Im Folgenden soll ein kurzer Überblick über die aktuellen Standards gegeben werden, wobei die Auflistung keinen Anspruch auf Vollständigkeit erhebt.

2.1.1 Containerformate

Bild- und Toninformationen besitzen völlig unterschiedliche Eigenschaften und müssen bei der Wiedergabe getrennt voneinander behandelt werden. Daher ist es nur nachvollziehbar, dass diese Informationsdaten an unterschiedlichen Orten gespeichert werden, um sie bei der Wiedergabe besser voneinander trennen und weiterverarbeiten zu können.

Damit aber nicht jeder Film aus zwei (oder mehr) Dateien besteht, werden diese in einen Container „verpackt“.

Dieser Container hat hauptsächlich die Aufgabe, alle enthaltenen Dateien zu organisieren und dem Wiedergabeprogramm korrekt zur Verfügung zu stellen. Die Art des Containers sagt daher auch nichts über den tatsächlichen Inhalt aus.

2.1.1.1 QuickTime

Mit der Veröffentlichung von QuickTime 1990 wurde es zum ersten Mal möglich, Filme ohne externe Hardware auf einem Apple Macintosh abzuspielen. Die Auflösung betrug damals 156 mal 112 Pixel bei einer Bildrate von 10 Bildern pro Sekunde.

In nachfolgenden Abschnitten soll noch detaillierter auf QuickTime eingegangen werden.

2.1.1.2 Audio Video Interleave (AVI)

Das AVI-Dateiformat (vorgestellt von Microsoft im Jahre 1992) basiert auf dem Resource Interchange File Format (RIFF) und teilt die einzelnen Bild- und Tonspuren in Blöcke ein, welche durch einen speziellen Code (FourCC) identifizierbar bleiben. Die Datenblöcke werden „verzahnt“ (interleaved) in die Datei geschrieben. Ein großer Nachteil des AVI-Formats (welches ursprünglich für Filme mit 15 Bildern pro Sekunde und einer Auflösung von 160 mal 120 Pixel konzipiert war) besteht darin, dass immer die gesamte Datei vorliegen muss, um sie abspielen zu können. Dies verhindert ein Streaming¹ und macht das Arbeiten mit Videoschnittprogrammen zu einer mühsamen Angelegenheit: Zur Betrachtung des Schnittresultats muss der gesamte Film jeweils neu erstellt, also aus den Einzelsegmenten zusammenkopiert werden.

2.1.1.3 MP4

Der MP4 Container wurde von der Motion Picture Experts Group für MPEG4-kodierte Inhalte vorgesehen. Er basiert auf dem QuickTime Container, darf allerdings, zu Gunsten der Kompatibilität, nur ganz bestimmte (überwiegend auf MPEG4 basierende) Video- und Tonformate enthalten.

2.1.1.4 Ogg

Das Ogg Container Format ist Bestandteil eines Open-Source Projects, welches nicht nur den Container, sondern u.a. auch Codecs für Multimedia Anwendungen entwickelt. Der bekannteste Vertreter ist wahrscheinlich der Ogg Vorbis Audio Codec.

2.1.1.5 Matroska

Das Open-Source Projekt Matroska dagegen fokussiert seine Bemühungen vollkommen auf die Etablierung seines eigenen Containerformats. Es genügt allen heutigen technischen Anforderungen (wie z.B. Streaming), unterstützt gängige Fremdcodecs und gilt als rückwärtskompatibel bei späterer Formaterweiterung.

¹ Vgl. Kap. 2.1.3: Streams

2.1.2 Video- und Tonformate (Codecs)

Der Codec (eine Abkürzung von „coding–decoding“) ist bei der Erstellung einer Filmdatei für die Kodierung eines Medieninhalts und bei der Wiedergabe des Films für die Dekodierung desselben verantwortlich.

In Filmdateien werden Codecs hauptsächlich dafür benutzt, die immensen Größen von Bild- und Tondateien zu verringern, um diese auf gängigen Medien speichern zu können.

Je nach Codec geschieht dies durch verschiedene Verfahren und in unterschiedlicher Qualität.

Es werden bei der Verkleinerung von Datenmengen zwei Grundprinzipien unterschieden: Die Kompression und die Reduktion.

Die Datenkompression arbeitet verlustfrei, d.h. nach der Dekodierung liegen wieder exakt dieselben Daten vor. (Vergleichbar mit der Komprimierung in .zip oder .sit Dateien.)

Die Datenreduktion dagegen analysiert die Informationen auf Anteile, welche die menschlichen Sinne nicht (oder in nur sehr geringem Maße) wahrnehmen würden, und entfernt diese. Je nach Stärke der Reduktion können dabei wahrnehmbare Artefakte auftreten. Diese äußern sich beispielsweise durch metallisches Klingeln im Ton oder unscharfe Kanten und „Klötzchenbildung“ im Bild.

Auf Datenreduktion basierende Codecs nutzen meist zusätzlich die Datenkompression, um die reduzierte Datenmenge noch weiter zu verkleinern.

Bekannte ältere Video-Codecs sind z.B. Cinepak, Sorenson und H.261. In den letzten Jahren setzten sich vor allem MPEG-4-basierende Codecs, darunter auch DivX, 3ivx oder XviD, durch. Beinahe alle Video-Codecs arbeiten mit Datenreduktion.

Im Audio-Bereich fanden sich Codecs basierend auf ADPCM² (z.B. IMA) oder a-Law / μ -Law³. Später wurden durch bessere Prozessorleistungen hochwertigere Codecs wie AC3, AAC oder MP3 möglich (ebenfalls alle verlustbehaftet).

Seit neuestem werden auch verlustfreie Audio-Codecs wie Apple Lossless oder FLAC (Free Lossless Audio Codec) eingesetzt.

² ADPCM (Adaptive Differential Pulse Code Modulation) variiert dynamisch die Quantisierung der Bitstufen.

³ Entwickelt für Telefonverbindungen. Das Signal wird nach festen Formeln komprimiert.

2.1.3 Streams

2.1.3.1 Real Media

Seit 1995 vertreibt die amerikanische Firma Real ihren Media Player. Schwerpunkt des Real Formats war und ist das Streaming⁴ von Bild- und/oder Audiodaten.

Der Hauptvorteil des Real Dateiformats ist die Möglichkeit, den Stream zu skalieren, d.h. die Größe des Streams wird der jeweils vorhandenen Internetverbindung angepasst. (Selbstverständlich ändert sich dabei auch die Qualität des darzustellenden Medieninhalts.)

Das Dateiformat und die dazugehörigen Video- und Audio-Codecs sind proprietär. Allerdings gibt es einige Fremdprogramme, die das Herstellen von Medieninhalten in Real zulassen.

Der Media Player selbst ist allerdings auch in der Lage, andere Formate (wie z.B. MP3, MPEG4 oder QuickTime) abzuspielen.

2.1.3.2 Windows Media Player

Der Windows Media Player aus dem Hause Microsoft stellt die Konkurrenz zum Real Player dar. Auch er besitzt ein eigenes Containerformat sowie proprietäre Codecformate und kann durch Fremdcodecs erweitert werden.

⁴ Streaming bezeichnet das Aussenden eines Films von einem Internet Server aus. Um den Film anzusehen, muss dieser nicht erst vollständig auf den Computer des Benutzers heruntergeladen worden sein, sondern es wird während des Abspielens ständig vom Server nachgeladen.

Eine besondere Art des Streamings ist die Live-Übertragung (z.B. der Echtzeit-Stream eines TV-Senders). Hierbei werden jeweils kurze Filmteile generiert, welche nach einmaligem Aussenden nicht wieder verfügbar sind.

2.2 QuickTime – Multimedia für Mac OS

Die Programmierung einer QuickTime-basierten Applikation setzt das Verständnis der QuickTime Architektur voraus. Im Folgenden soll daher ein Überblick über QuickTime aus der Sicht des Programmierers gegeben werden.

Die Abschnitte 2.2.1 bis 2.2.3 benutzen als Quelle die gekürzte, leicht bearbeitete und übersetzte Fassung des „QuickTime Overview“ der Firma Apple (Hrsg.).

Insbesondere die Angaben zu „QuickTime Atoms“ (die byteweise Beschreibung des Aufbaus einer QuickTime-Datei bzw. ihrer Unterstrukturen) und die Ausführungen zum Streaming wurden, auf Grund ihrer untergeordneten Relevanz im Zusammenhang mit dieser Arbeit, gekürzt.

2.2.1 Das Grundkonzept

QuickTime ist eine plattformunabhängige Multimedia Architektur für Mac OS und Windows. Es besteht aus einem Satz von Multimedia Systemerweiterungen, einer kompletten API⁵, einem Dateiformat und Programmen wie beispielsweise dem QuickTime Player oder dem QuickTime Browser Plug-In.

Es handelt sich bei QuickTime nicht nur um ein Abspielprogramm. Es unterstützt die Erstellung, Produktion und Verbreitung einer großen Vielzahl von Medientypen. QuickTime bietet dabei Unterstützung für den gesamten Prozess: Digitalisieren von Medien in Echtzeit, Import und Export von bestehendem Material, Bildbearbeitung und Schnitt; Datenkompression, Distribution und Wiedergabe beim Endnutzer.

Spezifische Aufgaben, für die QuickTime eingesetzt werden kann:

- Das Abspielen von Filmen und anderen Medien, wie z.B. Flash oder MP3 Audio.
- Nicht-destruktives Editieren von Filmen und anderen Medien.

⁵ API: Application Program Interface – Ein Satz aus Routinen, Protokollen und Werkzeugen zum Erstellen von Programmen. Diese vorgefertigten Teile erleichtern dem Programmierer die Arbeit, seinem Programm die Funktionen der Technologie, zu der die API gehört, hinzuzufügen.

- Import und Export von Bildern in verschiedene Formate, z.B. JPG und PNG.
- Das Einfügen und Arrangieren („Compositing“) von mehreren Medienelementen aus verschiedenen Quellen.
- Das Anlegen verschiedener zeitabhängiger Medien auf eine gemeinsame Zeitachse.
- Das Digitalisieren und Speichern von Material aus Echtzeit-Quellen, wie z.B. Audio- und Videoeingängen.
- Das Erzeugen von skript-gesteuerten Animationen („Sprites“).
- Erstellung von Präsentationen, die sowohl mit dem Benutzer als auch mit vernetzten Datenbanken und Servern interagieren.
- Das Erzeugen von Filmen mit individueller Gestaltung der Benutzeroberfläche.
- Das Abspielen von Filmen über ein Netzwerk oder das Internet in Echtzeit („Streaming“).
- Live-Übertragungen über Netzwerk oder das Internet.
- Die Distribution von Filmen auf Speichermedien, dem Netzwerk oder im Internet.

2.2.2 Die QuickTime Architektur aus Sicht des Programmierers

Die QuickTime API besteht aus zwei wesentlichen Bestandteilen: „Tool Sets“⁶ und „Components“⁷.

Die Tool Sets bilden die erste Schnittstelle zwischen einem Programm und QuickTime. Sie sind auf spezielle Aufgaben zugeschnitten. Die „Movie Toolbox“ ist beispielsweise für das Öffnen, Abspielen und Speichern von Filmen zuständig, während der „Image Compression Manager“ die Komprimierung und Dekomprimierung von Bilddaten vornimmt.

Alle diese Tool Sets arbeiten Hand in Hand, und es ist für den Programmierer nicht zwingend notwendig, mit allen Sets zu arbeiten, solange er spezielle Funktionen aus den einzelnen Sets nicht benötigt.

⁶ Tool Set = wörtlich: Werkzeug-Satz

⁷ Components = wörtlich: Komponente, Bestandteil, Zutat

Die Components dagegen sind Bausteine, welche in die QuickTime Architektur eingefügt werden können. Diese wird dadurch flexibel, modular und ausbaufähig. Components sind Schnittstellen zwischen den theoretischen Befehlen in den Tool Sets und den praktischen Anwendungen und können z.B. dafür verwendet werden, um QuickTime einen neuen Medientyp abspielen zu lassen oder eine bestimmte Streaming Variante auszuführen.

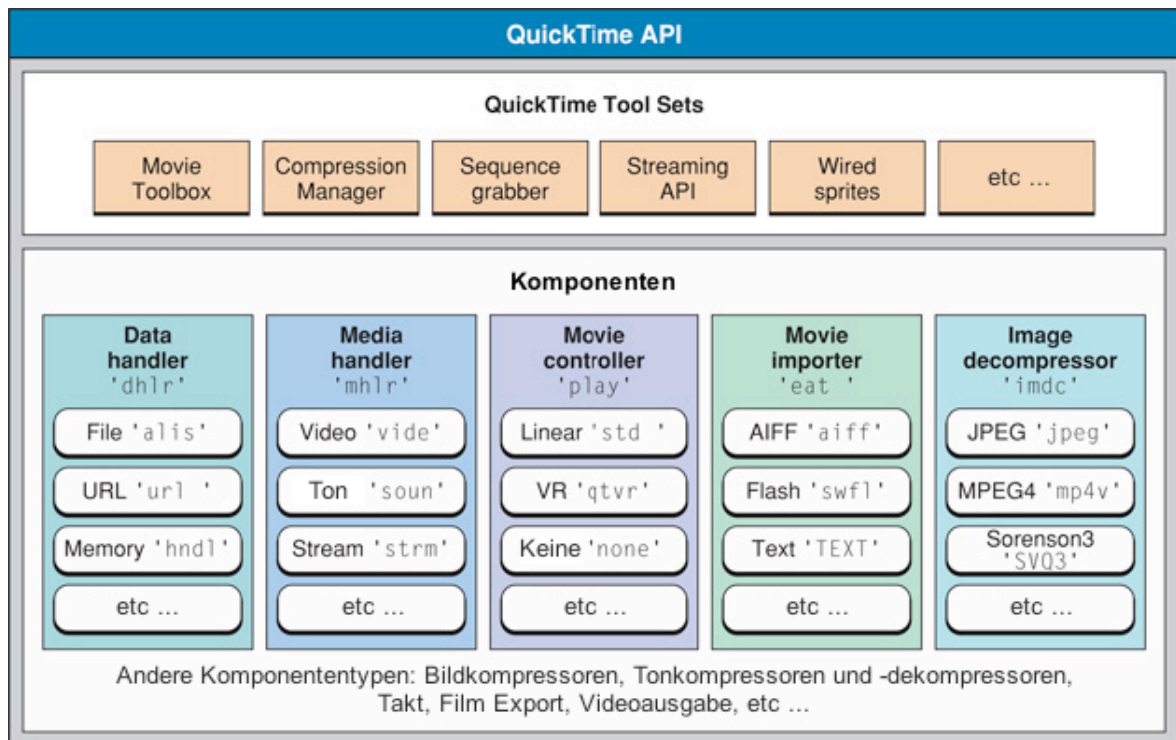


Abbildung 2.1: QuickTime API – Tool Sets und Komponenten⁸

Die Komponente „Media Handler“ beispielsweise ist während des Abspielens eines Films mit mehreren Aufgaben beschäftigt. Einerseits übernimmt sie von der Komponente „Data Handler“ die eigentlichen Mediendaten und beauftragt einen anderen Baustein mit deren Dekomprimierung. Danach sendet sie die Daten an den QuickTime Ausgang, von wo aus weitere Mac OS X Technologien die Darstellung übernehmen (z.B. „Core Audio“ die Ausgabe an Lautsprechern oder „Quartz“ die Darstellung auf einem Display). Ob die Daten nun von einer

⁸ Übersetzte Fassung von: http://developer.apple.com/documentation/QuickTime/RM/Fundamentals/QTOverview/art/toolsets_components.jpg

Festplatte oder dem Internet kommen, ist für den „Media Handler“ ohne Bedeutung. Damit hat sich vorher schon der „Data Handler“ beschäftigt und eine entsprechende Komponente zum Datenempfang genutzt.

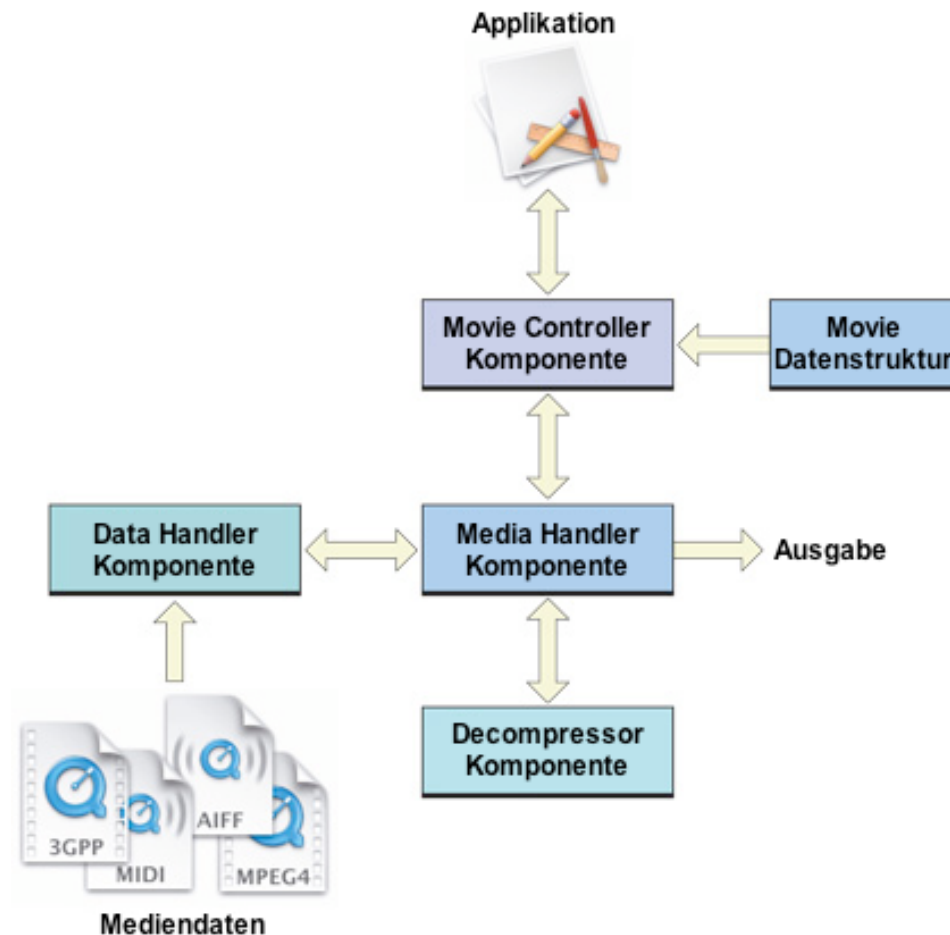


Abbildung 2.2: Das Zusammenspiel der QuickTime Komponenten während des Abspiels eines Films⁹

Der Programmierer trifft nun selbst die Entscheidung, bis zu welcher Tiefe er die Vorgänge in QuickTime mit seinem Programm beeinflussen will. Er hat die Möglichkeit, jeden einzelnen Vorgang und jedes Detail – bis hin zum einzelnen Daten-Sample – zu kontrollieren bzw. bestimmte Abläufe zu verändern (z.B. die Art der Ausgabe oder der Steuerung). Er kann aber auch mit nur wenigen

⁹ Übersetzte Fassung von: http://developer.apple.com/documentation/QuickTime/RM/Fundamentals/QTOverview/art/playing_a_movie.jpg

Befehlen an die Movie Toolbox einen Film abspielen und die vorgefertigten Routinen von QuickTime automatisch ausführen lassen.

Durch die Flexibilität der Mac OS X Architektur kann er dies in verschiedenen Varianten tun: entweder direkt durch die Programmierung in C, C++, Objective-C oder Java oder aber, bei der Benutzung einer anderen Programmiersprache wie Basic oder Pascal, durch Aufrufe von Frameworks¹⁰, die vom Mac OS bereit gestellt wurden.

2.2.3 Die Struktur von QuickTime Movies

Das QuickTime Movie ist eine Beschreibung einer multimedialen Präsentation. Es liefert dem abspielenden Gerät Informationen darüber, welcher Medientyp behandelt wird, wo die Daten vorhanden sind, wann und wie die einzelnen Daten- und Medientypen abgespielt werden sollen und wie die einzelnen Elemente bei der Darstellung angeordnet sind.

Ein QuickTime Movie enthält keine tatsächlichen Mediendaten, wie z.B. Videobilder oder Audiosamples. Ein Movie ist vielmehr Inhaltsverzeichnis und Arbeitsanweisung, welche es dem Computer erlauben, die Mediendaten zu finden und entsprechend auszuwerten.

2.2.3.1 Movies und Filmdateien

Es ist wichtig, zwischen der eben diskutierten Datenstruktur „Movie“ und einer Filmdatei (dem „Movie File“) zu unterscheiden.

Das Movie File kann eine gespeicherte Kopie dieser Datenstruktur enthalten: oder aber auch nur eine Referenz zu einer solchen Struktur.

Wenn das Movie File ein Movie enthält, kann es optional auch die benötigten Mediendaten enthalten. Dies wird dann ein „selbständiger Film“ (Self-Contained Movie File“) genannt und ist wahrscheinlich die häufigste Auftretensweise einer Filmdatei.

¹⁰ Jede Technologie (wie z.B. QuickTime, OpenGL, Quartz, Java, etc.) besitzt ein Framework. Dieses stellt eine Schnittstelle zur API dieser Technologie dar. Durch das Ansprechen des Frameworks aus einer beliebigen Programmierungsumgebung heraus, wird es möglich, die entsprechende Technologie einzubinden.

Es gibt drei Arten von Movie Files¹¹:

- Reference Movie File: Es enthält nur die Adresse des Movie Files. (z.B. die Pfadangabe zu einer anderen Datei, oder eine URL zu einer Datei im Internet.
- Movie File: Es enthält nur das „Movie“ (also die Datenstruktur).
- Self-Contained Movie File: Es enthält sowohl das „Movie“ als auch die Mediendaten. Diese sind ineinander verschachtelt angeordnet, um eine flüssigere Wiedergabe zu ermöglichen.

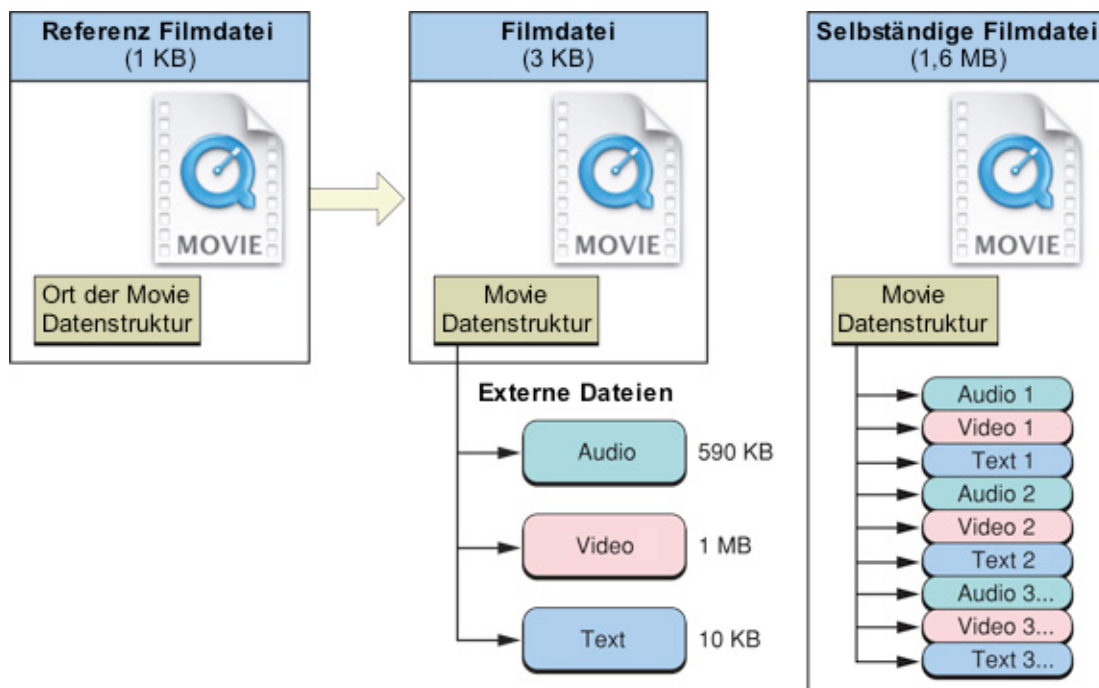


Abbildung 2.3: Die verschiedenen Arten von QuickTime Filmdateien¹²

¹¹ Der Autor möchte an dieser Stelle auf die unterschiedlichen Termini in der QuickTime Dokumentation und in der später nachfolgenden Dokumentation seiner Software hinweisen und bittet um Beachtung von „Bemerkungen zu QuickTime Termini“.

¹² Übersetzte Fassung von: http://developer.apple.com/documentation/QuickTime/RM/Fundamentals/QTOverview/art/movie_files.jpg

2.2.3.2 Spuren

Ein QuickTime Movie ist in Spuren organisiert. Die Anzahl der Spuren innerhalb eines Movies wird nur durch die Leistungsfähigkeit des Computers beschränkt.

Jede Spur ist einem Medientyp – z.B. Bild, Ton oder Text – zugeordnet und beinhaltet eine Datenreferenz, in welcher beschrieben wird, wo die tatsächlichen Mediendaten für diese Spur gefunden werden können. Eine Spur kann auch ein Kompressionsformat (wie MotionJPEG Video oder GSM Audio) vorschreiben.

Die Datenreferenz kann auf eine lokal vorhandene Datei, eine Datei im Netzwerk oder auf einem Internet Server, einem Datenstrom aus dem Netzwerk oder von einem Internet Server oder auf eine bestimmte Adresse im Arbeitsspeicher des Computers verweisen. Andere Datenreferenztypen sind auch möglich und der Typ selber ist erweiterbar. Einfach gesagt: Die Mediendaten können überall sein. Eine Datenreferenz identifiziert die Datenquelle.

Unterschiedliche Spuren können auf dieselbe Datenquelle verweisen oder aber auch auf jeweils unterschiedliche Quellen. Alle für den Film benötigten Mediendaten können in einer Datei untergebracht sein, oder aber auch beispielsweise die Tondaten in einer Datei und die Bilddaten in einer anderen. Möglich ist auch die Verbindung zwischen lokal vorhandenen und auf einem Internet Server gespeicherten Datenquellen.

Eine einzelne Spur kann nur einem Medientyp zugeordnet sein, und die meisten Spurtypen beziehen ihre Daten von einer einzelnen Quelle. Manche Medientypen unterstützen aber auch mehrere Datenquellen. Beispielsweise kann eine Videospur aus einer Serie von JPEG Bildern bestehen, die jeweils in einer separaten Datei gespeichert sind. In diesem Fall gibt es für jede dieser Bilderdateien eine eigene Datenreferenz.

Unterschiedliche Spuren können dieselben oder auch unterschiedliche Medientypen besitzen; zum Beispiel mehrere Video- und mehrere Tonspuren im selben Film oder mehrere Textspuren in verschiedenen Sprachen.

Eine einzelne Spur kann nur einen Kompressionstyp nutzen, aber mehrere Spuren desselben Medientyps können unterschiedliche Kompressionen innerhalb des Movies besitzen. Beispielsweise kann ein Film sowohl MP3- als auch MPEG4-komprimierte Audiospuren enthalten.

2.2.3.2.1 Spuren und ihre Eigenschaften

Spuren, die visuell dargestellt werden, besitzen Eigenschaften wie Höhe, Breite, Abstand zum Nullpunkt der Darstellungsebene, Anzahl der Darstellungsebenen und den graphischen Darstellungsmodus. Dies erlaubt die Darstellung von mehreren visuell sichtbaren Spuren gleichzeitig: nebeneinander, teilweise oder vollständig überlappend und mit einstellbaren Werten für Transparenz oder Transluzenz¹³. Auch enthalten ist eine Transformationsmatrix, die zur Rotation, Skalierung oder Kippung der visuellen Darstellung während des Abspielens genutzt werden kann.

QuickTime stellt Funktionen zum Abschneiden des Bildes an den Grenzen der Spur zur Verfügung. Dies wird mit Hilfe von Masken frei wählbarer Form realisiert.

