

# PSE 2012

## OQAT

Objective Quality Assessment Toolkit

Praxis der Softwareentwicklung

WS 2012

## P f l i c h t e n h e f t



Auftraggeber

Karlsruher Institut für Technologie

Institut für Technische Informatik

CES - Chair for Embedded Systems

Prof.Dr.J.Henkel

Betreuer: S. Kobbe

Auftragnehmer

Name	E-Mail-Adresse
Eckhart Artur	artur.eckhart@gmail.com
Ermantraut Georg	georg.ermantraut@gmail.com
Leidig Sebastian	sebastian.leidig@gmail.com
Monev Alexander	bcclan@mail.bg
Sailer Johannes	johsailer@gmail.com

Karlsruhe, 20.5.2012

# Inhaltsverzeichnis

<b>1</b>	<b>Einleitung</b>	<b>5</b>
<b>2</b>	<b>Zielbestimmung</b>	<b>7</b>
2.1	Musskriterien . . . . .	7
2.2	Wunschkriterien . . . . .	9
2.3	Abgrenzungskriterien . . . . .	9
<b>3</b>	<b>Produkteinsatz</b>	<b>10</b>
3.1	Anwendungsbereiche . . . . .	10
3.2	Zielgruppen . . . . .	10
<b>4</b>	<b>Produktumgebung</b>	<b>11</b>
4.1	Software . . . . .	11
4.2	Hardware . . . . .	11
4.2.1	Mindestanforderungen . . . . .	11
<b>5</b>	<b>Produktfunktionen</b>	<b>12</b>
5.1	Grundfunktionalität . . . . .	12
5.2	Optionale Funktionalität . . . . .	14
<b>6</b>	<b>Produktdaten</b>	<b>15</b>
<b>7</b>	<b>Qualitätsbestimmungen</b>	<b>16</b>
<b>8</b>	<b>Benutzeroberfläche</b>	<b>17</b>
<b>9</b>	<b>Globale Testfälle</b>	<b>20</b>
9.1	Testfälle . . . . .	20
9.2	Testszenarios . . . . .	20
<b>10</b>	<b>Systemmodelle</b>	<b>24</b>
10.1	MVVM-Architektur . . . . .	24
10.2	Anwendungsfälle . . . . .	25
<b>11</b>	<b>Entwicklungsumgebung</b>	<b>28</b>

<b>Glossar</b>	<b>29</b>
<b>Abkürzungsverzeichnis</b>	<b>31</b>

## Abbildungsverzeichnis

1.1	Diagramm zur Funktionsweise von OQAT <sup>1</sup> . . . . .	6
5.1	Projektexplorer . . . . .	13
8.1	Auswahl mehrerer Filter und des Namens unter dem das so entstandene Video im Projektordner gespeichert werden soll . . . . .	18
8.2	Auswahl einer Analysemetrik und des Namens unter dem die Ergebnisdaten im Projektordner abgelegt werden sollen . . . . .	18
8.3	Visuelle Darstellung der Ergebnisse nach einem Analysevorgang mit der PSNR <sup>2</sup> Metrik . . . . .	19
10.1	Architektur-Klassendiagramm des Grobentwurfs . . . . .	24
10.2	Anwendungsfall-Diagramm der Projektverwaltungs-Funktionalitäten . . . . .	25
10.3	Anwendungsfall-Diagramm der Filter-Funktionalitäten . . . . .	26
10.4	Anwendungsfall-Diagramm der Analyse-Funktionalitäten . . . . .	27

---

<sup>1</sup>Objective Quality Assessment Toolkit

<sup>2</sup>Peak signal-to-noise ratio

# 1 Einleitung

OQAT ist eine Anwendung, die im Rahmen des Praktikums der Softwareentwicklung am KIT<sup>1</sup> vom CES<sup>2</sup> des ITEC<sup>3</sup> in Auftrag gegeben wurde.

Die Erwartungen von Verbrauchern an die Qualität eines ihnen präsentierten Videos waren noch nie so hoch wie heute. Ob Internetvideoportale, Kinos oder das mit dem Smartphone aufgenommene Kurzvideo, die vom Verbraucher wahrgenommene Güte des angebotenen Dienstes, die sogenannte *Quality of Service* (QOS<sup>4</sup>), ist oft einer der wichtigsten Faktoren, die über den Erfolg des jeweiligen Diensteanbieters entscheiden. In der Vergangenheit wurde Videoqualität fast ausschließlich mit Hilfe subjektiver Verfahren beurteilt. Dabei wurden einer Gruppe von Testpersonen bestimmte Videosequenzen vorgespielt, worauf die Testpersonen die Sequenz nach einer vordefinierten Skala (z.B. MOS<sup>5</sup>) bewertete. Bei solch einer Untersuchung spielen sehr viele Faktoren eine Rolle, z.B. der Inhalt der vorgespielten Testsequenz, aber auch die Beleuchtung im Testraum, die Einstellungen des Vorführgeräts und die Erwartungen der jeweiligen Betrachter.

Aufgrund dieser Faktoren war und ist subjektive Qualitätsmessung einer Videosequenz sehr zeitaufwendig und teuer. Für viele Domänen ist es heutzutage schlichtweg unmöglich auf subjektive Methoden zurückzugreifen (Internetvideoportal mit mehreren Millionen Nutzern). Aus diesen und anderen Gründen setzt man heute vermehrt objektive Bewertungsverfahren ein.

---

<sup>1</sup>Karlsruher Institut für Technologie

<sup>2</sup>Chair for Embedded Systems

<sup>3</sup>Institut für technische Informatik

<sup>4</sup>Quality of Service

<sup>5</sup>Mean opinion Score

An dieser Stelle kommt OQAT ins Spiel. OQAT zielt darauf ab Entwickler von Video- oder Bild-verarbeitenden Werkzeugen bei der Evaluierung zu Unterstützen. Um dieser Aufgabe nachzukommen bedient sich OQAT bereits entwickelter Verfahren, was den Wert der Anwendung auf keinen Fall mindert, denn viele heute frei Verfügbaren Werkzeuge zur objektiven Videoqualitätsbewertung sind lediglich Forschungsprototypen.

Abbildung 1 demonstriert die prinzipielle Vorgehensweise von OQAT :

1. Der Benutzer stellt OQAT ein Referenzvideo im .YUV Format zur Verfügung.
2. OQAT verarbeitet das Referenzvideo mit Hilfe vom Benutzer zuvor gewählter Filter.
3. Der Benutzer wendet sein Werkzeug am verarbeiteten Video an und stellt es wiederum OQAT zur Verfügung.
4. OQAT analysiert das vom Nutzer bereitgestellte Video mit Hilfe eines Full Reference Verfahrens.
5. OQAT stellt die Ergebnisse der Analyse graphisch sinnvoll dar.

