

---

# Der Regensburger Marathon-Cluster

- Informationswissenschaftliche Betrachtung -

Hubert Feyrer <hubert@feyrer.de>

Juli 2002



# Marathon Cluster - Projektziel

- **Einstellen der Zieleinläufe des Regensburger Sports Experts Marathons ins Internet:**
  - **Zieleinlaufs-Bild**
  - **Zieleinlaufs-Video**
- **Zerlegen von 6 Stunden Filmmaterial in Einzelbilder**
- **Erstellen der einzelnen Bilder und Filme**
- **Für jeden der 5500 Läufer**



# Abgrenzung

- **Informatik:**

**Cluster Computing, Parametrisierung des Prozesses, Lastverteilung auf die einzelnen Knoten, Deployment der Software auf die Clients, technische Realisierung des Datenaustausches mittels zentralem File Storage System**

- **Informationswissenschaften:**

**Video Retrieval, Image Retrieval, Audio Retrieval, Indexierung & Durchsuchen**



## Zustandekommen des Projektes

- **R-KOM als Sponsor des Sports Experts Marathon**
- **Rechenleistung vor Ort reicht nicht aus**
- **Unterstützung durch FH Regensburg:**
  - **Bereitstellen von Rechnern und Personal**
  - **Installation und Management der Clients**
  - **Bereitstellen von File Storage**
  - **Optimieren und Parametrisieren der Software**

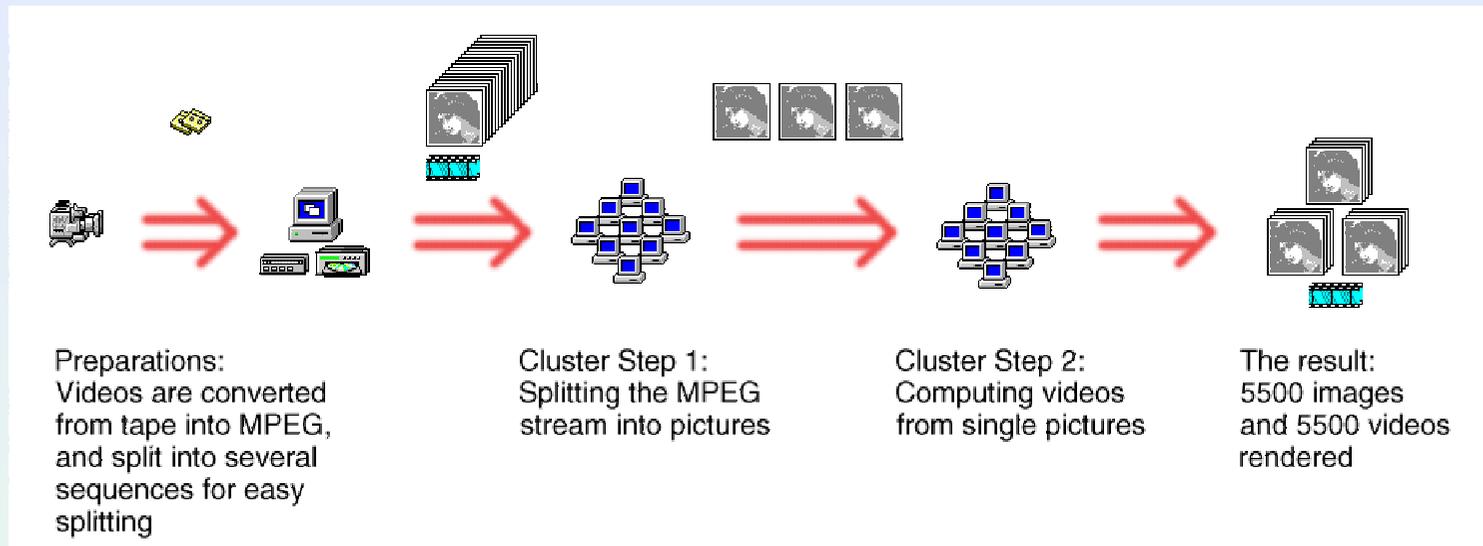


# Marathon Cluster - Hard- & Software (I)

- **45 frei installierbare PCs: 30\*400Hz, 15\*1GHz**
- **1 Control-Rechner, 1 File Server**
- **Betriebssystem: 100% Unix (NetBSD, Linux, Solaris)**
  