Tonspuren besitzen Eigenschaften wie Lautstärke, Panorama und Frequenzgangregelung. Mehrkanalige Tonformate wie Vierkanal oder 5.1 Surround Sound, werden noch nicht unterstützt.¹⁴

2.2.3.2.2 Spuren und ihre Zeitskala

Jede Spur hat ihre eigene Zeitskala: eine Zahl von Einheiten pro Sekunde, die passend dafür ist, den jeweiligen Inhalt der Spur angemessen zu beschreiben. Diese Zeitskala kann festgelegt werden, wenn das Movie neu angelegt wird. Beispielsweise könnte für eine PAL Videospur eine Skala von 25 angemessen sein, so dass jedes Vollbild mit einer Zähleinheit korrespondiert. Einer Tonspur von einer CD könnte eine Skala von 44100 zugewiesen werden, um mit jeder Einheit ein Sample zu beschreiben; oder aber auch eine Skala von 1000, um eine Millisekunde darzustellen.

Diese Einteilung der Zeitskalen kann bei der Erstellung eines Movies frei gewählt werden. Andernfalls wird von QuickTime eine angemessene Skala vorgegeben.

Das Movie selbst besitzt zusätzlich eine unabhängige Zeitskala (welche ebenfalls bei der Erstellung des Movies gewählt werden kann), um die verschiedenen Spuren anhand einer übergeordneten Zeitleiste koordinieren zu können.

¹³ Transluzenz: Grad der Lichtdurchlässigkeit.

¹⁴ Stand: QuickTime 6.5.2. QuickTime 7 unterstützt mehrkanalige Tonformate.

QuickTime besorgt die Synchronisierung der einzelnen Spuren automatisch, aber die QuickTime API stellt auch Funktionen zur Konvertierung von Zeiteinheiten in Zeitskaleneinheiten zur Verfügung, um mit Spur- und Movie-Zeiten direkt arbeiten zu können.

2.2.3.2.3 Spurdaten (Spur Media)

Genauso wie das Movie nur eine Datenstruktur ist und nicht die tatsächlichen Bild- und Tondaten enthält, enthält auch jede Spur eine ähnliche Datenstruktur: das Media. Es beschreibt den Ort, die Dauer und die natürliche Zeitskala der tatsächlichen Mediendaten. Dies kann verwirrend sein, denn im umgangssprachlichen Gebrauch werden diese Mediendaten manchmal als Spurdaten bezeichnet.

2.2.3.2.4 Schnittliste einer Spur

Jede Spur enthält eine Schnittliste, welche es erlaubt, die Reihenfolge der Darstellung der Mediendaten zu ändern ohne die Mediendaten selbst neu anordnen zu müssen (Nichtdestruktives Editieren). Es ist möglich ein Segment einer Spur heraus zu schneiden, ohne tatsächlich Daten zu löschen, oder aber ein Segment mehrmals zu wiederholen, ohne die Größe der Datenquelle durch duplizierte Daten zu vergrößern.

In der Schnittliste kann auch die Dauer eines Segments geändert werden, um es schneller oder langsamer abzuspielen. Es könnte ebenso ein leeres Segment eingefügt werden, welches für eine gewisse Dauer keinerlei Daten enthält.

Wenn eine Spur nicht editiert wurde, ist die Schnittliste leer und die gesamte Spur wird als ein zusammenhängendes Segment behandelt. Alle Daten werden in der abgespeicherten Reihenfolge und in der normalen Geschwindigkeit abgespielt.

2.2.3.2.5 Länge einer Spur

Jede Spur hat eine bestimmte Länge, welche aus den addierten Längen aller Segmente (typischerweise der Dauer aller verwendeten Mediendaten) und allen leeren Segmenten besteht.

Gleichzeitig hat auch jedes Movie eine bestimmte Länge, welche sich aus der Länge der längsten Spur ergibt.

2.2.3.3 Mediendaten (Samples)

Eine QuickTime Spur enthält außerdem einen Satz von Sample-Tabellen. Jeder Eintrag in einer Sample-Tabelle beschreibt den Ort und die Dauer eines kurzen Abschnitts (Chunk) von Mediendaten (Samples), wie z.B. einem Foto, einem Vollbild einer Bildspur, einer Sequenz von PCM Audio Samples oder einem einzelnen Textabschnitt.

Für jede Sampletabelle gibt es mindestens eine Samplebeschreibung (Sample Description). Diese Beschreibung stellt die benötigten Informationen bereit, um die gespeicherten Samples in ein Format zu übersetzen, mit dem der Media Handler (s.o.) arbeiten kann. Beispielsweise kann eine Samplebeschreibung die Höhe, Breite und das Pixelformat eines Bildes beinhalten, oder die Auflösung und Samplingrate einer Gruppe von PCM Audiosamples.

Für manche Medientypen, unter anderem Ton, teilen sich alle Mediendaten in einer Spur eine einzige Samplebeschreibung. Um Audiodaten verschiedener Bitlängen und/oder Samplingraten verwenden zu können, müssen sich diese in jeweils getrennten Spuren befinden.

Andere Medientypen können mehrere Samplebeschreibungen verwalten: eine Serie von Einzelbildern könnte unterschiedliche Höhen und Breiten besitzen, mit verschiedenen Samplebeschreibungen, die genutzt werden, sobald sich die Abmessungen ändern.

2.2.3.4 Sampledauer und Bildrate

Da jeder Abschnitt von Samples (Chunk) seine eigene Länge hat und ein Abschnitt so klein wie ein einzelnes Sample sein kann, muss eine QuickTime Spur nicht automatisch eine feste Bildrate besitzen. Eine Videospur kann aus einer Serie von Einzelbildern bestehen, die wie eine Diashow gezeigt werden und jedes „Dia“ wird dabei mit einer anderen Länge vorgeführt.

Dies kann zu Verwirrung führen; speziell wenn man das Arbeiten mit Medien mit fester Bildrate gewohnt ist. Eine feste Bildrate würde es erfordern, Bilder periodisch zu wiederholen, eventuell sehr oft, um sie für eine längere Zeit auf

dem Bildschirm darzustellen. Mit QuickTime kann jedes Bild als einzelnes Sample mit seiner eigenen Länge gespeichert werden.

Zusätzlich ist ein QuickTime Movie nicht an eine feste Bildrate gebunden. Eine PAL Videospur mit 25 Bildern pro Sekunde kann Seite an Seite mit einer NTSC Videospur mit 30 Bildern pro Sekunde dargestellt werden. Beispielsweise sind beide Spuren positioniert über einem stillstehenden Einzelbild, das für den gesamten Ablauf des Films dargestellt wird; oder über einer „Diashow“, deren Bilder in unregelmäßigen Abständen wechseln. Dies ist möglich, da die Darstellung erst während des Abspielens von einem Programm erzeugt und nicht nur mechanisch projiziert wird.

Natürlich kann eine Spur oder ein Movie eine feste Bildrate besitzen; es ist für eine Videospur üblich, dass sie eine Serie von Samples enthält, die alle dieselbe Länge besitzen, und es ist genauso für ein Movie üblich, nur eine Videospur mit einer konstanten Samplerate zu besitzen. Aber es ist keine Voraussetzung.

2.2.3.5 Zeitskalen

Wie schon erwähnt, besitzt ein Movie eine Zeitskala; genauso jede Spur und die Spurdaten (Media). Eine Zeitskala definiert eine gewisse Anzahl von Einheiten pro Sekunde. Für die Spurdaten ist die Zeitskala normalerweise die Samplerate. Für eine Spur oder ein Movie kann die Zeitskala ein angemessener Wert sein. (Falls bei der Erstellung eines Movies keine andere Angabe gemacht wird, benutzt QuickTime den Wert 600.)

Die Beziehung zwischen den Zeitskalen des Movies, der Spuren und der Spurdaten definiert das **Zeitkoordinatensystem**. QuickTime nutzt dieses Koordinatensystem zur Synchronisierung aller Spuren und ihrer Inhalte an der Zeitleiste des Movies.

Ein Movie besitzt immer eine **Jetzt-Zeit**, welche die Teile des Movies bestimmt, die gerade jetzt dargestellt werden sollen. Die Jetzt-Zeit wird in Einheiten der Zeitskala des Movies ausgedrückt. Wenn beispielsweise die Zeitskala des Movies 600 beträgt und das Movie für eine halbe Sekunde abgespielt wurde, beträgt die Jetzt-Zeit 300.

Die Jetzt-Zeit reicht von 0 bis zur kompletten Länge des Films. Sie ändert sich fortlaufend, während der Film abgespielt wird. Auch das Verziehen des

Abspielkopfes in einem Filmabspieلفenster verändert den Jetzt-Zeitwert des Movies.

Ein Movie besitzt zudem eine **Abspielrate**, welche 0 ist, wenn der Film gestoppt ist, und 1, wenn der Film in normaler Geschwindigkeit (definiert durch die Zeitskala des Movies) abgespielt wird. Ein Movie mit der Zeitskala 600 spielt beispielsweise mit 600 Einheiten pro Sekunde, wenn die Rate 1 ist. Negative Abspielraten führen zu einem Rückwärtslauf des Films und Raten größer oder kleiner als 1 bewirken die schnellere oder langsamere Wiedergabe als normal.

QuickTime etabliert eine Zeitbasis für die Wiedergabe, wenn ein Film abgespielt wird. Diese Zeitbasis ergibt sich aus einer Kombination des Zeitkoordinatensystems, der Abspielrate, der Jetzt-Zeit und der **Referenz zur System-Uhr**. Letztere versorgt QuickTime mit Daten über die Echtzeit. Dies macht es möglich, einen Film mit der korrekten Anzahl von Zeitskaleneinheiten entsprechend der Abspielrate in Echtzeit darzustellen.

Dies gewährleistet auch eine exakte Synchronisierung zwischen den einzelnen Spuren. Sollte die Leistung des wiedergebenden Computers nicht ausreichend sein, um alle Spuren perfekt wiederzugeben, kann QuickTime angemessene Maßnahmen ergreifen (z.B. das Auslassen von Einzelbildern), um die Wiedergabe in Echtzeit zu sichern.

2.2.4 Einbindung in das Betriebssystem

QuickTime ist ein integraler Bestandteil des Mac OS X Betriebssystems und daher auf jedem Apple Computer standardmäßig vorhanden. Jede Art von Multimedia Darstellung (sei es in Webbrowsern oder Dokumenten) greift an einem gewissen Punkt auf die QuickTime Technologie zur Darstellung des multimedialen Inhalts zurück.

Programme, mit denen sich Medieninhalte erstellen oder manipulieren lassen, sind jedoch im Standardumfang des Mac OS nur eingeschränkt enthalten.

Ein Beispiel hierfür ist der QuickTime Player, der in der normalen Variante nur Filme abspielen, nicht aber diese manipulieren oder speichern kann.

Andere Möglichkeiten (z.B. zur Nachbearbeitung von Fotos oder zum Schnitt von Video) sind nur in funktionsärmeren (weil älteren) Versionen enthalten. Die aktuellen Versionen dieser so genannten „iLife“ Programme sind kostenpflichtig.

Wichtig ist aber, dass der gesamte Funktionsumfang von QuickTime auf jedem Apple Computer prinzipiell vorhanden ist und durch entsprechende Programme (seien sie kostenpflichtig oder nicht) genutzt werden kann.

2.2.5 Bemerkungen zu den QuickTime Termini

Die QuickTime Dokumentation definiert drei verschiedene Arten von Filmdateien: das „Reference Movie File“, das „Movie File“ und das „Self-Contained Movie File“.¹⁵

Der QuickTime Player benutzt eine andere Terminologie. Diese wurde auf Grund des Vorbildcharakters des Programms von den meisten QuickTime-basierten Bearbeitungssaplikationen übernommen.

Der Autor möchte daher im Folgenden neue Termini einführen, die sich am QuickTime Player orientieren und so dem allgemeinen Verständnis der Begriffe eher gerecht werden:

Art des Films	Terminus in der QuickTime Dokumentation	Nachfolgend benutzter Terminus
Eine Datei, die nur den Speicherort der Movie Datenstruktur enthält (ähnlich einer URL).	Reference Movie File / Referenz Filmdatei	--
Eine Datei, welche nur die Movie Datenstruktur enthält.	Movie File / Filmdatei	Abhängiger Film / Reference Movie
Eine Datei, die sowohl die Movie Datenstruktur als auch die tatsächlichen Mediendaten enthält.	Self-Contained Movie File / Selbständige Filmdatei	Selbständiger Film / Movie / Film

¹⁵ Vgl. Abb. 2.3 im Abschnitt 2.2.3.1

3. Konzeption des Programms „QT Sync“

3.1 *Asynchronitäten*

Als Asynchronität wird das Vorhandensein einer zeitlichen Verzögerung zwischen zusammengehörigen Ereignissen in Bild und Ton bezeichnet.

Geübte Betrachter erkennen diese bereits ab einem Versatz von zehn bis zwanzig Millisekunden, während der durchschnittliche Zuschauer eine Beeinträchtigung erst ab Werten von fünfzig bis hundert Millisekunden wahrnimmt.

Zwei Arten von Asynchronität sind für diese Arbeit relevant:

- Gleichbleibende Asynchronität

Bild und Ton sind während der gesamten Laufzeit des Films um einen festen Wert gegeneinander verschoben.

Eine Korrektur kann durch die Änderung des Startzeitpunkts einer der beiden Spuren erreicht werden. Der entstehende zeitliche Versatz wird als „Offset“ (engl.: Abstand) bezeichnet.

In QuickTime wird der Offset einer Spur als der zeitliche Abstand zwischen ihrem Startzeitpunkt und dem Beginn des Films angesehen.

- Sich kontinuierlich verändernde Asynchronität

Die Asynchronität zwischen Bild und Ton verstärkt sich während der Laufzeit des Films um einen gleich bleibenden Wert. (Die Spuren „laufen auseinander“.)

Dies kann durch eine Veränderung der Abspielgeschwindigkeit einer der beiden Spuren korrigiert werden.

QuickTime Filme definieren die Abspielgeschwindigkeit einzelner Spuren unter anderem durch die Angabe der „Duration“ (engl.: Länge, Dauer). Eine Modifizierung dieses Werts resultiert in einer schnelleren oder langsameren

Wiedergabe – mit all ihren Nebenwirkungen: Tonspuren beispielsweise verändern ihre Tonhöhe („Pitch Shifting“)¹⁶

Diese Arten der Asynchronität können auch in Kombination auftreten.

3.2 Ursachen eines Bild-Ton-Versatzes

Es gibt zahlreiche Gründe, die eine Asynchronität in Filmdateien verursachen können:

- Das Ausgangsmaterial ist bereits asynchron

Zeitdruck und Geldmangel bei der Produktion können Ursachen für Synchronfehler sein. Gerade bei Fernsehproduktionen schleichen sich oftmals leichte Asynchronitäten ein, die vom durchschnittlichen Zuschauer meist nicht bemerkt werden.

Konvertierungsfehler oder Verzögerungen innerhalb von Übertragungswegen, die Bild und Ton getrennt transportieren, stellen eine weitere Fehlerquelle dar. Beispiele dafür wären das digitale Fernsehen DVB, bei dem durch das getrennte Dekodieren der Bild- und Toninformationen (speziell bei der Benutzung von sog. Tunerkarten für den Computer) Verzögerungen entstehen können, oder aber auch die teilweise noch übliche Praxis, bei Live-Übertragungen das Bild per Satellit und den Ton über Telefon zu übertragen.

- Fehlerhafte Kodierung der Einzelspuren

Bei der Herstellung von Filmdateien werden Bild und Ton völlig getrennt voneinander kodiert. Ein zeitlicher Bezug untereinander besteht während der Kodierung nicht.

Durch fehlerhafte Software oder eine Überlastung des arbeitenden Computers können Fehler entstehen, welche sich auf die Gesamtlänge der Spur auswirken.

¹⁶ Seit QuickTime 7 wird für die Längenveränderung von Tonspuren auch die Funktion des „Time Stretching“ unterstützt. Dabei handelt es sich um einen Algorithmus, der die Abspielgeschwindigkeit, nicht aber die Tonhöhe verändert.

(Als Beispiel seien hier die Störungen einer MP3-Datei genannt, welche sich als „Blupser“ oder „Knacker“ äußern.)

Auch eine fehlerhafte Angabe der Abspielgeschwindigkeit, welche durch den kodierenden Codec in die enkodierte Spur geschrieben wird, kann zu einem „Auseinanderlaufen“ von Bild und Ton führen. Selbst eine „geringe“ Abweichung um ein Hundertstel Prozent führt innerhalb von zwanzig Minuten zu einer Asynchronität von 120 Millisekunden. (Im PAL-Gebiet entspricht dies 3 Vollbildern.)

Dem Autor liegt das Beispiel eines Benutzers vor, der beim „Capturing“ (also der Digitalisierung von analogem Bild und Ton mit Hilfe einer entsprechenden Erweiterungskarte) ein Auseinanderlaufen von Bild und Ton des digitalisierten Materials beobachtet.¹⁷ Da, nach Meinung des Benutzers, die Karte ansonsten keinerlei Probleme aufweist, sondern sogar in Punkto Bildqualität und Bedienbarkeit in ihrer Preisklasse „unschlagbar“¹⁸ ist, würde hier ein Programm, welches die Asynchronität behebt, den Benutzer davon abhalten, die Karte auszutauschen.

- Unsachgemäße Herstellung der Filmdatei

War das Rippen¹⁹ von DVDs vor einigen Jahren noch eine langwierige Prozedur, die mit Hilfe von verschiedensten Programmen und größerer Sachkenntnis erledigt werden musste, so sind heute mehrere Programme erhältlich, die alle notwendigen Arbeitsschritte in einem Durchgang erledigen.

Diese Programme sind bewusst sehr einfach gehalten, um auch dem unerfahrenen Benutzer ein Instrument in die Hand zu geben, mit dem er, ohne viel von der Materie zu verstehen, ein vorzeigbares Resultat erzeugen kann.

¹⁷ Forum Maclife.de, Forum >> Apple-Talk >> Audio & Video >> „AlchemyTV DVR oder doch was anderes“ /

<http://maclife.de/index.php?name=PNphpBB2&file=viewtopic&p=296761>

¹⁸ ebd., Posting von „Mr. Ash“ am 9.3.2005, 00.59 Uhr

¹⁹ (engl.) to rip: zerreißen, aufreißen

Der Begriff beschrieb ursprünglich das Umgehen des Kopierschutzes einer DVD, wird aber heute für den kompletten Vorgang des Herstellens einer Filmdatei von einer Vorlage benutzt. Dabei kann die Vorlage sowohl eine DVD als auch ein anderes Medium (z.B. VHS, Fernsehen oder eine Satellitenübertragung) sein.

Es liegt in der Natur der Sache, dass automatisierte Prozesse nur dann einwandfreie Resultate liefern können, wenn das Ausgangsmaterial fehlerfrei vorliegt. Auf Besonderheiten oder „exotische“ Ausgangsbedingungen (wie z.B. unterschiedlich lange Bild- und Tonspuren innerhalb der DVD) können diese Programme nicht flexibel genug reagieren.

Viele Programme sind auch nicht ausgereift genug, um allen Anforderungen fehlerfrei genügen zu können. Da das Umgehen eines Kopierschutzes rechtlich eine bedenkliche Angelegenheit ist²⁰, werden viele dieser Programme als so genannte „Freeware“ oder „Shareware“ (d.h. kostenlos oder gegen eine geringe Gebühr) angeboten. Die Programmierer haben also meistens nicht den finanziellen Hintergrund, um ein professionelles Produkt herzustellen.

- Asynchronität durch unterschiedliche Latenzzeiten der wiedergehenden Codecs oder darstellenden Geräte

Abhängig von der verwendeten Software, dem Betriebssystem und der zur Verfügung stehenden Prozessor- und Grafikkartenleistung kann die Latenzzeit²¹ vom abspielenden Programm nicht korrekt ermittelt und ausgeglichen werden.

Auch die Ausgabe auf TFT-, LCD- oder Plasmabildschirmen kann eine Verzögerung des Bildes bis zu 100 Millisekunden²² bewirken.

- Inkompatibilitäten

Das Abspielen von QuickTime-fremdem Material gestaltete sich an Apple Computern schon immer schwierig, da einige benötigte Fremdcodecs gar nicht in einer Mac-Variante existieren (z.B. ein WMA-Codec für Mac OS X) und andere nicht immer einwandfrei arbeiten.

Ein wichtiger Codec für das Abspielen von AVIs ist der „DivX for Mac“ Codec. Er ist nicht nur für das Dekodieren von DivX Bildmaterial zuständig, sondern er ist es auch, der (unabhängig vom verwendeten Bild-Codec) die in AVIs oft genutzten

²⁰ Vgl. §95a, §108b UrhG / <http://transpatent.com/gesetze/urhg.html>

²¹ Latenzzeit: Die Zeit, die der Codec braucht, um Bild bzw. Ton zu dekodieren und dem abspielenden Programm zur Verfügung zu stellen.

²² YOSHIDA, J., 2004: Synching Lucy's lips with TV sound. EE Times, 19.4.2004 / <http://www.eetimes.com/showArticle.jhtml?articleID=18901792>

MP3-Tonspuren für QuickTime benutzbar macht. (Ohne installierten DivX Codec bleibt ein AVI mit MP3-Tonspur stumm). DivX für Mac gibt es erst seit Dezember 2001²³.

Ein anderes Beispiel wäre der Open-Source²⁴ Codec XviD. Dieser wird zwar für die Windows-Welt von einer großen Programmierergruppe entwickelt; für den Mac existierte allerdings bis vor einem Jahr keine native Version. Stattdessen mussten ähnlich arbeitende Fremdcodecs zur XviD Dekodierung verwendet werden.²⁵

3.3 Analyse vorheriger Lösungen

Das nachträgliche „Synchronziehen“ von Filmdateien war bisher schon mit mehreren Programmen möglich. Als Beispiele seien hier „QT Mutator“ und „Synchole“ genannt.

Außerdem soll der im Betriebssystem mitgelieferte „QuickTime Player“ vorgestellt werden, da er eine Referenz für alle anderen Filmabspielprogramme darstellt.

²³ Der erste Eintrag im „DivX for Mac“-Forum datiert auf den 19.12.2001. Mit folgenden Worten wird die Mac-Version des Codecs vorgestellt:

„So take your Powerbooks, iMac's, G3's, G4's... Install DivX and soon, very soon. The world will be yours!!“ („So nehmt Eure Powerbooks, iMac's, G3's, G4's... Installiert DivX und bald, sehr bald, wird die Welt Euch gehören!!“)

<http://forums.divx.com/viewtopic.php?topic=22901&forum=14>

²⁴ Open-Source („offener Quelltext“) bezeichnet Software, deren Code der Öffentlichkeit zugänglich ist. Jeder, der qualifiziert ist, kann zu der Weiterentwicklung der betreffenden Software beitragen. Die Software ist durch eine spezielle Lizenz geschützt und wird in der Regel kostenlos im Internet angeboten.

²⁵ Eine Möglichkeit war die Übergabe der XviD Daten an den DivX Codec, welcher wegen der Ähnlichkeit der beiden Verfahren relativ passable Resultate erzielte. Auch der 3ivx Codec ist in der Lage, XviD Material zu dekodieren.

Seit 2004 allerdings bietet Christoph Nägeli, einer der Programmierer im XviD Projekt, den ersten tatsächlichen XviD Codec für Mac an.

<http://people.ee.ethz.ch/~naegelic/index.php>

3.3.1 QT Mutator

Autor: unbekannt

Homepage: <http://www.geocities.com/phredreic/Downloads.html>

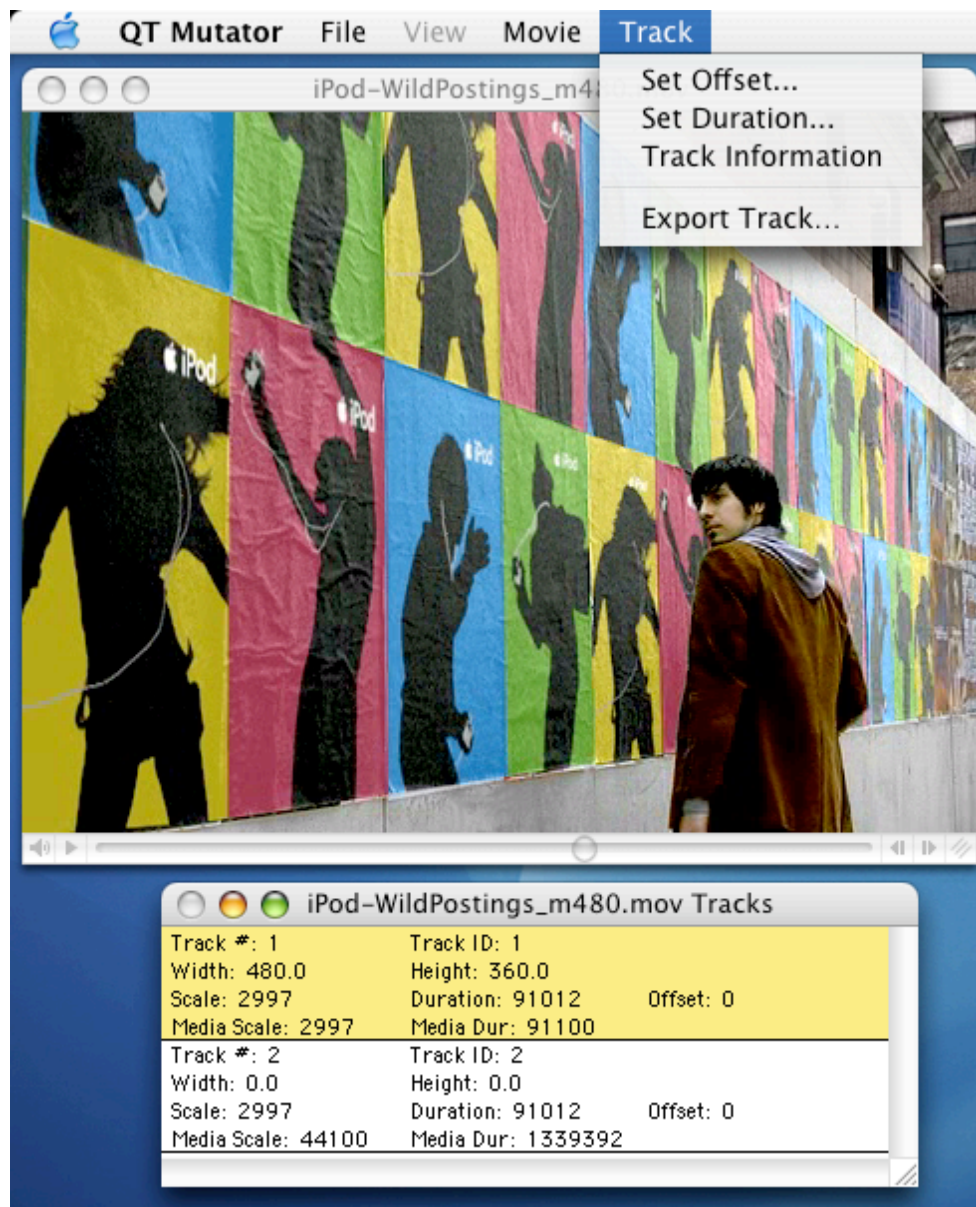


Abbildung 3.1: Die Bedienoberfläche des „QT Mutator“

QT Mutator bietet ein Abspielfenster, das den Film in halber, normaler oder doppelter Größe abspielt. Die in QuickTime üblichen Tastaturkommandos werden unterstützt.

Ein weiteres Fenster stellt die Informationen über die enthaltenen Spuren des Films dar. Per Selektion der gewünschten Spur und Auswählen eines Menübefehls erhält der Benutzer die Möglichkeit, die Parameter zu bearbeiten.

Veränderbar und abspeicherbar sind: Darstellungsgröße und Länge des Films, Offset und Länge der einzelnen Spuren.²⁶

Zusätzlich können einzelnen Spuren an- bzw. ausgeschaltet und exportiert werden.

Vorteile von QT Mutator:

- Umfangreiche Information über die Eigenschaften des Films und der einzelnen Spuren.
- Eingabe sowohl in Sekunden als auch in TimeScales möglich.

Nachteile von QT Mutator:

- Das Programm wird nicht mehr weiter entwickelt. Die letzte Version v1.1b stammt vom 25.10.2001.
- Diese Version ist eine Beta, welche mehrere gravierende Fehler aufweist, z.B. Abstürze beim Speichern.
- Nicht Drag & Drop-fähig.
- Es werden nur QuickTime Filme mit der Dateierweiterung „.mov“ geöffnet.
- Kein Full Screen Modus.
- Umständliche Bedienung: Es existieren keine Kurzbefehle für Menüeinträge und das Fenster mit den einzelnen Parametern ist schlecht bedienbar. (Apples Richtlinien für die Gestaltung von Bedienelementen empfehlen für solch eine Art von Fenster die Benutzung eines „Fliegenden“ Fensters – eines Fensters, das sich immer im Vordergrund befindet und über allen anderen Fenstern schwebt²⁷. Diese Richtlinie hat der Entwickler von QT Mutator missachtet.)