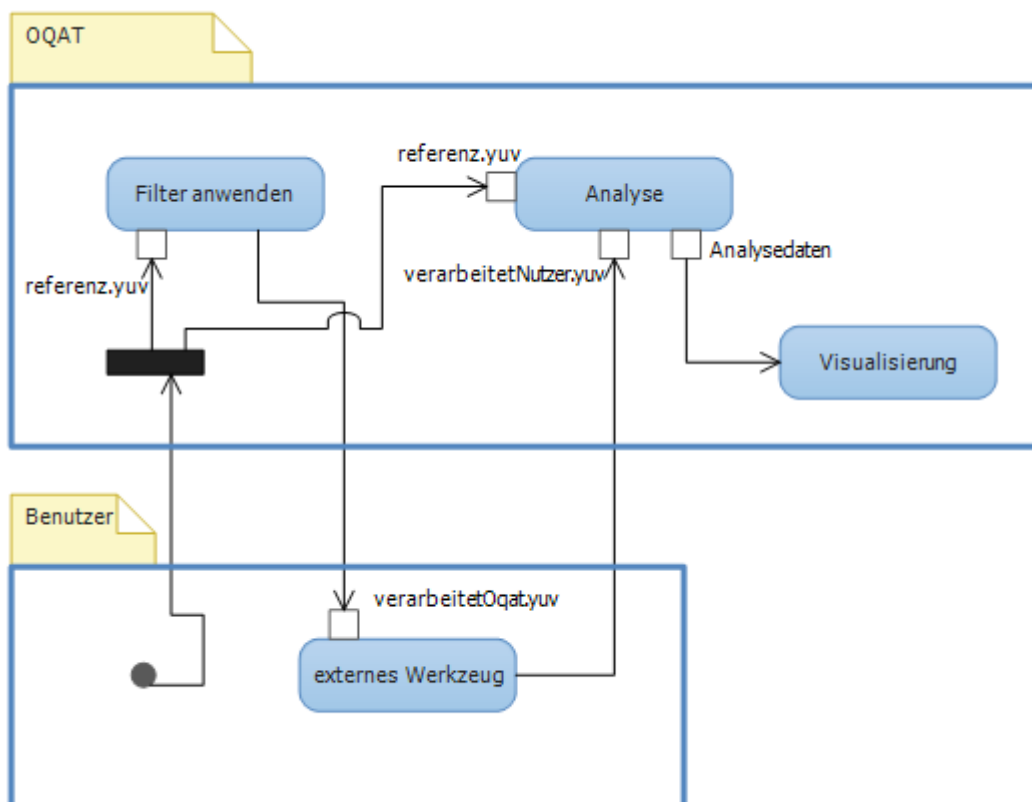


Abbildung 1.1: Diagramm zur Funktionsweise von OQAT

## 2 Zielbestimmung

### 2.1 Musskriterien

/MK-10/ OQAT verfügt über eine interaktive graphische Oberfläche.

/MK-20/ Die Anwendung kann Videos im .YUV Format von der Festplatte lesen, manipulieren und wieder auf die Festplatte schreiben.

/MK-30/ OQAT kann ein bereitgestelltes Video darstellen und bietet grundlegende Kontrollfunktionalitäten an wie:

- zeige das nächste Frame
- zeige das vorherige Frame
- zeige ein vom Nutzer gewähltes Frame
- spiele das Video mit einer Variablen Geschwindigkeit ab
- halte das Video an

/MK-40/ Ein Satz an Filtern steht der Anwendung bereits zum Auslieferungszeitpunkt zur Verfügung. Diese sind:

- Rauschgenerator
- Verschiedene Farbfilter (Graustufen, Sepia, Farbkanäle ein/ ausblenden)
- Faltungs-Filter (Scharfzeichner, Weichzeichner, Sobel-Filter)

/MK-50/ Ein Satz an Analysemetriken steht der Anwendung bereits zum Auslieferungszeitpunkt zur Verfügung. Diese sind:

- Mean square error  
Der Mean Square Error dient zur Berechnung der Abweichung eines Schätzers von dem zu schätzenden Wert. Es ergibt sich aus der Differenz zwischen den Werten und dem arithmetischem Mittel. Die Differenz wird quadriert und durch die Anzahl der Werte dividiert.

- Signal-to-noise ratio

Das Signal-to-noise Ratio ist ein Maß für die Qualität eines Signals, das von einem Rauschsignal überlagert ist. Es ist definiert als das Verhältnis der mittleren Leistung des Signals zur mittleren Rauschleistung des Störsignals.

- Peak signal-to-noise ratio PSNR ist ein Maßstab zur Messung von Unterschieden zwischen zwei Bildsignalen. Er wird unter anderem dazu verwendet, um mittels eines unkomprimierten Original-Videos und eines verlustbehaftet komprimierten entsprechenden Vergleichsvideos den Qualitätsverlust zu quantifizieren. Der PSNR basiert auf der MSE für die Pixel des Original-Videos und des Vergleichsvideos und wird durch eine Formel errechnet. Bei Farbsignalen wird der PSNR separat je Farbkomponente errechnet und anschliessend der Mittelwert gebildet.

/MK-60/ Falls es für einen Filter sinnvolle Einstellungsmöglichkeiten gibt, kann man diese verändern. Beispielsweise kann die Stärke eines Weichzeichners eingestellt werden.

/MK-70/ Einstellungsparameter für Filter können abgespeichert und geladen werden.

/MK-80/ Ein Filteervorgang kann mehrere Filter beinhalten. Die Reihenfolge, in der Filter angewandt werden, ist variabel und kann vom Nutzer bestimmt und abgespeichert werden. Bei einem solchen Vorgang wird nur ein Video für den kompletten Filteervorgang abgespeichert.

/MK-90/ Benutzer haben die Möglichkeiten Projekte anzulegen. Ein Projekt dient als Ablageort für absolvierte Analysevorgänge, relevante Projekteinstellungen und die vom Benutzer bereitgestellte Ressourcen.

/MK-100/ Der Nutzer hat die Möglichkeit aktuelle und oder zurückliegende Analyseergebnisse in einem portablen Format (z.B. .CSV<sup>1</sup>) zu exportieren.

/MK-110/ Die Ergebnisse eines Analysedurchlaufs werden graphisch aufbereitet (z.B. als Differenzbild).

/MK-120/ OQAT unterstützt Lokalisierung, d.h. der Nutzer hat die Möglichkeit die Sprachfunktionalität der Anwendung durch Sprachpakete zu erweitern. Es wird nur die deutsche Sprachdatei standardmäßig bereitgestellt.

---

<sup>1</sup>comma separated value



## 2.2 Wunschkriterien

/WK-10/ OQAT kann die Daten, die der Video Encoder des ITEC bereitstellt, erkennen und darstellen.

/WK-20/ Ein angelegtes Projekt ist auch unter einer anderen OQAT Installation lauffähig, vorausgesetzt die nötigen Ressourcen (z.B. Video Dateien die dem Projekt zuvor zur Verfügung gestellt wurden) sind vorhanden und es handelt sich um die selbe Version von OQAT .

/WK-30/ Der Benutzer kann eigene Analysemetriken implementieren und diese als Plugins einbinden.

/WK-40/ Der Benutzer kann eigene Filter implementieren und diese als Plugins einbinden.

/WK-50/ Während eines Analysevorgangs wird eine Fortschrittsanzeige und die voraussichtliche Dauer des Vorgangs angezeigt.

/WK-60/ Für einen ausgewählten Filter wird im Visualisierungsbereich eine Vorschau generiert.

/WK-70/ Der Projektextplorer unterstützt Drag and Drop.

## 2.3 Abgrenzungskriterien

/AK-10/ OQAT beherrscht nur Full Reference Analysemetriken.

/AK-20/ OQAT arbeitet nicht mit Audiodaten.