- **Anwendungssoftware 100% Open Source**
- **dumpmpeg: MPEG->JPEGs**
- **mpeg\_encode: JPEGs->MPEG**
- **ImageMagick: Abspann-Titel und Zieleinlaufvideo**
- **Cluster control: selbstgeschriebene perl-Scripte**



# Marathon Cluster - Projektablauf & Schritte



- **Segmentierung - Zerlegung des Eingabedatenstroms**
- **Selektion - Auswählen „interessanter“ Daten**
- **Synthese - Wieder-Zusammenfügen zum Ergebnis**
- **Indexierung und Durchsuchbar-Machen**



# Marathon Cluster - Vorarbeit

- **Zwei Videokameras liefern insgesamt ca. 6 Stunden Videomaterial (mit Overlap)**
- **Digitalisieren der Bänder**
- **Aufteilen in MPEG-Sequenzen zu je ca. 11 Minuten**
- **Brennen der MPEG-Sequenzen auf CD**



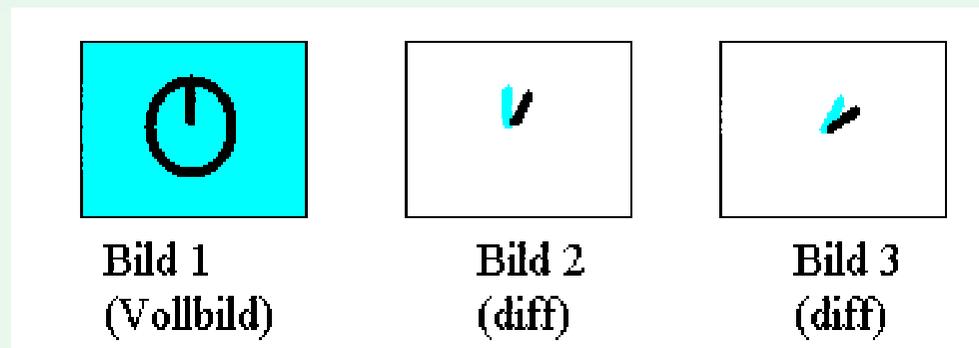
# Marathon Cluster - Erster Rechenschritt

- **Zerlegen der MPEG-Sequenzen in JPEG-Bilder**
- **Transport der MPEG-Sequenzen via CD**
- **Berechnung mittels dumpmpeg und 1GHz Maschinen**
- **Länge Video-Sequenz: 11 Minuten**
- **Dauer Zerlegungsprozess: 45 Minuten**
- **Anzahl JPEG-Bilder: ca. 16.500**
- **Ablegen der JPEG-Bilder auf zentralem Fileserver**



# Abstrakte Schritte - Segmentierung (I)

- **Marathon Cluster:**  
**Zerlegung eines MPEG-Streams in einzelne JPEGs**
- **Bei uncodierten/unkomprimierten Eingabedaten rein zeit-/volumenbasierte Segmentierung möglich**
- **Dekompression nötig für MPEG:**



## Abstrakte Schritte - Segmentierung (II)

- **Viele Ein- und Ausgabeformate denkbar**
- **Bild- , Video- und Audio-Daten**
- **Auswahl abhängig von Anwendung und Umfeld**
- **Generell: Aufsplitten eines Datenstroms in kleine, verarbeitbare Einheiten**



## Marathon Cluster - Zwischenschritt

- **Notieren der genauen Start- und Endzeit jeder MPEG-Sequenz**
- **Hilfe durch Digitaluhr im Blick auf das Ziel**
- **Nötig aufgrund von Schankungen der Bildfolge**
- **Es sind keine 25 Bilder pro Sekunde garantiert**
- **Ablegen der Start-/Endzeiten in MySQL-Datenbank, für zweiten Rechenschritt**



