



AKADEMIA GÓRNICZO-HUTNICZA IM. STANISŁAWA STASZICA W KRAKOWIE

WYDZIAŁ INFORMATYKI, ELEKTRONIKI I TELEKOMUNIKACJI

KATEDRA TELEKOMUNIKACJI

**Proces tworzenia środowiska do testów subiektywnych
video wysokiej rozdzielczości w oparciu o dyplomową pracę
magisterską**

Autorzy:	Konrad Jagielski, Bartosz Orliński
Kierunek studiów:	Sieci i Usługi
Opiekun pracy:	dr inż. Lucjan Janowski

Kraków, 2017

1. WSTĘP

Niniejsza praca magisterska jest wynikiem pracy autorów nad kwestią jakości wideo wysokiej rozdzielczości poprzez przeprowadzenie testów subiektywnych oraz oceny jakości przeprowadzanych standardowych testów subiektywnych. Badania wideo są nieodzownym elementem rozwoju telekomunikacji, jako dziedziny nauki zajmującej się transmisją informacji na odległość. Aby usługobiorcy mogli uzyskać doskonały obraz istotne jest wysłanie sygnału wideo o jak najlepszej jakości dlatego też usługodawca (nadawca telewizyjny, bądź internetowy) stosuje szereg testów przed uruchomieniem nowej usługi. Obecnie standardem staje się wideo w jakości *FullHD*, jednakże niemal wszyscy producenci sprzętu dostarczają na rynek urządzenia gotowe do odtwarzania obrazu w jakości *UltraHD*. Ze względu na możliwość wprowadzenia do użytku codziennego, w postaci telewizji bądź filmów, wideo w jakości lepszej niż *FullHD* twórcy pracy postanowili stworzyć oprogramowanie pozwalające na przeprowadzania testów w możliwie jak najlepszej jakości. Same testy subiektywne stosuje się od wielu lat, jednak raz opracowane nie są poddawane zbyt częstej weryfikacji należy więc zastanowić się czy jest możliwość oceny jakości testów subiektywnych będących z założenia zależnych od grupy testerów i jeżeli tak to jak tego dokonać.

2. CEL PRACY

Celem pracy było zaprojektowanie i stworzenie stanowiska do przeprowadzenia oceny jakości wideo w standardzie UHD wraz z przygotowaniem oprogramowania i scenariuszy testowych, a także przeprowadzenie badań pozwalających na weryfikację poprawnego działania zaproponowanego rozwiązania. Kolejnym aspektem było porównanie różnych standardowych metod przeprowadzania testów subiektywnych.

3. STRESZCZENIE

Głównym problemem przedstawionym w pracy jest porównanie wybranych metod testów subiektywnych nieskompresowanych sekwencji wideo. Zbadanie głównego problemu było możliwe poprzez sformułowanie szeregu pomniejszych problemów i następnie ich rozwiązanie. Część z niżej przedstawionych zostało zdefiniowanych już podczas fazy planowania i analizy wstępnej problemu, inne pojawiły się dopiero po znalezieniu odpowiedzi na poprzednie lub w toku pracy. Kluczowe dla pracy było rozwiązanie ich wszystkich.

Problem wymagań sprzętowych wymagał wnikliwej analizy istniejących rozwiązań stosowanych w elektronice, która służy do przetwarzania obrazu, a następnie rozważenia dostępnych konfiguracji sprzętowych oraz, co najważniejsze, ich dostępności. Ustalony został maksymalny budżet na zakup

nowego sprzętu elektronicznego, kolejno podjęto rozmowy dotyczące wypożyczenia części wymaganego zestawu. W sprawie pozostałych przestudiowano dostępność wybranych produktów.

Przyjętym standardem podczas przeprowadzania testów subiektywnych wideo jest użycie nieskompresowanych sekwencji wideo. Konieczne było przygotowanie oprogramowania, które pozwoli odtwarzać kolejne klatki reprezentowane jako surowe dane. Przeprowadzono analizę istniejących już rozwiązań, a następnie stworzono autorskie rozwiązanie programistyczne oraz przetestowano zarówno jego wydajność jak i niezawodność.

Przed przeprowadzeniem badań należało zastanowić się nad scenariuszami testowymi które zostaną wykonane. Scenariusze musiały zostać przygotowane bardzo drobiazgowo, ze względu na zarówno potrzebę stworzenia oprogramowania zdolnego do ich przeprowadzenia jak i analizy wyników uzyskanych w ich trakcie. Ważnym elementem było dobranie scenariuszy w taki sposób, aby możliwe było ich bezpośrednie porównanie.

Zgodnie z zaleceniami środowisko eksperymentu powinno być przygotowane starannie. Osoba poddawana badaniu, dzięki naszym zabiegom, powinna odczuwać odpowiedni poziom komfortu, a jednocześnie przestrzeń, w której przebywa, nie powinna burzyć jej koncentracji i negatywnie wpływać na skupienie. Przeanalizowano dostępne lokacje, w których test mógłby się odbyć, a następnie wybrano najlepszą z nich i przystosowano ją tak, by jak najlepiej spełniała powyższe kryteria

Zebrane wyniki poddane zostały analizie. Porównano czułość poszczególnych scenariuszy, omówiono trendy pojawiające się wśród grupy osób badanych. Zaobserwowane wnioski zostały poddane weryfikacji oraz odpowiednio omówione.