²⁶ Zur Erinnerung: In QuickTime Filmen bedeutet die Änderung der Länge eine Veränderung der Abspielgeschwindigkeit.

²⁷ Firma Apple (Hrsg.), 2005: Apple Human Interface Guidelines. Kapitel „Types of Windows“ Stand: Mai 2005, http://developer.apple.com/documentation/UserExperience/Conceptual/OSXHIGuidelines/XHIGWindows/chapter_17_section_2.html#/apple_ref/doc/uid/20000957-20000961-TPXREF48

- Es ist nicht möglich, nur Teile einer Spur zu bearbeiten.
- Die werksseitig eingestellten (und nicht ausreichend dokumentierten) Voreinstellungen ergeben beim Abspeichern eines abhängigen Films eine unverhältnismäßig große Datei, da die Movie Datenstruktur unnötigerweise zweimal in die Datei geschrieben wird.

3.3.2 Synchhole

Autor: bantha

Homepage: <http://www.applesolutions.com/bantha/Sync.html>

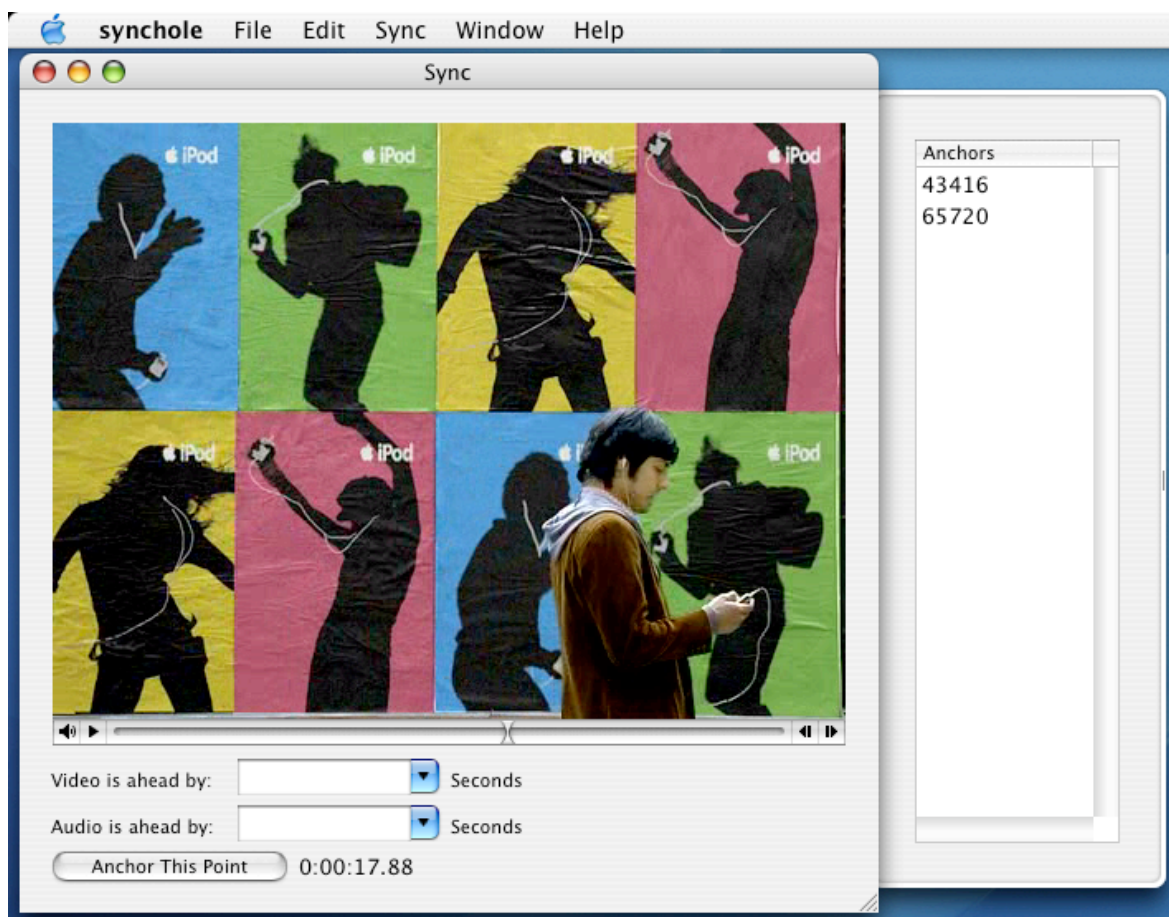


Abbildung 3.2: Die Bedienoberfläche des Programms „Synchhole“

Im Gegensatz zu den Zahlenkolonnen von QT Mutator setzt das Programm Synchhole auf Übersichtlichkeit und richtet sich damit primär an den unerfahrenen Nutzer.

Der Film wird zusammen mit den Eingabefeldern in einem gemeinsamen Fenster dargestellt und besitzt zur Navigation die bekannte QuickTime–Controller Leiste. Skalierungsmöglichkeiten bietet das Programm nicht, sondern es stellt den Film lediglich in Originalgröße dar.

Zwei Eingabefelder erlauben die Angabe, um wie viele Sekunden Bild bzw. Ton zu früh abgespielt werden. Synchole korrigiert daraufhin entweder den Offset der Audiospur oder die Abspielgeschwindigkeit der Bildspur.

Durch den Einsatz von Ankerpunkten ist es möglich, nur bestimmte Teile des Films zu editieren. Diese Punkte sind beliebig setzbar und dienen nur der Selektion des zu bearbeitenden Abschnitts. Alle aktuell existierenden Ankerpunkte können in einem separaten Fenster angezeigt werden.

Vorteile von Synchole:

- Übersichtliche Benutzerführung.
- Bearbeitungen von Teilen des Films möglich.

Nachteile von Synchole:

- Durch fehlende Anzeigen wird ein Nachvollziehen der Arbeitsschritte unmöglich.
- Zu wenig Information für den fachkundigen Benutzer.
- Größe des Filmbildes ist nicht veränderbar.
- Kein Full Screen Modus.
- Der Programmierer von Synchole weist in der beiliegenden Dokumentation auf einen Bug hin, der innerhalb des Arbeitens auftreten kann und zu einem Absturz des Programms führt. Im praktischen Test ist dieser Fehler jedoch nicht reproduzierbar.

3.3.3 QuickTime Player / QuickTime Player Pro

Autor: Apple Computer, Inc.

Homepage: <http://www.apple.com/quicktime/>



Abbildung 3.3: Das Abspielfenster und das Informationsfenster des QuickTime Players (QuickTime Version 6.5.2)

Der QuickTime Player wird von der Firma Apple zusammen mit dem Mac OS X Betriebssystem ausgeliefert und ist Bestandteil jeder Systeminstallation. Er stellt somit eine Referenz dar und dient jeglichem Programm zum Abspielen von Filmen als Vorlage.

Das Abspielfenster bietet eine komfortable Abspielleiste, Zeiteinblendung und Analyzeranzeige mit integrierter Klangregelung. Die Bedienoberfläche ist dem Apple-typischen Stil angepasst. Ausführliche Informationen über den Film sind über Menüs abrufbar.

Der Standard QuickTime Player ist allerdings nur in der Lage, Filme abzuspielen. Funktionen wie Abspeichern und erweitertes Editieren werden erst durch die

kostenpflichtige Erweiterung QuickTime Player Pro²⁸ ermöglicht. Die Wiedergabe von Video im MPEG2-Standard bedarf einer nochmaligen Erweiterung.²⁹

Vorteile des QuickTime Player / QuickTime Player Pro

- In jedem Mac OS vorinstalliert.
- Nahtlose Einbindung in das Betriebssystem.
- Umfangreiche Informationsmöglichkeiten über den Film und seine Datenstruktur.
- Bildschirmfüllende Darstellung (Full Screen) (nur QuickTime Player Pro).
- Editiermöglichkeiten des Films (nur QuickTime Player Pro).
- Speichermöglichkeiten als abhängiger und selbständiger Film (nur QuickTime Player Pro).
- Umfangreiche Möglichkeiten zum Import und Export von Spuren / Filmen (nur QuickTime Player Pro).

Nachteile des QuickTime Player / QuickTime Player Pro

- Pro Variante nicht kostenlos.
- Behebung von Asynchronitäten nur mit QuickTime Player Pro und unter erheblichem Aufwand möglich³⁰.
- Kein Tastaturkommando für schnelles Spulen (Bildsuchlauf).

²⁸ QuickTime 7 Pro für Mac OS X: €29,99 (Stand: Mai 2005) im Apple Store Deutschland
<http://www.apple.com/de/thestore/>

²⁹ QuickTime MPEG-2 Wiedergabe-Komponente für Mac OS X €19,99 (Stand: Mai 2005)
im Apple Store Deutschland
<http://www.apple.com/de/thestore/>

³⁰ Ein möglicher Arbeitsweg wäre, zunächst eine Spur in einen separaten Film zu exportieren, diese dort zu beschneiden, um sie dann in den Film zurückzuimportieren. Erschwerend kommt hierbei hinzu, dass der QuickTime Player Pro bei dem Editieren von Audiospuren nur Schritte von 0,5 Sekunden zulässt.

3.3.4 Fazit

QT Mutator und Synhole bieten gute Ansätze für die Arbeit mit asynchronen Filmdateien, sind aber beide für den praktischen Alltagseinsatz nicht ausgereift genug. Während Synhole durch seine Simplizität nur den Amateur-Sektor abdeckt, ist QT Mutator durch seine Instabilität und unkomfortable Bedienoberfläche für den professionelleren Bereich unbrauchbar.

Der QuickTime Player ist für eine solche Aufgabe erst gar nicht ausgelegt, setzt aber durch seine Verbreitung Maßstäbe für die Bedienung und das Verhalten³¹ eines Abspielprogramms für Filme.

3.4 *Mögliche Einsatzgebiete von QT Sync*

Ausgangspunkt für die Entwicklung von QT Sync war der Bedarf nach einer Möglichkeit zur Korrektur eines Bild-Ton-Versatzes während der laufenden Darstellung des Films im Vollbildmodus. Zusätzliche Funktionen sollten Änderungen der Filmeigenschaften (wie Seitenverhältnis oder Größe) zulassen sowie die Speicherung des Resultats in eine Referenzdatei (ohne Änderung der Originaldatei) ermöglichen.

Daraus ergeben sich unterschiedliche Einsatzgebiete:

- Zur Bearbeitung von asynchronen Filmdateien

Primär sollte sich das Programm an jeden Benutzer richten, der eine asynchrone Filmdatei zu bearbeiten hat. Speziell Benutzer von QT Mutator sollten sich schnell in das Programm einarbeiten können.

Durch die beiliegende Dokumentation sollte es auch einem nicht mit der Materie vertrauten Benutzer möglich sein, relativ schnell ein gutes Ergebnis zu erzielen – eine gewisse Begabung zur Beurteilung von Asynchronitäten vorausgesetzt.

³¹ Apple nennt die Eigenschaft von Programmoberflächen, dem Benutzer einen intuitiven Zugang zu ermöglichen, „Look and Feel“ und hält Programmierer dazu an, ihre Programme dementsprechend zu gestalten.

- Als alternativer Full Screen Player

Durch die Fähigkeit, Filme auch im „Full Screen“ Modus abspielen zu können, richtet sich das Programm auch an diejenigen Benutzer, die „nur“ ein kostenloses Programm zum bildschirmfüllenden Abspielen von Filmen benötigen. (Der QuickTime Player der Firma Apple ist zwar im Mac OS Betriebssystem enthalten, bietet aber erst in der kostenpflichtigen Pro-Version einen „Full Screen“ Modus.)

- Als Hilfsprogramm in der Post Produktion

Ein weiteres Einsatzgebiet ergibt sich dort, wo Filmdateien in der Post Produktion verwendet werden. Speziell die Möglichkeit, mehrere Akte eines Films in eine Filmdatei zusammenfassen (und trotzdem die einzelnen Originaldateien unverändert zu lassen), sowie die TimeCode-Einblendung, welche bei weiterer Benutzung des Films in einem Audioschnitt-Programm sehr hilfreich sein kann, machen das Programm zu einem nützlichen Werkzeug.

3.5 Wahl der Programmiersprache

Der Autor möchte darauf hinweisen, dass eine ausführliche Diskussion über Programmiersprachen, ihre Funktionalität und Eigenheiten weit außerhalb des Rahmens dieser Arbeit liegt und beschränkt sich daher auf eine kurze Vorstellung.

Die Firma Apple bietet Entwicklern, die für den Macintosh programmieren möchten, umfangreiche Möglichkeiten. Auf der firmeneigenen Internetseite unter <http://developer.apple.com> findet der interessierte Besucher alle Informationen über das Macintosh Betriebssystem und über Möglichkeiten, für den Mac zu programmieren. Auch eine eigene Programmierumgebung namens „Xcode“ steht registrierten Mitgliedern³² zur Verfügung. In Xcode kann mit mehreren Programmiersprachen (auch parallel) gearbeitet werden, darunter sind Java, Cocoa und Carbon.

³² Die einfache Mitgliedschaft in der „Apple Developer Connection“ (ADC) ist kostenfrei.

Carbon wurde ursprünglich entwickelt, um Programmierern den Umstieg von Mac OS 9 auf Mac OS X (sprich: Zehn) zu erleichtern.³³ Es basiert auf der Sprache C / C++ und ist speziell für das Mac OS X Betriebssystem optimiert. Für professionelle Entwickler, welche die überaus komplexe Sprache C beherrschen, ist Carbon sicherlich die erste Wahl. Die umfangreiche Dokumentation und der Zugriff auf tausende von Mac OS X Befehlen machen Carbon zu der mächtigsten aller Mac OS X Programmiersprachen.

Cocoa dagegen deckt nicht die gesamte Palette der Möglichkeiten ab, bietet dafür aber eine etwas einfachere Programmiersprache. Es basiert auf Objective-C (einer Unterart von C) und bietet objekt-orientierte³⁴ Programmierung. Viele Apple-eigenen Programme wie TextEdit (ein Textverarbeitungsprogramm), Safari (ein Internet-Browser) und Mail (ein E-Mail Programm) sind in Cocoa programmiert.

Das Programm QT Sync dagegen ist in REALbasic programmiert. Die Entscheidung für diese Sprache wurde weniger durch das Abwägen von Vor- und Nachteilen einzelner Sprachen, sondern durch die fehlende Erfahrung des Autors mit der komplexen Sprache C begründet.

Während der Entwicklung von QT Sync wurde versucht, die Entwicklungsumgebung zu wechseln. Es folgten mehrere Wochen, in denen zunächst mit Carbon, später intensiv mit Cocoa experimentiert wurde. Der Versuch wurde mit der Einsicht beendet, dass es nicht möglich ist, Objective-C innerhalb einer kurzen Zeit zu erlernen.

REALbasic – entwickelt und vertrieben von der Firma Real Software³⁵ – ist eine Mischung aus Basic und objektorientierter Programmierung. Im Vergleich zu

³³ Mac OS X ist nicht einfach nur eine Weiterentwicklung des alten, „klassischen“ Macintosh Betriebssystems. Es wurde von Grund auf neu programmiert (auch unter Nutzung von Open-Source Software) und stellt somit eher ein völlig neues Betriebssystem dar. Programme, die auf diesem System arbeiten sollten, mussten teilweise stark verändert werden.

³⁴ Objekt-orientiert bedeutet, dass die verschiedenen Komponenten (Objekte) des Programms voneinander unabhängig sind. Die Objekte sind in sich geschlossen und geben sich gegenseitig nur Kommandos. Je nach Geschick des Programmierers gestaltet sich dadurch ein Modifizieren einzelner Komponenten eines Programms sehr problemlos.

³⁵ <http://www.realsoftware.com>

Cocoa ist es jedoch viel einfacher zu erlernen, da es viele betriebssystem-interne Vorgänge ohne Zutun des Programmierers erledigt und ihn dadurch von umständlichen Aufgaben befreit.

Dem Programmierer werden viele vorgefertigte Module (z.B. Bedienelemente) angeboten, welche sofort nach Einfügen in das Programm benutzbar sind. Während z.B. bei Cocoa das Auffangen eines Mausklicks auf einen Bedienknopf mehrere komplexe Vorbereitungen erfordert, stellt REALbasic den entsprechenden „Event Handler“ (also ein Mechanismus, der den Klick auf den Knopf auffängt und eine weitere Aktion auslöst; beispielsweise das Ausführen einer Subroutine) direkt zur Verfügung.

Der Nachteil hierbei ist, dass der Programmierer an die vorgegebenen Möglichkeiten gebunden ist und die Funktionalität der Module nur auf Umwegen erweitern kann.³⁶

Es bietet jedoch dem Unerfahrenen die Möglichkeit, sehr schnell und unkompliziert zu einem Ergebnis zu kommen. Auch die Verwendung von herkömmlichem Basic (anstatt des komplexen C) erleichtert die Arbeit.

3.6 Exkurs: Eine kurze Entstehungsgeschichte von QT Sync

Die Programmierung von QT Sync war zunächst ein einfaches Experiment. Aus früheren Erfahrungen mit AmigaBasic³⁷ und Delphi³⁸ war eine Vertrautheit sowohl mit Basic als auch mit objektorientierter Programmierung vorhanden, so dass es möglich war, innerhalb weniger Tage eine erste lauffähige Version herzustellen. Mit dieser ersten Fassung konnten schon Filme an den Bildschirm angepasst dargestellt (der spätere „Fill Screen“ Modus) und der Offset einzelner Spuren verschoben werden.

In weiteren Programmierungszyklen entstanden Informationsfenster, Voreinstellungen und Tastaturkommandos.

³⁶ Vgl. Kap. 5

³⁷ eine Basic-Variante auf dem Commodore Amiga System

³⁸ eine objektorientierte Programmieroberfläche für Windows basierend auf Pascal

Am 30. Mai 2004 wurde das Programm unter der Versionsnummer 0.1.0b1 zum ersten Mal unter der Internetadresse qtsync.de.vu zum kostenlosen Herunterladen angeboten.

Angeregt durch die Rückmeldungen der ersten Benutzer erschienen innerhalb einer Woche drei neue Versionen. Die Neuerungen bis zur Version 0.1.0b5 vom 7. Juni 2004 umfassten unter anderem eine größere Anzahl Tastaturkommandos (darunter auch ein Kurzbefehl für den „Bildsuchlauf“, eine Funktion, die der QuickTime Player nicht bietet), die Möglichkeit zum Ändern des Seitenverhältnisses des Films und eine englischsprachige Anleitung.

Am 21. Juni wurde das Programm in die Liste der „Downloads des Tages“ von macwelt.de aufgenommen. Das Internetportal der auflagenstärksten deutschen Mac-Zeitschrift „MacWelt“ stellt auf dieser Liste ausgewählte Share- und Freeware-Programme vor.

Die Version 0.2.0b1 wurde am 14. Juli veröffentlicht. In dieser war neben neuen Funktionen (Möglichkeit zur Längenveränderung einzelner Spuren und neue Tastaturkommandos für die Arbeit im „Full Screen“ Modus) auch darauf geachtet worden, ein Apple-konformes Verhalten des Programms zu erreichen. War QT Sync zunächst nur auf private Wünsche und Arbeitsweisen des Autors zugeschnitten, so wurde nun versucht, allen Anwendern gerecht zu werden. Dazu gehörten unter anderem eine Verbesserung der Fehlermeldungen und die Möglichkeit, Filme von schreibgeschützten Medien (z.B. CD-Rom) abzuspielen. Zum selben Zeitpunkt wurde auch das Aussehen der Webseite komplett überarbeitet, eine eigene Domain geschaltet (qtsync.com) und ein Forum eingerichtet.

Im September erschien die nächste Version 0.2.1b1, welche nun sowohl mehrere Filme aneinanderfügen als auch Bild- oder Tonspuren aus anderen Filmen in den aktuell geöffneten Film importieren konnte. In dieser Version wurde auch das Aussehen der Informationseinblendungen geändert, die während des Full Screen Modus die Offset- und Längenänderungen anzeigen: Um sich dem Aussehen von Mac OS X anzupassen, werden diese nun halb transparent und an den Ecken abgerundet dargestellt. Eine neu hinzugefügte „Einblendung“ war die zuschaltbare Zeitanzeige, welche die abgelaufene und die Restspielzeit des Films angibt. Neu war auch die Funktion zur Insertierung eines TimeCodes in das laufende Filmbild.

Erst im Frühjahr 2005 erschien die Version 0.2.1b6, in welcher nach einer längeren Programmierpause lediglich einige in der Zwischenzeit gefundene Fehler korrigiert wurden.

Im April ergab sich ein enger Mailkontakt mit einem argentinischen Sounddesigner, der mit vielen Vorschlägen und weiteren Fehlermeldungen zu einer neuen Version beitrug. Es wurde der TimeCode Inserter um die Möglichkeit der Feet und Frame Einblendung ergänzt und die Funktion zum Beschneiden des Filmbilds eingeführt.

Bis Anfang Mai 2005 ist QT Sync ca. 7.500 Mal über die Downloadseiten macupdate.com und versiontracker.com heruntergeladen worden. Hinzu kommen die nicht gezählten Abrufe, welche direkt von der Homepage sowie von anderen Downloadseiten (wie z.B. videohelp.com) aus durchgeführt wurden. Es handelt sich wohl um eine geschätzte Gesamtzahl von ca. 9.000 Downloads.

War die Zielsetzung bei der Programmierung zunächst nur gewesen, einen funktionierenden Ersatz für den instabilen QT Mutator herzustellen, so ergab sich während der fortschreitenden Entwicklung durch Wünsche und Anregungen von Benutzern (sowie Experimentierfreude bei der Programmierung) ein großer Funktionskatalog, der inzwischen nicht nur die Möglichkeiten des QT Mutators beinhaltet, sondern auch Teile des Funktionsumfangs von QuickTime Player und Synchole.

Dieser gewachsene Katalog soll im Folgenden vorgestellt werden.

3.7 *Funktionsumfang*

Die Grundansätze:

- Ein Programm, das dem Benutzer die Möglichkeit gibt, während des Betrachtens eines Films Asynchronitäten zwischen Bild- und Tonspur zu korrigieren und diese Änderungen entweder als neue Filmdatei oder als einen von der Originaldatei abhängigen Film zu speichern.
- Rudimentäre Bearbeitungsmöglichkeiten der Filmstruktur und der enthaltenen Spuren. Hierbei soll keine Fokussierung auf Filmschnitt oder andere tiefer gehende Manipulationen stattfinden, sondern auf die qualitative Verbesserung des Abspielergebnisses.
- Funktionen, die zur Bearbeitung des angelieferten Bildmaterials innerhalb der Audiopostproduktion nützlich sein könnten.

Die gewünschten Funktionen:³⁹

- Änderung des Startzeitpunkts einzelner Spuren relativ zum Beginn des Films (Offset).
- Änderung der Länge einzelner Spuren. Dies hat eine Veränderung der Abspielgeschwindigkeit zur Folge.
- Darstellungsfähigkeit aller QuickTime-kompatiblen Filmformate (Im Besonderen: .mov, .avi, .mpg und .dv).
- Drag & Drop – Fähigkeit.
(Möglichkeit, einen Film durch das Doppelklicken bzw. Ziehen einer Filmdatei auf das Programmsymbol zu öffnen.)
- Unterstützung von Filmen mit mehr als zwei Spuren und Filmen ohne Bildspur.
- Ein zuschaltbares Informationsfenster mit allen relevanten Informationen über den Film und seine Spuren. (Länge, Bildrate, Größe, Spurlängen, Spuroffsets, interne QuickTime Zeitskaleneinheiten, usw.)

³⁹ Alle *kursiv gehaltenen Forderungen* sind in der aktuellen Version des Programms noch nicht vorhanden.

- Darstellung des Films in verschiedenen Größen (halbe, normale, doppelte Größe).
- Darstellung des Films in bildschirmangepasster Größe.
(In QT Sync wird dieser Modus „Fill Screen“ genannt. Die Darstellungsgröße des Films passt sich dabei der Größe des verwendeten Bildschirms an und lässt gleichzeitig das komfortable Arbeiten mit dem Dock von Mac OS X zu.)
- Unterstützung der üblichen Tastaturkommandos für das Abspielen eines Films.
- Ergänzung dieser Tastaturkommandos um einen Kurzbefehl für das schnelle Vor- und Rückspulen (Bildsuchlauf).
- Eine automatische Lautstärkeanpassung beim Abspielen von Filmen des MPEG1-VideoCD-Standards.
(QuickTime spielt diese Art von Filmdatei zu laut ab und produziert dadurch Verzerrungen.)
- Ein möglichst kompaktes, nichtsdestotrotz informatives und praktikables Fenster zur Eingabe der Werte zur Offset- und Längenkorrektur einzelner Spuren.
- Zusätzlich zu der Eingabe von numerischen Werten, die Möglichkeit, die Spuren in Schritten verschieben zu können.
(Entsprechend den bekannten „Nudge“-Tasten, die von vielen Audioschnittsystemen verwendet werden.)
- Eingabe der Werte für Offset und Länge sowohl in den Einheiten Bilder und Millisekunden als auch in den QuickTime-internen Zeitskaleneinheiten (TimeScales).
- Eine komfortable Umschaltung für die Anzeige der verschiedenen Einheiten.
- Option zur Bearbeitung mehrerer Spuren gleichzeitig.

- *Die Möglichkeit, nur Teile von Spuren bearbeiten zu können.*
(Es sollen dazu Synchronpunkte benutzt werden, die ähnlich einer EDL⁴⁰ arbeiten und angezeigt werden.)
- Darstellung des Films im so genannten „Full Screen Mode“.
(Hierbei werden sämtliche Menüleisten und Navigationsleisten ausgeblendet und der Film in größtmöglicher Darstellung gezeigt.)
- Tastaturkommandos für wichtige Kommandos im „Full Screen“ Modus, um auch ohne Maus und zusätzliche Fenster Korrekturen an der Synchronität des Films vornehmen zu können.
- Zuschaltbare informative Einblendungen im Full Screen Modus, um Änderungen an der Synchronität nachverfolgen zu können.
- Zuschaltbare Zeitanzeige im Full Screen Modus.
- Abspeicherbare Veränderung der Größe und des Seitenverhältnisses des Bildes.
- Abspeicherbare Beschneidung („Cropping“) des Bildes.
- Einblendungen von TimeCode oder „Feet + Frames“ Anzeigen in den Film. Diese sind nicht in das Bild „eingebrennt“ (also hineingerechnet), sondern als Textspur, ähnlich zu Untertiteln, vorhanden.
- Die Möglichkeit, externe Spuren in den aktuellen Film einzufügen.
- Option zum Verbinden mehrerer einzelner Filmdateien zu einer langen Filmdatei.
- *Das An- bzw. Abschalten der Darstellung einzelner Spuren.*
- Das Entfernen einzelner Spuren.
- *Die Bearbeitung von Start- und Endpunkten der einzelnen Spuren oder des ganzen Films.*

⁴⁰ EDL = Edit Decision List (engl.) – Schnittliste.

In einer Schnittliste stehen detaillierte Informationen darüber, welche Teile des Quellmaterials in welcher Reihenfolge für ein Endprodukt (z.B. Film) verwendet werden sollen.

- Das Abspeichern der Änderungen als abhängigen Film.
(Diese Datei beinhaltet nicht die tatsächlichen Mediendaten, sondern nur die Steuerdaten; also die Movie Struktur. Die originale Datei bleibt dabei unverändert und wird zum Abspielen immer benötigt.)
- Das Speichern des bearbeiteten Films in eine neue eigenständige Filmdatei.
(In diesem Fall werden auch die Mediendaten in die neue Datei abgespeichert. Dabei handelt es sich jedoch nicht um einen neuen Kodiervorgang, d.h. die Mediendateien werden nur von einem Container in den anderen kopiert.)
- *Den Export des bearbeiteten Films in ein neues Format. („Rendering“)*
(Hierbei werden die Daten neu enkodiert. Alle Änderungen und Einblendungen werden permanent in das entsprechende Bild eingebrannt. Auch nützlich für evtl. notwendige Formatkonvertierungen.)

4. Das Programm „QT Sync“

4.1 Referenz

Dieser Abschnitt ist eine ausführliche Beschreibung aller Funktionen in QT Sync. Er soll als Handbuch und Referenz bei der Benutzung des Programms dienen.