/AK-30/ OQAT untersucht nicht die Performanz von Videobearbeitungswerkzeugen

## 3 Produkteinsatz

### 3.1 Anwendungsbereiche

Die Anwendung kann in sehr unterschiedlichen Bereichen eingesetzt werden, einige Beispiele sind:

- Ein Entwickler von Videoencodern kann mit Hilfe von OQAT die Qualität der von seinem Werkzeug erzeugten Videos auf vielfältige Art und Weise analysieren und die Daten entwicklungsunterstützend einsetzen.
- Ein Internetvideoportal steht vor der Entscheidung welchen Videocodec es verwenden möchte. Eine aussagekräftige subjektive Qualitätsbewertung ist mit sehr hohen Kosten verbunden. Mit Hilfe von OQAT lassen sich diese Kosten vermeiden ohne sich blind für einen bestimmten Videocodec zu entscheiden.
- Ein Forschungsteam des KIT möchte die Qualität ihrer Videoencoder analysieren, stellt aber fest, dass es für ihren Encoder noch keine aussagekräftigen Analysemetriken gibt. Das Forschungsteam möchte aus Zeitgründen keine vollständige Analyseanwendung entwickeln. Mit OQAT muss das Team lediglich eine passende Analysemetrik als Plugin implementieren und kann sich dadurch auf ihr eigentliches Ziel konzentrieren.
- Entwickler von Filteralgorithmen(z.B. Rauschunterdrückung) können OQAT dazu nutzen, Filter (z.b. Rauschen) auf standardisierte Testsequenzen anzuwenden. Ihr eigenes Filterwerkzeug dann auf das daraus entstandene Video anzuwenden und anschließend die Ergebnisse mit Analysemetriken von OQAT zu untersuchen.

### 3.2 Zielgruppen

- Entwickler von Videoencodern.
- Entwickler von Filteralgorithmen.

## **4 Produktumgebung**

### **4.1 Software**

- Die Lauffähigkeit von OQAT wird nur für das Windows7 Betriebssystem garantiert.
- .NET 4.0 oder höher

### **4.2 Hardware**

#### **4.2.1 Mindestanforderungen**

- 2,27-GHz-Prozessor oder höher mit 64 Bit (x64)
- 4 GB RAM
- 100MB verfügbaren Festplattenspeicher
- Grafikkarte mit 1024 MB
- DirectX10-Grafikgerät mit WDDM 1.0- oder höherem Treiber

## 5 Produktfunktionen

### 5.1 Grundfunktionalität

#### /PF-10/ Startbildschirm

Nach dem Start der Anwendung werden dem Benutzer die zuletzt geöffneten Projekte angezeigt. Der Benutzer hat dann die Möglichkeit ein bereits bestehendes Projekt zu öffnen oder ein neues anzulegen.

#### /PF-20/ Filtercontainer

Der Filtercontainer ist die Stelle der grafischen Oberfläche, an der der Nutzer einen der Anwendung zur Verfügung stehenden Filter auswählen kann (durch einen Klick mit der Maus). Um einen Filter auszuwählen muss ein gültiges Video im Projektextplorer gewählt worden sein. Durch die Auswahl eines Filters wird der entsprechende Filter als der nächste anzuwendende Filter markiert. Dem Benutzer steht die Möglichkeit offen, einen bereits gewählten Filter wieder abzuwählen.

#### /PF-30/ Projektextplorer

Nachdem ein bestehendes Projekt geöffnet oder ein neues erstellt worden ist wird dem Benutzer unter anderem ein Projektextplorer zur Verfügung gestellt. Ein Projektextplorer ist ein Fenster das zwei Registerreiter enthält.

Der erste Reiter, die *SmartTree*, enthält dem Projekt zugewiesene Videosequenzen. Dabei wird grundsätzlich zwischen Referenz- und erzeugten Sequenzen unterschieden. Wie in Abb.5.1 ersichtlich, hat die *SmartTree* eine Baumstruktur, dabei ist jedes Väterelement ein Referenzvideo im Bezug zu seinen Kindelementen. Beispielsweise ist in Abb.5.1 »Carphone.yuv« ein Referenzvideo im Bezug zum Kindelement »Gaussian Blurr«. Gaussian Blurr ist ein vom Nutzer gewählter Name für die Videosequenz »Carphone.yuv« an der der gaußsche Weichzeichner angewandt wurde und »processed« ist der vom Nutzer gewählter Name für die »Gaussian Blurr« Videosequenz auf der er sein Videobearbeitungswerkzeug ausgeführt hat. Jedes der in der *SmartTree* aufgeführten Videos wird im Projektordner oder in einem vom Benutzer angegebenen Stelle seiner Festplatte abgelegt.

Der zweite Reiter *Dateiexplorer* stellt eine Schnittstelle zum Dateisystem des Benutzers dar. Mit Hilfe des Dateiexplorers kann der Nutzer OQAT neue Videosequenzen bekannt machen.

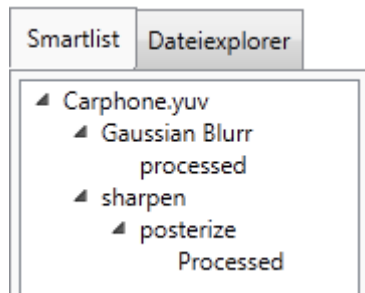


Abbildung 5.1: Projektexplorer

#### /PF-40/ Filtervorgang

Nachdem der Nutzer einen oder mehrere Filter gemäß der PF Filtercontainer markiert hat, kann er den Filtervorgang durch den Klick auf das entsprechende Symbol in der Toolbar oder im Hauptmenü einleiten. Die Dauer des Filtervorgangs ist abhängig von der Anzahl der gewählten Filter und der Auflösung und Länge des gewählten Videos. Nach einem Filtervorgang wird das neu entstandene Video als Kindelement des für den Filtervorgang ausgewählten Videos aufgeführt.

#### /PF-50/ Graphische Oberfläche

OQAT stellt dem Nutzer eine Interaktive Benutzeroberfläche zur Verfügung, die einzelnen Bestandteile dieser werden im Kapitel *Graphische Oberfläche* näher erläutert.

#### /PF-60/ Projekte anlegen

Um OQAT verwenden zu können, muss der Benutzer ein neues Projekt anlegen oder ein bereits existierendes öffnen.

Ein neues Projekt kann entweder über das Hauptmenü, die Toolbar oder mit Hilfe des sich im Startfenster befindenden Buttons angelegt werden. Nachdem der Nutzer eine dieser Möglichkeiten in Anspruch genommen hat, öffnet sich Projekterstellungs-Dialog. In diesem Dialog muss der Benutzer folgende Daten angeben:

- Projektname
- Ein Pfad unter dem die Projektdaten abgelegt werden sollen. Dieses kann auch leer gelassen werden, dann wird ein Defaultpfad verwendet.

#### /PF-70/ MetriList

Die *MetriList* enthält alle der Anwendung zur Verfügung stehenden Analysemetriken. Sobald

im SmartTree ein Video und ein gültiges Referenzvideo ausgewählt wurde, kann der Benutzer eine oder mehrere Metriken auswählen. Eine erfolgreiche Auswahl wird durch das hervorheben des jeweiligen Eintrags der *MetriList* und einen entsprechenden Vermerk in dem Visualisierungsbereich gekennzeichnet. Eine Metrik kann erst ausgewählt werden, wenn möglicherweise zuvor gewählte Filter z.B. durch den Klick auf das entsprechende Symbol der Toolbar bereits angewandt wurden. Versucht der Benutzer eine Metrik auszuwählen ohne ein Video und Referenzvideo entsprechend markiert zu haben oder während markierte aber nicht abgearbeitete Filter existieren wird er darauf hingewiesen.

