

Joonas Kannisto

Android-kamerarajapinnan tuki automaattiseen kameratestausjärjestelmään

Opinnäytetyö

Kevät 2015

SeAMK Tekniikka

Tietotekniikan Koulutusohjelma

SeAMK 

SEINÄJOEN AMMATTIKORKEAKOULU
SEINÄJOKI UNIVERSITY OF APPLIED SCIENCES

SEINÄJOEN AMMATTIKORKEAKOULU

Opinnäytetyön tiivistelmä

Koulutusyksikkö: Tekniikan yksikkö

Tutkinto-ohjelma: Tietotekniikan Koulutusohjelma

Suuntautumisvaihtoehto: Ohjelmistotekniikka

Tekijä: Joonas Kannisto

Työn nimi: Android-kamerarajapinnan tuki automaattiseen kameratestausjärjestelmään

Ohjaaja: Petteri Mäkelä

Vuosi: 2015

Sivumäärä: 42

Liitteiden lukumäärä: 1

Opinnäytetyö toteutettiin Sofica Oy:n SoMA automaattiseen kameratestausjärjestelmään. Sofica Oy kehittää ja markkinoi SoMA automaattista kameratestausjärjestelmää matkapuhelinvalmistajille. SoMA-järjestelmä mahdollistaa tuotekehitysvaiheen ongelmien ja virheiden löytymisen mahdollisimman aikaisessa vaiheessa matkapuhelinten kameroiden kehityksessä.

Uuden Android-käyttöjärjestelmän päivityksen myötä SoMA-järjestelmän testiohjelma ei ollut enää ajankohtainen uusien matkapuhelimien kanssa. Työn toteutustavaksi valittiin uuden testiohjelman tekeminen, koska Android-käyttöjärjestelmän kameraominaisuudet kokivat täydellisen muutoksen. Työn tavoitteena oli saada tehtyä täysin uusi testiohjelma, joka tukee uuden kamerarajapinnan toimintoja.

Lopputulokseksi saatiin uusi testiohjelma, jolla uuden Android-version kamerarajapinnan toiminnot saadaan testattua.

Avainsanat: Android, testaus, kamerapuhelimet, testausjärjestelmä

SEINÄJOKI UNIVERSITY OF APPLIED SCIENCES

Thesis abstract

Faculty: School of Technology

Degree programme: Information Technology

Specialisation: Software Engineering

Author: Joonas Kannisto

Title of thesis: Android camera interface support for an automatic camera testing system

Supervisor: Petteri Mäkelä

Year: 2015

Number of pages: 42

Number of appendices: 1

This thesis was made for the SoMA automatic camera testing system of Sofica Oy. Sofica Oy develops and markets the SoMA automatic camera testing system for mobile phone manufacturers. The SoMA system enables finding the problems and mistakes of the product development stages as early as possible.

Because of the new Android operating system upgrade, the SoMA system test program was no longer suitable for new mobile phones. Creating a new test program was chosen to be the method of implementation, because the Android operating system's camera features had experienced a complete change. The objective of the thesis was to make whole new test programs which support the new camera interface features. The final result was a new test program which enables the testing of the new camera interface features.

Keywords: Android, testing, camera phone, testing system

SISÄLTÖ

Opinnäytetyön tiivistelmä.....	2
Thesis abstract.....	3
SISÄLTÖ.....	4
Kuva-, kuvio- ja taulukkoluettelo.....	6
Käytetyt termit ja lyhenteet.....	8
1 JOHDANTO.....	11
1.1 Työn tausta.....	11
1.2 Työn tavoite.....	11
1.3 Työn rakenne.....	12
1.4 Yritysesittely.....	12
2 ANDROID-KÄYTTÖJÄRJESTELMÄ.....	13
2.1 Android 5.0 Lollipop -versio.....	14
3 ANDROID-KAMERA.....	17
3.1 Kamera API.....	17
3.2 Uusi kamera API.....	18
3.3 Kamera HAL v3.....	20
4 SoMA AUTOMAATTINEN KAMERATESTAUSJÄRJESTELMÄ....	23
4.1 API-testaus.....	25
4.2 Toiminnallinen testaus (functional testing).....	26
4.3 Suorituskykytestaus (performance testing).....	28
4.4 Rasitustestaus (stress testing).....	28
4.5 Laadun regressiotestaus (quality testing).....	28
4.6 Testausympäristö.....	29
5 TESTIOHJELMA.....	31
5.1 Laitteisto.....	31
5.2 Vanha testiohjelma ja SoMA-palvelin.....	31
5.3 Uusi testiohjelma.....	33
5.3.1 Kameratoiminnot.....	34
5.3.2 Video- ja äänitoiminnot.....	36
6 TULOKSET JA YHTEENVETO.....	38