Zusätzlich wird ein detaillierter Einblick in die Funktionsweise von QT Sync gewährt, ohne dabei den tatsächlichen Quellcode zu analysieren.⁴¹

Die Benutzeroberfläche von QT Sync ist nur in Englisch verfügbar. Mac OS X bietet Programmierern komfortable Möglichkeiten, ihre Programme in mehreren Sprachfassungen anzubieten, allerdings unterstützt REALbasic diese Funktionen nicht. Es wäre notwendig gewesen, auf relativ kompliziertem Wege verschiedene Versionen des Programms zu erstellen.

Der Autor geht allerdings davon aus, dass die angesprochene Zielgruppe soweit mit der Materie vertraut ist, um auch mit einer englischen Benutzerführung umgehen zu können. Zumal sind Begriffe wie „Offset“, „Duration“ oder „Track“ auch schon aus anderen Programmen bekannt und gehören zum allgemein üblichen technischen Fachvokabular.

4.1.1 Programmstart

Beim Start von QT Sync wird zunächst überprüft, ob QuickTime auf dem Computer installiert ist.⁴² QT Sync wurde unter den QuickTime Versionen 6.5.1, 6.5.2 und 7 entwickelt und getestet. Bei Verwendung anderer Versionen kann keine Aussage über die Funktionalität von QT Sync getroffen werden.

Als nächstes wird die installierte Systemsoftware abgefragt. QT Sync wurde unter Mac OS X 10.3.3 bis 10.4.1 entwickelt und getestet. Auch hier kann der Autor bei Benutzung anderer Systemversionen keine Garantie für die Funktionalität geben.

⁴¹ Ausgewählte Beispiele aus dem Quellcode werden in einem späteren Abschnitt der Arbeit ausführlich diskutiert.

⁴² Zur Erinnerung: Dies sollte eigentlich bei jedem Computer mit Mac OS X der Fall sein, da QuickTime zu den Grundbestandteilen einer Systeminstallation gehört.

Sollte QuickTime nicht installiert sein oder das Programm auf einem Computer mit einer Systemsoftware laufen, die älter als Mac OS X 10.3 ist, wird QT Sync mit einem entsprechenden Hinweis beendet.

Nach einer erfolgreichen Abfrage der beschriebenen Systemparameter beginnt das Programm interne Speicherbereiche anzulegen, in denen später sowohl der Film als auch die anzuzeigenden Daten abgelegt werden.

Der nächste Schritt ist die Suche und Analyse der Voreinstellungsdatei, welche im privaten Ordner des Benutzers unter Library/Preferences/QTSync.plist abgespeichert ist. Sie besteht aus einem Text, dessen einzelne Felder durch einen Asterisk getrennt sind. Im ersten Feld steht die Versionsnummer der Voreinstellungsdatei, im zweiten Feld ein Warnhinweis für denjenigen, der mit Hilfe eines Textbearbeitungsprogramms diese Datei verändern möchte. Danach folgen die eigentlichen Voreinstellungen, welche meist durch Zahlenwerte beschrieben werden.

Es stehen 300 Felder zur Verfügung, wovon zurzeit nur knapp zwanzig in Benutzung sind.

Sollte die Voreinstellungsdatei fehlen oder eine ältere Versionsnummer tragen, so werden neue Voreinstellungen erstellt. Im Falle des Überschreibens einer älteren Version der Voreinstellungsdatei wird ein Informationsfenster angezeigt.

4.1.2 Das Laden eines Films

Der Benutzer hat mehrere Möglichkeiten, einen Film in das Programm zu laden. Entweder er benutzt das Apple-übliche Menükommando „File→Open“ bzw. das dazugehörige Tastaturkommando Apfel+O oder er zieht die zu ladende Filmdatei auf das Programm-Icon („Drag & Drop“) im Finder bzw. im Dock.

Durch den Mac OS X – Befehl „Alle Dateien dieser Art mit folgendem Programm öffnen...“ ist es auch möglich, QT Sync als Standardprogramm für Filmdateien auszuwählen, so dass die Datei durch einen simplen Doppelklick mit QT Sync geöffnet werden kann.

In der Programmiersprache REALbasic werden zwei verschiedene Typen von Filmdateien definiert: Einmal das „Movie“ und das „EditableMovie“ (bearbeitbarer Film“). Der Unterschied liegt darin, dass mit dem Typ EditableMovie wesentlich mehr Bearbeitungsfunktionen möglich sind als mit dem einfachen Movie. Der Nachteil ist, dass die Datei nicht schreibgeschützt oder in Benutzung durch ein

anderes Programm sein darf. (Auch wenn QT Sync selbst keinerlei Veränderung an dieser Datei vornimmt.)

Es wird beim Öffnen des Films also zunächst überprüft, ob dieser als ein EditableMovie geöffnet werden kann. Ist dies nicht der Fall, wird mit Hilfe einer anderen Methode eine Filmdatei erstellt, die im (unsichtbaren) temporären Ordner des Mac OS gespeichert wird. Diese ist ein so genannter „abhängiger Film“ und beinhaltet ausschließlich die Steuerdaten des ursprünglich zu öffnenden Films sowie den Verweis, wo die tatsächlichen Mediendaten (also Bild und/oder Ton) zu finden sind. Der abhängige Film wird dann geöffnet. Das Erstellen dieser temporären Datei läuft für den Benutzer völlig unsichtbar ab und hat für die spätere Bearbeitung keinerlei Relevanz.

Nach dem erfolgreichen Ladevorgang wird der Film analysiert. Es werden derzeit nur Filme mit maximal fünf Spuren unterstützt, wobei nur eine davon eine Bildspur sein darf. (Sollte der Film diese Anforderung nicht erfüllen, wird der Ladevorgang mit einer entsprechenden Fehlermeldung abgebrochen.)

Anschließend erfasst QT Sync automatisch alle relevanten Filmdateien. Dazu gehören Name, Länge, Bildrate, Größe, Speicherort (sowohl der Original-Datei als auch einer evtl. temporären Datei) und TimeScales⁴³. Auch die Eigenschaften der einzelnen Spuren werden gesammelt, u.a. Offset, Länge, Art und TimeScales der Spuren⁴⁴.

Der Film wird dann innerhalb von QT Sync dupliziert und zwei getrennten Speicherorten zugewiesen. Der erste Film ist derjenige, mit dem tatsächlich gearbeitet wird. Der zweite dient als Vorlage für spätere Bearbeitungsvorgänge und bleibt immer unverändert.

Schließlich wird das Film- und das „Controller“-Fenster geöffnet, und der Benutzer kann mit der Bearbeitung beginnen.

Es bleibt noch zu erwähnen, dass nur jeweils ein Film zur selben Zeit geöffnet sein kann. Sollte versucht werden, einen weiteren Film zu laden, wird durch ein Informationsfenster auf diese Tatsache hingewiesen.

⁴³ TimeScale: Zeitskaleneinheiten. Vgl. Kapitel 2

⁴⁴ Vgl. Kapitel 2

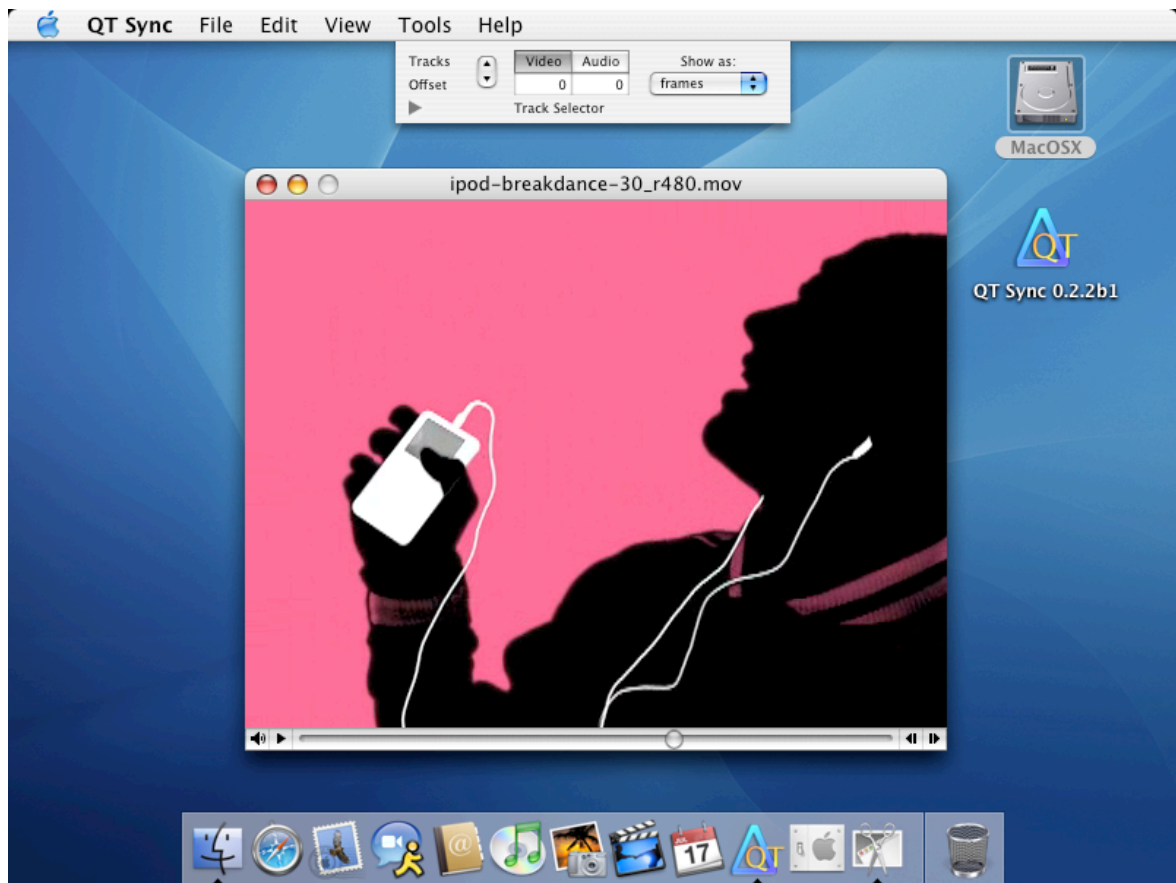


Abbildung 4.1: Die Bedienoberfläche von „QT Sync“ nach dem Laden eines Films

4.1.3 Das Filmfenster

Die Erscheinungsform des Filmfensters wurde unverändert vom QT Mutator übernommen. Das Bedienelement am unteren Rand des Fensters ist allerdings keine Erfindung von QT Mutator, sondern wird schon vom Betriebssystem (bzw. der QuickTime Architektur) vorgegeben.

Auf der linken Seite befindet sich der Lautstärkeregler (durch Tastaturkommando ↓ (Pfeil nach unten) und ↑ (Pfeil nach oben) fernbedienbar). Standardmäßig ist die Lautstärke auf 100% eingestellt, mit Ausnahme bei Filmen des VideoCD-MPEG1-Standards: Beim Öffnen von Filmen dieser Art wird die Lautstärke auf 50% reduziert.⁴⁵

Die Lautstärke kann generell bis auf 200% (entspricht +6 dB) erhöht werden. Veränderungen der Lautstärke werden allerdings nicht abgespeichert.

⁴⁵ QuickTime spielt diese Filme zu laut ab, was zu einer Übersteuerung des Computer-Ausgangs führt. Durch die Reduzierung der Abspiellautstärke wird dieses Problem behoben und die Lautstärke auf ein normales Maß begrenzt.

Daneben befindet sich der Knopf zum Abspielen bzw. Anhalten des Films (Tastaturkommando: Leertaste). Auch durch Klicken in den Film kann dieser angehalten (einfacher Klick) oder gestartet werden (Doppelklick).

Auf der angrenzenden Zeitleiste wird die aktuelle Position des Films angezeigt. Durch ein Verschieben des Knopfes kann der Abspielpunkt geändert werden.

An der rechten Seite befinden sich zwei Bedienflächen zum schnellen Vor- und Rücklauf. Es wird allerdings von einer Benutzung abgeraten, da der Gebrauch zu einer Störung führt, welche sich durch leichte Knackser (selbst nach Beendigung des Spulvorgangs) äußert. Durch einen Neustart von QT Sync kann diese Beeinträchtigung jedoch wieder beseitigt werden. (Es handelt sich hierbei nicht um einen Fehler von QT Sync. Das beschriebene Verhalten zeigt sich ebenfalls beim Spulen im QuickTime Player.)

Zusammenfassung aller für das Filmfenster relevanten Tastaturkommandos:

- Leertaste: Abspielen bzw. Anhalten des Films (Play / Pause)
- ↑ / ↓ : Regulierung der Gesamtlautstärke.
- ← / → : Einzelbildfortschaltung
- SHIFT + ← / SHIFT + → : Vorwärts bzw. Rückwärts abspielen (in 1-facher Geschwindigkeit)
- ALT + ← / ALT + → : Sprung zum Beginn bzw. Ende des Films

Diese Tastaturkommandos entsprechen dem Verhalten des QuickTime Player.

Das zusätzliche Kommando

- CTRL + ← / CTRL + → : Bildsuchlauf

erleichtert das Bewegen durch den Film. Drücken und Halten dieser Tastenkombination bewirkt das schnelle Vor- bzw. Rückspulen. (Die Geschwindigkeit ist in den Voreinstellungen veränderbar.) Loslassen der Tastenkombination führt zu einer Rückkehr zum vorherigen Abspielmodus.

Um die oben genannte Störung durch Knackser zu vermeiden, wird nicht tatsächlich die Abspielgeschwindigkeit erhöht, sondern (während der Film in einfacher Geschwindigkeit weiterläuft) durch den Film „gesprungen“. Je nach Geschwindigkeit des verwendeten Computers kann es dabei dazu kommen, dass das Loslassen der Tasten nicht sofort zu einem Stoppen des Suchlaufs führt.

4.1.4 Die verschiedenen Betrachtungsmodi

Zum Betrachten des Films bietet QT Sync sechs verschiedene Modi. Zwei davon nutzen den so genannten „Full Screen“ Modus, bei dem alle Mac OS spezifischen Bedienelemente wie Fensterrahmen, Menüleisten und das Dock ausgeblendet sind und der Film den gesamten Bildschirm einnimmt. Eventuell nicht genutzte Bildschirmteile sind geschwärzt, so dass insgesamt der Eindruck eines Fernsehers entsteht. Diese beiden Modi sollen in einem separaten Kapitel, zusammen mit allen anderen Funktionen des „Full Screen“ Modus von QT Sync genauer vorgestellt werden.

Drei weitere Betrachtungsmodi sind jene aus QT Mutator und QuickTime Player angelehnt.

Durch das Menü „View“ sind die einzelnen Modi abrufbar. Der Benutzer findet dort die drei Menüpunkte „Half“ (Tastaturkommando: Apfel+0 (Null)), „Normal“ (Apfel+1) und „Double“ (Apfel+2). Die Auswahl eines dieser Menüpunkte führt zu der Darstellung des Films in halber, Original-, bzw. doppelter Größe.

Das Filmfenster wird dabei jeweils der Abbildungsgröße des Films angepasst und auf dem Bildschirm zentriert dargestellt. Sollte das Fenster für den Bildschirm zu groß sein (zum Beispiel bei Benutzung des „Double“ Modus), wird die linke obere Ecke des Fensters in die linke obere Ecke des Bildschirms gelegt, um nach wie vor Zugriff auf die Bedienknöpfe und die Titelleiste des Fensters zu ermöglichen. (Bei einer Zentrierung wäre beides außerhalb des Bildschirmrandes und damit für den Benutzer unerreichbar.)

Die drei beschriebenen Modi arbeiten unabhängig von der benutzten Bildschirmauflösung (sprich: Bildschirmgröße). Deshalb enthält QT Sync einen vierten Modus namens „Fill Screen“ (nicht zu verwechseln mit „Full Screen“). In diesem wird der Film auf die maximal mögliche, vollständig auf dem Bildschirm darstellbare Größe skaliert.

Beim Aufrufen des „Fill Screen“ Modus wird zunächst die vorhandene Bildschirmauflösung analysiert. Die Höhe des verfügbaren Platzes richtet sich zusätzlich danach, ob das Dock von Mac OS X während des Aufrufens des „Fill Screen“ Modus eingeblendet ist. Sollte es vorhanden sein, wird die Höhe des Docks von der verfügbaren Bildschirmhöhe abgezogen.

Da am unteren Rand des Filmfensters die Bedienelemente des Films (Zeitleiste, etc.) untergebracht sind und diese mit der Maus erreichbar bleiben sollen, wird außerdem ein gewisser Sicherheitsabstand zum Dock oder zum unteren Bildschirmrand eingerechnet. Dies soll vermeiden, dass bei einem leichten „Abrutschen“ der Maus das Dock aufgezoomt bzw. eingeblendet wird.

Es sei darauf hingewiesen, dass alle diese Betrachtungsmodi nur die Darstellung des Films innerhalb von QT Sync beeinflussen, nicht aber seine tatsächlichen Proportionen. Sollen diese geändert werden, bietet sich hierfür die Funktion „Change Ratio...“ an, welche im Kapitel 4.1.9 beschrieben wird.

4.1.5 Das Informationsfenster („Info Window“)



Abbildung 4.2: Informationsfenster – Spalte „General“

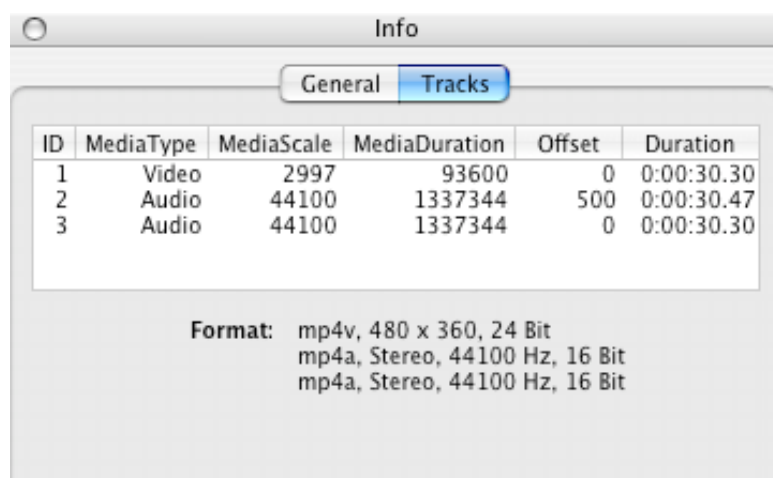


Abbildung 4.3: Informationsfenster – Spalte „Tracks“

Über das Menü „File → Info“ (Tastaturkommando Apfel+i) lässt sich das Informationsfenster öffnen. Dort werden ausgewählte Angaben über einzelne Parameter des Films angezeigt, welche während des Öffnens der Datei gesammelt wurden. Durch die Schaltflächen „General“ und „Tracks“ kann zwischen zwei Seiten umgeschaltet werden.

Auf der Seite „General“ finden sich allgemeine Informationen über den Film: Dateiname, Länge, Größe, Bildrate und TimeScales.

Die Seite „Tracks“ bietet mehr Informationen zu den einzelnen Spuren des Films. In einer Tabelle werden TrackID, Art der Spur, MediaScales (also die TimeScales der jeweiligen Spur), Offset und Länge der Spur angezeigt. Unterhalb der Tabelle befinden sich Informationen über die verwendeten Codecs der Spuren.

Das Errechnen der Bildrate gestaltete sich etwas komplexer als zunächst vermutet. Durch die Eigenschaften verschiedener Formate mussten zwei unterschiedliche Methoden der Berechnung angewendet werden⁴⁶: Für MPEG1- und MPEG2-Filmdateien gibt es die Möglichkeit, durch einen speziellen Befehl die in den Datenstrom hineinkodierte Bildrate abzufragen. Bei allen anderen Formaten wird einfach nur die Anzahl der Bilder des gesamten Films gezählt und durch die Länge in Sekunden geteilt.

Auch das Abfragen des verwendeten Codec war nicht ohne weiteres möglich. Obwohl diese Funktion schon in den allerersten Versionen von QT Sync angelegt war, konnte sie erst durch die Verwendung eines Plug-Ins⁴⁷ korrekte Ergebnisse liefern. Ab Version 0.2.1b1 ist die Anzeige der entsprechenden Werte frei geschaltet.

⁴⁶ Diese Vorgehensweise wird von Apple in einem Programmierbeispiel empfohlen:

<http://developer.apple.com/qa/qa2001/qa1262.html>

Das Beispiel ist in der Sprache Carbon verfasst und wurde vom Autor in entsprechende REALbasic Befehle „übersetzt“. Vgl. Kap. 5.1.3: „Das Berechnen der Bildrate für verschiedene Filmformate“

⁴⁷ Zur Notwendigkeit von Plug-Ins vgl. Kapitel 5

4.1.6 Das „Controller Window“



Abbildung 4.4: Der „QT Sync“ Controller

Das Controller Fenster ist das Herzstück von QT Sync und dient der Eingabe und Anzeige aller relevanten Informationen für die Bearbeitung des geöffneten Films.

Um eine effiziente Benutzung sicher zu stellen, wird die Bedienung dieses Fensters stark eingeschränkt. Unter anderem ist nicht gestattet, die Größe oder die Position des Fensters auf dem Bildschirm zu verändern. Der Grund für diese starke Reglementierung liegt in dem Wunsch nach einer möglichst kompakten und dennoch hocheffizienten Anzeige- und Eingabemöglichkeit begründet.⁴⁸

In der Mitte des Fensters befindet sich der „Track Selector“, also das Element zur Anwahl und Bearbeitung der einzelnen Spuren. Jeder Spur wird eine Spalte zugeordnet.

Die oberste Zeile dient einerseits zur Identifizierung der Spuren und ist, je nach Art der Spur, eindeutig beschriftet (z.B. „Video“, „Audio“, „text“ oder „MPEG“). Sollten mehrere Spuren derselben Art vorhanden sein, wird dem Namen eine fortlaufende Nummer hinzugefügt.

Zum anderen ist die Zeile eine Schaltfläche, welche zur Scharfschaltung einzelner oder mehrerer Spuren dient. (Die Scharfschaltung bezieht sich auf die Verwendung der Pfeilschaltflächen. (s.u.))

Darunter befinden sich Anzeige- und Eingabefelder für die Offsets der einzelnen Spuren (relativ zum Filmanfang). Hier können Werte eingegeben werden, welche nach dem Drücken der Return- oder Tabulator-Taste eine entsprechende Änderung des Films bewirken. Es werden hier nur positive Werte angenommen.

⁴⁸ Ausgehend von Erfahrungen mit dem Programm QT Mutator sah der Autor die unbedingte Notwendigkeit gegeben, ein neues Konzept für dieses Fenster zu entwickeln und umzusetzen. Dass hierdurch bewusst eine Apple-untypische Arbeitsweise gefördert und die Richtlinien zur Programmierung einer Bedienoberfläche verletzt werden, wird in diesem Fall hingenommen.

Auf der rechten Seite befindet sich eine Auswahlbox, welche die Anzeige der Werte beeinflusst. Es kann zwischen drei Einheiten ausgewählt werden. Diese sind „frames“ (Bilder), „msec“ (Millisekunden) und „TimeScales“ (die interne Einheit von QuickTime). Bei Benutzung der Auswahlbox werden die anzuzeigenden Werte entsprechend umgerechnet.

(Anmerkung: Intern arbeitet QT Sync nur mit TimeScales. Die Auswahlbox beeinflusst nur die Anzeige. Allerdings können durch die Umstellung der Einheit Rundungsfehler bis zu einem halben Bild entstehen. Daher wird der Film nach einer Umstellung noch einmal mit den aktuellen Werten bearbeitet.)

Auf der linken Seite befindet sich eine Schaltfläche mit zwei Pfeilen. Diese bewirkt eine Änderung des Offsets der jeweils scharf geschalteten Spuren. Das Besondere an dieser Methode ist, dass auch mehrere Spuren gleichzeitig verändert werden können, wenn diese scharf geschaltet sind.

Der Betrag, welcher jeweils zu einem Offset hinzuaddiert bzw. abgezogen wird, kann durch ein Kontextmenü beeinflusst werden. Der Benutzer klickt dazu mit gedrückter gehaltenen CTRL-Taste auf die Pfeilschaltfläche und wählt aus dem erscheinenden Menü die gewünschte Schrittgröße aus.

4.1.7 Erläuterungen zur Behandlung von Offsets innerhalb von QT Sync

In diesem Abschnitt werden zwei Mechanismen erläutert, die QT Sync bei Benutzung der Pfeile zur Änderung des Offsets anwendet.

Der Startzeitpunkt einer Spur innerhalb des QuickTime Films wird durch den Offset bestimmt. Es ist offensichtlich, dass ein Offset nicht negativ sein kann, da sich der Anfang der betreffenden Spur dann vor dem Beginn des eigentlichen Films befinden würde. Deswegen sind bei der numerischen Eingabe des Offsets in die Eingabefelder des Controller Fensters nur positive Werte erlaubt.⁴⁹

⁴⁹ Diese Aussage steht im scheinbaren Widerspruch zu der Möglichkeit, im Programm QT Mutator einen negativen Offset eingeben zu können. Der QT Mutator allerdings schneidet bei der Eingabe eines negativen Offsets die entsprechende Länge am Beginn der entsprechenden Spur ab. Der Offset bleibt daher trotzdem positiv bzw. Null.

Um aber trotzdem einen scheinbar negativen Offset zu erzeugen, wendet QT Sync eine besondere Verfahrensweise an: Bei der schrittweisen Verringerung des Offsets durch die Benutzung der Pfeilschaltfläche wird überprüft, ob einer der Werte negativ werden würde. In diesem Fall wird nicht mehr der Offset der „scharfen“ Spur(en) verringert, sondern derjenige aller anderen erhöht. Auf diese Weise bleiben alle Offsets positiv oder Null. Relativ gesehen jedoch wurde die betroffene Spur um den gewünschten Betrag verschoben.

Dieser Mechanismus soll anhand eines Beispiels verdeutlicht werden:

Annahme: Es ist ein Film mit drei Spuren geöffnet: Video, Audio1 und Audio2. Der Offset der beiden Audiospuren ist 3 Frames, der Offset der Videospur 0 Frames.

Video	Audio1	Audio2
0	3	3

Es sei weiterhin angenommen, der Benutzer möchte den Beginn der Videospur um 2 Frames nach vorne verschieben, so dass die Videospur 2 Frames früher beginnt. Der Benutzer wird nun die Videospur scharf schalten und durch die Pfeilschaltfläche die Videospur um 2 Felder nach vorne bewegen. Das erzeugte Resultat sähe theoretisch aus wie folgt:

Video	Audio1	Audio2
-2	3	3

Wie oben erläutert ist ein negativer Offset nicht möglich. Das tatsächliche Resultat wird so angezeigt:

Video	Audio1	Audio2
0	5	5

Es ist unschwer zu erkennen, dass der Offset der Videospur nach wie vor den Wert von 0 Frames besitzt, dagegen aber die beiden anderen Spuren in ihrem Wert verändert wurden. Dadurch wird sowohl dem Wunsch des Benutzers

entsprochen, die Videospur 2 Frames früher beginnen zu lassen, als auch die Forderung erfüllt, dass ein Offset nicht negativ sein darf.

Ein weiterer intelligenter Mechanismus von QT Sync ist das automatische Heranziehen aller Spuren an den Nullpunkt. Es ist rein theoretisch möglich, dass alle enthaltenen Spuren eines Films einen Offset größer als Null besitzen. Es ist allerdings kein praktischer Nutzen darin zu erkennen, eine Filmdatei mit einem „Loch“ beginnen zu lassen.⁵⁰ Daher wird nach jeder Benutzung der Pfeilschaltfläche überprüft, ob mindestens noch ein Offset bei Null ist. Sollte dies nicht der Fall sein, wird vom Offset aller Spuren der Offsetbetrag der Spur abgezogen, welche dem Nullpunkt des Films am nächsten ist.

Die Fortführung des obigen Beispiels soll hier zum besseren Verständnis dieser Funktion beitragen:

Angenommen, der Benutzer ist mit seiner oben gemachten Änderung unzufrieden und möchte gerne den Ausgangszustand wieder herstellen.

Zur Erinnerung der jetzige Zustand:

Video	Audio1	Audio2
0	5	5

Nach wie vor ist nur die Videospur scharf geschaltet und der Benutzer erhöht den Offset der Videospur mit Hilfe der Pfeilschaltfläche um 2 Frames. Das theoretische Resultat wäre:

Video	Audio1	Audio2
2	5	5

⁵⁰ Der Mechanismus des „Heranziehens“ wird nur bei Benutzung der Pfeile ausgelöst. Sollte tatsächlich ein „Loch“ zu Beginn des Films gewünscht werden, so wird empfohlen, nicht die Pfeilschaltflächen, sondern die Möglichkeit der manuellen Eingabe in die Eingabefelder zu nutzen.