#### /PF-80/ Analysevorgang

Nachdem der Nutzer eine oder mehrere Metriken aus der *MetriList* und ein Video samt einem Referenzvideo ausgewählt hat, kann der Benutzer einen Analysevorgang starten. Hat der Nutzer mehrere Metriken ausgewählt werden diese im Batchverfahren abgearbeitet. Nachdem ein Analysevorgang beendet wurde, stehen dem Nutzer die Ergebnisse im Visualisierungsbereich zur Verfügung (wurden mehrere Metriken angewandt, so werden diese unter verschiedenen Registerkarten des Visualisierungsbereichs abgelegt). Die Art der Darstellung der Analyseergebnisse hängt ganz und gar von der Art der gewählten Metrik ab. Wenn Beispielsweise die Metrik MSE<sup>1</sup> gewählt wurde, wird neben dem Differenzvideo, in dem jedes Pixel den jeweiligen MSE Wert dieser Stelle trägt, auch der globale MSE für das jeweilige Frame und für die gesamte Videosequenz dargestellt.

## 5.2 Optionale Funktionalität

#### /OF-10/ Motion Vektoren des .H264 Encoders des ITEC

Der .H264 Encoder des Instituts liefert, neben der encodierten Videosequenz, auch die verwendeten Motion Vektoren in einem Datensatz. OQAT kann diese Vektoren im Visualisierungsbereich darstellen.

#### /OF-20/ Drag and Drop

Der Projektextplorer ist *Drag and Drop* fähig, d.h. eine gültige Videodatei kann nicht nur über die Toolbar oder das Hauptmenü hinzugefügt werden, sondern darf auch nach dem *Drag and Drop* Prinzip in die SmartTree eingefügt werden. Das einzubindende Video kann dabei als ein neues Element (ohne Väterelement) oder aber als Kindelement eines bereits vorhandenen Videos eingefügt werden.

---

<sup>1</sup>Mean squared error

## 6 Produktdaten

Anwendungsdaten:

/PD-10/ Globale Einstellungen

/PD-20/ Filterdateien

Enthalten zur Verfügung stehende Filter und relevante Einstellungen zu diesen.

/PD-30/ Metriodateien

Enthalten zur Verfügung stehende Analysemetriken.

/PD-40/ Sprachdateien

Die Lokalisierungsdateien der Anwendung werden extern gespeichert. Die deutsche Sprachdatei wird standardmäßig bereitgestellt.

/PD-50/ Projektdaten

Ein Projekt fasst folgende Daten zusammen:

- Projektname und -beschreibung
- Referenzen zu dem Projekt zugewiesenen Videos
- Ergebnisse absolvierter Analysevorgänge
- Videodateien des Benutzers  
Die durch Filtervorgänge generierten Videodateien werden vom Programm generiert und gespeichert.
- Logdateien  
Enthalten Informationen darüber, welche Filter verwendet wurden, um die im Ordner liegenden Videodateien zu erzeugen.

## 7 Qualitätsbestimmungen

- Die GUI soll falsche Benutzereingaben weitestgehend vermeiden.
- Fehlerhafte Eingaben für Pfad, Frames per second, Filter und Analysemetriken werden vom Programm nicht angenommen und der Benutzer wird gefordert, sie zu korrigieren.
- Ein Filter- oder Analysevorgang findet nur für gültige YUV-Dateien statt, d.h. wenn eine YUV-Datei nicht gelesen werden kann, stürzt das Programm nicht ab und gibt eine entsprechende Fehlermeldung aus.
- OQAT wird Benutzern eine Hilfe, in Form von Tooltips und einer kurzen Einleitung die wichtigsten Programmfunktionen anbieten.
- Die Anwendung wird ausführlich getestet.



## 8 Benutzeroberfläche

OQAT stellt dem Nutzer eine intuitive graphische Oberfläche zur Verfügung. Die Graphische Oberfläche setzt sich aus folgenden Bestandteilen zusammen:

/GUI-10/ Hauptmenüleiste

Über das Hauptmenü sind nahezu alle Funktionen von OQAT abrufbar. Das Hauptmenü ist das oberste Element das auf den untenstehenden Abbildungen zu sehen ist.

/GUI-20/ Toolbar

Über die Toolbar sind häufig benötigte Funktionen wie z.B. »Neues Projekt erstellen«, »Projekt öffnen« abrufbar. Die Toolbar ist das Element unterhalb der Hauptmenüleiste.

/GUI-30/ Projektextplorer

Der Projektextplorer ist der Tabcontainer und auf der linken Seite der unten aufgeführten Screenshots zu sehen. Im Projektextplorer befindet sich der SmartTree und der Dateieexplorer.

/GUI-40/ Filter und Metriken

Auf Filter und Metriken kann man über den rechten Tabcontainer zugreifen.

/GUI-50/ Hauptvisualisierungsbereich

Im Hauptvisualisierungsbereich befindet sich der .YUV Player und außerdem werden hier die Ergebnisse einer Analyse dargestellt.

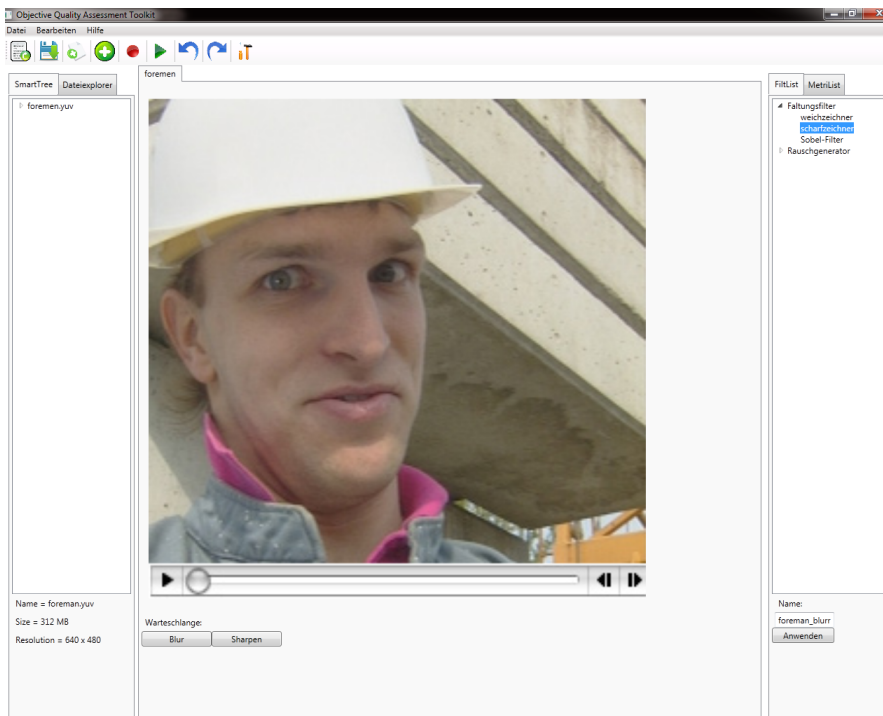


Abbildung 8.1: Auswahl mehrerer Filter und des Namens unter dem das so entstandene Video im Projektordner gespeichert werden soll