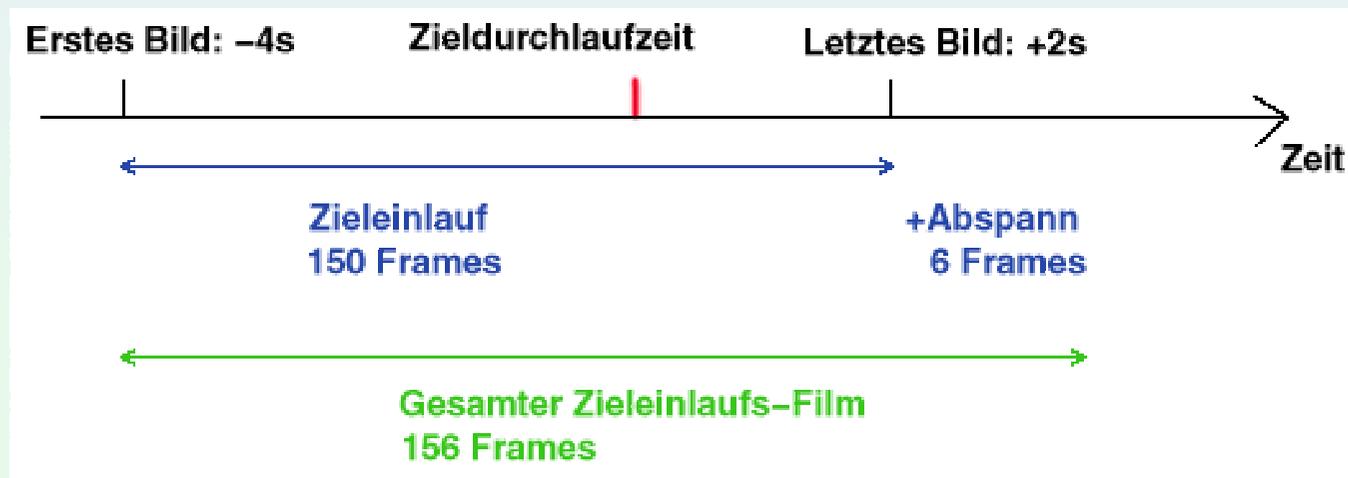
## Marathon Cluster - Zweiter Schritt (I)

- **Berechnen der Zieleinlaufs-Videos und -Bilder**
- **ca. 7000 Läufer, davon 5500 durch's Ziel**
- **Disziplin:**
  - **Marathon (42km)**
  - **Halbmarathon (21km)**
  - **Speedskating (21km)**
- **Getrennte Ergebnislisten für Damen und Herren und pro Disziplin**



## Marathon Cluster - Zweiter Schritt (II)

- **Zieldurchlaufzeit aus Ergebnisliste**
- **Ermitteln des Zieldurchlauf-Bildes mit Datenbank**
- **Ermitteln der Bilder des Zieleinlaufs-Videos:**



## Abstrakte Schritte - Selektion (I)

- **Auswahl „interessanter“ Daten**
- **Marathon Cluster: Zieldurchlaufzeit bestimmt Bilder**
- **Andere Methoden denkbar:**
  - **Gesichtserkennung mit Künstlicher Intelligenz**
  - **Erkennen von Bildelementen durch Vergleich**
  - **Erkennen von Bewegungsabläufen**
  - **Szenenerkennung**
  - **Überlagern mit weiteren, nicht-bildlichen Daten**



## Abstrakte Schritte - Selektion (II)

- **Für nicht-bildliche Verfahren gelten andere Methoden:**
  - **Frequenz- und Spektrumsanalyse bei Audiodaten**
  - **Szenenerkennung bei Videos**
  - **Allgemein: Filtern von Daten**



# Marathon Cluster - Zweiter Schritt (III)

- Eintragen der Läufer-Daten für den Abspann:



- Eintragen der Daten aus der Datenbank
- Automatisiert via perl & ImageMagick

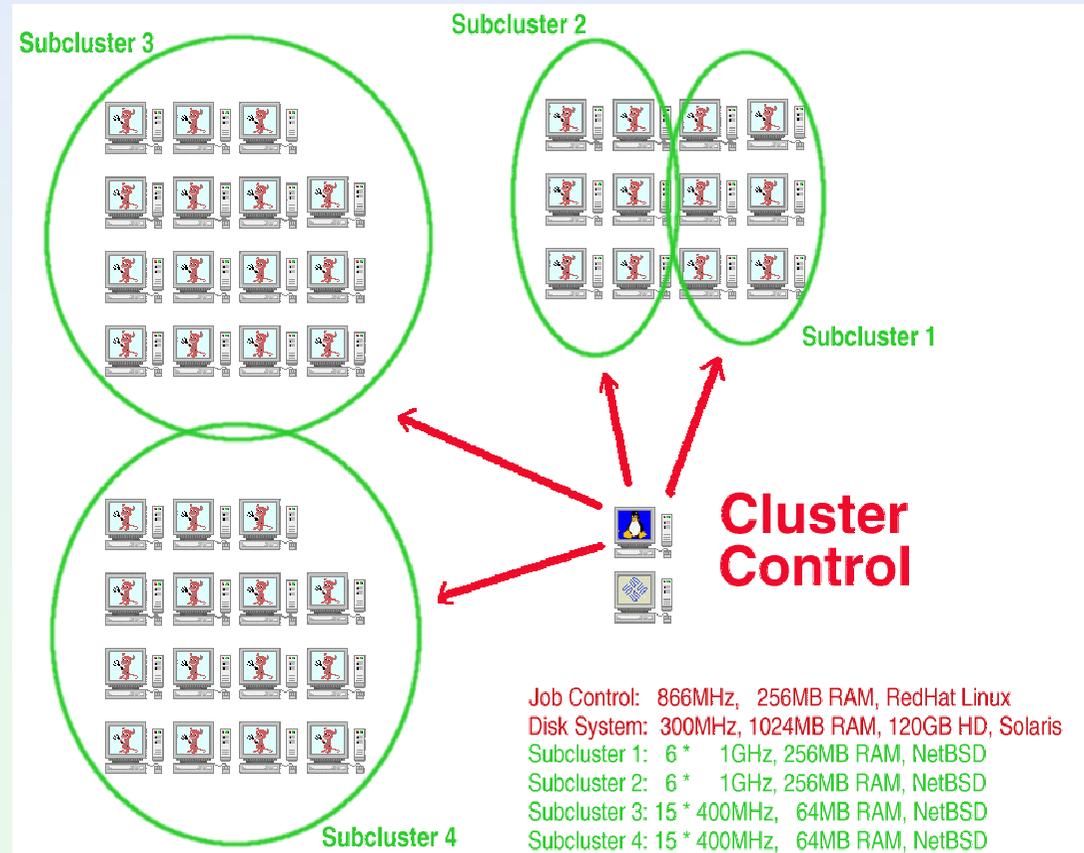


## Marathon Cluster - Zweiter Schritt (IV)

- **übergabe der Bilder an mpeg\_encode**
- **Bilder werden vom zentralem Fileserver gelesen**
- **Ergebnis-Video wird auf zentralem Fileserver abgelegt**
- **Automatisches Load-Balancing**
- **mpeg\_encode kann bei wenigen (156) Eingabe-Bilder die Rechenlast nicht auf beliebig viele Knoten verteilen**
- **Aufteilen der Rechner in mehrere Subcluster**
- **Manuelle Lastverteilung auf die einzelnen Subcluster**



# Marathon Cluster - Zweiter Schritt (V)



## Abstrakte Schritte - Synthese

- **Überführen in streambasiertes Format**
- **Marathon-Cluster: Errechnen von MPEGs aus JPEGs**
- **Sinnvoll bei weiterer Verarbeitung oder Transport**
- **Hoher Rechenaufwand für komprimiertes/codiertes Zielformat**
- **Alternativ direktes blitting in Framebuffer denkbar, um Ergebnis sofort anzusehen**



# Marathon Cluster - Zweiter Schritt (VI)

- **Zielfoto:**



- **Eintragen der Daten aus der Datenbank**
- **Automatisiert via perl & ImageMagick**



# Marathon Cluster - Ergebnis

- **5500 Zieleinlaufs-Videos**
- **5500 Zieleinlaufs-Bilder**
- **Webseite zum Abfragen Bilder und Videos:**
  - **Eingabe der Startnummer**
  - **Auswahl von Bild oder Video**



# Indexieren & Durchsuchen

- **Aufbereiten der Inhalte zum Durchsuchen**
- **Marathon-Cluster:**  
**Suchen nach Startnummer, Auswahl Video oder Bild**
- **Abfrage über HTML-Formular**
- **Direktzugriff über Dateisystem als Ordnungssystem**
- **Startnummer als Dateiname:**  
**startnummer.jpg, startnummer.mpg**
- **Unterschiedliche Größenordnung zwischen indexierten und Selektierten Daten (Sekunden vs. -Bruchteile)**



# Weitere Anwendung der Abstrakten Schritte

- **Marathon-Zieleinlauf**
- **Rad- und Autorennen**
- **Werbepausen im Fernsehen entfernen**
- **Augmented Reality**
- **„Knackser“ aus Schallplatten-Aufnahmen entfernen**
- **Voice over IP**