4. OPROGRAMOWANIE

4.1 Informacje ogólne

Kluczowymi elementami pracy było stworzenie oprogramowania umożliwiającego odtwarzanie nieskompresowanych sekwencji wideo oraz przeprowadzenie testów subiektywnych z użyciem autorskiego odtwarzacza.

4.2 Wybór narzędzi

Wybór narzędzi wiązał się z koniecznością przeglądu dostępnych zasobów ludzkich pod kątem umiejętności tworzenia oprogramowania w danym języku. Kolejną niezbędną kwestią była analiza istniejących rozwiązań w celu znalezienia jak najlepszego narzędzia do rozwiązania problemu.

Odtwarzacz nieskompresowanych sekwencji wideo musi charakteryzować się jak największą wydajnością. Poprzez wydajność rozumiane jest jak najlepsze zarządzanie zasobami w taki sposób, aby na dostępnym sprzęcie komputerowym uzyskać jak najlepsze parametry wyświetlania kolejnych klatek. Zdecydowano się zrezygnować z języków korzystających z maszyn wirtualnych na korzyść takich, które pozwalają na dużą swobodę w zarządzaniu pamięcią. Wybór padł na język C++ ze względu na doświadczenie autorów w pracy z nim, zarówno zawodowe jak i nabyte podczas studiów.

Wybrany systemem operacyjnym został Linux, dystrybucja *Xubuntu* [1] 16.04.2 posiadająca jądro w wersji 4.4.0-72-generic. Decydującymi czynnikami były łatwość instalowania kolejnych pakietów bibliotek, dostęp do narzędzi konwertujących parametry filmów oraz niskie zużycie zasobów sprzętowych przez biernie działający system.

Zdecydowano się użyć środowiska programistycznego *CLion* [2] dostarczanym przez firmę *JetBrains* [3], korzystając z licencji studenckiej [4], która pozwala na użycie IDE (ang. *Integrated Development Environment*) w celach edukacyjnych.

Analiza istniejących odtwarzaczy wideo pozwoliła wyselekcjonować bibliotekę, która dała możliwość darmowego użycia i jak największej możliwości modyfikacji, czyli posiadała licencję *open source*. Ze względu na popularność odtwarzacza VLC w systemach operacyjnych *Linux*, podjęto decyzję o wykorzystaniu biblioteki z użyciem której powyższy odtwarzacz został stworzony – *libVLC* [5].

FFmpeg [6], czyli narzędzie pozwalające na edycję parametrów wzorcowego wideo zostało wybrane ze względu na łatwość użycia, swobodę w wyborze zmienianych aspektów filmu oraz licencję *open source*. Wszystkie użyte w pracy sekwencje wideo zostały wygenerowane przy pomocy *FFmpeg* z filmów wzorcowych.

QT [7] to zestaw bibliotek dedykowanych dla m.in. języka C++ pozwalający na tworzenie zaawansowanych interfejsów użytkownika. *QT* zawiera również elementy pozwalające na obsługę procesów, sieci i grafiki trójwymiarowej a także integrację z bazami danych, posiada także narzędzia pozwalające na przeprowadzenie lokalizacji dla innych wersji językowych programu. Wykorzystane narzędzia zostały szczegółowo omówione w części pracy poświęconej graficznemu interfejsowi użytkownika. Wybór bibliotek *QT* jako narzędzia do tworzenia *GUI* (ang. *Graphic User Interface*) był niejako oczywisty ze względu na fakt, iż VLC wykorzystuje jako *QT* jako jeden z podstawowych interfejsów użytkownika w odtwarzaczu podstawowym. Kluczowa była także opinia Jean-Baptiste Kempfa jednego z autorów i moderatorów biblioteki *VLC*, a także prezydenta i administratora forum jej poświęconego. Po przeanalizowaniu innych możliwości uznano, iż jest to jedyny dostępny produkt dający możliwość swobodnego tworzenia paneli np. do oceny obejrzanego filmu.

QT Creator [8] [9] to jedno z narzędzi udostępnianych w ramach zestawu bibliotek *QT*. Jest to kolejne użyte w pracy środowisko programistyczne pozwalające nie tylko na edycję kodu źródłowego, ale także (w narzędziu *QT Designer*) na projektowanie graficznego interfejsu użytkownika w

okienkowym edytorze za pomocą widżetów z biblioteki QT. Za pomocą dodatkowych narzędzi *uic* i *moc* każda klasa korzystająca z sygnałów i slotów QT (reprezentująca element GUI) otrzymuje dodatkowe pliki z rozszerzeniem *cpp* i *ui* reprezentujące rozmieszczenia okien, ustawienia grafiki, standardowe zdarzenia i przypisane do nich metody.