LÄHTEET	40
LIITTEET	42

Kuva-, kuvio- ja taulukkoluettelo

Kuvio 1. Android 5.0 -version pika-asetukset-valikko.....	15
Kuvio 2. Kameraoikeuksien määrittely.....	17
Kuvio 3. Ilmoitus kameran käytöstä	17
Kuvio 4. SurfaceHolderin määrittely.....	17
Kuvio 5. Kameran alustus	18
Kuvio 6. Kameran esikatselun aloitus	18
Kuvio 7. Kameran sulkeminen	18
Kuvio 8. Uuden kameran Preview-luokka	18
Kuvio 9. Esikatselun määrittäminen pääluokassa.....	19
Kuvio 10. Kameran avaus.....	19
Kuvio 11. Esikatselun asetusten määrittely	20
Kuvio 12. Esikatselun käynnistäminen.....	20
Kuvio 13. Kameran sulkeminen	20
Kuvio 14. Kameran ytimen toimintamalli	22
Kuvio 15. SoMA automaattinen kameratestausjärjestelmä.....	23
Kuvio 16. SoMA:n testaustasot.....	24
Kuvio 17. Suoritusajon sekvenssi	25
Kuvio 18. API-testaus	26
Kuvio 19. Toiminnallinen testaus	27

Kuvio 20. Kalibrointitestin shakkilautakuva	27
Kuvio 21. SoMA-testilaatikko	29
Kuvio 22. Laitteiden kiinnitysteline	30
Kuvio 23. Testikonfiguraatio-tiedosto	33
Kuvio 24. Esikatselun ja kasvojentunnistuksen layout	33
Kuvio 25. Oma kuuntelija kameran toiminnoille	34
Kuvio 26. Salama-tilan määrittäminen.....	35
Kuvio 27. Valotuksen kompensointin määrittäminen.....	36
Kuvio 28. Esikatselukuvan resoluution määrittäminen	36
Taulukko 1. Androidin eri versiot ja julkaisuajat (Hildenbrand 2013.).....	14
Taulukko 2. LG Nexus 5 -puhelimien ominaisuudet (PhoneArena 2015.)	31

Käytetyt termit ja lyhenteet

3GPP	Videon pakkaukseen käytetty formaatti.
AAC	AAC mahdollistaa häviämättömän äänen pakkauksen.
AAC-ELD	Mahdollistaa maksimaalisen äänenlaadun erittäin alhaisella koodausviiveellä.
ADT Bundle	Eclipsen lisäosa, joka mahdollistaa Android-sovellusten kehittämisen.
Algoritmi	Yksityiskohtaista kuvausta toimintaohjeista, joiden avulla useasta erillisestä työvaiheesta koostuva tehtävä saadaan suoritettua.
AMR-NB	AMR-NB on ääniformaatti optimoituun puheen pakkaukseen.
AMR-WB	Ääniformaatti optimoituun puheen pakkaukseen, jossa on tavallista laajempi puhekaista.
API	Ohjelmointirajapinta, jolla on mahdollista kutsua toisen ohjelman rajapintaa.
Arduino	Avoimeen lähdekoodiin perustuva mikrokontrolleri kehitysalusta.
C-kieli	Ohjelmointikieli, joka on tarkoitettu järjestelmäohjelmointiin, mutta voidaan käyttää myös sovellusohjelmointiin.
Dalvik-virtuaalikone	Android-käyttöjärjestelmän osa, joka vastaa sovellusten suorittamisesta.
Dekooderi	Purkaa tai erottelee jonkin algoritmin mukaan pakattua tietoa alkuperäiseen muotoon esittämistä varten.