Der automatische Mechanismus von QT Sync allerdings erkennt nun, dass keine der drei Spuren mehr am Nullpunkt des Films beginnt und zieht alle drei Spuren auf den Nullpunkt zurück. Der relative Offset der Spuren untereinander bleibt dabei erhalten. Das Resultat sieht so aus:

Video	Audio1	Audio2
0	3	3

4.1.8 Weitere Funktionen des „Controller Window“

Auf der linken Seite des Fensters befindet sich in der unteren Ecke ein kleines Dreieck. Der Benutzer kann hiermit die Eingabefelder zur Längenänderung einer Spur auf- und zuklappen. Wird das Programm zum ersten Mal geöffnet, sind diese geschlossen.

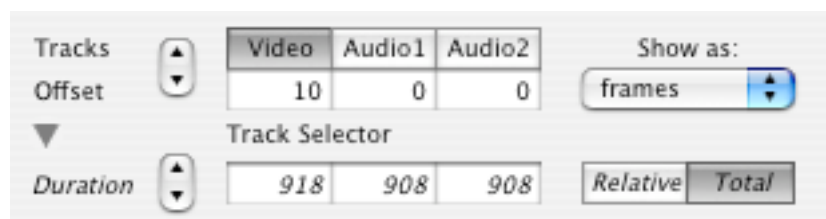


Abbildung 4.5: Der aufgeklappte „QT Sync“ Controller

Wird das Fenster aufgeklappt, erscheinen weitere Bedienelemente. In der Mitte befinden sich die Anzeige- und Eingabefelder zur Längenänderung der einzelnen Spuren. Die Anzeige ist (analog zu den Anzeigefeldern des Spuroffsets) von der Wahl der entsprechenden Einheit (frames, msec oder TimeScale) abhängig.

Auf der rechten Seite befinden sich zwei weitere Schaltflächen: „Relative“ und „Total“.

Wenn letztere ausgewählt ist, wird die komplette Länge der Spur angezeigt. Dabei kann (je nach Länge des Films und ausgewählter Anzeige-Einheit) die Zahl so groß sein, dass sie nicht mehr vollständig in dem Anzeige-/Eingabefeld dargestellt werden kann.

Bei Auswahl von „Relative“ wird dagegen nur angezeigt, welche Änderungen an der Spurlänge seit Öffnen des Films vorgenommen wurden. Die Errechnung des relativen Wertes erklärt sich wie folgt:

Beim Öffnen eines Films fragt QT Sync u.a. die Längen der Spuren ab und speichert diese als Referenzwert. Bei der relativen Darstellung der Länge wird von der aktuellen Länge der Spur dieser Referenzwert abgezogen.

QuickTime versteht unter der Länge einer Spur den Zeitraum von Beginn des Films bis zum Ende der Spur. Dies bedeutet, dass auch ein evtl. hinzugefügter Offset zu der Länge einer Spur beiträgt. Um Verwirrungen an dieser Stelle vorzubeugen, beschloss der Autor, die relative Länge unabhängig vom verwendeten Offset anzuzeigen. Daher wird sowohl bei der Errechnung des Referenzwertes als auch bei der Errechnung des relativen Anzeigewertes ein eventueller Offset nicht berücksichtigt.

Die absolute Länge dagegen wird inklusive des vorhandenen Offsets dargestellt. Sie ist also genauso definiert wie in QuickTime.

Dem Autor ist bewusst, dass er hier inkonsequent handelt. Er geht aber davon aus, dass diese Inkonsequenz weniger Verwirrung erzeugen wird, als eine Miteinberechnung des Offsets in die relative Längendarstellung. Die praktische Konsequenz dieser Darstellung wäre nämlich, dass bei einer Offsetänderung auch bei der relativen Längenanzeige Änderungen entstehen (nämlich um den Wert der Offsetänderung). Der Autor geht weiterhin davon aus, dass ein Benutzer den Begriff „Spurlänge“ eher als den Zeitraum vom Beginn bis zum Ende der jeweiligen Spur versteht als den Zeitraum vom Beginn des Films bis zum Ende der jeweiligen Spur.⁵¹

Auf der linken unteren Seite des aufgeklappten Controller Fensters befindet sich schließlich eine weitere Pfeilschaltfläche. Diese funktioniert genauso wie die Schaltfläche für den Spuroffset, bezieht sich aber auf die Spurlänge. Auch hier ist die Schrittgröße durch ein Kontextmenü einstellbar. Allerdings gilt die Schrittgröße, welche bei einer der beiden Schaltflächen eingestellt wurde, auch automatisch für die andere.

⁵¹ Dieser Gedanke orientiert sich an dem Prinzip der aus Schnittprogrammen bekannten „Regions“.

Eine weitere Funktion ist die Anpassung an die eingestellte Darstellungsgröße des Films. Abbildung 4.5 (s.o.) zeigt die Ansicht des Fensters in den Abspielmodi „Half“, „Normal“ oder „Double“.

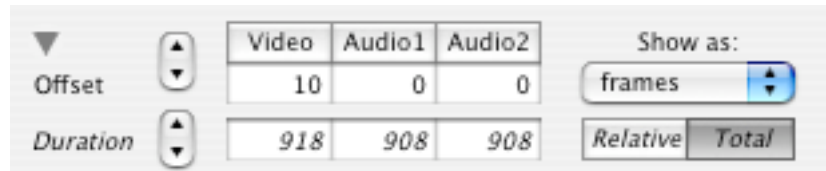


Abbildung 4.6: Der aufgeklappte „QT Sync“ Controller im Fill-Screen-Modus.

In Abbildung 4.6 wird die Darstellung des Fensters im „Fill Screen“ Modus gezeigt. Das Fenster ist kompakter und nimmt nur noch drei Viertel des vorherigen Bildschirmplatzes ein. Erreicht wird dies durch teilweises Entfernen der Beschriftungen. Das Dreieck zum Auf- bzw. Zuklappen des Fensters befindet sich nun in der linken oberen Ecke.

Der Autor geht davon aus, dass der unkundige Benutzer zunächst im voreingestellten „Normal“ Modus arbeitet. In diesem Falle sind die Beschriftungen sichtbar und erleichtern den Einstieg in die Bedienung des Controller Fensters.

Sobald der Benutzer den „Fill Screen“ Modus benutzt, wird er vermutlich soweit mit dem Fenster vertraut sein, dass er auf den wegfallenden Teil der Beschriftung verzichten kann. Sollte dies nicht der Fall sein, vertraut der Autor darauf, dass mit der immer verfügbaren Kontexthilfe⁵² die meisten Bedienungsschwierigkeiten gelöst werden können.

4.1.9 Das Ändern des Seitenverhältnisses (Change Ratio...)

Gerade bei der Arbeit mit DV- oder 16:9 Material (prinzipiell bei jeder Art von Videoformat mit nicht-quadratischen Pixeln) kann es sinnvoll sein, das Seitenverhältnis des Materials zu verändern, um die Stauchung des Bildes auszugleichen. QT Sync bietet dafür eine Funktion, die durch den Menübefehl „Edit→Change Ratio...“ aufgerufen werden kann.

⁵² Lässt der Benutzer den Mauszeiger einige Zeit über einem bestimmten Bedienelement stehen, so erscheint eine Einblendung mit einem speziellen Hilfetext für dieses Bedienelement.

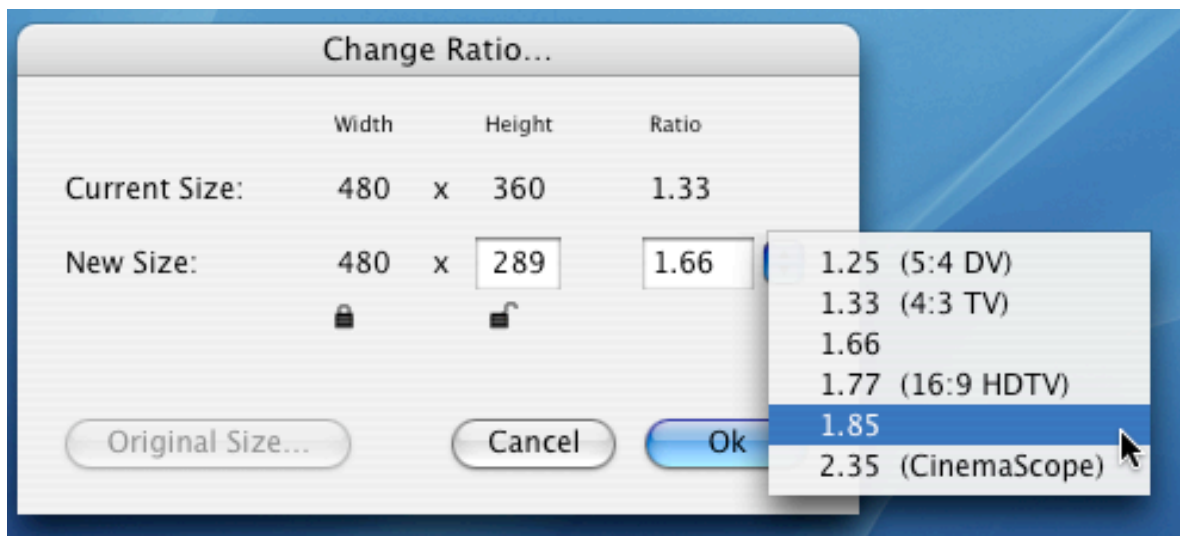


Abbildung 4.7: Das „Change Ratio...“ Fenster

In dem erscheinenden Dialogfenster wird in der oberen Zeile die momentane Höhe und Breite des Films angezeigt. An der rechten Seite befindet sich die Anzeige „Ratio“⁵³ (der Quotient aus Höhe und Breite).

In der unteren Zeile können die gewünschten Werte für Höhe und Breite eingegeben werden. Durch Klicken in eines der Schlösser unterhalb der Eingabefelder kann der korrespondierende Wert fixiert und stattdessen eine neue Ratio direkt eingegeben werden. Neben dem Eingabefeld für die neue Ratio befindet sich auch ein Auswahlménü für die gängigsten Ratiowerte.

Alle eingegebenen Werte müssen mit der Return- oder TAB-Taste bestätigt werden. Erst dann werden die korrespondierenden Werte aktualisiert.

Durch Klick auf „OK“ bestätigt der Benutzer die Änderungen am Film. QT Sync prüft daraufhin, ob die neuen Werte für Breite und Höhe Vielfache von 4 sind und weist gegebenenfalls darauf hin, dass eine Nichteinhaltung dieser Konvention zu Problemen in der weiteren Bearbeitung (z.B. bei einer nachfolgenden Neu-Encodierung durch ein anderes Programm) führen könnte.⁵⁴

⁵³ Ratio (engl.): Seitenverhältnis

⁵⁴ Man beachte, dass hierbei manchmal der Wert 4 nicht ausreichend sein kann. Bei einer Encodierung in das 3ivX Format wird beispielsweise ausdrücklich empfohlen, Vielfache von 8 für die Höhe und Breite des Films zu verwenden; ansonsten drohen gravierende Qualitätseinbußen.

Es wird außerdem ein Höhen- und Breitenwert vorgeschlagen, der durch 4 teilbar ist und trotzdem möglichst nah an der gewünschten Zielgröße liegt. Dieser Vorschlag kann ignoriert werden, sodass der Film in die ursprünglich gewählte Größe skaliert wird.

Die Skalierung, welche QT Sync daraufhin vornimmt, betrifft nur die Darstellung des Films. Die Mediendaten werden nicht verändert. Man beachte, dass durch eine Skalierung die benötigte Prozessorleistung des abspielenden Computers ansteigt.

Bei einem erneuten Aufruf der „Change Ratio...“ Funktion wird in der linken unteren Ecke zusätzlich die Schaltfläche „Original Size“ angezeigt. Beim Klicken auf diese Schaltfläche werden in die Eingabefelder für die neue Filmgröße die Werte eingegeben, welche der Film besaß, als er von QT Sync geöffnet wurde. Dies kann hilfreich sein, um einen Film wieder in seine ursprüngliche Darstellungsgröße zu versetzen, um z.B. die Prozessorlast für das Abspielen zu verringern oder DV-Material wieder in seinen (evtl. anamorphotischen) Ursprungszustand zu versetzen.

Das Seitenverhältnis lässt sich nur ändern, solange keine Textspur im Film enthalten ist, da Textspuren nicht skalierbar sind.

4.1.10 Festlegung des sichtbaren Bildanteils („Cropping“)

QT Sync bietet über „Edit→Crop Movie...“ die Möglichkeit, das Filmbild zu beschneiden. Sinnvoll ist dies vor allem bei Material, welches in ein anderes Seitenverhältnis umgewandelt werden soll, ohne dabei verzerrt zu werden. Bei der Beschneidung gehen stattdessen Bildanteile verloren. Ein Beispiel wäre hierbei die Arbeit mit Super 35mm Material.⁵⁵

⁵⁵ Bei dem Super 35mm Format wird das 35mm Negativ ohne anamorphotische Verzerrung voll belichtet. Für die Cinemascope Kinokopie wird durch einen Kopierprozess nur ein vorher definierter Teil des Bildes verwendet. Der Vorteil liegt darin, dass für eine spätere TV-Fassung das 4:3 Vollbild verwendet werden kann.

Ein anderer Anwendungsfall wäre die Beschneidung um Bildanteile an den Rändern, die außerhalb des auf einem Fernseher sichtbaren Bildbereichs liegen. (wie z.B. bei Material, das sichtbare und evtl. störende VITC-Zeilen enthält)

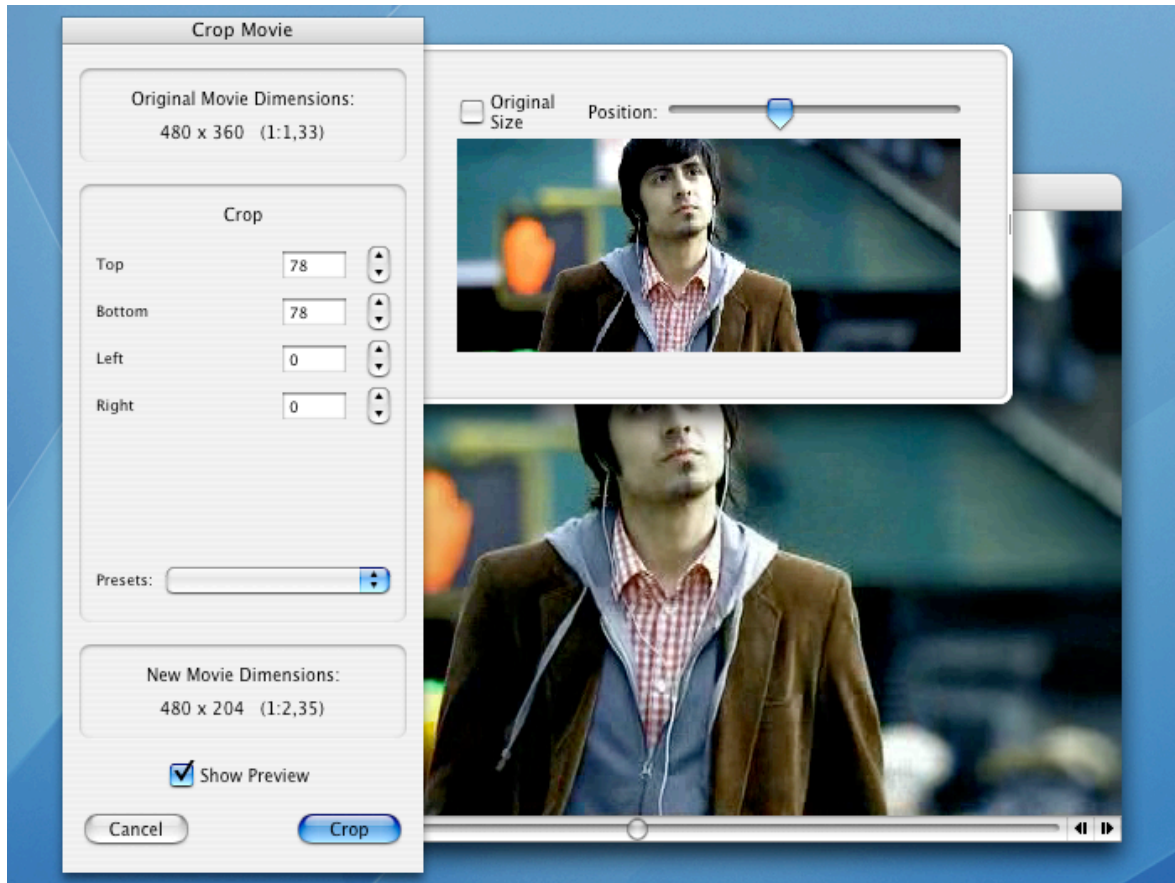


Abbildung 4.8: Das „Crop Movie“ Fenster mit geöffneter Vorschau

Das Fenster, welches sich nach dem Auswählen von „Edit→Crop Movie...“ öffnet, bietet dem Benutzer die Möglichkeit, alle vier Ränder einzeln zu beschneiden. Die Werte sind Pixelwerte und können entweder von Hand eingetragen oder durch Pfeilschaltflächen verändert werden. Diese erhöhen oder verringern die Zahl jeweils um 10 Pixel.

Zusätzlich sind einige Presets anwählbar, welche die üblichen Seitenverhältnisse beinhalten. Eine Besonderheit sind das „Cinemascope Common Center“ und „Cinemascope Common Top“ Preset. Beide sind auf die Arbeit mit Super 35mm Material optimiert. Während „Common Center“ eine zentrierte Beschneidung vornimmt, wählt „Common Top“ einen Bildausschnitt, der etwas oberhalb des Mittelpunktes liegt.

Mit dem Betätigen des „Show Preview“ Knopfes wird ein Vorschau Fenster geöffnet, welches dem Benutzer visuell veranschaulicht, wie seine Einstellungen später aussehen werden.

In dem Fenster der Vorschau befindet sich außerdem eine Auswahlbox mit dem Titel „Original Size“, die bei Aktivierung das Filmbild in seiner Originalgröße anstatt einer verkleinerten Abbildung darstellt sowie ein Positionselement, das dem Benutzer ermöglicht, andere Stellen des Films für die Vorschau auszuwählen. Sollten Änderungen an den Einstellungen gemacht werden, wird die Vorschau sofort automatisch aktualisiert. Dies kann (je nach Bildgröße des Films und verwendetem Computer) einige Sekunden in Anspruch nehmen.

Mit „Cancel“ wird der Dialog abgebrochen; mit „Crop“ wird die Beschneidung durchgeführt.

Die Beschneidung bezieht sich nur auf die Darstellung des Films. Das ursprüngliche Material wird nicht verändert. Man beachte, dass durch die Beschneidung die benötigte Prozessorleistung des abspielenden Computers ansteigt.

Die Beschneidung erfolgt durch das Hinzufügen einer Maske („Matte“), die QuickTime den gewünschten sichtbaren Bereich angibt. Prinzipiell wäre für die Maske jede zweidimensionale Form denkbar. Allerdings erscheint in diesem Zusammenhang nur eine rechteckige Form sinnvoll.

Ein Film kann nur beschnitten werden, solange noch keine Textspur enthalten ist, da diese nicht beschnitten werden können.

4.1.11 Aufheben des Croppings

Mit dem Menübefehl „Edit→Uncrop Movie“ kann eine vorherige Beschneidung (die auch durch ein anderes Programm stattgefunden haben kann) wieder aufgehoben werden.

Die Funktion testet, ob eine Maske vorhanden ist und entfernt diese.

4.1.12 Hinzufügen einer Spur

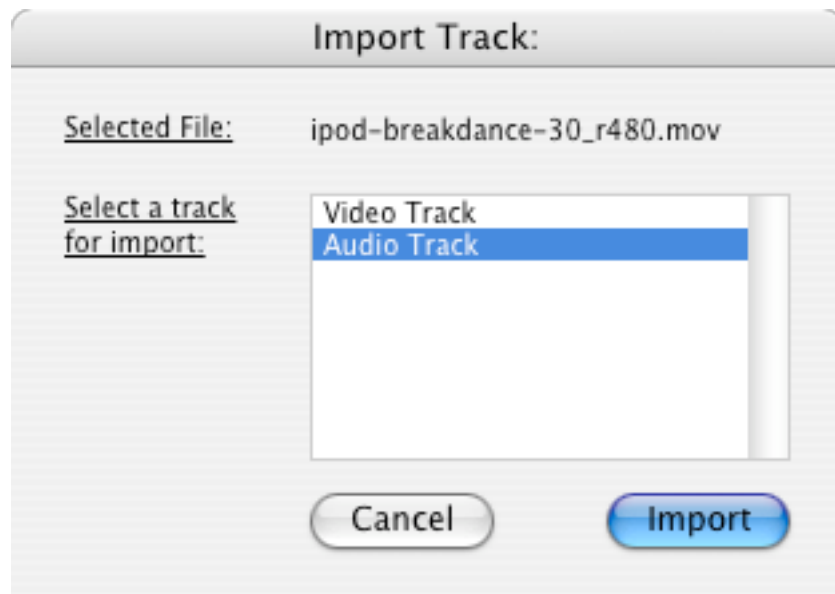


Abbildung 4.9: Das „Import Track“ Fenster

QT Sync bietet die Möglichkeit, dem aktuell geöffneten Film eine Spur aus einer anderen Datei hinzuzufügen, die parallel zu den bislang vorhandenen Spuren abgespielt werden soll. Dies kann beispielsweise eine Audiospur sein, die einer Filmdatei, welche nur Video enthält, beigefügt werden soll.

Über den Menüpunkt „Edit→Add Track...“ öffnet sich der in Abb. 4.9 dargestellte Dialog. Hier wird zunächst die Filmdatei ausgewählt, welche die hinzuzufügende Spur enthält. Danach bestimmt der Benutzer, welche der in dieser Datei vorhandenen Spuren eingefügt werden sollen.⁵⁶

Die entsprechende Spur wird importiert und bei erfolgreichem Abschluss des Vorgangs im Controller- und Informationsfenster angezeigt.

⁵⁶ In der aktuellen Version ist QT Sync nicht in der Lage, Videospuren einzufügen. Diese Einschränkung resultiert aus dem generellen Unvermögen von QT Sync, mit Filmen zu arbeiten, die mehr als eine Videospur besitzen. Es wird daher empfohlen, möglichst den Film mit der Videospur zu öffnen und die gewünschten Audiospuren zu importieren; nicht umgekehrt.

4.1.13 Entfernen einer Spur

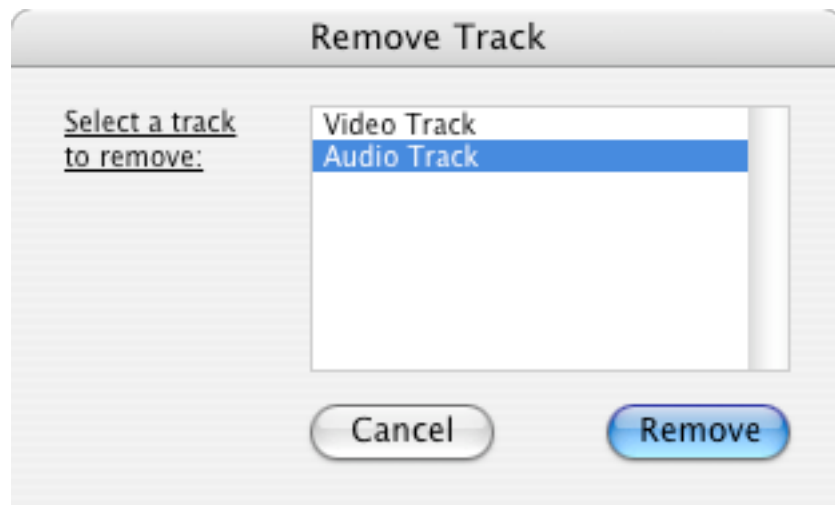


Abbildung 4.10: Das „Remove Track“ Fenster

Um eine Spur aus dem aktuell geöffneten Film zu entfernen, wählt der Benutzer aus dem Menü „Edit“ den Punkt „Remove Track...“. In dem erscheinenden Dialogfenster werden alle Spuren des geöffneten Films angezeigt. Nach der Auswahl der gewünschten Spur wird diese entfernt und ist auch nicht mehr im Controller- und Informationsfenster sichtbar.

Es ist zu erwähnen, dass nur die Referenz auf die entsprechende Spur gelöscht wird. Das tatsächliche Medienmaterial bleibt erhalten und kann über den Befehl „Edit→ Add Track...“ jederzeit aus der ursprünglichen Filmdatei wieder importiert werden.

4.1.14 Zusammenfügen mehrerer Filme

Während bei dem Einfügen einer Spur diese parallel zum bereits geöffneten Film angelegt wird, werden hier mehrere Filme hintereinander gehängt. Ein mögliches Einsatzgebiet wäre das Zusammenfügen eines Films aus verschiedenen Einzelsegmenten, welche aus organisatorischen Gründen voneinander getrennt wurden (z.B. die verschiedenen Akte eines Films). Es wäre auch denkbar, dass Filme wegen ihrer Dateigröße in mehrere Teile zerlegt wurden und erst dadurch ein Transport auf z.B. mehreren CD-ROMs ermöglicht wurde.

Für das Zusammenfügen mehrerer Filme bietet QT Sync seit der Version 0.2.2b1 einen Assistenten an, der (wie eine unabhängige Applikation) die Quellfilme in

einen Zielfilm zusammenfügt, welcher nicht in QT Sync geöffnet, sondern direkt auf die Festplatte gespeichert wird. Selbstverständlich kann dieser neue Film sofort nach der Erstellung in QT Sync geöffnet und bearbeitet werden.

Der Assistent wird durch den Menübefehl „Tools → Combine Movies...“ aufgerufen. Er kann nur geöffnet werden, solange in QT Sync kein Film geladen ist.

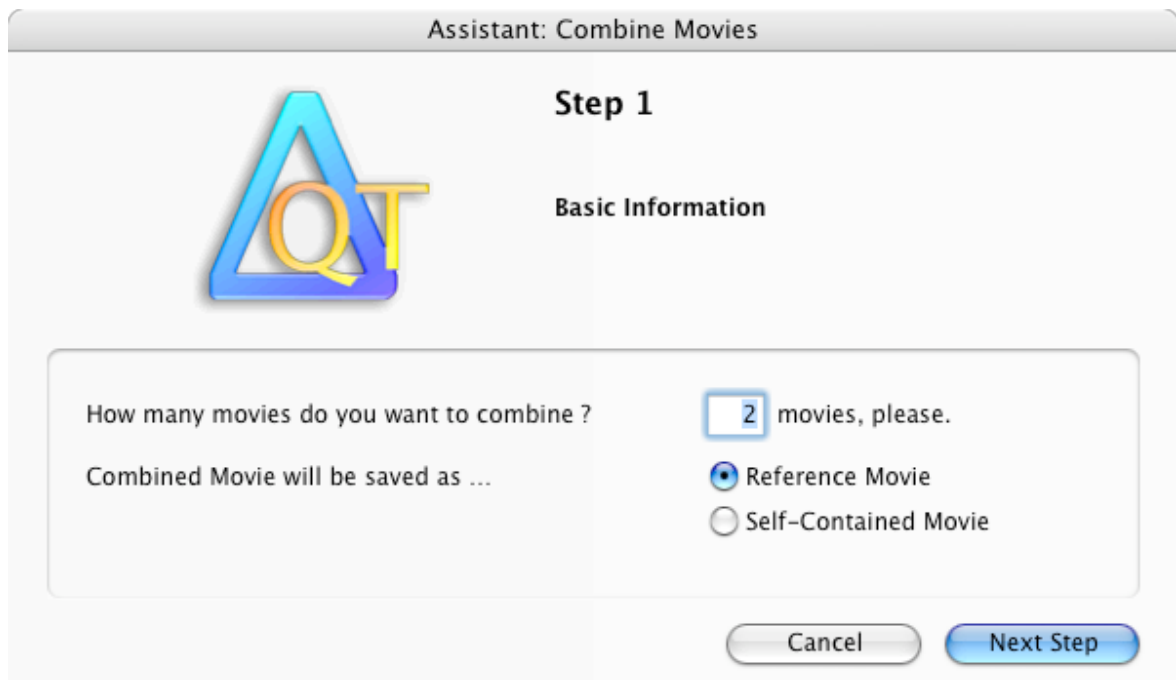


Abbildung 4.11: Das Fenster des „Combine Movies“ Assistenten (erste Seite)

Zunächst wird die Anzahl der Quellfilme und die Art des Zielfilms (selbständig oder von den Quellfilmen abhängig) erfragt.