VLC-QT [10] darmowa biblioteka autorstwa Tadej Novaka służąca do połączenia bibliotek QT z biblioteką *libvlc*. Pozwala ona na stworzenie prostego odtwarzacza wraz z dowolnym interfejsem pozwalającym na kontrolowanie odtwarzania. *VLC-QT* nakrywa metody i klasy upraszczając użycie *libvlc* w oknach QT. Ze względu na ograniczenia spowodowane przez użycie interfejsu *imem* do wczytywanie klatek nieskompresowanych sekwencji wideo, użycie *VLC-QT* zostało ograniczone do użycia widżetu wideo dającego większe możliwości niż standardowe *QFrame*.

5. PRZYGOTOWANIE BADAŃ

5.1 Wstęp

Po wstępie teoretycznym i części implementacyjnej naturalnym następnym krokiem jest część badawcza. Wartość stworzonego środowiska testowego można sprawdzić tylko w jeden sposób – przeprowadzając badania. Część badawczą postanowiono rozpocząć od eksperymentu mającego na celu zadanie sprawdzenie poprawnego działania odtwarzacza, a więc weryfikację czy wyświetlane wideo jest zgodne z tym co zostało wysłane przez program do karty graficznej. Właściwe testy subiektywne miały na celu porównanie standardowych metod badawczych pod kątem wpływu wyboru metody testu na jego wynik.

6. TESTY SUBIEKTYWNE

6.1 Cel eksperymentu

Za cel przeprowadzanego eksperymentu przyjęto zbadanie wpływu wyboru metody przeprowadzania testu na otrzymane wyniki. Przeprowadzenie różnych testów i porównanie wyników pozwoliłoby na wyłonienie metody najbardziej efektywnej. Postanowiono zastanowić się nie tylko nad samymi wynikami, ale także nad łatwością obsługi, czasem koniecznym na przestawienie sposobu działania danego testu testerowi, opinią testerów dotyczącą poszczególnych testów oraz czasem pozyskiwania wyników z poszczególnych testów.

7. ANALIZA DANYCH

7.1 Informacje ogólne

Pierwszym krokiem, aby porównać wybrany zbiór metod testowych jest ich przeprowadzenie oraz zebranie wyników. Jednak same wyniki liczbowe nie przynoszą żadnej wiedzy na temat testów. Dopiero po ich analizie można zacząć wyciągać wnioski. W poniższym rozdziale przedstawione zostaną użyte metody porównawcze. Opisany zostanie również tok myślenia, kierujący autorami podczas analiz.

Przeprowadzone zostały trzy różne scenariusze testowe. Dwie z nich można porównać w sposób bezpośredni, ponieważ korzystają z tej samej puli sekwencji wideo. Filmy te oceniane są w tej samej skali w obu scenariuszach. Trzecia metoda badawcza, polegająca na porównywaniu dwóch następujących po sobie filmów, oceniana była w innej skali. Co więcej dostarczała wiedzy o filmach nie w porównaniu ze wszystkimi dostępnymi, ale tylko w zestawieniu z wybranym jednym.

W części badawczej pracy magisterskiej zbadano prawdziwość tezy wpływu doboru scenariusza testowego na otrzymane wyniki.

BIBLIOGRAFIA

- [1] The Xubuntu team, „Xubuntu,” The Xubuntu team, [Online]. Available: <https://xubuntu.org/>. [Data uzyskania dostępu: 1 Czerwiec 2017].
- [2] JetBrains, „A cross-platform IDE for C and C++ :: JetBrains CLion,” JetBrains, [Online]. Available: <https://www.jetbrains.com/clion/>. [Data uzyskania dostępu: 1 Czerwiec 2017].
- [3] JetBrains, „JetBrains: Development Tools for Professionals and Teams,” JetBrains, [Online]. Available: <https://www.jetbrains.com/>. [Data uzyskania dostępu: 1 Czerwiec 2017].
- [4] JetBrains, „Free for Students: Professional Developer Tools from JetBrains,” JetBrains, [Online]. Available: <https://www.jetbrains.com/student/>. [Data uzyskania dostępu: 1 Czerwiec 2017].
- [5] VideoLAN, „libVLC media player, Open Source video framework for every OS! - VideoLAN,” VideoLAN, [Online]. Available: <http://www.videolan.org/vlc/libvlc.html>. [Data uzyskania dostępu: 1 Czerwiec 2017].
- [6] FFmpeg, „FFmpeg,” FFmpeg, [Online]. Available: <https://ffmpeg.org/>. [Data uzyskania dostępu: 1 Czerwiec 2017].

- [7] QT, „qt.io,” [Online]. Available: <https://www.qt.io>. [Data uzyskania dostępu: Czerwiec 2017].
- [8] QT, „<https://www.qt.io/ide/>,” [Online]. Available: <https://www.qt.io/ide/>. [Data uzyskania dostępu: Czerwiec 2017].
- [9] W. Commons, „[pl.wikipedia.org](https://pl.wikipedia.org/wiki/Qt_Creator),” [Online]. Available: https://pl.wikipedia.org/wiki/Qt_Creator. [Data uzyskania dostępu: Czerwiec 2017].
- [10] T. Novak, „<https://vlc-qt.tano.si/>,” [Online]. Available: <https://vlc-qt.tano.si/>. [Data uzyskania dostępu: Czerwiec 2017].