## **Anwendung: Rad- und Autorennen**

- **Bilder und Videos vom Zieleinlauf**
- **Andere Auswahlkriterien: Zielzeit, Rundenzeit, bestimmte Waypoints**
- **Segmentierung und Synthese basierend auf Bild- und Video-Retrieval Methoden**
- **Bild- und Video-Formate anhängig von Anwendung**
- **Höhere Bild“auflösung“ als beim Marathon - 25 Bilder/Sekunden reichen nicht, auch genaue(re) Zeitmessung**
- **Indexierung: Name oder Startnummer**



# Anwendung: Werbepausen im TV entfernen

- **Segmentierung und Synthese wie gehabt**
- **Problem: Selektion - Was ist eine Werbepause**
  - **Bild/Videoerkennung**
  - **Weitere Daten hinzunehmen - Ton**
  - **Vergleich mit Spots aus Datenbank**
- **Desweiteren: Akzeptanzproblem, Werbungs-Ersatz**
- **Durchsuchen wenig sinnvoll**
- **Weitere Anwendung: Kindersicherung**

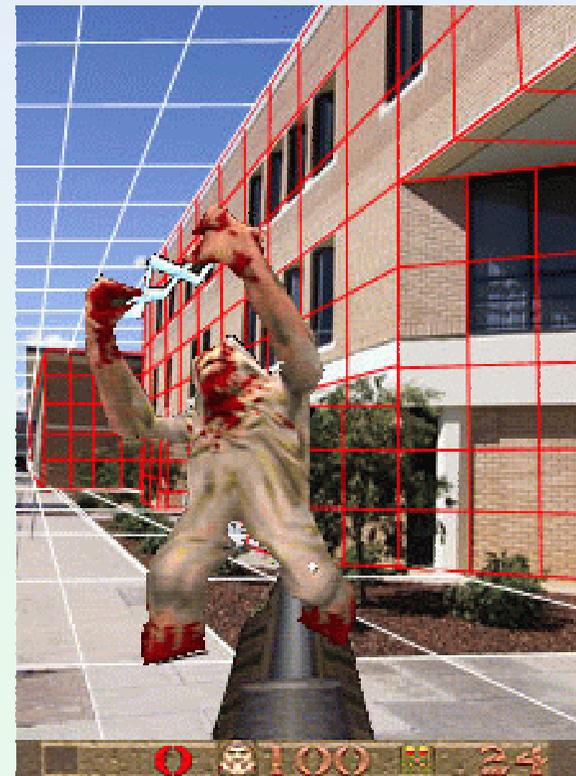


# Anwendung: Augmented Reality (I)

- **Segmentierung via Framegrabber, Synthese via Framegrabber (kein digitaler Datenstrom als Zwischenformat)**
- **Selektion: Es werden alle Daten weitergereicht**
- **Ergänzung um weitere Informationen („Augmentierung“):**
  - **Verbesserte Bildqualität**
  - **Einzeichnen von Gebäude-Elementen**
  - **Kennzeichnen von Personen**



# Anwendung: Augmented Reality (II)



## Anwendung: Augmented Reality (III)

- **Ergänzung basiert auf**
  - **Bildlichen Daten (Muster, Textur, Kontur)**
  - **Standort-Koordinaten (GPS, Kreiselkompass, ...)**
- **Überlagerung der Daten aus beiden Datenquellen für optimales Ergebnis**
- **Durchsuchen: Kernpunkt der Anwendung, Ein/Ausblenden von Klasse von Gegenständen/Daten (Personen, Raum, Bildverbesserung, ...)**
- **Beispielanwendung: Rettungseinsätze**



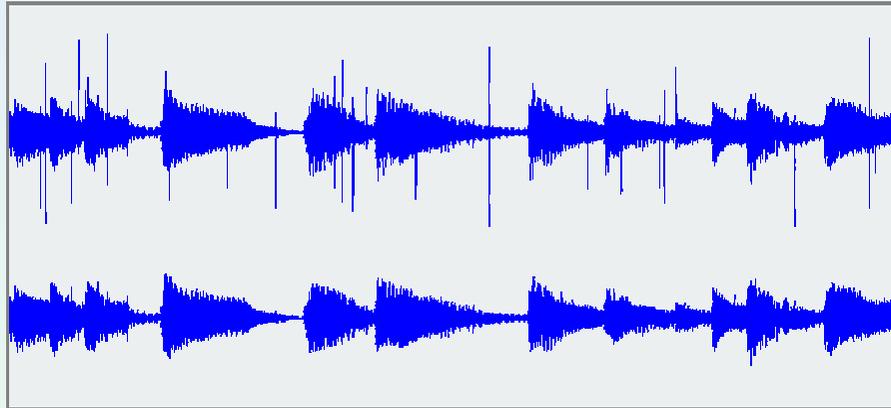
## Anwendung: Knackser entfernen (I)