Eclipse	Kehitysympäristö Java-ohjelmointikielelle.
Enkooderi	Pakkaa tai salaa tiedon jonkin algoritmin mukaisesti, jolloin tieto voidaan tallentaa tai siirtää helpommin.
H263	Videonpakkausstandardi, joka suunniteltiin matalan bittinopeuden pakkaamiseen.
H264	Videonpakkausstandardi, joka pystyy tarjoamaan hyvän videon laadun pienemmällä bittinopeudella.
HAL	Vastaa kamerarajapinnan toiminnasta ajureiden ja laitteiston kanssa.
HDR	HDR tuottaa kuvan, jonka sävyalue on laajempi kuin mitä yhdellä valotuksella voidaan saada aikaan.
HE-AAC	Häviöllinen tiedonpakkaus, joka on optimoitu matalan bittinopeuden sovelluksille.
IDE	Ohjelmointiympäristö on tekstieditori ja ohjelmointikielen kääntäjä.
ISO	Valoherkkyys, kuvaa linssin herkkyyttä eli kykyä reagoida valoon.
Java	Oliopohjainen ohjelmointikieli.
Linux	Avoimeen lähdekoodiin perustuva käyttöjärjestelmä.
Mikrokontrolleri	Mikrokontrolleri eli IC-piiri, jossa on mikroprosessori, muistia ja liityntälohkoja. Käytetään sulautetuissa järjestelmissä.
MPEG4	Videonpakkausstandardi, joka aluksi suunniteltiin matalan bittinopeuden videoille. Laajennettiin myöhemmin sopimaan myös muille videoille.
Native Development Kit	

Mahdollistaa Android-ohjelmien kehityksen C- ja C++-kieltä käyttäen.

RAW

Raakakuva-tiedostomuoto, jossa resoluutio on sama kuin kameran kennon resoluutio.

Referenssi

Työssä referenssikuva on mallikuva, jossa tiedetään mitä kuva sisältää ja siihen verrataan kaapattua kuvaa.

Regressio

Regressio on menetelmä, jonka avulla tarkastellaan vastemuuttujan riippuvuutta valituista selittävästä muuttujista.

SDK

Software Development Kit on tyypillisesti joukko työkaluja, joilla voidaan luoda sovelluksia tietyille kehitysalustoilla.

Sekvenssi

Sekvenssi on kaavio, joka kuvaa olioiden välistä vuorovaikutusta ajan kuluessa.

SoMA

Sofica Oy:n kehittämä automaattinen kamerantestausjärjestelmä.

Ubuntu

Linux-pohjainen käyttöjärjestelmä.

VORBIS

Patentiton ja lisenssimaksuton häviöllinen äänenpakkausmenetelmä, joka pohjautuu avoimeen lähdekoodiin.

VP8

Avoimeen lähdekoodiin perustuva videonpakkausmenetelmä.

Wi-Fi

Wi-Fi on langaton lähiverkkotekniikka.

1 JOHDANTO

1.1 Työn tausta

Android-matkapuhelimia on valmistettu vuodesta 2008 lähtien ja niitä on ollut markkinoilla siitä lähtien. Android-matkapuhelinvalmistajia on maailmalla monia, esimerkiksi Samsung, Sony ja HTC.

Android-käyttöjärjestelmästä tulee uusia versioita keskimäärin kerran vuodessa. Pienempiä versiopäivityksiä tulee aina tarvittaessa.

Työ tehtiin Sofica Oy:lle, koska yritys tarvitsi uuden testiohjelman SoMA automaattiseen kameratestaussjärjestelmään. Työn toteutustavaksi valittiin kokonaan uusi testiohjelma suuren Android-käyttöjärjestelmän päivityksen takia. Aina uuden version kanssa tulee testata testiohjelman toiminta sekä tutkia uudet ominaisuudet ja muokkaukset.

Sofica Oy tarjoaa matkapuhelinvalmistajille automaattista kameratestaussjärjestelmää. Android-käyttöjärjestelmän päivityksen myötä vanha testiohjelmisto ei ollut enää ajanmukainen uusille puhelimille. Uudet puhelimet käyttävät Android-versioita 5.0 tai uudempaa. Uusi kamerarajapinta tuo vakautta ja nopeutta kameran toimintaan.