Im zweiten Schritt müssen die Quellfilme ausgewählt werden. Der Assistent überprüft daraufhin, ob die Spur- und Codec-Informationen der Quellfilme identisch sind. Nur wenn diese übereinstimmen, ist ein Zusammenfügen möglich.⁵⁷ Andernfalls wird durch einen Warnhinweis auf diesen Fakt hingewiesen.

⁵⁷ Es ist theoretisch möglich, auch Filme mit unterschiedlichen Spur-Formaten miteinander zu verbinden. Allerdings zeigt die Praxis, dass derartig verknüpfte Filme meistens nicht zuverlässig abgespielt werden können. Dies ist auch im QuickTime Player zu beobachten.

Schließlich wird der gewünschte Speicherort des Zielfilms erfragt und das Zusammenfügen ausgeführt.

4.1.15 Hinzufügen einer TimeCode / Feet + Frames Einblendung

Das Einblenden eines TimeCodes kann vor allem bei Filmen nützlich sein, die während der Post Produktion in Audioschnittsystemen als Bildzuspiel verwendet werden. QT Sync bietet eine Möglichkeit, TimeCode nachträglich in ein Videobild „einzublenden“.

Dieser ist nur eine Anzeige. Weder ist QT Sync TimeCode-fähig (also imstande, sich zu einem anliegenden TimeCode zu synchronisieren), noch wird dem Film eine echte TimeCode-Fähigkeit verliehen.⁵⁸

Durch den Aufruf des Menüpunktes „Tools→TimeCode Inserter“ öffnet sich ein Dialogfenster, in dem verschiedene Darstellungsparameter konfiguriert werden können.

Zunächst muss die Wahl zwischen einer TimeCode- und einer Feet + Frames Einblendung getroffen werden. Bei Feet + Frames gibt es wiederum die Unterscheidung zwischen 16mm (40 Frames = 1 foot) oder dem üblichen 35mm Format mit 4 Perforationslöchern (16 Frames = 1 foot).⁵⁹

⁵⁸ QuickTime selber bietet gewisse TimeCode-Fähigkeiten. Da aber die Funktion der TimeCode-Einblendung primär für Filme gedacht ist, die später in Audiotbearbeitungssystemen weiter verwendet werden, sieht der Autor keinen Bedarf für eine (von der Programmierung her sehr aufwändige) echte TimeCode-Anbindung; die Problematik der Verkopplung lösen die entsprechenden Systeme eigenständig.

⁵⁹ Nicht ganz so häufig verwendet ist ein weiteres 35mm Format, bei dem die Einzelbilder nur drei statt vier Perforationslöcher hoch sind. Dieses Material wird häufiger für (amerikanische) Fernsehproduktionen verwendet, um Kosten zu sparen. Der Autor hat vorerst auf die Unterstützung dieses Formats verzichtet.



Abbildung 4.12: Das Fenster des TimeCode Inserters im TimeCode Modus

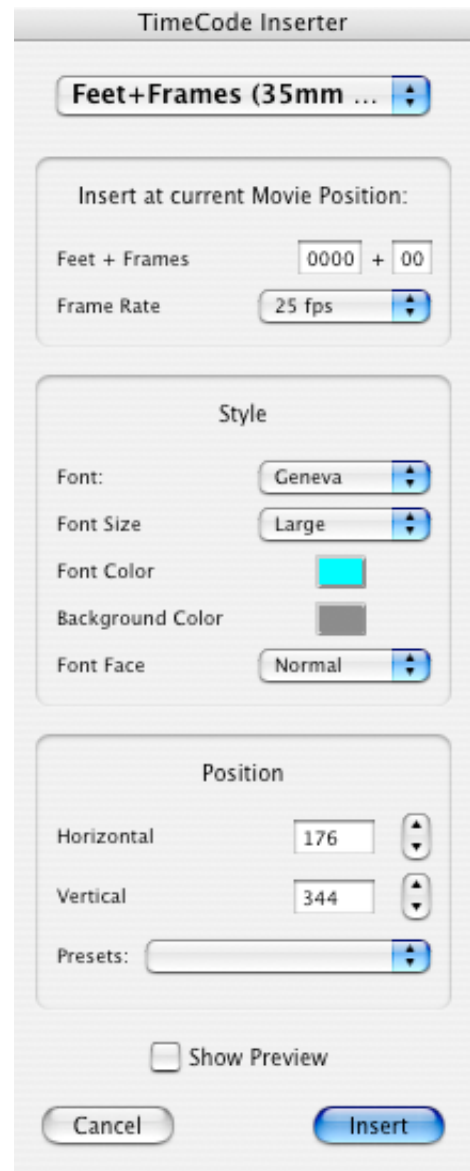


Abbildung 4.13: Das Fenster des TimeCode Inserters im Feet + Frames Modus (35mm / 4perf)

Nach der Entscheidung für eine TimeCode Anzeige bietet sich in der obersten Zeile die Eingabemöglichkeit für den TimeCode-Wert, welcher an der momentanen Jetzt-Position des Films eingesetzt werden soll.⁶⁰ Direkt darunter findet sich die Einstellungsmöglichkeit für die gewünschte Bildrate des

⁶⁰ Eine typische Vorgehensweise wäre daher, den Film zunächst an eine Stelle zu fahren, an der ein eindeutiges Synchronzeichen vorliegt (beispielsweise 2 Sekunden vor Bildstart), danach den TimeCode Inserter aufzurufen und den zu insertierenden TimeCode auf den entsprechenden Wert (beispielsweise 09:59:58:00) zu setzen.

TimeCodes. Sollte es notwendig sein, für den TimeCode einen anderen Bildratenwert als den der Videospur zu verwenden, ist mit einer stark erhöhten Prozessorlast beim Wiedergeben des Films zu rechnen.

Bei der Entscheidung für Feet + Frames ändern sich die Eingabefelder für die Zahlenwerte entsprechend. Auch hier wird der gewünschte Wert für die momentane Jetzt-Position des Films eingegeben. Die Bildrate sollte hier korrespondierend zu der Bildrate der Videospur gesetzt werden, da andernfalls die Geschwindigkeit des Frame Zählers nicht mit den tatsächlichen Einzelbildern korrespondiert.

Alle weiteren Einstellungen sind sowohl für TimeCode als auch für Feet+Frames identisch. Es wird daher im Folgenden exemplarisch nur der TimeCode Fall beschrieben.

Im zweiten Feld, welches mit „Style“ betitelt ist, können Einstellungen zum Aussehen des TimeCodes getroffen werden. Schriftart⁶¹, Schriftgröße, Farbe der Schrift, Farbe des Hintergrundes und Schriftstil sind auswählbar.

Das „Position“ Feld darunter bietet Möglichkeiten zur freien Positionierung der Einblendung innerhalb des Filmbildes. Die Werte für „Horizontal“ und „Vertical“ bezeichnen jeweils den Abstand der linken oberen Ecke der Einblendung zur linken oberen Ecke des Films.

Im Auswahlmenü „Presets“ sind einige typische Positionseinstellungen vorgegeben, die nach ihrer Anwahl individuell bearbeitet werden können.

Am unteren Rand des Fensters befinden sich die Schaltflächen für das Abbrechen des TimeCode Inserter Dialogs, das Erstellen der Einblendung sowie die Auswahlbox für eine Vorschau. Ist diese Box angekreuzt, wird in einem weiteren Fenster visuell veranschaulicht, wie die Einblendung mit den ausgewählten Einstellungen später aussehen wird.

In dem Fenster der Vorschau befindet sich außerdem eine Auswahlbox mit der Bezeichnung „Original Size“, welche bei Aktivierung das Filmbild in seiner

⁶¹ Der Autor hat nicht alle in Mac OS X verfügbaren Schriften zur Verwendung freigegeben, sondern beschränkt sich auf eine Auswahl von Schriften, die eine feste Zeichenbreite besitzen. Bei einer Verwendung von Schriften mit einer variablen Zeichenbreite würde der TimeCode, je nachdem aus welchen Ziffern er momentan zusammengesetzt ist, in der Darstellungsbreite variieren und damit eine sehr unruhige Anzeige verursachen.

Originalgröße anstatt einer verkleinerten Abbildung darstellt sowie ein Positionselement, das dem Benutzer ermöglicht, andere Stellen des Films für die Vorschau auszuwählen.⁶²

Sollten Änderungen an den Einstellungen im TimeCode Inserter Dialog gemacht werden, wird die Vorschau sofort automatisch aktualisiert. Dies kann (je nach Bildgröße des Films und verwendetem Computer) einige Sekunden in Anspruch nehmen.

Nach dem Klick auf die Schaltfläche „Create“ wird die TimeCode Einblendung erzeugt und in den Film importiert.⁶³ Dieser Vorgang dauert (je nach Länge des Films) einige Sekunden bis zu einer halben Minute.

Die TimeCode Einblendung ist technisch gesehen eine Textspur nach QuickTime Standard. QT Sync erzeugt einen eigenen QuickTime-Film, welcher nur die Textspur mit den TimeCode Informationen enthält, und legt diesen in dasselbe Verzeichnis wie die Originaldatei ab. Der Name dieses Films besteht aus dem Namen der Originaldatei (ohne seine Dateierweiterung), dem Zusatz „_TC“ und der Dateierweiterung „.mov“. Danach fügt QT Sync die Textspur dieses Films in den aktuell geöffneten Film ein (analog zur „Add Track“ Funktion).

Es ist wichtig zu wissen, dass die Informationen für die TimeCode-Anzeige nur in dieser separaten Datei enthalten sind. Ohne sie kann die erstellte Textspur nicht abgespielt werden. Allerdings besteht durch die Verwendung der „Save As Self-Contained Movie“-Funktion (s.u.) die Möglichkeit, die TimeCode Informationen zusammen mit allen anderen benötigten Daten in eine neue selbständige Filmdatei speichern zu lassen.

⁶² Die Auswahl dieses Positionselements bezieht sich nur auf die Anzeige der Vorschau und steht nicht im Zusammenhang damit, an welcher Stelle im Film der eingegebene TimeCode Wert erscheinen wird.

⁶³ Manchmal kann es dazu kommen, dass der angegebene TimeCode Wert nicht genau an der gewünschten Stelle platziert wird. In einem solchen Fall wird empfohlen, den Offset der nach dem Erstellen der TimeCode Einblendung im Controller-Fenster neu erscheinenden Textspur in gewohnter Art zu verändern.

4.1.16 Das „About“ Fenster

Über den Befehl „About...“ bzw. „Über dieses Programm...“ im Apfel-Menü erreicht der Benutzer das in Mac OS X obligatorische Fenster mit Informationen über das Programm und den Autor. Angezeigt werden das Programmlogo, die Versionsnummer, der Name des Autors und die Internetadresse der Webseite, auf der QT Sync erhältlich ist. Bei einem Klick auf die Internetadresse öffnet sich der im System voreingestellte Browser und versucht, eine Internetverbindung zu dieser Webseite herzustellen.

Auf der deutsch- und englischsprachigen Webseite findet der Benutzer neben dem Programm selbst die komplette Anleitung, nützliche Internetadressen und ein Forum, in dem Hilfen zur Bedienung des Programms gegeben werden.

4.1.17 Überprüfung der Verfügbarkeit einer neuen Programmversion

Über den Menüpunkt „Help → Check for Updates...“ gibt es die Möglichkeit festzustellen, ob eine neue Programmversion verfügbar ist. Dies erfordert eine Verbindung mit dem Internet.

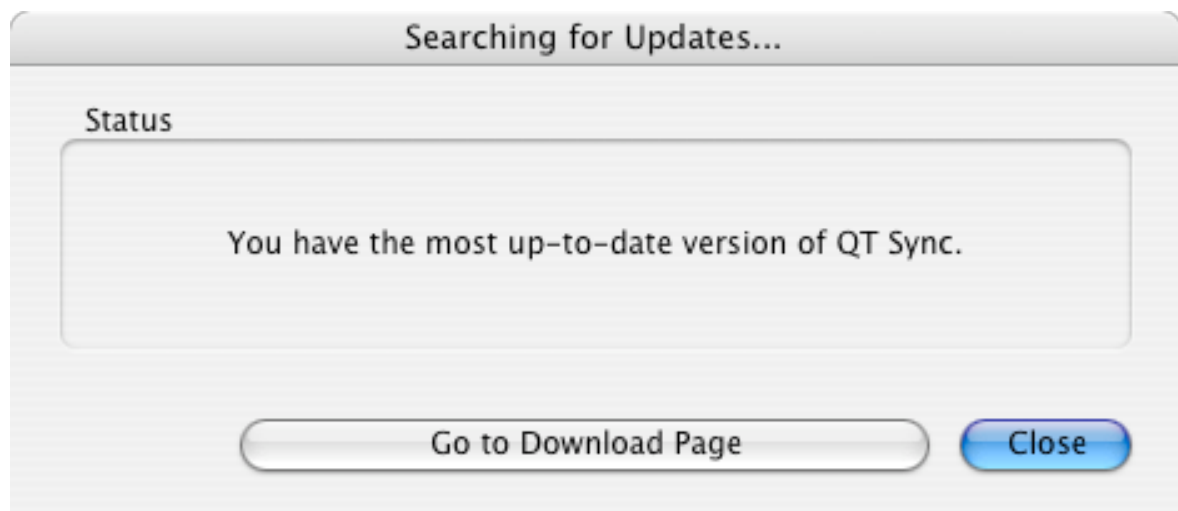


Abbildung 4.14: Das „Check for Updates...“ Fenster

Sollte eine neue Version verfügbar sein, kann mit einem Klick auf den „Go to Download Page“ Knopf die Webseite von QT Sync aufgerufen und die dort verfügbare neue Version heruntergeladen werden.

4.1.18 Schließen des Films

Ein Film wird geschlossen durch Auswahl des Menüs „File→Close“ bzw. das Tastaturkommando Apfel+W oder durch das Schließen des Filmfensters.

Sollten Änderungen am Film vorgenommen worden sein, wird der Benutzer durch eine Sicherheitsabfrage aufgefordert, den Film abzuspeichern oder zu bestätigen, dass die Änderungen tatsächlich verworfen werden sollen. Diese Abfrage bietet zusätzlich die Möglichkeit zum Abbruch des Schließvorgangs an.

4.1.19 Speichern eines Films

Das Konzept von QT Sync basiert darauf, dass die Original-Filmdatei grundsätzlich nicht verändert wird. Analog zu den heute üblichen nicht-destruktiven Bearbeitungssystemen (seien es Video- oder Audibearbeitungsprogramme) soll das Original nur als eine Vorlage dienen, welche dann innerhalb des Programms virtuell verändert wird. Eine weitere Begründung für die Anwendung dieses Verfahrens ergab sich aus dem praktischen Problem, dass ein Abspeichern von Änderungen in AVI Dateien nicht ohne größere Umwege möglich ist.

QT Sync speichert daher den bearbeiteten Film in einem so genannten „abhängigen Film“. Dieser Film beinhaltet ausschließlich Steuerdaten, nicht die tatsächlichen Mediendaten, und ist daher (relativ zur Originaldatei gesehen) fast vernachlässigbar klein. (Der abhängige Film, den QT Sync nach der Bearbeitung einer Filmdatei mit 400 MB Größe⁶⁴ erstellt, ist schätzungsweise 1,5 MB groß; also unter einem halben Prozent im Vergleich zur Originaldatei.)

Der abgespeicherte abhängige Film ist immer im QuickTime-Format und damit kompatibel zu allen QuickTime-Anwendungen. Wichtig ist allerdings, dass zum Abspielen des abhängigen Films immer die Originaldatei vorhanden sein muss. Sollten Spuren aus anderen Filmen hinzugefügt oder ein TimeCode eingeblendet worden sein, so müssen auch diese Dateien verfügbar sein.

⁶⁴ Eine 50minütige Filmdatei mit 3ivx-Video und MP3-Audio

Der Dialog zum Speichern wird über den Menüpunkt „File→Save As...“ bzw. das Tastaturkommando Apfel+S⁶⁵ aufgerufen. Der Benutzer wird in der Überschrift darauf hingewiesen, dass er einen abhängigen Film abspeichert. Gleichzeitig wird ein veränderter Name vorgeschlagen.⁶⁶ Das Verzeichnis, welches als Speicherort angeboten wird, ist dasselbe wie jenes, in dem sich die originale Filmdatei befindet.⁶⁷ Alle diese Angaben können selbstverständlich je nach Wunsch modifiziert werden mit der einen Ausnahme, dass der Name des Films auf jeden Fall die Endung .mov nachgestellt bekommt.

Sollte das Speichern nicht erfolgreich abgeschlossen worden sein, wird eine entsprechende Fehlermeldung ausgegeben.

4.1.20 Speichern eines Films in eine neue selbständige Filmdatei

Während ein Speichern von Änderungen in die Originaldatei hinein nicht möglich ist, bietet QT Sync die Möglichkeit, eine neue Filmdatei zu erstellen. In diese werden sowohl die vorgenommenen Änderungen als auch die tatsächlichen Mediendateien gespeichert. Der neu erstellte Film ist selbständig („Self-Contained“) und erspart das Bereithalten der Originaldatei(en). Dies ist vor allem dann nützlich, wenn durch Hinzufügen von Spuren aus anderen Filmdateien und/oder das Hinzufügen einer TimeCode-Spur der Film von mehreren Dateien gleichzeitig abhängig ist oder aber aus einer großen Datei viele Spuren herausgelöscht wurden, welche nun ungenutzt Festplattenspeicherplatz belegen.

⁶⁵ Dies ist Apple-untypisch, da das Tastenkürzel Apfel+S normalerweise dem „Save“ Kommando zugeordnet ist und dagegen der Befehl „Save As...“ kein eigenes Tastaturkürzel besitzt. Da aber in QT Sync kein „Save“ Befehl existiert, hat sich der Autor zu Gunsten der Bedienungsfreundlichkeit dafür entschieden, die Apple Richtlinien diesbezüglich zu missachten.

⁶⁶ In der Grundeinstellung wird dem Namen das Kürzel „_QT“ nachgestellt. Dies ist aber in den Voreinstellungen individuell einstellbar.

⁶⁷ Auch hier liegt ein leichter Verstoß gegen die Apple-Richtlinien vor. Von Apple gewünscht ist, dass das Verzeichnis angeboten wird, in welches von dem entsprechenden Programm zuletzt hinein gespeichert wurde.

Auch beim Wechsel des Arbeitsplatzes, Übergabe des fertigen Ergebnisses an die nächste Bearbeitungsstation, zur Archivierung oder beim Versenden über Internet kann es sinnvoll sein, nur eine Datei transportieren zu müssen.

Das Speichern in einen selbständigen Film wird über das Menü „File → Save As Self-Contained Movie“ aufgerufen. Das erscheinende Auswahlfenster ähnelt dem normalen „Save As...“-Dialog stark und bietet dieselben Funktionen (Automatische Namensvorgabe, Verzeichnis der Originaldatei als vorgeschlagenes Verzeichnis zum Abspeichern).

Nach der Auswahl eines Zielordners beginnt die Speicherung des Films. Der Status des Vorgangs wird durch den QuickTime-üblichen Fortschrittsbalken angezeigt. Je nach Länge des Films kann dieser Vorgang bis zu mehreren Minuten dauern.

Während des Speicherns kopiert QT Sync alle benötigten Mediendaten aus den entsprechenden Original-Dateien und erstellt schließlich eine neue Filmdatei, welche sowohl die bearbeitete Filmstruktur als auch alle benötigten Mediendaten enthält.

Zu beachten ist dabei, dass tatsächlich nur die benötigten Mediendaten kopiert werden. Evtl. nicht benutzte Spuren (oder Teile) der Originaldatei(en) werden nicht kopiert.

Ebenso wichtig ist der Fakt, dass es sich hierbei nicht um eine Neukodierung des Materials handelt.

4.1.21 Der Full Screen Modus

Der Menüpunkt „View→Full Screen“ bzw. das Tastaturkommando Apfel+F öffnet den Full Screen Modus. (Die Wahl dieses Befehls ist auch bei laufendem Film möglich. Während des Wechsels kommt es allerdings zu einem systembedingten Tonaussetzer.)

Nach dem Aufruf werden alle Mac OS-typischen Bedienelemente wie das Dock oder die Menüleiste ausgeblendet und der Film wird auf die Fläche des gesamten Bildschirms vergrößert. Evtl. unbenutzte Bildschirmbereiche werden geschwärzt, so dass insgesamt der Eindruck eines Fernsehers entsteht. Des Weiteren wird der Benutzer in seiner Handlungsfreiheit leicht eingeschränkt. So ist es beispielsweise nicht möglich, den Film aus dem Full Screen Modus heraus zu schließen oder zu speichern.

Nur das Controller Fenster ist als einziges Bedienelement weiterhin verfügbar. Um möglichst wenig störend aufzufallen, wird es an den obersten Bildschirmrand geschoben und seine Farben werden geändert: Anstatt grau ist die Hintergrundfläche des Fensters nun schwarz (womit auch die Beschriftungen nicht mehr sichtbar sind) und die Hintergründe der Eingabefelder wechseln von weiß zu einem mittleren Grauton. Die Arbeitsweise im Controllerfenster bleibt identisch.

Sämtliche transportbezogenen Tastaturkommandos sind nach wie vor verfügbar.

4.1.22 Der Full Screen Modus mit verstecktem Controller Fenster

Dieser Modus ist das Herzstück von QT Sync und war ursprünglich die Grundanforderung an das Programm.

In diesem Modus wird der Film bildschirmfüllend abgespielt und es besteht parallel die Möglichkeit, per Tastaturkommando die Synchronität des Films zu korrigieren.

Das im Full Screen Modus eingeblendete Controller Fenster ist mit dem Tastaturkommando Apfel+G aus- und wieder einblendbar.

Bei ausgeblendetem Controller ist zusätzlich der Mauszeiger unsichtbar. (Bei evtl. Fehlermeldungen innerhalb dieses Modus wird mitgeteilt, wie die Meldungen auch ohne Maus bestätigt werden können; nämlich durch Betätigen der Return Taste.)

Der Grundgedanke dieses Modus ist es, dieselbe Funktionalität durch Tastaturkommandos zu ermöglichen, wie bei der Arbeit mit dem Controller Fenster. Dies kann QT Sync mit einer Ausnahme gewährleisten: Es ist nicht möglich, mehrere Spuren gleichzeitig „scharf“ zu schalten. (Sollten vor dem Umschalten in den „Full Screen Modus mit verstecktem Controller Fenster“ mehrere oder gar keine Spuren scharf geschaltet worden sein, so wird der Benutzer bei dem Versuch, mit den Tastaturkürzeln zu arbeiten, auf diesen Umstand hingewiesen.)

Im „Full Screen Modus mit verstecktem Controller Fenster“ sind nach wie vor alle transportbezogenen Tastaturkürzel verfügbar. Folgende neue Kommandos sind zusätzlich aktiv:

- 0** Rücksprung im Film um sechs Sekunden. Der Wert ist in den Voreinstellungen veränderbar. (Dieses Tastaturkürzel ist auch mit eingeblendetem Controllerfenster verfügbar.)
- TAB** (die Tabulatortaste): Scharfschalten der jeweils nächsten Spur. (Es kann zeitgleich immer nur eine Spur scharf geschaltet sein.)
- T** Einblendung der abgelaufenen Zeit des Films. Ein weiterer Tastendruck schaltet auf die noch verbleibende Filmlaufzeit um. Ein dritter schaltet die Anzeige wieder aus. (Dieses Tastaturkürzel ist auch mit eingeblendetem Controllerfenster verfügbar.)

Befehle zur Offsetänderung:

- 1** erniedrigt den Offset der scharfen Spur
- 2** zeigt den Offset aller Spuren
- 3** erhöht den Offset der scharfen Spur

Befehle zur Längenänderung

(bewirken eine vom Original abweichende Wiedergabegeschwindigkeit):

- 7** verkürzt die scharf geschaltete Spur
- 8** zeigt die Länge aller Spuren
- 9** verlängert die scharf geschaltete Spur

Befehle zur Schrittgröße:

- 4** Umschaltung zwischen den verschiedenen Einheiten (frames, msec und TimeScales).
(Durch die Umstellung könnten interne Rundungsfehler bis zu einem halben Bild entstehen. Nach einer Änderung der Einheit wird daher der Film noch einmal mit den aktuellen Werten bearbeitet.)
- 5** Auswahl der Schrittgröße für die Tastaturbefehle „1“, „3“, „7“ und „9“.

Nach Anwendung eines Tastaturbefehls wird in der rechten oberen Ecke des Bildschirms für einige Sekunden eine Statusanzeige eingeblendet. Diese

informiert, je nach bearbeitetem Parameter, über die aktuellen Werte der Offsets oder Längen der Spuren bzw. die gewählte Einheit, Schrittgröße oder scharf geschaltete Spur.

Die Einblendungen für die Tastaturkürzel „1“, „3“, „7“ und „9“ sind in den Voreinstellungen deaktivierbar.

Spurlängenwerte sind vom angewählten Status der Schaltfläche „Relative/Total“ im Controller Fenster abhängig und zeigen entsprechend die relative oder absolute Dauer der Spuren an.

Zur besseren Unterscheidung zwischen Offset- und Längenangaben wird grüne bzw. blaue Schrift benutzt. Informationen zur Schrittgröße, Einheit oder Spurscharfschaltung sind in gelb gehalten.

4.1.23 Verlassen des Full Screen Modus

Der Full Screen Modus wird mit der ESC (Escape) – Taste verlassen.

Daraufhin wird der bildschirmfüllende Modus aufgehoben und der normale Arbeitszustand (inklusive Dock und Menüleiste des Mac OS) wieder hergestellt. Unabhängig davon, welcher Betrachtungsmodus vor dem Umschalten in den Full Screen Modus aktiv war, wird wieder in den „Normal Mode“ geschaltet.

Der Befehl zum Verlassen des Full Screen Modus kann auch während der Filmwiedergabe ausgeführt werden. Während des Wechsels kommt es allerdings zu einem systembedingten Tonaussetzer.

4.1.24 Voreinstellungen

Der Benutzer erreicht die Voreinstellungen durch den Menüpunkt „Preferences“ im Apfel-Menü oder durch das Tastaturkürzel Apfel+Komma.