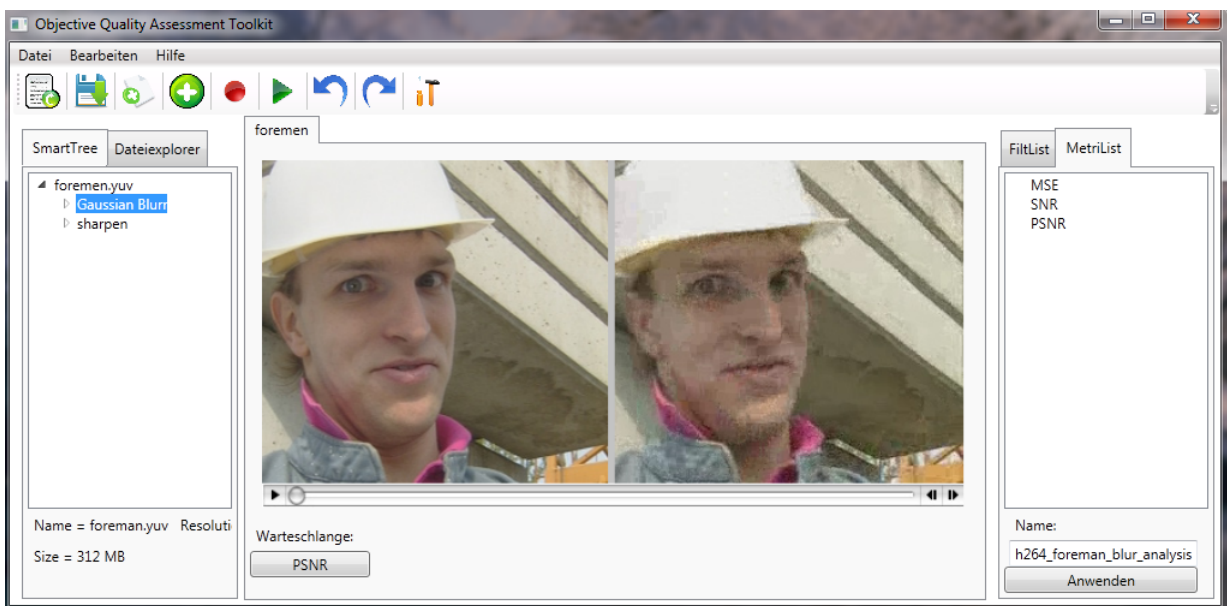


Abbildung 8.2: Auswahl einer Analysemetriks und des Namens unter dem die Ergebnisdaten im Projektordner abgelegt werden sollen

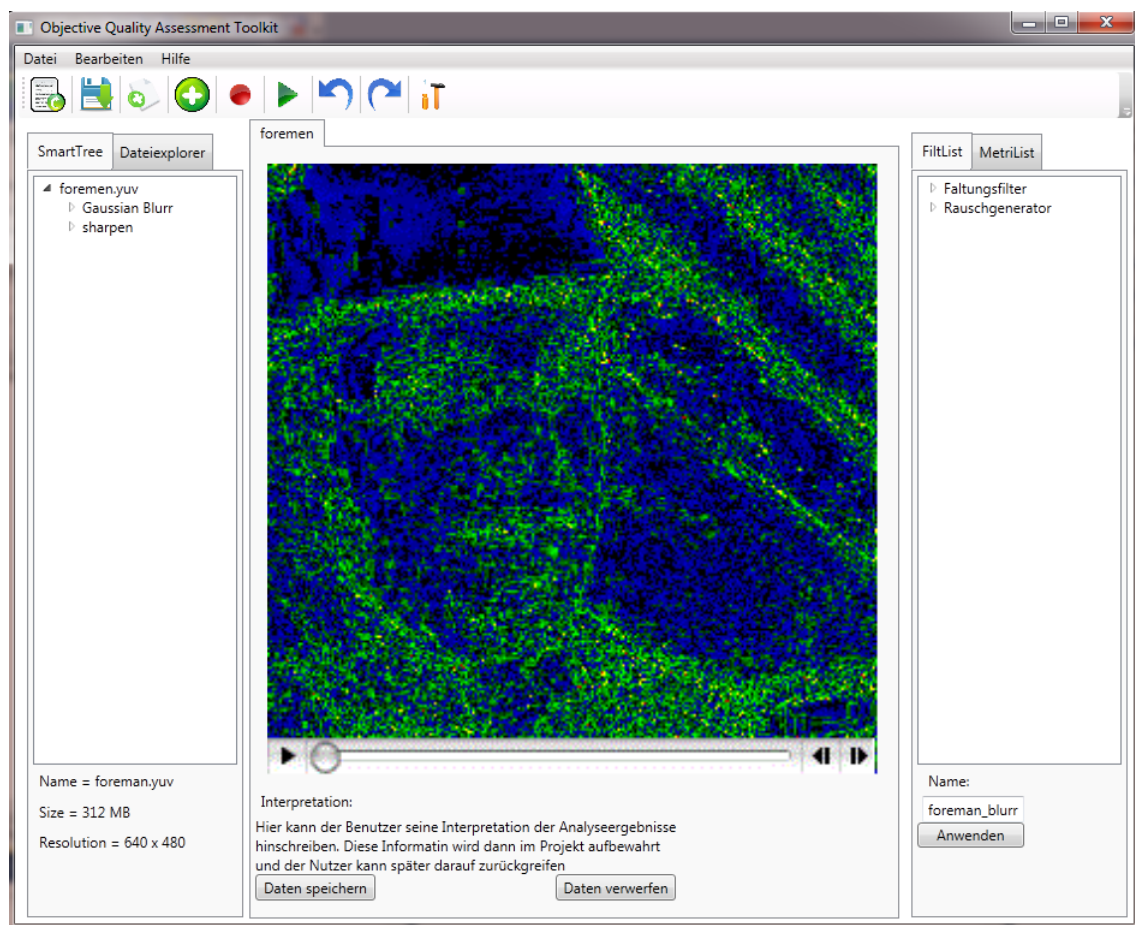


Abbildung 8.3: Visuelle Darstellung der Ergebnisse nach einem Analysevorgang mit der PSNR Metrik

## 9 Globale Testfälle

### 9.1 Testfälle

- /T-10/ Projekt erstellen, speichern und laden
- /T-20/ Projekt bearbeiten, speichern und laden
- /T-30/ Projekt auf anderem Rechner öffnen
  
- /T-40/ Filter auswählen ohne Video ausgewählt zu haben - Fehlermeldung
- /T-50/ Filter auswählen, Vorschau betrachten
- /T-60/ Filter anwenden, generierte Video-Datei überprüfen
- /T-70/ Filter-Einstellungen verändern
- /T-80/ Mehrere Filter auf ein Video anwenden, Reihenfolge verändern
  
- /T-90/ Analysemetrik auswählen ohne zwei Videos ausgewählt zu haben - Fehlermeldung
- /T-100/ Analyse starten ohne Metrik auszuwählen - Fehlermeldung
- /T-110/ Analyse durchführen, Ergebnisse anzeigen
- /T-120/ Analyseergebnisse speichern und laden
- /T-130/ Analyseergebnisse exportieren (CSV)
  
- /T-140/ GUI Funktionalität

### 9.2 Testszenarien

- /TS-10/ Neues Projekt und neues Video zum Projekt hinzufügen.

