

Inhaltsverzeichnis

1	Live-Streaming – Eine vielschichtige Problemstellung	1
2	Überblick über die Media Internet Streaming Toolbox	5
2.1	Entwurf von Streaming-Anwendungen	5
2.2	Konzepte und Funktionsweise	6
2.3	Komponenten der TOOLBOX	11
2.3.1	MIST-Kerndienste	13
2.3.2	Compresslet-Bibliotheken	14
2.3.3	Videogewinnung	15
2.3.4	Java WebVideo-Klient	16
2.4	Fallstudie eines Streaming-Szenarios	17
3	Video als Perzeptionsmedium	25
3.1	Das menschliche visuelle Wahrnehmungssystem	25
3.2	Repräsentation von Videosignalen	28
3.2.1	Mathematische Repräsentation	28
3.2.2	Digitale Repräsentation	29
3.2.3	Farbräume	29
3.3	Grundlagen der Videokompression	31
3.3.1	Datenreduktion	32
3.3.2	Dekorrelation	32
3.3.3	Codierung	33
3.3.4	Bewertung der Bildqualität	35
3.4	Verteilung von Videoströmen über das Internet	36
3.4.1	Das Transmission Control Protocol (TCP)	37
3.4.2	Das Real-time Transport Protocol (RTP)	40
3.4.3	IP-Multicast	42
3.4.4	Eignung von TCP zur Verteilung von Videoströmen	43
3.5	Aktuelle Live-Streaming Technologien	45
3.5.1	Multimedia-Frameworks	45
3.5.2	Standards zur Videocodierung	47
3.5.3	Streaming-Server	48
4	Konzepte und Algorithmen	51
4.1	Formalisierung des graphenbasierten Ansatzes	51
4.1.1	Charakterisierung des Konzeptes	51
4.1.2	Medienformate	52
4.1.3	Pins	57
4.1.4	Knotentypen	57

4.1.5	Modellierung von Formatabhängigkeiten	59
4.1.6	Beschreibung von Medienverarbeitung und Stromerstellung durch Anwendungsgraphen	61
4.1.7	Formatvorgaben	64
4.1.8	Flussgraphen	66
4.2	Konstruktion des Flussgraphen durch Formatverhandlung	67
4.2.1	Verhandlungsgraphen	68
4.2.2	Charakterisierung des Problems	70
4.2.3	Konstruktion des Flussgraphen in NMM	72
4.2.4	Plausibilitätsbetrachtungen zur Aufwandsabschätzung	74
4.2.5	Charakterisierung der Lösungsstrategie	76
4.2.6	Heuristik zur Lösung des Steinerbaum-Problems	78
4.2.7	Betrachtungen zur verwendeten Metrik	80
4.2.8	Konstruktion des Flussgraphen	82
4.2.9	Auflösung von Wildcard-Parametern	84
4.2.10	Optimierte Formatverhandlung	85
4.2.11	Validierung und Aktivierung	87
4.3	Component Encoding Stream Construction	89
4.3.1	Skalierbarkeitsbegriff	89
4.3.2	Ansatz der CESC-Architektur	91
4.3.3	Nutzung von Synergien	93
4.3.4	Nutzung temporaler Redundanz	93
4.3.5	Änderungsdetektion	98
4.3.6	Entkopplung von Verarbeitung und Versand	100
4.3.7	Übertragung der Entkopplung auf das graphenbasierte Konzept	102
4.3.8	Gruppierung von Frameanforderungen	104
4.3.9	Abschätzung des Berechnungsaufwandes für die verschiedenen Ansätze	105
4.4	Compresslet-Konzept	108
4.4.1	Compresslets in der Media Internet Streaming Toolbox	109
4.4.2	Klientengesteuerte Stromkonstruktion	111
5	Architektur und Design	113
5.1	Das Ein-/Ausgabe-Subsystem	113
5.1.1	Die Socket-API	114
5.1.2	Das Acceptor-Connector Entwurfsmuster	116
5.1.3	Designalternativen für die gleichzeitige Bedienung mehrerer Verbin- dungen	118
5.1.4	Das Reactor-Framework	122
5.2	ADAPTIVE Communication Environment (ACE)	124
5.3	Die Compresslet-Schnittstelle	126
5.3.1	Der Compresslet-Lebenszyklus	128