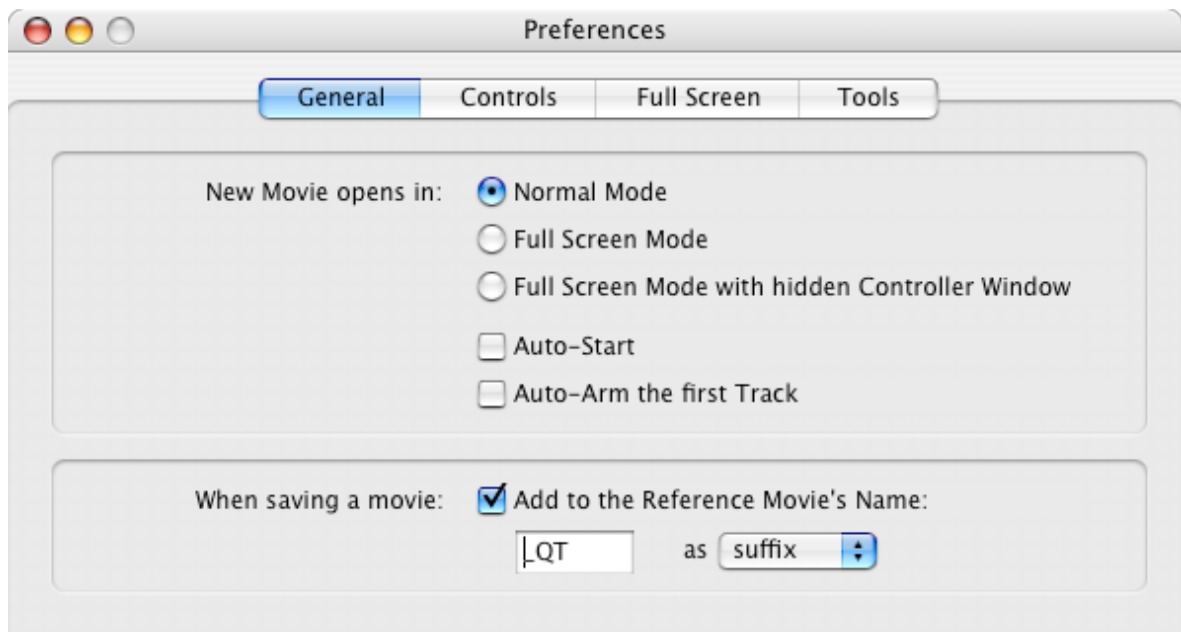


Abbildung 4.15: Voreinstellungsfenster (Spalte „General“)

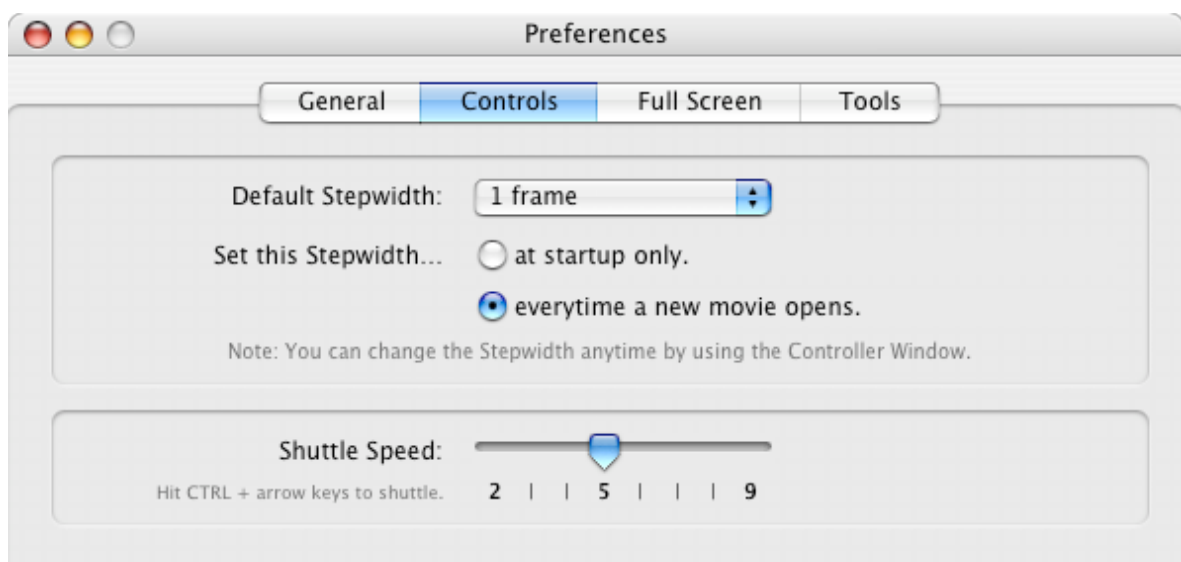


Abbildung 4.16: Voreinstellungsfenster (Spalte „Controls“)

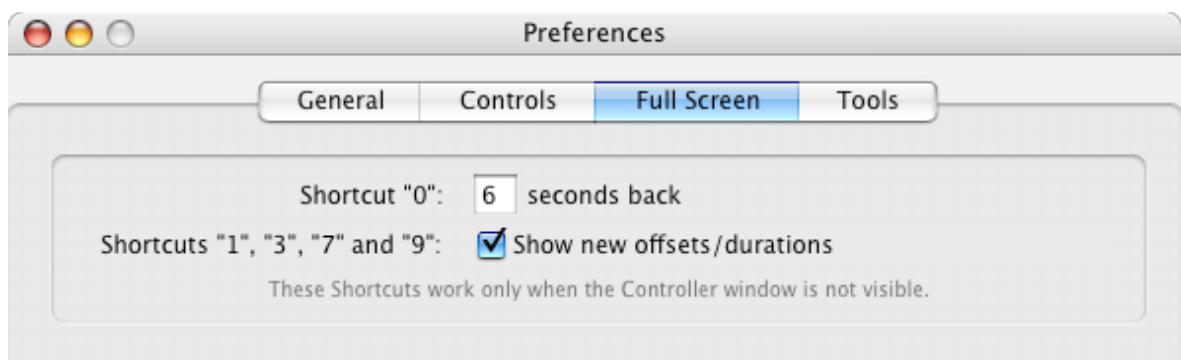


Abbildung 4.17: Voreinstellungsfenster (Spalte "Full Screen")

4.1.24.1 Kategorie „General“ (Allgemeines)

New Movie opens in... : Es kann zwischen drei Betrachtungsmodi gewählt werden, in denen ein neuer Film geöffnet wird. Dies sind der „Normal Mode“, der „Full Screen Mode“ und der „Full Screen Mode with hidden Controller“ (Full Screen Modus mit verstecktem Controller Fenster)

Auto-Start : Bei Aktivierung dieser Option wird ein neu geöffneter Film sofort abgespielt.

Auto-Arm the first Track : Ist diese Option aktiviert, wird die erste Spur eines neu geöffneten Films automatisch scharf geschaltet, allerdings nur, wenn der Film mehr als eine Spur besitzt.

When saving a movie... : Bei dem Speichern eines abhängigen Films kann dem vorgeschlagenen Namen ein Zusatz beigefügt werden. Es besteht hierbei die Wahl zwischen einer vor- oder nachgestellten Ergänzung („Suffix“ bzw. „Präfix“) sowie die freie Eingabemöglichkeit der zu verwendenden Zeichenfolge. Bei Werkseinstellung wird dem abhängigen Film ein „_QT“ nachgestellt, um ein versehentliches Überschreiben der ursprünglichen Filmdatei zu verhindern und die Referenzdatei als solche kenntlich zu machen.

4.1.24.2 Kategorie „Controls“ (Vorgaben für das Controller Fenster)

Default Stepwidth : Hier kann zwischen 12 möglichen Schrittgrößen für die Offset- und Längenbearbeitung ausgewählt werden. (0,5 / 1 / 2 / 3 frames, 20 / 40 / 80 / 160 msec, 10 / 20 / 40 / 80 TimeScales)

Set this stepwidth... : Diese Option entscheidet, ob die oben ausgewählte Schrittgröße jedes Mal eingestellt wird, wenn ein neuer Film geladen wurde oder nur, wenn QT Sync gestartet wird.

Shuttle : Hier kann die Geschwindigkeit für das schnelle Vor- und Rückspulen (Bildsuchlauf) eingestellt werden.

4.1.24.3 Kategorie „Full Screen“

Shortcut „0“ : Die numerische Angabe bestimmt, um wie viele Sekunden der Film beim Betätigen der „0“ zurückspringt.

Shortcuts „1“, „3“, „7“, „9“ – Show new offsets/durations : Ist diese Option **deaktiviert**, werden nach dem Betätigen der Tastaturbefehle „1“, „3“, „7“ oder „9“ (und der damit verbundenen Offset- oder Längenänderung) die nun aktuellen Offsets bzw. Längen aller Spuren **nicht** in einer Einblendung oben rechts im Film gezeigt.

4.1.24.4 Kategorie „Tools“

Hier sind noch keine Voreinstellungen verfügbar.

Alle Änderungen sind ab dem Moment aktiv, in dem sie ausgewählt werden.

4.1.25 Das Beenden des Programms

Das Programm wird durch den Befehl „Quit QT Sync“ im Menü „QT Sync“ beendet, bzw. durch das korrespondierende Tastaturkommando Apfel+Q.

Sollte ein Film geöffnet sein, wird dieser zuvor geschlossen. Sind an dem Film Änderungen vorgenommen worden, wird vor dem Schließen durch eine Sicherheitsabfrage darauf hingewiesen und die üblichen Möglichkeiten zum Speichern des Films oder zum Abbruch des Schließens angeboten.

5. Programmierung

In diesem Abschnitt soll auf die Programmierung mit REALbasic im Hinblick auf die Applikation QT Sync eingegangen werden.

Die REALbasic Programmierumgebung macht es möglich, innerhalb kurzer Zeit mit Hilfe von vorgefertigten Modulen ein gewünschtes Ziel zu erreichen.

Die Erzeugung einer graphischen Benutzeroberfläche erfolgt im Baukastenprinzip. Vorgefertigte Elemente wie Fenster, Knöpfe und Eingabefelder werden in einem Editor zusammengefügt und über ein Menü mit den gewünschten Eigenschaften versehen. Der Code, der schließlich dafür sorgt, dass sich die Elemente im laufenden Programm wie erwartet verhalten, wird von REALbasic beim Kompilieren⁶⁸ automatisch erzeugt.

Nachteilig ist hierbei, dass der Programmierer an den Funktionsumfang der Module gebunden ist. Sobald eine Aufgabe ausgeführt werden soll, für die kein vorgefertigtes Modul besteht bzw. das vorhandene Modul nicht alle benötigten Funktionen bietet, hat der Programmierer zwei Möglichkeiten zur Erweiterung des Funktionsumfangs:

- Shared Libraries

Jedes Betriebssystem stellt Befehlsbibliotheken zur Verfügung, die von Programmen genutzt werden können. Diese bieten Zugriff auf systemnahe Vorgänge und Technologien (z.B. auf QuickTime).

REALbasic bietet dem Programmierer die Möglichkeit, über Frameworks⁶⁹ auf diese Bibliotheken zuzugreifen und deren Funktionen in seine Applikation einzubinden.

Nachteil der Nutzung dieser Libraries ist, dass innerhalb des REALbasic Codes mit Konventionen der Programmiersprache Carbon⁷⁰ gearbeitet werden muss. Dies

⁶⁸ Das Kompilieren (engl.: „compile“) beschreibt den Vorgang, bei dem der geschriebene Code in „Maschinensprache“ (die Sprache, die vom Computer tatsächlich verstanden wird) umgesetzt wird.

⁶⁹ Vgl. Kap. 2.2.2

⁷⁰ Vgl. Kap. 3.5

erfordert eine Konvertierung von Variablentypen und eine spezielle Syntax bei dem Aufruf von Funktionen.

- Plug-Ins

Die zweite Möglichkeit, den Funktionsumfang von REALbasic zu erweitern, besteht in der Anwendung von Plug-Ins. Sie ergänzen vorhandene REALbasic Module oder führen neue ein.

Der Hauptunterschied zwischen Plug-Ins und dem direktem Zugriff auf die Shared Libraries besteht darin, dass die Module des Plug-Ins in gewohnter REALbasic Syntax angesprochen und damit komfortabler in den Quellcode eingearbeitet werden können.

Das QuickTime Modul von REALbasic beispielsweise bietet die Standardfunktionen für das Abspielen und für einfache Manipulationen eines QuickTime Movies. Allerdings fehlen einige tiefer gehenden Funktionen, welche zur Erfüllung des Aufgabenkatalogs von QT Sync notwendig sind.

Kompensiert wurde dies sowohl durch die Einbindung der Shared Library „QuickTime Lib“, als auch durch den Einsatz des „QTPlugin“ von Dr. Riccardo Favilla und des „MBS Plugin“ von Christian Schmitz.

5.1 Programmierbeispiele

Im Folgenden soll der Quellcode einiger Funktionen eingehender diskutiert werden. Die Kenntnis der Syntax und der Arbeitsweise von REALbasic wird hierbei vorausgesetzt.

Um jedoch auch Lesern, die nicht mit dieser Programmierumgebung vertraut sind, einen Einblick zu geben, wird im ersten Codeauszug zusätzlich eine Kommentierung der normalen REALbasic Syntax vorgenommen. Grundkenntnisse der Programmierung werden aber auch hierfür benötigt.

Die Erläuterungen sind als Kommentare in den betreffenden Code eingefügt.

Um eine bessere lineare Präsentation zu ermöglichen, wurden einige Subroutinen in den Quellcode eingearbeitet und unrelevante Passagen gekürzt.

Formatierung der Kommentare:

Alle Erläuterungen zur prinzipiellen REALbasic Syntax werden kursiv und in blau, die Ausführungen zu einzelnen Funktionen in roter Kursivschrift dargestellt.

5.1.1 Vorbemerkungen

Globale Variablen:

Variablen besitzen einen Gültigkeitsbereich. Dieser Bereich beschreibt, welche Instanzen innerhalb des Programms auf die Variable zugreifen dürfen. Vereinfacht formuliert: Private Variablen sind nur in der Funktion gültig, in welcher sie definiert wurden, wohingegen auf globale Variablen von jeder Funktion aus zugegriffen werden kann.

Variable „em“ (definiert als EditableMovie)

Die Repräsentation der QuickTime Moviestruktur des in QT Sync geöffneten Films.

Variable „Tracks(x,y)“ (definiert als String)

Ein Array zur Speicherung von Daten der Moviestruktur. Diese werden beim Öffnen der Filmdatei eingelesen. Auf eine Auflistung der Struktur des Arrays wird verzichtet, stattdessen wird im Quellcode auf den jeweiligen Inhalt hingewiesen.

5.1.2 Öffnen von Filmen

Vorbemerkungen:

- Namen von Klassen, Objekten, Variablen, etc. sind von der Groß- und Kleinschreibung unabhängig. „OpenTheFile“ bezeichnet dasselbe Objekt wie „openthefile“.*
- Die Syntax von REALbasic ist der des normalen Basic sehr ähnlich. Die Kenntnis um If-Then-Else, While-Wend und andere Operationen wird in diesem Abschnitt vorausgesetzt.*
- REALbasic basiert auf objektorientierter Programmierung. Objekte (Klassen oder Module) sind voneinander unabhängig und besitzen jeweils definierte Eigenschaften und/oder Methoden. Die Syntax zum Ansprechen von Eigenschaften oder Methoden in anderen Objekten ist üblicherweise:*

Objekt.Eigenschaft = ...

Objekt.Methode

Objekt.Methode(Parameter1,Parameter2,...)

Ergebnis = Objekt.Methode(Parameter1,Parameter2,...)

Parameter werden standardmäßig als Wert übergeben. Durch den Zusatz „ByRef“ in der aufgerufenen Methode ist es allerdings möglich, die Variable selbst an die Methode zu übergeben.

Sollte eine Methode eine Eigenschaft oder Methode innerhalb desselben Objekts ansprechen, kann das Objekt ausgelassen werden.

Es sind auch globale Eigenschaften oder Methoden definierbar, die ebenfalls ohne Angabe des Objekts angesprochen werden können.

- Methoden, die ein Ergebnis zurückliefern, heissen „Function“. Methoden ohne Ergebnis werden „Sub“ genannt.

- Methoden werden auch Routinen genannt.

Sub openthefile (f as folderitem)

// Beginn der Subroutine „openthefile“. Das beim Aufruf als Parameter an die

// Funktion übergebene Folderitem wird als „f“ definiert.

// Ein Folderitem enthält Informationen über eine bestimmte Datei,

// z.B. Name, genauer Speicherort, Größe und Typ.

// Um eine Filmdatei zu öffnen, hat der Benutzer 3 Möglichkeiten. Er kann über

// den Menübefehl „File→Open“ eine Datei auswählen, eine Datei auf das

// Programmicon ziehen oder den Film per Doppelklick öffnen (vorausgesetzt, QT

// Sync ist als Standardprogramm hierfür voreingestellt).

// In jedem Fall wird die zu öffnende Datei in einem Folderitem gespeichert

// und an die Subroutine „openthefile“ übergeben.

// Zunächst werden private Variablen definiert. Dies sind Variablen, welche nur

// innerhalb dieser Subroutine bestehen.

// Ein Integer ist eine ganze Zahl zwischen 2^{31} und -2^{31} .

// Ein String ist eine beliebig lange Folge aus alphanumerischen Zeichen.

dim i as integer

dim s as string

If f<>Nil and f.exists then

// Es wird getestet, ob f ein gültiges Folderitem enthält und ob die durch das

```

// Folderitem beschriebene Datei existiert.
// („Nil“ meint „Null“ oder „leer“. Ein Folderitem kann leer sein, wenn der
// Benutzer den „Datei öffnen“-Dialog aufruft und dann abbricht, anstatt eine
// Datei auszuwählen.)
em = f.OpenEditableMovie
// Es wird versucht, die Filmdatei als EditableMovie zu öffnen.
// Ein EditableMovie ist per Definition beschreibbar. Die dazugehörige Datei
// muss sich daher auf einem Datenträger mit Schreibzugriff befinden.
if em = nil then
    // Wenn sich die Datei auf einem nur lesbaren Datenträger (z.B. einer CD-
    // Rom) befindet, wird der Öffnungsversuch scheitern. In diesem Fall wird eine
    // andere Methode verwendet:

    if TryTempfile(f,s) then Tracks(0,1) = s
    // Die Funktion TryTempfile versucht, einen lokalen Referenzfilm mit
    // Schreibrechten anzulegen und gibt im Erfolgsfall das Folderitem dieser
    // neuen Datei zurück. s ist dabei der Name der alten Filmdatei. Dieser
    // wird in Tracks(0,1) eingelesen und später im Abspielfenster als Titel
    // angezeigt.
    // If-Then Befehle müssen nicht mit End If abgeschlossen werden,
    // solange sie sich in einer Zeile befinden.
    // Die Funktion TryTempfile gibt als Ergebnis „True“ oder „False“
    // zurück. f und s sind nicht als Wert, sondern als Variablen übergeben
    // worden. Sie wurden also innerhalb der TryTempfile Routine
    // geändert.
    em = f.OpenEditableMovie
    // Das neue Folderitem wird geöffnet.
end if

if em <> Nil then
    // Ist em nicht leer, wird mit dem Öffnen der Datei begonnen:
    em.saveOnClose = false
    // Es wird festgelegt, dass beim Schliessen des Films keine Änderungen
    // in diesen Film gespeichert werden.
    if not (em.trackcount > 5) then
        // em.trackcount zählt die im Film enthaltenen Spuren.
        // QT Sync ist nicht in der Lage, mehr als fünf Spuren zu verarbeiten.
        Mov.MoviePlayer1.Movie = em
        Mov.MoviePlayer2.Movie = em
    end if
end if

```

```

// Das EditableMovie em wird an die zwei MoviePlayer im Hauptfenster
// übergeben. Die MoviePlayer spielen später den Film ab.
// Es existieren zwei Instanzen des MoviePlayers. Eine enthält eine Controller
// Leiste, die andere nicht. Die MoviePlayer werden abwechselnd sichtbar,
// je nachdem, ob eine Controllerleiste vorhanden sein soll (normaler
// Darstellungsmodus) oder nicht (Full Screen Modus).
// Dieser Weg ist notwendig, da sich die Controllerleiste während des
// Programmablaufs nicht ein- bzw. ausblenden lässt.
infoget (f)
// In der Routine „infoget“ werden Informationen über den Film und die
// enthaltenen Spuren gesammelt und in das Tracks Array eingetragen.
// Diese Routine wird ausschnittsweise später diskutiert.
Mov.title = Tracks(0,1)
// Der Titel des Abspielfensters wird festgelegt.
volume_start
// Durch die globale Methode volume_start wird die Lautstärke des
// Films festgelegt. 50% für .mpg, 100% für alle anderen.

// Die Fenster (Abspielfenster und Controllerfenster) werden geöffnet.
con.visible = true
mov.visible = true
// Der Darstellungsmodus „Normal“ wird aktiviert.
Mov.moviePlayer1.visible = true
mov.size(1)
// Das Fenster mov enthält die Methode size(), die alle Größen-
// veränderungen des Abspielfensters durchführt.

// Das Array pref enthält Werte aus den Voreinstellungen.
// Abhängig davon wird nun in den Full Screen Modus oder in den
// Full Screen Modus mit verstecktem Controllerfenster
// geschaltet (pref(11)) und der Film gestartet (pref(12)).
select case pref(11)
case "1"
  app.doEvents
  // DoEvents unterbricht die Abarbeitung und lässt dem System Zeit,
  // eventuell aufgelaufene Ereignisse abzuarbeiten. (z.B. das Darstellen
  // von Fenstern)
  mov.sizef
  // sizef wechselt in den Full Screen Mode.

```

```

        if pref(12)="1" then mov.moviePlayer2.play
    case "2"
        app.doEvents
        mov.sizef
        mov.sizeg
        // sizeg blendet im Full Screen Modus den Controller aus.
        if pref(12)="1" then mov.moviePlayer2.play
    else
        if pref(12)="1" then mov.moviePlayer1.play
    end select

else
    msgbox("Sorry, couldn't open the file."+endofline+endofline+ _
        "QT Sync can't open movies with more than 5 tracks.")
    // msgbox öffnet eine Dialogbox mit einer Mitteilung für den Benutzer.
    // Die beiden „EndOfLine“ Bestandteile trennen die Mitteilung in eine fett
    // gedruckte Überschrift und normalen Hinweistext.
end if
else
    msgbox("Sorry, couldn't open the file."+endofline+endofline+ _
        "This shouldn't have happened. Please write me a bug report.")
    // Der Unterstrich am Ende der Zeile zwingt REALbasic, den nachfolgenden
    // Zeilenumbruch zu ignorieren.
    // Bei sehr langen Befehlszeilen kann so die Übersichtlichkeit verbessert
    // werden.
end if
end if

End Sub

```

```

Function TryTempfile(ByRef f as folderitem, ByRef DisplayName as string) _
    as Boolean
    // Liefert im Erfolgsfall als Ergebnis „True“, sonst „False“.
    // Durch den Zusatz ByRef wird festgelegt, dass nicht die Werte der Variablen
    // sondern die Variablen selbst übergeben werden.
    // Eine Änderung des Variablenwertes innerhalb dieser Routine wirkt sich daher
    // auch auf die Variable in der aufrufenden Routine aus.

```

// Definition der privaten Variablen:

```
dim origfile, newfile as folderitem
dim origmovie, newmovie as movie
dim error as integer
dim origname as string
dim testem as EditableMovie
```

*// Der Sinn der Routine TryTempfile ist es, den anscheinend schreibgeschützt
// gespeicherten Film als Variablentyp Movie einzulesen (für den keine
// Schreibrechte benötigt werden) und eine lokale Referenzdatei zu erstellen,
// welche auf die Spurdaten (Bild, Ton, etc.) der Originaldatei zurückgreift.*

```
origfile = f
newfile = TemporaryFolder.child("qtstemp.mov")
// Da diese Methode des Öffnens vom Benutzer unbemerkt geschehen soll,  
// wird die lokale Referenzdatei im unsichtbaren Temporärordner des Systems  
// angelegt.
```

```
dim i as integer = 0
while newfile.exists
    i=i+1
    newfile = TemporaryFolder.child("qtstemp"+str(i)+".mov")
    if i > 1000 then goto bail
wend
```

// Die Originaldatei wird geöffnet und ihr Name zwischengespeichert.

```
origmovie = origfile.openAsMovie
if origmovie = nil then goto bail
origname = origfile.displayname
```

// Am Ort der Temporärdatei wird eine neue Moviestruktur erstellt.

// Die Funktion CreateAnEmptyMovie entstammt dem QTPlugin.

```
newmovie = CreateAnEmptyMovie(newfile)
if newmovie = nil then goto bail
```

// Das Referenzmovie wird erstellt und auf die Festplatte geschrieben;

// ebenfalls eine Funktion des QTPlugins.

```
error = InsertAMovieByReference(origmovie,newmovie)
if error <> 0 then goto bail
```

```

// Alle geöffneten Dateien werden geschlossen.
// REALbasic kümmert sich normalerweise selbst um das Abmelden von
// Variablen bzw. das Schliessen von Dateien, sobald diese nicht mehr
// benötigt werden.
// In diesem Fall soll die Datei sofort geschlossen werden, damit durch das
// nachfolgende erneute Öffnen keine Fehlermeldung ausgelöst wird.
origmovie = nil
newmovie = nil
origfile = nil

// Der neue Referenzfilm wird erneut geöffnet, um den Erfolg der Funktion
// festzustellen.
testem = newfile.OpenEditableMovie
if testem = nil then goto bail

// Erst jetzt werden die ursprünglich übergebenen Variablen verändert.
DisplayName = origname
f = newfile
return true

// Mit Return wird der Wert festgelegt, den diese Funktion als Ergebnis
// zurückliefert. Return bedeutet auch das sofortige Ende der Funktion.
// (Eventuell nachfolgender Code wird nicht mehr ausgeführt.)

bail:
// Hiermit wird ein Anker namens „bail“ festgelegt, der innerhalb der Routine
// mit „goto bail“ jederzeit angesprungen werden kann.
return false

```

End Function

Bemerkung:

Die Benutzung des Goto Befehls ist unter Programmierern sehr verpönt, da durch seine Anwendung der Quellcode sehr unübersichtlich werden kann. Manche Programmiersprachen haben daher gar keine entsprechende Anweisung in ihrem Befehlssatz.

Eine Anwendungsmöglichkeit des Goto Befehls, welche von Programmierern toleriert wird, ist die oben gezeigte: Bei Nichterfüllung einer geforderten Bedingung wird zu einem Anker gesprungen, der die Routine abbricht und eine Meldung über den Misserfolg zurückgibt. Dadurch wird der Einsatz von verschachtelten If-Then-Else-Blöcken innerhalb der Routine vermieden und die Lesbarkeit des Quellcodes verbessert.

5.1.3 Das Berechnen der Bildrate für verschiedene Filmformate

Sub infoget (f as folderitem)

*// Diese Subroutine wird direkt nach dem Laden aufgerufen. Sie ist
// zuständig für das Sammeln von Informationen über den geladenen Film.*

dim i,j as integer

dim s as string

dim fps as single

*// Durch den Aufruf der Routine infobasics werden die Daten für Tracks(0,y)
// gesammelt; unter anderem die Spureanzahl und Spurart.
// Diese Routine wird nicht näher vorgestellt.*

infobasics

*// Hier werden die Daten für die einzelnen Spuren gesammelt.
// Tracks(0,5) enthält die Spureanzahl als String.
// Die Routine infoTrack wird ebenfalls nicht weiter erläutert.*

for i=1 to val(Tracks(0,5))

infoTrack(i)

next i

*// Im Folgenden wird versucht, aus der Videospur des Films die Bildrate zu
// ermitteln.
// Tracks(0,5) enthält die Spureanzahl.
// Tracks(0,9) enthält die Anzahl von Videospuren.
// Tracks(0,2) soll die Bildrate erhalten.*

```

// Tracks(i,2) wobei i > 0, enthält den OSType71 der Spur i.
if val(Tracks(0,9)) >= 1 then
  for i=1 to val(Tracks(0,5))
    if tracks(i,2) = "vide" then
      Tracks(0,2)=str(infogetfps(i,"vide"))
      // Der tatsächliche Wert wird in der Routine infogetfps ermittelt.
      // Diese Routine wird nachfolgend erläutert.
      exit
    end if
    if tracks(i,2) = "MPEG" then
      Tracks(0,2)=str(infogetfps(i,"mpeg"))
      // Der Bildratenwert für MPEG Datenströme wird anders bestimmt
      // als der Wert für Bildspuren. Daher wird der Subroutine infogetfps
      // auch die Information über die Art der Spur übergeben.
      exit
    end if
  next
else
  Tracks(0,2)="25"
  // Wenn das Movie keine Videospur erhält, wird der Bildratenwert 25
  // zugewiesen. Tracks(0,2) wird zur Definition der Werte, um welche die
  // einzelnen Spuren verschoben werden können, benutzt. Er kann daher
  // nicht Null oder leer sein.
end if

// Im weiteren (hier nicht gezeigten) Verlauf dieser Routine werden noch
// andere Daten über den Film gesammelt.
End Sub

```

Function infogetfps (index as integer, type as string) as Single

```

//index bezeichnet die Spurnummer.
//type ist entweder „vide“ oder „MPEG“.