1. Bob startet OQAT .
2. Der Willkommensbildschirm öffnet sich.

3. Bob klickt auf den *Neues Projekt erstellen* Button.
4. Ein Projekt-Erstellungsdialog wird geöffnet.
5. Bob trägt einen Projekt-Titel und eine kurze Beschreibung ein und bestätigt den Vorgang.
6. Bob klickt auf den Dateexplorer und wählt *carphone.yuf* mit einem Rechtsklick aus. Aus dem erscheinenden Dialog wählt er die *Ressource hinzufügen* Option.
7. Das Video wird nun im Projektextplorer angezeigt.
8. Bob beendet das Programm, da sein Ziel erreicht ist.

/TS-20/ Bestehendes Projekt öffnen, mehrere Filter anwenden und deren Reihenfolge abspeichern.

1. Bob startet OQAT
2. Der Willkommensbildschirm öffnet sich.
3. Bob wählt aus der Liste seiner zuletzt geöffneten Projekten das *H264Test* Projekt aus.
4. Bob klickt auf das *carphone.yuf* Video im Smarttree.
5. Bob wählt den Weichzeichner aus der Filterliste aus.
6. Bob erhöht die Stärke des Weichzeichners und fügt ihn der Warteliste anzuwendender Filter hinzu.
7. Bob wählt den Scharfzeichner aus der Filterliste aus und fügt ihn der Warteliste anzuwendender Filter hinzu.
8. Bob schiebt den Scharfzeichner vor den Weichzeichner in der Warteliste.
9. Bob speichert die Reihenfolge unter dem Namen *ScharfWeich* ab.
10. Bob klickt auf Filter anwenden.
11. Das Video wird im *SmartTree* als Kindelement von *carphone.YUF* aufgelistet.
12. Bob beendet OQAT .

/TS-30/ Analyse durchführen.

1. Bob startet OQAT .
2. Bob öffnet das *H264Test* Projekt.

3. Bob wählt im Dateieexplorer die *carphoneH264encoded.YUF* und fügt diese als Kindelement des *carphone.yuv* Video (im *emphsmartTree*) hinzu.
4. Bob wählt die *PSNR* Metrik aus der *MetriList* aus.
5. Bob markiert das *carphone.yuv* Video aus dem *SmartTree* als Referenzvideo und *carphoneH264encoded.YUF* als das zu analysierende Video.
6. Bob startet die Analyse durch einen Klick auf das entsprechende Symbol der Toolbar.
7. Bob wartet bis der Analysevorgang beendet wurde.
8. Bob schaut sich die Ergebnisse der Analyse an und gibt eine Beschreibung ein, die er zusammen mit den Analyseergebnissen abspeichert um auch später auf diese zurückgreifen zu können.
9. Bob schließt das Programm.

#### /TS-40/ Kompletter Analyse-Ablauf

1. Bob startet OQAT .
2. Es erscheint ein Willkommensbildschirm.
3. Bob wählt die Option *neues Projekt erstellen*.
4. Im geöffneten Projekterstellungsdialog trägt Bob den Projekttitel (mpeg-4-TestProjekt) und den Pfad( /bob/Videos/) zu einem Videoverzeichnis in seinem Arbeitsverzeichnis.
5. Bob klickt auf den *Projekt anlegen* Button.
6. Im *smartTree* des *ProjektExplorers* werden nun alle kompatiblen (.YUV) Videos, die sich im Videoverzeichnis von Bob befinden, aufgelistet.
7. Bob wählt den *foreman.yuv* Eintrag im Smarttree aus.
8. Bob klickt auf das *play* Symbol des Videoplayers (im Hauptvisualisierungsbereich).
9. Nachdem Bob sich vergewissert hat, dass die Videosequenz seinen Ansprüchen genügt pausiert er das Video (durch einen Klick auf das *pause* Symbol des Videoplayers).
10. Bob wählt einen Rauschfilter aus der Filterliste.
11. Bob wählt den *Weichzeichner* Filter aus der Filterliste aus und fügt ihn der Warteschlange der auf das Video anzuwendenden Filter hinzu.
12. Bob gibt einen Namen an (foremanRauschenWeich.yuv), klickt auf den *Filter anwenden* Button und wartet bis der Vorgang beendet wurde.

13. Im SmartTree wird nun *foremanRauschenWeich.yuv* als Kindelement von *foreman.yuv* aufgeführt (und befindet sich außerdem im Projektordner von Bob).
14. Bob startet seinen MPEG-4 Encoder, wendet ihn auf das *foremanRauschenWeich.yuv* Video aus seinem Projektverzeichnis an und speichert das erzeugte Video (*foremanRauschenWeichEncoded.yuv* Video in seinem Projektverzeichnis ab.
15. Bob wählt das *foremanRauschenWeichEncoded.yuv* Video aus dem Dateieexplorer (ein Tab im Projektextplorer) und fügt es als Kindelement von *foremanRauschenWeich.yuv* hinzu.
16. Bob wählt den Analyse-Tab (neben dem Filter-Tab) aus, im Videoplayer sind nun zwei leere Boxen zu sehen.
17. Bob wählt das *foreman.yuv* Video als Referenzvideo und das *foremanRauschenWeichEncoded.yuv* als das zu vergleichende Video aus. In der linken (zuvor leeren) Box wird nun das Referenzvideo und in der rechten Box das zu vergleichende Video angezeigt.
18. Bob wählt die PSNR Metrik aus der Metriken Liste aus und klickt auf den *Analyse starten* Button.
19. Bob wartet bis der Analysevorgang fertig ist und startet das Ergebnisvideo (im Player).
20. Nachdem Bob sich das Ergebnisvideo angeschaut hat schreibt er eine kurze Beschreibung der Analyseergebnisse und speichert diese ab.
21. Bob schließt OQAT .

## 10 Systemmodelle

### 10.1 MVVM-Architektur

Es wird die MVVM-Architektur zur Realisierung von OQAT verwendet.

/S-10/ Model

Hier befinden sich die unter Produktdaten aufgeführten Punkte, Filter sowie Analysemetriken. Wenn die Daten sich ändern, schickt das Model eine Benachrichtigung an sein ViewModel.

/S-20/ ViewModel

Im ViewModel werden alle vom Benutzer erhaltenen Eingaben entsprechend verarbeitet und gegebenenfalls an das Model weitergeleitet. Das ViewModel leitet auch die Daten des Models an das View weiter.

/S-30/ View

Ein View ist die Präsentation der Daten des Models. In unserem Fall zeigt die GUI den aktuellen Zustand des Models.