- **Schema ist auch auf Klang-Daten übertragbar**
- **Segmentierung: zeitbasiert,  
bei CD-Qualität 44.100 Samples a 16bit, Stereo**
- **Abhängig vom Eingabeformat, Entkomprimierung  
Decodierung (mp3)**
- **Selektion: algorithmische Erkennung via Fourier Ana-  
lyse des Frequenzspektrums**



## Anwendung: Knackser entfernen (II)

- **Knackser sind optisch leicht erkennbar:**



- **Können ersatzlos entfernt werden**
- **Weitere Datenselektion durch Filter: Hochpass, Tiefpass**
- **Problematisch: Noise > Signal**



## Anwendung: Knackser entfernen (III)

- **Synthese: Zurückführen der Audiosamples in Audio-Stream, codiert (mp3) oder uncodiert (WAV)**
- **Durchsuchen nicht sinnvoll, evtl. bei Spracherkennung**



## Anwendung: Voice over IP (I)

- **Segmentierung durch Analog-Digital-Wandlung**
- **Selektion:**
  - **Filterung: Hochpass, Tiefpass**
  - **Bandpass: Analog 300Hz-3.4kHz, ISDN: 0-120kHz**
  - **Telefonie-Verfahren: Mu-LAW, A-LAW**
- **Chunking der Daten für Übertragung in paketorientierten Netzen (TCP/IP, GPRS/i-Mode, UMTS)**
- **Übertragung inkl. Fehlersicherung (Retransmit, Erkennung von doppelten Paketen, richtige Reihenfolge)**



## Anwendung: Voice over IP (II)

- **Synthese:**
  - **Zusammensetzung der Datenpakete am Endgerät**
  - **Digital-Analog-Wandlung im Handy-Lautsprecher**
- **Durchsuchen: Wenig sinnvoll; evtl. in Zusammenarbeit mit Spracherkennungssystemen, z.B. Echelon**



## Fazit

- **Das vorgestellte Projekt bietet interessante Aspekte sowohl aus dem Bereich der Informatik als auch der Informationswissenschaften**
- **Die beschriebenen Schritte zur Segmentierung, Selektion und Synthese von streambasierten Daten sind auf einer breiten Palette von Anwendungen implementierbar**
- **Der diesjährige Stadtmarathon lief ohne R-KOM und FH Regensburg - vielleicht nächstes Jahr wieder!**



**Danke!**

**Weitere Infos:**

**<http://www.feyrer.de/marathon-cluster/>**

**Fragen?**



## Inhalt

Marathon Cluster - Projektziel	2
Abgrenzung	3
Zustandekommen des Projektes	4
Marathon Cluster - Hard- & Software (I)	5
Marathon Cluster - Projektablauf & Schritte	6
Marathon Cluster - Vorarbeit	7
Marathon Cluster - Erster Rechenschritt	8
Abstrakte Schritte - Segmentierung (I)	9
Abstrakte Schritte - Segmentierung (II)	10
Marathon Cluster - Zwischenschritt	11
Marathon Cluster - Zweiter Schritt (I)	12
Marathon Cluster - Zweiter Schritt (II)	13
Abstrakte Schritte - Selektion (I)	14
Abstrakte Schritte - Selektion (II)	15
Marathon Cluster - Zweiter Schritt (III)	16
Marathon Cluster - Zweiter Schritt (IV)	17
Marathon Cluster - Zweiter Schritt (V)	18
Abstrakte Schritte - Synthese	19
Marathon Cluster - Zweiter Schritt (VI)	20
Marathon Cluster - Ergebnis	21
Indexieren & Durchsuchen	22
Weitere Anwendung der Abstrakten Schritte	23
Anwendung: Rad- und Autorennen	24
Anwendung: Werbepausen im TV entfernen	25
Anwendung: Augmented Reality (I)	26
Anwendung: Augmented Reality (II)	27
Anwendung: Augmented Reality (III)	28
Anwendung: Knackser entfernen (I)	29
Anwendung: Knackser entfernen (II)	30
Anwendung: Knackser entfernen (III)	31
Anwendung: Voice over IP (I)	32
Anwendung: Voice over IP (II)	33
Fazit	34