```

⁷¹ Der OSType ist eine vierstellige Buchstabenkombination, welche die Art der Spur definiert. „vide“ steht für Bild, „soun“ für Ton, „MPEG“ für einen MPEG Datenstrom, „txt“ für Text (z.B. Untertitel oder Kapitelspur).

```

Dim mediahandle, trackhandle, THEmediahandler, tracksamples, _
    trackdurTS, trackTS as integer
dim fps as single
dim rate,ratesize as memoryblock
// Die folgenden Zeilen definieren den Zugriff auf die Shared Library
// „QuickTime Lib“.
declare function GetMediaSampleCount Lib "QuickTime" (mh as integer) _
    as integer
declare function MoviesTask Lib "QuickTime" (mh as Integer, a as integer) _
    as integer
declare function MediaGetPublicInfo Lib "QuickTime" (mh as integer, _
    infoselector as otype,rate as ptr,ratesize as ptr) as integer
declare function GetMediaHandler Lib "QuickTime" (mh as integer) as integer

// em ist die Repräsentation der Datenstruktur des QuickTime Movies.
// Sein Handle ist die Beschreibung des Speicherortes.
// Der mediahandle beschreibt den Speicherort der Datenstruktur TrackMedia.
trackhandle = GetMovieTrack(em.Handle,val(Tracks(index,1)))
mediahandle = GetTrackMedia(trackhandle)

select case type
case "vide"
    tracksamples = GetMediaSampleCount(mediahandle)
    trackdurTS = em.Track(index).TimeDuration // Dauer in TS-Units
    trackTS = em.Track(index).TimeScale
    fps = tracksamples / ( trackdurTS / trackTS )
    // tracksamples: Anzahl der Samples in der Spur. Ein Sample entspricht dabei
    // einem (Voll-)Bild.
    // trackdurTS: Dauer der Spur in TimeScale Einheiten.
    // trackTS: TimeScale der Spur.

case "MPEG"
    // Ein MPEG Datenstrom besitzt innerhalb seiner Datenstruktur einen als
    // „PublicInfo“ definierten Bereich. Aus diesem kann der Bildratenwert
    // ausgelesen werden.
    rate = newmemoryBlock(4)
    ratesize = newmemoryBlock(4)
    // Memoryblocks sind Bereiche im Speicher. Da die Funktion
    // MediaGetPublicInfo den Bildratenwert nicht als Ergebnis, sondern in

```

```

// einem ihrer Argumente zurückgibt, muss ein solcher Memoryblock
// benutzt werden.
THEmediahandler = GetMediaHandler(mediahandle)
call MoviesTask(em.handle,0)
// Der OSType „orat“ fordert aus der PublicInfo den Bildratenwert an.
call MediaGetPublicInfo(THEmediahandler,"orat",rate,ratesize)
// Der Bildratenwert wird als Float72 in den Memoryblock „rate“
// zurückgegeben und muss zur Weiterverarbeitung um 2 Bytestellen
// verschoben werden („Bit Shifting“).
fps = rate.long(0)/&h10000
// &h10000 bezeichnet die Hexadezimalzahl 10000.
end select

// Der Wert wird auf 3 Nachkommastellen gerundet.
fps = (round(fps*1000))/1000
// Nach erfolgreicher Bestimmung des Wertes wird dieser nun zurückgegeben.
return fps

// Sollte während der Ausführung der Routine ein Fehler aufgetreten sein
// (z.B. während der Aufrufe an die Shared Library) wird auf eine andere
// Methode zurückgegriffen, die allerdings nicht immer zufriedenstellende
// Ergebnisse liefert.
exception
// Eine Mitteilung an den Benutzer
msgbox("Couldn't determine the exact frame rate of the movie."+ _
      endofline+endofline+"The frame rate displayed in the Info window "+_
      "could be wrong.")
// In Tracks(index,3), wobei index > 1, steht die TimeScale Einheit der
// betreffenden Spur. Diese kann unterschiedliche Werte haben. Oftmals
// findet sich hier die hundert- oder tausendfache Bildrate.
fps = val(tracks(index,3))
select case fps
case is >32000
  // *shrug*
case is > 3200
  testfps = fps/1000
case is > 320

```

⁷² Float ist ein Carbon Variablentyp. REALbasic besitzt kein entsprechendes Gegenstück.

```

    testfps = fps/100
case is > 32
    testfps = fps/10
case is > 3
    testfps = fps
end select
// Es wird geprüft, ob durch die Teilung durch ein Mehrfaches von 10 ein
// sinnvolles Ergebnis (zwischen 31 und 22) entstanden ist.
// Sollte dies der Fall sein, wird dieser Wert zurückgegeben,
// ansonsten der Wert 26.
if testfps<31 and testfps>22 then return testfps else return 26

// Der Wert 26 indiziert im weiteren Verlauf des Programms, dass die Bildrate
// nicht eindeutig festzustellen war.
// Es wird der Wert 26 benutzt, da die ermittelte Bildrate zur Definition der
// Schrittgröße, um welche die einzelnen Spuren verschoben werden können,
// gebraucht wird. Er kann daher nicht Null oder leer sein.
// Die Abweichung um 1 von der üblichen Bildrate 25 hat für die spätere
// Funktionalität keine Auswirkung.

End Function

```

5.1.4 Das Ändern des Seitenverhältnisses (Change Ratio...)

Der Benutzer hat mit der Funktion „Change Ratio...“ die Möglichkeit, das Seitenverhältnis des Films zu verändern. Er kann dazu zwischen vorgefertigten Werten auswählen oder eigene eingeben. Dadurch ist es auch möglich, z.B. das Seitenverhältnis beizubehalten und stattdessen die Größe zu ändern.

Auf den Quellcode der Bedienelemente in dem Dialogfenster soll an dieser Stelle verzichtet werden. Stattdessen wird der Code diskutiert, welcher ausgeführt wird, nachdem der Benutzer Werte eingegeben und die Ausführung der Bearbeitung bestätigt hat.

```

Sub Action(index as integer)
// Bei einem Pushbutton handelt es sich um eine Schaltfläche, die angeklickt
// werden kann. Ist dies der Fall, so wird ein Action Ereignis ausgelöst, welches
// die Ausführung der zugehörigen Action Routine bewirkt.

// Die nachstehende Action Routine gehört zu den zwei Pushbuttons, die den
// Benutzer entweder die Ausführung der Seitenverhältnisänderung
// durchführen („OK“) oder abbrechen lassen („Cancel“).
// Die Pushbuttons sind als Array definiert. Index 0 bezeichnet den Cancel
// Knopf, Index 1 den „OK“ Knopf.

dim ws,hs as string
dim w,h as integer

// Es wird festgestellt, welchen Knopf der Benutzer gedrückt hat.
if index=1 then
    // Es wurde OK geklickt. Die Änderung des Seitenverhältnisses wird
    // durchgeführt.

    // Zunächst werden die vom Benutzer eingegebenen Werte aus den
    // Eingabefeldern „newedit“ ausgelesen, welche ebenfalls ein Array sind.
    ws=newedit(1).text
    hs=newedit(2).text
    if ws<>"" and hs<>"" then

        // Die Werte wurden zunächst als String ausgelesen. Im Folgenden wird
        // geprüft, ob sie tatsächlich numerische Werte enthalten.
        if IsNumeric(ws) and IsNumeric(hs) then
            // Falls ja, werden sie in Integer umgewandelt.
            w = val(ws)
            h = val(hs)
            // Es wird überprüft, ob die Werte Vielfache von vier sind.
            if CheckForMultipleOf4(w,h) then
                // Wenn die Funktion „true“ zurückgegeben hat, wird die Änderung
                // durchgeführt.
                // Die Routine „SizeChange“ ist eine Funktion. Sie gibt also ein Ergebnis
                // zurück. Durch die Benutzung von Call wird dies unterdrückt,
                // da eine Rückmeldung hier nicht benötigt wird: Das Dialogfenster
                // würde in jedem Fall geschlossen.
            end if
        end if
    end if
end if

```

```

        call SizeChange(w,h)
        winclose
    end if
else
    // Die Funktion „blink“ lässt den Bildschirm aufblinken. Dies signalisiert,
    // dass die Aktion, welche der Benutzer ausführen will, momentan nicht
    // möglich ist, da noch eine andere Eingabe von ihm erwartet wird.
    // In diesem Fall wird die Eingabe gültiger Zahlen gefordert.
    blink
end if
else
    blink
end if

else
    // Der Benutzer möchte das Ändern des Seitenverhältnisses nicht durchführen
    // („Cancel“). Die Routine „winclose“ schliesst das Fenster.
    winclose
end if
End Sub

```

```

Private Function CheckMultipleOf4(byRef w as integer, byRef h as integer) _
    As Boolean
    // Eine private Funktion steht nur innerhalb jenes Objekts zur Verfügung,
    // in welchem es definiert wurde.
    // Diese Funktion überprüft, ob die Werte w und h Vielfache von 4 sind.
    // Ist dies nicht der Fall, wird der Benutzer auf diesen Umstand hingewiesen.
    // Er hat daraufhin drei Auswahlmöglichkeiten:
    // 1. Er übernimmt die vorgeschlagenen neuen Werte. Dies liefert als Ergebnis
    // „true“ und löst damit in der aufrufenden Routine die Änderung des Seiten-
    // verhältnisses und das Schliessen des Dialogfensters aus.
    // 2. Er besteht auf seinen Werten. (Ergibt ebenfalls „true“)
    // 3. Er bricht ab. Diese Handlung liefert „false“. Die Änderung wird nicht
    // durchgeführt und das Dialogfenster bleibt für eine erneute Eingabe von
    // Werten geöffnet.

```

```

dim ok as boolean
// Neu definierte Booleans besitzen immer den Wert „false“.
dim w1,h1,w2,h2,wrest,hrest as integer
dim ratio as single
dim d as MessageDialog
dim b as MessageDialogButton

wrest = w mod 4
hrest = h mod 4
if wrest = 0 and w <> 0 and hrest = 0 and h <> 0 then
    ok = true
else
    // Der folgende Code soll Werte für Höhe und Breite des Filmes ermitteln,
    // welche durch vier teilbar sind und möglichst nah an denen des Originals
    // bleiben.
    w1 = w
    h1 = h
    if wrest = 0 then
        w2 = w
        h2 = (round(h/4))*4
    else
        if hrest = 0 then
            h2 = h
            w2 = (round(w/4))*4
        else
            ratio = w/h
            h2 = (round(h/4))*4
            w2 = (round((ratio*h2)/4))*4
        end if
    end if

    if h2 = 0 then h2 = 4
    if w2 = 0 then w2 = 4

    // Der Benutzer wird durch einen Dialog auf die nicht passenden Werte
    // hingewiesen.
    d = new messageDialog
    d.icon=1
    d.ActionButton.Caption="Use "+str(w2)+"x"+str(h2)

```

```

d.CancelButton.Visible=True
d.CancelButton.Cancel = False
d.CancelButton.Caption="Use "+str(w1)+"x"+str(h1)+" anyway"
d.AlternateActionButton.Visible=True
d.AlternateActionButton.Caption="Cancel"
d.AlternateActionButton.Cancel=true
d.Message="The values for width and/or height are not multiples of 4."
d.Explanation="Choosing values which are not "+ _
    "multiples of 4 can cause:"+endofline+ _
    "- a performance loss"+endofline+ _
    "- problems when the movie is exported to another "+ _
    "format."+endofline+endofline+ _
    "QT Sync suggests you use "+str(w2)+"x"+str(h2)+" instead."
b=d.ShowModal
// Die Dialogbox wird dargestellt. In der als MessageDialogButton definierten
// Variable b wird die Auswahl des Benutzers zurückgegeben.
Select Case b
Case d.ActionButton
    // Der Benutzer wählt die neuen Werte.
    w=w2
    h= h2
    ok = true
Case d.CancelButton
    // Der Benutzer behält seine alten Werte bei.
    w=w1
    h=h1
    ok = true
else
    // Der Benutzer bricht ab.
    // do nothing.
End select
end if
return ok

End Function

```

Function SizeChange(w as integer,h as integer) as Boolean

*// SizeChange ist eine globale Methode, die auch von anderen Objekten des
// Programms aufrufbar wäre.*

```
dim trackhandle,i,ID as integer
dim w2,h2 as integer
dim wblock,hblock as memoryBlock
dim successful as boolean
successful = false
```

// Es werden Zugriffe auf die Shared Library „QuickTime Lib“ definiert.

```
declare function GetMovieTrack Lib "QuickTime" _
    (mh as integer,id as integer) as integer
declare function SetTrackDimensions Lib "QuickTime" _
    (mh as integer, w as integer, h as integer) as integer
declare function GetTrackDimensions Lib "QuickTime" _
    (mh as integer, w as ptr, h as ptr) as integer
```

*// REALbasic speichert Filme in einem speziellen, null-basierten Array, wobei
// jeder Index für eine Spur steht.
// Zusätzlich besitzt jede Spur eine ID, die von QuickTime vergeben wird.
// Um die Spur an die Befehle der Shared Library zu übergeben, muss diese ID
// benutzt werden.*

ID = 0

*// In der hier benutzten Variable namens ID wird allerdings der Index des
// REALbasic Movie Arrays gespeichert.*

```
for i = 1 to val(Tracks(0,5))
    if Tracks(i,2) = "vide" or Tracks(i,2) = "MPEG" then
        ID = i
        exit
    end if
next
if ID <> 0 then
    trackhandle = GetMovieTrack(em.Handle,val(Tracks(ID,1)))
    // Tracks(i,1), wobei i > 0, enthält die ID der Spur.
```

```

hblock = newMemoryBlock(4)
wblock = newMemoryBlock(4)
call GetTrackDimensions(trackhandle,wblock,hblock)
w2 = wblock.long(0)
h2 = hblock.long(0)
// Zunächst werden noch einmal die tatsächlichen Dimensionen der Spur
// abgefragt. Die Resultate werden in den Argumenten der Methode
// zurückgegeben. Daher müssen Memoryblocks benutzt und diese in
// Integer umgewandelt werden. Die zurückgegebenen Werte sind im
// Float Format. Daraus folgt das Bitshifting in der folgenden Zeile:

if not (((w2/10000)=w) and ((h2/10000)=h)) then
  call SetTrackDimensions(trackhandle,w*10000,h*10000)
  call GetTrackDimensions(trackhandle,wblock,hblock)
  w2 = wblock.long(0)
  h2 = hblock.long(0)
// Die neuen Bildspurdimensionen werden gesetzt und zur Kontrolle
// direkt danach wieder abgerufen.
// Erst wenn die jetzt zurückgegebenen Werte mit den gewünschten
// übereinstimmen, gibt die Funktion eine Erfolgsmeldung zurück.

if ((w2/10000)=w) and ((h2/10000)=h) then
  Tracks(0,7) = str(h)
  Tracks(0,8) = str(w)
// Tracks(0,7) und Tracks(0,8) enthalten die aktuellen Dimensionsdaten.
// Die ursprünglichen Werte, denen der Film beim Öffnen entsprach, sind
// in anderen Variablen zwischengespeichert und werden im weiteren
// Programmablauf noch benötigt.
  mov.size(1)
// Das Abspielfenster wird an die neue Größe angepasst.
  successful = true
else
  msgbox("Sorry. Couldn't complete the operation." + _
    "This is a bug. Please report this.")
end if

else
  successful = true
end if

```

```
else  
    msgbox("No video track to resize.")  
end if  
return successful
```

End Function

Bemerkungen:

Diese Routine verändert nicht das Format der tatsächlichen Bilddaten, sondern die Abspielanweisungen in der QuickTime Movie Datenstruktur. Die Bilddaten werden vom Codec nach wie vor in die alten Dimensionen dekodiert und von QuickTime während der Wiedergabe in die neue Größe skaliert.

Die Überprüfung auf die Teilbarkeit der Werte durch vier wird durchgeführt, um Qualitätseinbußen bei einer eventuellen Neuenkodierung des bearbeiteten Films zu vermeiden: Einige Video-Codecs liefern schlechtere Qualität oder unbrauchbare Resultate bei nicht dieser Bedingung entsprechenden Filmhöhen und -breiten. Die Prüfung der Teilbarkeit durch vier ist theoretisch nicht vollkommen ausreichend. Diverse Codecs verlangen eine Teilbarkeit durch 8 oder 16.

In den Routinen „CheckForMultipleOf4“ und „SizeChange“ wurde nicht mit der „Goto Bail“ Methode⁷³, sondern mit herkömmlichen If-Then-Else Verschachtelungen gearbeitet. Speziell in der gedruckten, kommentierten Darstellung kann es dabei leicht zu Verwechslungen darüber kommen, in welchem Zweig man sich gerade befindet.

⁷³ Vgl. Routine „TryTempfile“ in Kap. 5.1.2

6. Fazit

QuickTime ist nicht nur ein gängiges Format zur Verbreitung von Filmen über digitale Medien, sondern kann auch ein wichtiger Bestandteil in der Tonpostproduktion sein. Fertig geschnittenes Bildmaterial wird häufig als QuickTime Movie angeliefert und in den üblichen Tonschnittprogrammen als Bildzuspiel verwendet. Mit entsprechenden Codecs kann eine Bildqualität erreicht werden, die jener von digitalen Videobändern sehr nahe kommt. Der Vorteil gegenüber einem Bildzuspiel von Band liegt in den schnellen Zugriffszeiten und der besseren Verwaltbarkeit der benötigten Medien: Bild und Ton können auf demselben Datenträger gespeichert werden. Auch das Ausspiel des QuickTime Movies auf externe Monitore oder Videoprojektoren ist mit Hilfe von vergleichsweise kostengünstigen Videokarten möglich. Eine eingehendere Beschäftigung mit diesem Format liegt also nahe, wird jedoch im Alltag eines Tonmeisters selten in die Tat umgesetzt.

Durch die Auseinandersetzung mit der QuickTime Technologie war der Autor in der Lage, nicht nur die ursprünglich angestrebten Funktionen in QT Sync zu implementieren, sondern auch diesbezügliche Bedürfnisse der Tonpostproduktion zu erkennen und eine Erweiterung des Programms um entsprechende Fähigkeiten in Betracht zu ziehen.

Konkretisiert wurde dieser Ansatz nach der Veröffentlichung von QT Sync auf den Downloadplattformen versiontracker.com, macupdate.com sowie videohelp.com⁷⁴. Entsprechende Reaktionen und Vorschläge der Anwender sind seitdem mit großer Freude aufgenommen worden.

Aus einem nächtlichen Experiment hatte sich ein Programm entwickelt, welches eine vom Autor zuvor nicht wahrgenommene Marktlücke auszufüllen begann. Steigende Downloadzahlen und Erwähnungen in weiteren Fachforen zeigten die Akzeptanz des Programms innerhalb der angesprochenen Benutzergruppen.

⁷⁴ videohelp.com widmet sich allen Arten von Problemen, welche im Umgang mit Filmdateien auftreten können. Durch Anleitungen, breit gefächerte Foren und Programmempfehlungen bietet es Benutzern aller Computerplattformen Hilfe beim Abspielen, Konvertieren und Bearbeiten jeglicher Art von Filmdatei.

Positive Kommentare („Simply fantastic!“⁷⁵) sowie konstruktive Kritiken („I suggest a way to "clean" (discard) the TimeCode file if it is not accepted by the user.“⁷⁶) führten zu der Implementierung neuer Funktionen, welche von den Benutzern wiederum dankbar angenommen wurden.

Aus der entstehenden Kommunikation ergaben sich aufschlussreiche Diskussionen und zusätzliche Anregungen. Dieser Prozess des Gedankenaustauschs hat zur Entwicklung einer Applikation beigetragen, welche ihren Platz in der Landschaft der Mac-Freewareprogramme gefunden hat und einigen Kollegen den beruflichen Alltag erleichtern kann.

Interessant zu beobachten ist die Zweiteilung der Anwendergruppe: Der Privatanutzer nimmt anscheinend die Möglichkeiten zur Bildbearbeitung kaum wahr, da er keinen praktischen Nutzen daraus ziehen kann. Die Gruppe der professionellen Anwender hingegen ist nicht primär an einem Abspielprogramm interessiert, sondern nutzt hauptsächlich die Funktionen zur Modifizierung des Bildes.

Dieser Zweiteilung soll auch innerhalb der Strukturierung der Applikation Rechnung getragen werden. Der Autor sieht QT Sync noch nicht als fertiges Produkt an, sondern als einen viel versprechenden Ausgangspunkt mit großem Erweiterungspotential. Die Software befindet sich immer noch im Test-Stadium, wird laufend verbessert und um neue Funktionen ergänzt.

In Planung ist die Auftrennung von QT Sync in zwei Varianten: Die Standard Version würde nach wie vor die Abspielmöglichkeit im Full Screen Modus und die Fähigkeit zur Korrektur von Asynchronitäten beinhalten. Der Schwerpunkt der professionellen Variante läge weniger auf der Wiedergabe eines Films, als auf dessen Bearbeitung (also Änderung des Seitenverhältnisses, TimeCode Insertierung, Formatkonvertierung, etc.).

Für diese Weiterentwicklung von QT Sync wird vor allem der Lernerfolg bezüglich der objektorientierten Programmierung von Nutzen sein. Wählte der Autor anfangs aus Unerfahrenheit häufig den umständlichen Programmierweg, so ist

⁷⁵ „Einfach fantastisch!“ / <http://www.macupdate.com/reviews.php?id=15245>

⁷⁶ „Es sollte eine Möglichkeit geben, die TimeCode Spur [von der Festplatte] zu löschen, wenn sie vom Anwender nicht benutzt wird.“ / aus einer privaten E-Mail an den Autor

heute ein tieferes Verständnis für den prinzipiellen Aufbau einer Applikation innerhalb einer objektorientierten Entwicklungsumgebung vorhanden.

Zurückblickend wird klar, dass viele grundlegende Ansätze in QT Sync anders hätten strukturiert werden müssen. Das Erweitern um neue Funktionen gestaltet sich aufgrund der am Anfang nicht optimal angelegten Objekte mühsamer als notwendig. Eine diesbezügliche Umstrukturierung der Applikation steht jedoch außer Frage, da dies eine fast vollständige Neuprogrammierung bedeuten würde. In weiteren Projekten wird sich der Lernfortschritt allerdings bemerkbar machen.

Neben der reinen Programmierung ist vor allem die Präsentation für den Erfolg eines Programms wichtig.

Für den Internetauftritt von QT Sync wurden Texte (u.a. ein zweisprachiges Handbuch) verfasst und eine Corporate Identity erstellt. Diese beinhaltet Icons, Logos sowie das komplette graphische Erscheinungsbild.

Ergänzend zur objektorientierten Programmierung wurden für die Erstellung der Webseite Kenntnisse in HTML und CSS aufgefrischt. Durch den Einsatz der zusätzlich erlernten Skriptsprache PHP konnten Funktionalität und graphische Darstellung des Internetauftritts verbessert werden.

Die Auseinandersetzung mit Marketingstrategien und unterschiedlichen Distributionskanälen war eher theoretischer Natur, da QT Sync zumindest bis zum jetzigen Zeitpunkt kostenfrei im Internet erhältlich ist. Es konnten dabei aber erste Einblicke in die Abläufe und Gesetze des Shareware Marktes gewonnen werden.

Die dargestellten Erkenntnisse sieht der Autor als eine wertvolle Ergänzung zu dem fachspezifischen Wissen an, welches er in seiner bisherigen akademischen und beruflichen Laufbahn gesammelt hat: Eine Erweiterung des Horizonts über das Fachgebiet des Tonmeisters hinaus eröffnet die zusätzliche Möglichkeit, den eigenen Tätigkeitsbereich aus einem anderen Blickwinkel heraus zu betrachten.

Gerade am Beispiel von QuickTime, welches sowohl den Fachbereichen Bild als auch Ton zugehörig ist, wird erkennbar, dass das Wissen um fächerübergreifende Technologien auf beiden Seiten (also Bild und Ton) vorhanden sein sollte, um den gemeinsamen Arbeitsprozess optimal gestalten zu können.

QT Sync nutzt nur einige der vielen Möglichkeiten von QuickTime und ermöglicht so eine schrittweise Annäherung an dieses Medienformat. Damit trägt die Applikation hoffentlich dazu bei, eventuelles Misstrauen abzubauen und Interesse an der Beschäftigung mit einer Technologie zu wecken, welche eine wertvolle Ergänzung im Bereich der Tonpostproduktion darstellen kann.

7. Anhang

Alle Markennamen, Produktbezeichnungen, Warenzeichen und eingetragenen Warenzeichen sind Eigentum der jeweiligen rechtmäßigen Besitzer.

7.1 Literatur- und Quellenverzeichnis

FIRMA Apple (Hrsg.), 2005: Apple Human Interface Guidelines.

<http://developer.apple.com/documentation/UserExperience/Conceptual/OSXHIGuidelines/OSXHIGuidelines.pdf>

FIRMA APPLE (Hrsg.), 2002: Introduction to Carbon.

<http://developer.apple.com/documentation/Carbon/Conceptual/newtocarbon/ch01.html>

FIRMA APPLE (Hrsg.), 2004: QuickTime Overview.

<http://developer.apple.com/documentation/QuickTime/RM/Fundamentals/QTOverview/QTOverview.pdf>

FIRMA APPLE (Hrsg.), 2005: What Is Cocoa ?

<http://developer.apple.com/documentation/Cocoa/Conceptual/CocoaOverview/index.html>

FIRMA REALSOFTWARE (Hrsg.): REALbasic Quick Start.

<http://highspeed.realsoftware.com/REALbasic555/REALbasicQuickStart.sit>

GLOSSAR.DE (Hrsg.): AVI – Audio Video Interleave.

http://www.glossar.de/glossar/z_avi.htm

WIKIPEDIA.ORG (Hrsg.): AVI. <http://en.wikipedia.org/wiki/AVI>

WIKIPEDIA.ORG (Hrsg.): List Of Codecs.

http://en.wikipedia.org/wiki/List_of_codecs

WIKIPEDIA.ORG (Hrsg.): Matroska. <http://de.wikipedia.org/wiki/Matroska>

WIKIPEDIA.ORG (Hrsg.): Microsoft Windows Media.

http://de.wikipedia.org/wiki/Microsoft_Windows_Media

WIKIPEDIA.ORG (Hrsg.): MP3. <http://en.wikipedia.org/wiki/MP3>

WIKIPEDIA.ORG (Hrsg.): MP4. <http://de.wikipedia.org/wiki/MP4>

WIKIPEDIA.ORG (Hrsg.): Ogg. <http://de.wikipedia.org/wiki/Ogg>

WIKIPEDIA.ORG (Hrsg.): QuickTime. <http://de.wikipedia.org/wiki/QuickTime>

WIKIPEDIA.ORG (Hrsg.): Real Player. http://de.wikipedia.org/wiki/Real_Player

WIKIPEDIA.ORG (Hrsg.): Windows Media Player.

http://de.wikipedia.org/wiki/Windows_Media_Player

YOSHIDA, J., 2004: Synching Lucy's lips with TV sound. EE Times, 19.4.2004.

<http://www.eetimes.com/showArticle.jhtml?articleID=18901792>

Stand aller genannten Weblinks: Mai 2005

7.2 *Abbildungsverzeichnis*

Abbildung 2.1: QuickTime API – Tool Sets und Komponenten	12
Abbildung 2.2: Das Zusammenspiel der QuickTime Komponenten während des Abspielen eines Films.....	13
Abbildung 2.3: Die verschiedenen Arten von QuickTime Filmdateien	15
Abbildung 3.1: Die Bedienoberfläche des „QT Mutator“	28
Abbildung 3.2: Die Bedienoberfläche des Programms „Synchole“	30
Abbildung 3.3: Das Abspielfenster und das Informationsfenster des QuickTime Players (QuickTime Version 6.5.2)	32
Abbildung 4.1: Die Bedienoberfläche von „QT Sync“ nach dem Laden eines Films	47
Abbildung 4.2: Informationsfenster – Spalte „General“	50
Abbildung 4.3: Informationsfenster – Spalte „Tracks“	50
Abbildung 4.4: Der „QT Sync“ Controller	52
Abbildung 4.5: Der aufgeklappte „QT Sync“ Controller	56
Abbildung 4.6: Der aufgeklappte „QT Sync“ Controller im Fill-Screen-Modus.....	58
Abbildung 4.7: Das „Change Ratio...“ Fenster	59
Abbildung 4.8: Das „Crop Movie“ Fenster mit geöffneter Vorschau	61
Abbildung 4.9: Das „Import Track“ Fenster	63
Abbildung 4.10: Das „Remove Track“ Fenster	64
Abbildung 4.11: Das Fenster des „Combine Movies“ Assistenten (erste Seite).....	65
Abbildung 4.12: Das Fenster des TimeCode Inserters im TimeCode Modus	67

Abbildung 4.13: Das Fenster des TimeCode Inserters im Feet + Frames Modus (35mm / 4perf)	67
Abbildung 4.14: Das „Check for Updates...“ Fenster	70
Abbildung 4.15: Voreinstellungsfenster (Spalte „General“)	77
Abbildung 4.16: Voreinstellungsfenster (Spalte „Controls“)	77
Abbildung 4.17: Voreinstellungsfenster (Spalte "Full Screen").....	77

In den Darstellungen der verschiedenen Programme wurden Standbilder folgender Filme verwendet: „Wild Postings“ und „Breakdance“. (Amerikanische TV Werbespots für Apples iPod.) Beide waren im Sommer 2004 unter <http://www.apple.com/ipod/ads/> zum freien Download verfügbar, sind inzwischen aber durch andere Werbefilme ersetzt worden.

7.3 Danksagung

Der Autor möchte folgenden Personen seinen herzlichen Dank für ihre Unterstützung aussprechen:

Prof. Ingo Kock, Dipl.-Ing. (FH) Robert Gräf, Dipl.-Ing. (FH) Christoph Engelke, Marie-Anne von Busse, Karl-Heinz und Ursula Gerbes sowie allen Benutzern von QT Sync, die durch ihre Kommentare und Anregungen zur Weiterentwicklung und Verbesserung des Programms beigetragen haben.

7.4 Selbständigkeitserklärung

Hiermit erkläre ich, die vorliegende Arbeit selbständig und nur unter Verwendung der angegebenen Quellen verfasst zu haben.

Potsdam-Babelsberg, 4. Juli 2005