1.2 Työn tavoite

Työn tavoitteena oli saada uusi valmis testiohjelma SoMA automaattiseen kameratestaussjärjestelmään. Uuden testiohjelman täytyi pystyä kommunikoimaan SoMA-järjestelmän kanssa samalla tavalla kuin vanha testiohjelman kommunikoi.

1.3 Työn rakenne

Luvussa 2 kerrotaan Android-käyttöjärjestelmästä sekä käydään läpi Androidin historiaa ja Androidin eri versiot. Luvussa tutustutaan Android 5.0 Lollipop -versioon ja käydään läpi muutamia sen uusia ominaisuuksia.

Luvussa 3 kerrotaan Android-käyttöjärjestelmän kamerarajapinnat koodillisesti sekä kerrotaan lyhyesti HAL-järjestelmästä.

Luvussa 4 kerrotaan Sofica Oy kehittämästä SoMA automaattisesta kameratestaajajärjestelmästä. SoMA automaattisesta kameratestaajajärjestelmästä käydään läpi eri testiominaisuuksia ja sen testiympäristöä.

Luvussa 5 keskitytään vanhaan sekä uuteen testiohjelmaan. Lisäksi on kerrottu kehityslaitteisto ja testeissä käytetty matkapuhelin.

Luvussa 6 on yhteenveto ja tulokset. Luvussa käydään läpi ongelmia työn teossa ja työn onnistuminen.

1.4 Yritysesittely

Sofica Oy on perustettu vuonna 2009. Yritys on erikoistunut sulautettujen kameralaitteiden automaattisten testausjärjestelmien kehittämiseen sekä kameralaitteiden vertailemiseen. Yritys kehittää ja markkinoi SoMA automaattista kameratestaajajärjestelmää. Työntekijöitä yrityksessä on tällä hetkellä 5 henkilöä. Yrityksen päätoimi paikka on Seinäjoella, jonka lisäksi yrityksellä on myyntiedustaja Aasiassa. (Myllyperkiö 2015.)

2 ANDROID-KÄYTTÖJÄRJESTELMÄ

Android on Linux-pohjainen avoimeen lähdekoodiin perustuva ohjelmistopino (Compile Time Error 2013). Android Inc. yritys perustettiin vuonna 2003 ja sen tarkoitus oli kehittää entistä älykkäämpiä mobiililaitteita. Elokuussa 2005 Google osti Android Inc. -yrityksen. Vuonna 2007 joukko matkapuhelinalan yrityksiä perusti Open Handset Alliancen (OHA). OHA:n tavoitteiden ensimmäinen tulos oli Android. Android on lisensoitu kokonaan avoimen Apache Licencen alaiseksi. Marraskuussa 2007 julkaistiin Androidista ensimmäinen versio. Lokakuussa 2008 julkaistiin ensimmäinen Android-puhelin. (Harju 2013, 11-12.)

Suurin osa Android-sovelluksista on ohjelmoitu Javalla. Javan byte-koodia ei kuitenkaan tueta Androidissa. Jotta Android voi suorittaa Java-koodia, täytyy Java-koodi esikäntää Dalvik-tiedostoiksi. Native Development Kitin avulla on mahdollista toteuttaa ohjelmia C-kieltä käyttäen. Androidille on suunniteltu oma Dalvik-virtuaalikone, ja se on optimoitu akkuvirtaa käyttäville mobiililaitteille, joissa on muistia rajallinen määrä ja prosessoritehoa. Java-sovellus Androidissa ajetaan omassa prosessissaan. (Harju 2013, 13-14.)

Android on vastakohta aiemmille suljetuille tai vain osittain avoimille käyttöjärjestelmille. Avoimuus mahdollistaa käyttöjärjestelmän täyden muunneltavuuden ja oman käyttöjärjestelmän tekemisen. (Harju 2013, 15.)

Eclipse IDE oli ensimmäinen kunnollinen kehitysympäristö Androidille. Nykyään löytyy myös muita ympäristöjä Android kehitykseen. Kehitysympäristön voi ladata suoraan Androidin kehittäjä sivuilta. Kehittäjä sivuilta voi ladata suoraan ADT Bundle -paketin, joka sisältää kaiken tarvittavan sovelluksen kehitykseen. (Harju 2013, 17-18.)