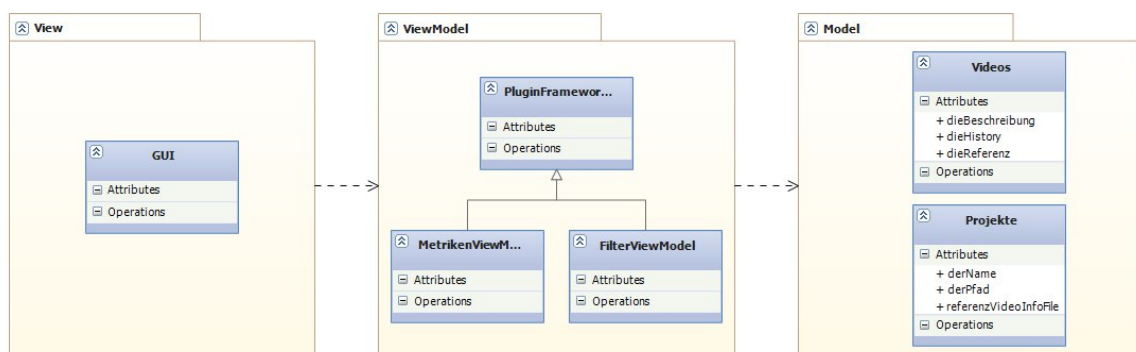


Abbildung 10.1: Architektur-Klassendiagramm des Grobentwurfs



## 10.2 Anwendungsfälle

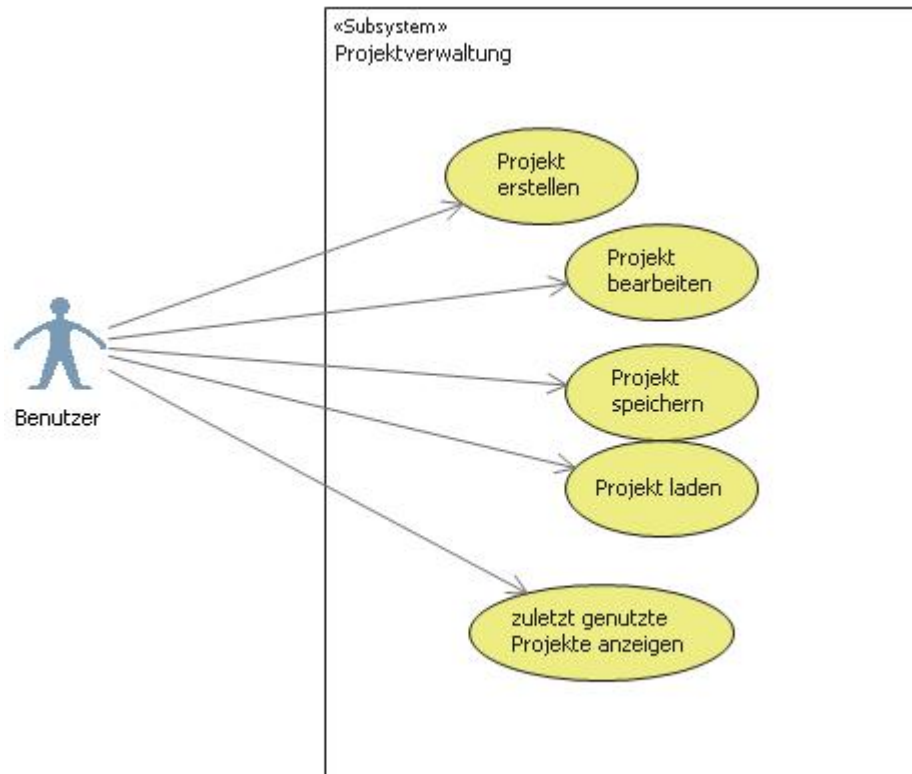


Abbildung 10.2: Anwendungsfall-Diagramm der Projektverwaltungs-Funktionalitäten

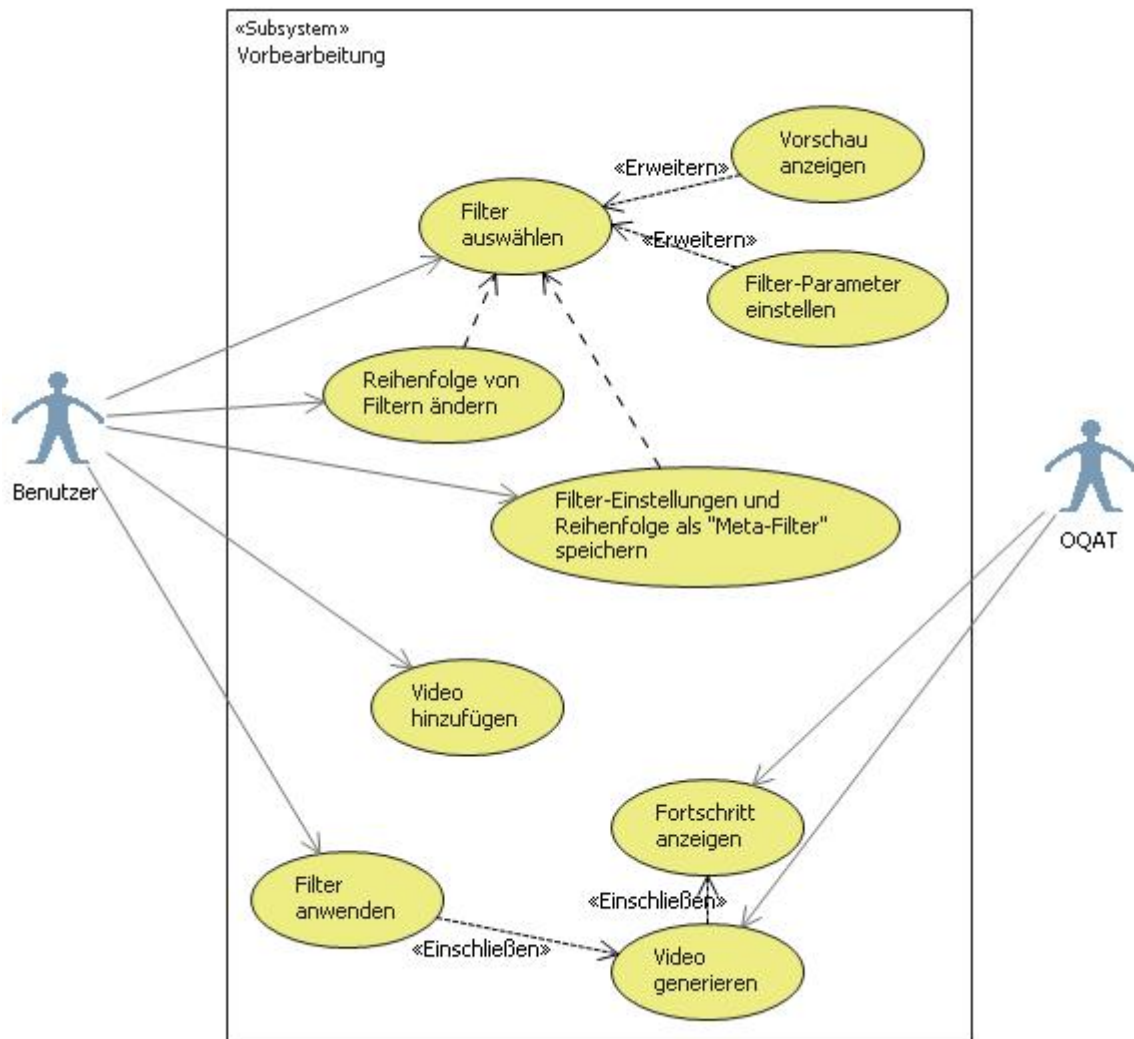


Abbildung 10.3: Anwendungsfall-Diagramm der Filter-Funktionalitäten

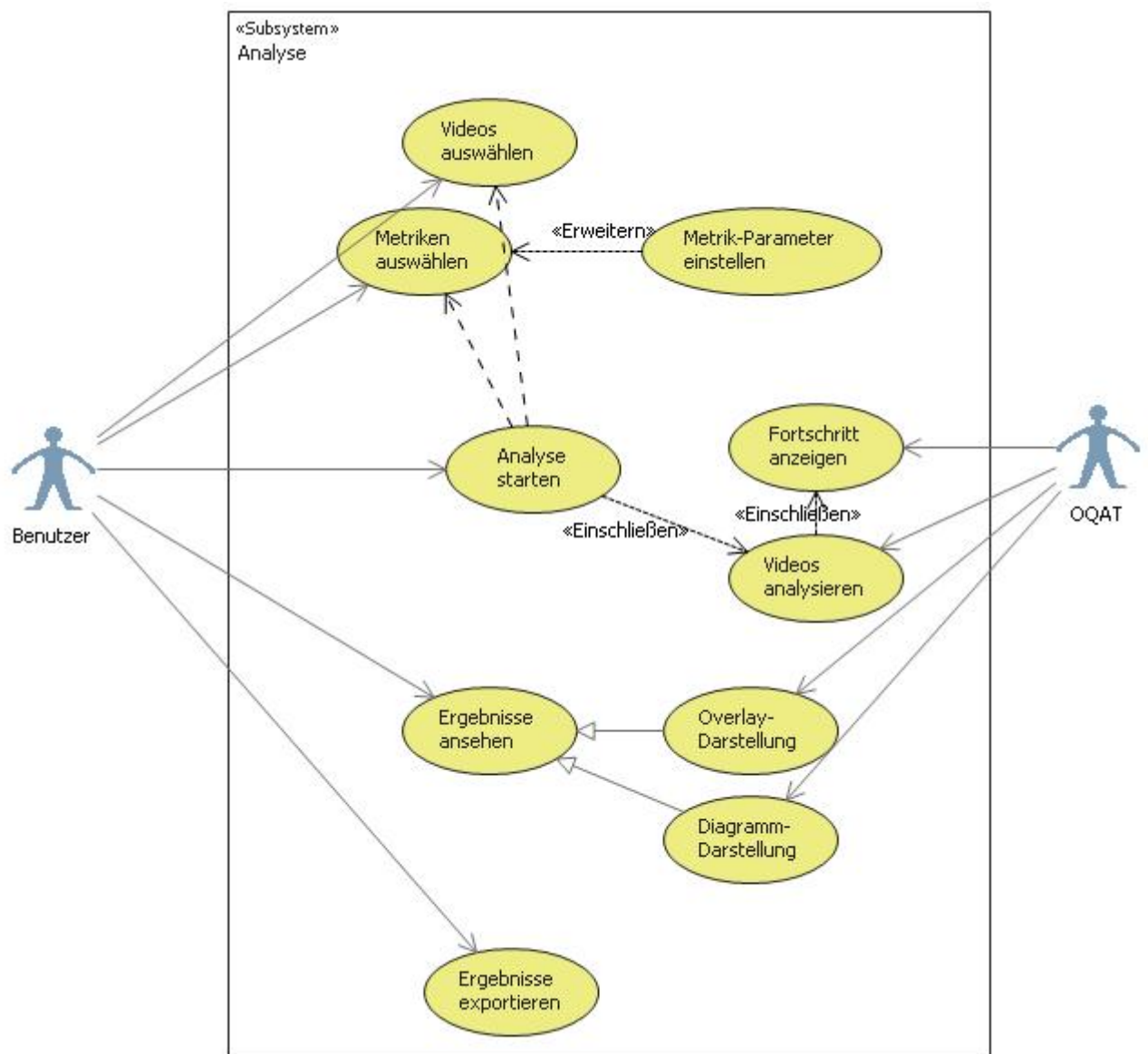


Abbildung 10.4: Anwendungsfall-Diagramm der Analyse-Funktionalitäten

## 11 Entwicklungsumgebung

UML-Tool	Visual Studio 2011
Entwicklungsumgebung	Visual Studio 2011
Programmiersprache	C Sharp
Code-Verwaltungssystem	Git
Dokumenteneditor	TexMakerX
Libraries	AForge, Managed Extensibility Framework

## Glossar

### **.YUV**

.yuv ist ein Dateiformat, in dem Videosequenzen unkomprimiert gespeichert werden. YUV teilt die Farbinformationen der Pixel dabei in Helligkeit (Luminance) und Farbigkeit (Chrominance) auf.

### **Filter**

Ein Werkzeug, mit dem OQAT ein Video manipuliert, um anschließend bestimmte Eigenschaften eines Videobearbeitungswerkzeugs gezielt auf diesem Video zu testen.

### **Full Reference**

Bei Full Reference Qualitätsanalyse ist (im Gegensatz zu einem No-Reference oder Reduced-Reference) das unveränderte Originalvideo vorhanden. Das heißt, eine Analyse bedarf immer zwei Videodateien, eine Referenzvideo und ein verarbeitetes Video(das mit Hilfe des Referenzvideos erstellt wurde)

### **Mean opinion score**

Der Mean Opinion Score ist das arithmetische Mittel einzelner Bewertungen und kann in einem Bereich von 1 (schlechtester Wert) bis 5 (bester Wert) liegen.