5.3.2	Kommunikation der Compresslets mit der TOOLBOX	130
5.3.3	Abbildung von Verbindungsanfragen auf die Senken	133
5.3.4	Austausch von Mediendaten über die Klasse <code>ComponentBuffer</code>	134
5.4	Ausführung von Flussgraphen	137
5.4.1	Das Entwurfsmuster <i>Active Objekt</i>	138
5.4.2	Nebenläufige Ausführung auf Knotenebene	140
5.4.3	Nebenläufige Ausführung auf Flussgraphenebene	141
5.4.4	Initiieren der Flussgraphen-Ausführung	142
5.5	Die Anwendungsschnittstelle	144
5.6	Einbindung von Java-Compresslets	145
5.6.1	Java und das Java Native Interface	146
5.6.2	Simplified Wrapper and Interface Generator (SWIG)	147
5.6.3	Der Java Compresslet Proxy	149
5.7	Video-Gewinnung mit der <i>avcap</i> -Bibliothek	149
5.7.1	Videogewinnung unter GNU/Linux	150
5.7.2	Videogewinnung unter Windows	152
5.7.3	Videogewinnung unter Mac OS X	152
5.7.4	Die Capture-API der <i>avcap</i> -Bibliothek	152
6	Anwendungsbeispiele	155
6.1	Beispiel DeltaWCV: Ein waveletbasiertes Stromformat für CESC	155
6.1.1	Blockweise Aktualisierung der Wavelet-Transformation	156
6.1.2	Entropiecodierung	158
6.1.3	Übertragung des Verfahrens in die TOOLBOX	160
6.2	Beispiel MotionWCV: Ein waveletbasiertes Stromformat mit klientenindividueller Bewegungsschätzung	162
6.2.1	Entwurf des Stromformates	162
6.2.2	Bewegungsschätzung	164
6.2.3	Codierung der Bewegungsvektoren	168
6.2.4	Skalierbarkeit des Verfahrens	169
6.3	Anwendungsbeispiel: Visualisierung der Bewegungsschätzung	170
6.4	Anwendungsbeispiel: Übersichtsvideo und Auswahl einer Quelle	171
6.5	Weitere Beispiele	173
7	Evaluation	175
7.1	Ausführungszeit zur Konstruktion des Flussgraphen	175
7.2	Vergleich der Stromformate DeltaWCV und MotionWCV	176
7.3	Skalierbarkeit bezüglich der Zahl bedienbarer Klienten	179
7.3.1	Versuchsaufbau zur Bestimmung der maximalen Framerate	179
7.3.2	Skalierbarkeit des Stromformates JPEG-ServerPush	181
7.3.3	Skalierbarkeit des Stromformates DeltaWCV	181

7.3.4	Skalierbarkeit des Stromformates MotionWCV	184
7.3.5	Bewertung der Ergebnisse	185
8	Zusammenfassung	187
A	Compresslet-Referenz	193
A.1	VidCapSource	193
A.2	WCVFileSource	195
A.3	RAWFileSource	195
A.4	HTTPSsource	196
A.5	StaticImageSource	197
A.6	JVideoRenderer	197
A.7	XVideoRenderer	198
A.8	HTTPSink	198
A.9	FileSink	199
A.10	DropSink	200
A.11	TextScroller	200
A.12	DateTimeRenderer	201
A.13	Flip	202
A.14	NullFilter	202
A.15	ClientChangeDetector	203
A.16	ClientMotionEstimator	204
A.17	FrameCompareMux	205
A.18	PIPMux	206
A.19	MultiPIPMux	206
A.20	MultiSwitch	208
A.21	ME_Demonstrator	209
A.22	ColorspaceConverter	210
A.23	ForwardDWT	211
A.24	InverseDWT	212
A.25	ExplicitDWTQuantizer	212
A.26	NEWDREncoder	213
A.27	NEWDRDecoder	214
A.28	ForwardDCT	214
A.29	DCTQuantizer	215
A.30	JpegEncoder	215
A.31	Scaler	215
A.32	MotionCompensatorClet	216