Androidin vähimmäisvaatimukset ovat:

- Home-, Menu- ja Back-painikkeet
- 240x320-resoluutiolla oleva kosketusnäyttö
- virtuaalinen näppäimistö
- 2-megapikselin kamera
- 92 Mb keskusmuistia

- 150 Mb sisäistä muistia
- tuki Wi-Fi-yhteyttä varten. (Dolcourt 2010.)

Taulukossa 1 (Hildenbrand 2013) on esitetty kaikki Android-versiot ja niiden julkaisuajat. Jokaisella merkittävällä Android päivityksellä on jälkiruokaan perustuva nimi.

Taulukko 1. Androidin eri versiot ja julkaisuajat (Hildenbrand 2013.)

Android-versio	Version nimi	Julkaisu aika
Android 1.5	Cupcake	Huhtikuu 2009
Android 1.6	Donut	Lokakuu 2009
Android 2.0/2.01/2.1	Eclair	Marraskuu 2009
Android 2.2	Froyo	Toukokuu 2010
Android 2.3	Gingerbread	Joulukuu 2010
Android 3.0	Honeycomb	Helmikuu 2011
Android 4.0	Ice Cream Sandwich	Joulukuu 2011
Android 4.1-4.3	Jelly Bean	Kesäkuu 2012
Android 4.4	KitKat	Lokakuu 2013
Android 5.0	Lollipop	Lokakuu 2014

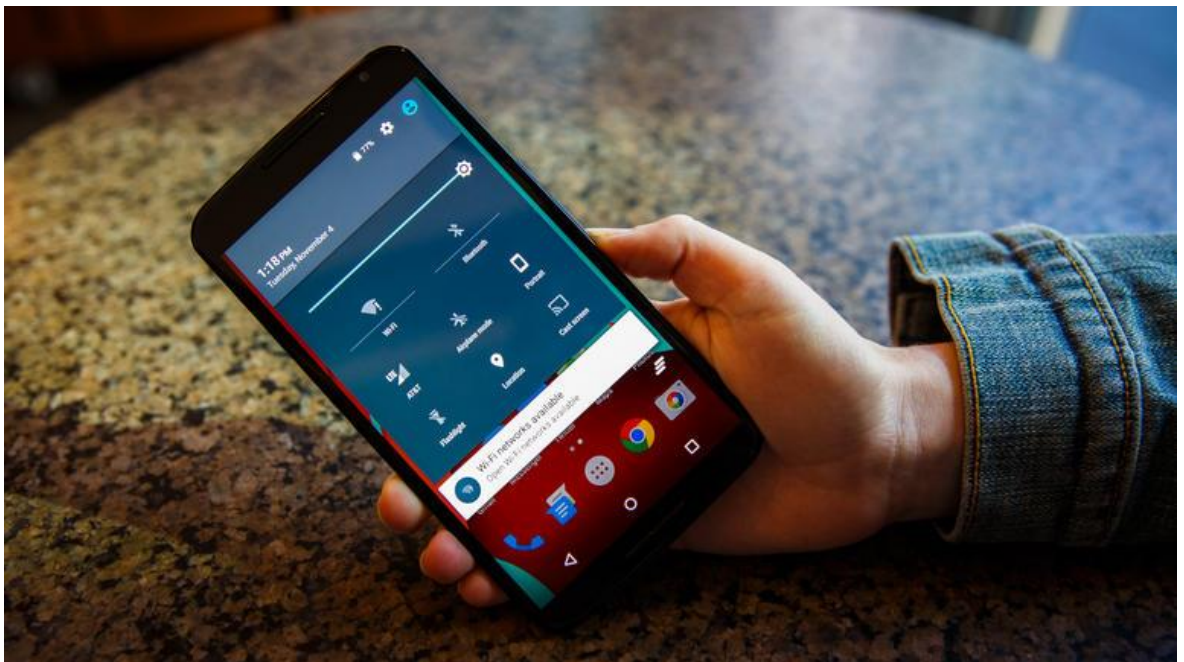
Android-älypuhelimien markkinaosuus on noussut joka vuosi. Vuonna 2011 Androidin markkinaosuus oli 57 prosenttia. Vuonna 2012 se oli 74 prosenttia. Vuonna 2013 se oli 81 prosenttia. Vuonna 2014 Androidin osuus oli jo 84 prosenttia. (IDC 2015.)