### **Mean squared error**

Der Mean Square Error dient zur Berechnung der Abweichung eines Schätzers von dem zu schätzenden Wert. Es ergibt sich aus der Differenz zwischen den Werten und dem arithmetischem Mittel. Die Differenz wird quadriert und durch die Anzahl der Werte dividiert.

### **Peak signal-to-noise ratio**

PSNR ist ein Maßstab zur Messung von Unterschieden zwischen zwei Bildsignalen. Er wird unter anderem dazu verwendet, um mittels eines unkomprimierten Original-Videos und eines verlustbehaftet komprimierten entsprechenden Vergleichs-Videos den Qualitätsverlust zu quantifizieren. Der PSNR basiert auf der MSE für die Pixel des Original-Videos und des Vergleichsvideos und wird durch eine Formel errechnet. Bei Farbsignalen wird der PSNR separat je Farbkomponente errechnet und anschliessend der Mittelwert gebildet.

### **Quality of Service**

Quality of Service oder Dienstgüte beschreibt die Güte eines Kommunikationsdienstes aus der Sicht der Anwender, das heißt, wie stark die Güte des Dienstes mit deren Anforderungen übereinstimmt.

### **Video Encoder**

Videoencoder werden hauptsächlich genutzt um Videodaten zu komprimieren.

### **Videobearbeitungswerkzeug**

Ein Werkzeug, dessen Qualität getestet werden soll; z.B. ein Video Encoder, aber auch die Analyse von auch Bildbearbeitungswerkzeugen ist mit OQAT möglich(vorausgesetzt das Bild wird als ein .YUV Video mit einem Frame encodiert).

## Abkürzungsverzeichnis

.CSV	comma separated value
CES	Chair for Embedded Systems
ITEC	Institut für technische Informatik
KIT	Karlsruher Institut für Technologie
MOS	Mean opinion Score
MSE	Mean squared error
OQAT	Objective Quality Assessment Toolkit
PSNR	Peak signal-to-noise ratio
QOS	Quality of Service