



QUALITÄTSSICHERUNG

Multimediatool zum Testen von Videoencodern

Johannes Werner, Noel Schuhmacher, Sascha Rapp,
Simon Grafenhorst, Carina Weber, Jan Benedikt Schwarz

19. März 2016

Inhaltsverzeichnis

1	Einleitung	1
2	Bugs	2
2.1	PSNRDifferenceCalculator bug	2
2.2	move Filter bug	2
2.3	VideoLoader bug	2
2.4	VideoSaver bug	2
2.5	Bug in mementos	2
2.6	Loading bug	2
2.7	FPS bug	2
2.8	Filter bug	2
2.9	Broken video file bug	3
2.10	Project loading bug	3
2.11	Video saving bug	3
2.12	Project loading bug 2	3
2.13	Filterlist undo bug	3
3	Testüberdeckung	4
4	Zusätzliche Szenarien und Testfälle	4
4.1	CrazyMonkeyTest	4
4.2	AddFilter ohne Video	5

1 Einleitung

In der Qualitätssicherungsphase geht es darum die Qualität des Projectes durch intensives testen zu prüfen. Dazu werden sowohl Unit-Tests als auch Manuelles Testen verschiedener Szenarien sowie Testen durch dritte Personen hinzugezogen.

2 Bugs In diesem Kapitel werden gefundene Bugs und deren Reparatur dokumentiert.

3 Testüberdeckung Dieses Kapitel gibt einen kurzen Einblick in die Testüberdeckung des Codes.

4 Zusätzliche Szenarien und Testfälle In diesem Kapitel werden zusätzliche Szenarien die wir zum testen benutzt haben.

2 Bugs

2.1 PSNRDifferenceCalculator bug

Symptome: Falsche Anzeigen bei den PSNR Graphen.

Ursache: Fehlende if Bedingung ob es sich um ähnliche Videos handelt.

Behebung: Hinzufügen einer Überprüfung ob die Videos die selbe Auflösung haben.

2.2 move Filter bug

Symptome: Fehlermeldung beim verschieben der Filter in der Filterliste.

Ursache: Falsche newposition in den if cases der moveFilter Methode.

Behebung: Ändere filters_size() zu filters_size()-1.

2.3 VideoLoader bug

Symptome: Encodierte Videos werden nicht korrekt geladen.

Ursache: Geladene Videos wurden nicht zu RGB24 umgewandelt.

Behebung: Umwandeln von yuv420 zu rgb24.

2.4 VideoSaver bug

Symptome: Alle abgespeicherten RGB Histogrammvideos haben die selbe Farbe.

Ursache: Falsches setzen der Farbe, in denen die Histogramme gerendert werden.

Behebung: Korrektes setzen der Farben.

2.5 Bug in mementos

Symptome: Comment wird nicht mit abgespeichert.

Ursache: Comment wird nicht in den String, der zum Speichern benutzt wird geschrieben.

Behebung: Comment wird mit memento->getComment() an den String angehängt.

2.6 Loading bug

Symptome: Im Analysistab werden beim Laden eines Projektes manche Videos nicht geladen (inkonsistent).

Ursache: Entscheidung, ob ein Konverter auf das vollständig geladene Video wartet hängt vom Timing der threads ab.

Behebung: Timingsunabhängig prüfen, ob gewartet werden muss.

2.7 FPS bug

Symptome: Gespeicherte Videos der RGB Histogramme haben nicht die korrekte FPS.

Ursache: Fps wird nicht korrekt gesetzt.

Behebung: Setzen der Fps.

2.8 Filter bug

Symptome: Filtertab ist beim Hinzufügen eines Filters zur Filterpreview gewechselt, obwohl kein Video geladen war.

Ursache: Es wird nicht geprüft, ob ein Video geladen ist.

Behebung: Prüfen, ob ein Video geladen ist.

2.9 Broken video file bug

Symptome: Unzulässige Dateien können als Videos geladen werden.

Ursache: Fehlende Überprüfung ob es sich bei der Datei um ein Video handelt.

Behebung: Hinzufügen einer Überprüfung ob die Datei Frames enthält.

2.10 Project loading bug

Symptome: Beim Laden eines Projektes stürzt das Programm im Analysistab ab.

Ursache: UndoStack wird nach dem Laden des Projektes geleert, somit auch die geladenen Videos.

Behebung: UndoStack wird vor dem Laden des Video geleert.

2.11 Video saving bug

Symptome: Gespeicherte Videos können nicht von einem externen Player abgespielt werden.

Ursache: Metadaten zum Video werden nicht gespeichert.

Behebung: Speichern von Metadaten.

2.12 Project loading bug 2

Symptome: Beim Laden eines Projekts, wenn bereits ein Video geladen ist, stürzt das Programm ab.

Ursache: Der Undostack wird gelöscht, der Player aber nicht gestoppt. Der Player hat versucht ein gelöscht Video abzuspielen.

Behebung: Stoppen des Players vor dem Löschen des UndoStacks.

2.13 Filterlist undo bug

Symptome: Benutzen von undo nach hinzufügen/entfernen/reseten bei der Filterliste.

Ursache: Alter Zustand wird mit restore(memento) wiederhergestellt.

Behebung: Statt restore(memento) wird eine Handlung simuliert die das Ganze rückgängig macht.

3 Testüberdeckung

Durch das hinzufügen weiterer Unit tests und automatischer GUI tests konnten wir die Testabdeckung von den 23% aus der Implementierungsphase auf ca. 74% erweitern. Durch Manuelle Tests konnten wir die Abdeckung nochmal auf bis zu 80% erweitern. Zu diesen gehören das durchgehen der Szenarien die im Pflichtenheft beschrieben sind. Des weiteren haben wir auch noch das speichern und laden des Projektes, der Results und Filterconfigurationen sowie diverser Videos in verschiedenen Formaten manuell getestet.

4 Zusätzliche Szenarien und Testfälle

Zum testen der GUI haben wir zusätzliche zu den aus Szenarien aus dem Pflichtenheft folgende automatisch ausgeführte Mini-Szenarien erstellt.

4.1 CrazyMonkeyTest

Hintereinanderausführung 'zufälliger' Aktionen.

Ablauf:

Starte im *Filtertab*

Klicke Button: "Apply to video"

Klicke Button: "Filter up"

Klicke Button: "Apply to video"

Klicke Button: "Filter down"

Klicke Button: "Save configuration"

Klicke Button: "Add Edge filter"

Wechsle zu *Analysistab*

Klicke Button: "PSNR"

Klicke Button: "Bitrate"

Klicke Button: "Red histogram"

Lade Video

Klicke Button: "PSNR"

Klicke Button: "Bitrate"

Klicke Button: "Red histogram"

Wechsle zu *FilterTab*

Klicke Button: "Remove filter"

Klicke Button: "Save configuration"

Klicke Button: "Add Contrast filter"

Lade Video

Klicke Button: "Add Vintage filter"

Klicke Button: "Apply to video"

Klicke Button: "Redo"

Klicke Button: "Redo"

Klicke Button: "Redo"

Klicke Button: "Add Vintage filter"

Klicke Button: "Redo"

Klicke Button: "Undo"

Wechsle *Analysistab*

Klicke Button: "New"

Klicke Button: "Add video"

4.2 AddFilter ohne Video

Filter hinzufügen ohne zuvor ein Video geladen zu haben.