2.1 Android 5.0 Lollipop -versio

Kesäkuussa 2014 Google ensiesitteli uuden Android 5.0 Lollipop -version. Lokakuussa Google julkaisi uudet Nexus-laitteet, jotka käyttävät uutta Android 5.0 Lollipop -versiota. Vanhemmat Nexus-laitteet saivat myös Lollipop-päivityksen. (Laakso 2014.)

Ilmeisin muutos on tapahtunut Android-käyttöliittymän muotoilussa, jota kutsutaan nimellä Material. Material-muotoilu muutti Android-käyttöjärjestelmä ulkoasun kokonaan. Esimerkiksi asetukset- ja ohjelma-valikot ovat täysin uudistettu. Toinen tärkeä teema on kortit. Kortit ovat tehty suosittuun Google Now -palveluun. Kortit ilmoittavat esimerkiksi paikannuksen avulla suosittuja nähtävyyksiä lähialueelta. (Mitroff 2015.)

Uudessa Android-versiossa animaatioita on joka puolella. Esimerkiksi pyyhkäisemällä näyttöä alas saa esiin yläreunan ilmoitusvalikon, jolloin esiintyy laajennus- ja tihennysefektejä. (Mitroff 2015.)



Kuvio 1. Android 5.0 -version pika-asetukset-valikko (Miller 2014.)

Uusi tärkeys-tila mahdollistaa äänettömän-tilan, jossa on mahdollista valita yhteystietoja tai hälytyksiä, joilla on oikeus hälyttää myös äänettömässä-tilassa. Ei mitään

-tilassa millään ohjelmalla ei ole oikeutta hälyttää millään tavalla. (Mitroff 2015.)

Lollipop-versiossa moniajo-ominaisuus sai muodonmuutoksen. Edellisissä versioissa viimeaikaiset sovellukset -toiminnolla on Lollipop-versiossa uusi ominaisuus, jota kutsutaan *yleistä*-toiminnoksi. Yleistä-toiminto osoittaa ohjelmia, jotka ovat käynnissä taustalla. (Mitroff 2015.)

Android Lollipop -versio tuo mukanaan muutamia uusia lukitusmenetelmiä, joiden avulla on helpompaa pitää laite turvassa. Lollipop tarjoaa uuden Smart Lock -ominaisuuden. Ominaisuus toimii siten, että se käyttää laiteparina esimerkiksi älykelloa, joka antaa luvan avata laitteen. Jos lukitusominaisuus on aktivoitu laitteesta ja älykelloa ei ole lähettyvillä laite pyytää lukituksen avaukseen käytetyn PIN-koodin tai muunlaisen yhdistelmän laitteen avaamista varten. (Etherington 2014.)

Uudessa päivityksessä on myös mukana täysin uusi kamerarajapinta. Uuden kamerarajapinnan avulla kehittäjien sovellukset pääsevät käyttämään laitteiston kamera-asetuksia paremmin. (Whitwam 2014.)

Uudella kamerarajapinnalla on mahdollista kaapata täyden resoluution kuvia. Kamerarajapinta myös parantaa kamerasuorituskykyä ja tuo uusia ominaisuuksia, esimerkiksi RAW-kuvan kaappauksen, HDR-tilan ja manuaalisen ohjauksen kamera-asetuksille. Manuaaliohjauksella voi muun muassa kontrolloida:

- valotusaikaa
- ISO-herkkyyttä
- linssin tarkennuksen etäisyyttä
- salaman käyttöä
- värienkorjaus matriisia
- automaattisen valotuksen, valkotasapainon tai tarkennuksen tiloja
- automaattisen valotuksen ja valkotasapainon lukitusta. (Toombs 2014.)

3 ANDROID-KAMERA

Android sisältää tuen eri kameroille ja ominaisuuksille, joiden avulla voi kaapata kuvia ja videoita. Ennen kuin ohjelma pystyy käyttämään kameraa, pitää kameralle antaa oikeudet. Ohjelmien oikeudet määritetään manifest-tiedostoon. (Android API Guides [Viitattu 8.2.2015].)

```
<uses-permission android:name="android.permission.CAMERA" />
```

Kuvio 2. Kameraoikeuksien määrittely
(Android API Guides [Viitattu 8.2.2015].)

Manifest-tiedostossa on ilmoitettava kameran käytöstä (Android API Guides [Viitattu 8.2.2015]).

```
<uses-feature android:name="android.hardware.camera" />
```

Kuvio 3. Ilmoitus kameran käytöstä
(Android API Guides [Viitattu 8.2.2015].)

3.1 Kamera API

Vanhassa rajapinnassa kameran käyttöönotto aloitetaan määrittelemällä SurfaceHolder-luokka. SurfaceHolderille luodaan oma luokka, jossa kaikki kameran esikatselutoiminnot suoritetaan. SurfaceHolderi on abstrakti käyttöliittymä, joka pitelee näytön pintaa. SurfaceHolder mahdollistaa pinnan koon ja muodon muokkauksen. (Android Developers 2015.) SurfaceHolderin määrittely suoritetaan kuvio 4 perusteella.

```
SurfaceHolder surfaceHolder = this.getHolder();  
surfaceHolder.addCallback(this);  
surfaceHolder.setType(SurfaceHolder.SURFACE_TYPE_PUSH_BUFFERS);
```

Kuvio 4. SurfaceHolderin määrittely

Kun SurfaceHolder on luotu, alustetaan kamera ja asetetaan luotu SurfaceHolder kameran esikatselun alustaksi.

```
Camera camera = Camera.open();
camera.setPreviewDisplay(surfaceHolder);
```

Kuvio 5. Kameran alustus

Seuraavaksi määritetään esikatselun koko. Esikatselun koon määrittäminen tapahtuu kameran parametreissa. Koon määrittämisen jälkeen käynnistetään kameran esikatselu puhelimen näytölle.

```
Camera.Parameters cameraParameters = camera.getParameters();
cameraParameters.setPreviewSize(width, height);
camera.setParameters(cameraParameters);
camera.startPreview();
```

Kuvio 6. Kameran esikatselun aloitus

Omasta ohjelmasta poistuessa täytyy kamera vapauttaa muille ohjelmille. Jos kameraa ei vapauteta, kamera jää taustalle päälle, eikä sitä pysty käyttämään ennen kuin se on suljettu.

```
camera.stopPreview();
camera.release();
camera = null;
```

Kuvio 7. Kameran sulkeminen

3.2 Uusi kamera API

Uudessa kamerarajapinnassa luodaan Preview-luokka, joka laajennetaan TextureView-luokalla. Preview-luokassa määritellään konstruktori. Preview-luokassa tehdään kaikki kameran esikatseluun tarvittavat muokkaukset ja komennot.

```
public class Preview extends TextureView{
    .....
    public Preview(Context context){
        this(context, null);
    }
}
```

Kuvio 8. Uuden kameran Preview-luokka

Kun Preview-luokka on luotu, alustetaan Preview-luokka pääluokassa. Luotu Preview-luokka liitetään RelativeLayouttiin, joka määritetään ohjelman layoutiksi.

```
RelativeLayout relativeLayout = new RelativeLayout(this);
Preview preview = new Preview(this);
relativeLayout.addView(preview);
setContentView(relativeLayout);
preview.setSurfaceTextureListener(surfaceTextureListener);
```

Kuvio 9. Esikatselun määrittäminen pääluokassa.

Kuviossa 10 on esimerkki kameran avauksesta. Avaus tapahtuu CameraManager-luokkaa käyttäen. Ennen kuin voidaan avata kamera, täytyy esikatselulle määrittää koko. Esikatselun koko määritetään valitsemalla optimaalinen koko sallituista ko'osta. Esikatselulle täytyy myös määrittää kuvasuhde. Seuraavaksi kamera avataan CameraManageria käyttäen.

```
CameraManager manager = (CameraManager) this
    .getSystemService(Context.CAMERA_SERVICE);
CameraCharacteristics characteristics
    = manager.getCameraCharacteristics("0");
StreamConfigurationMap map = characteristics
    .get(CameraCharacteristics
        .SCALER_STREAM_CONFIGURATION_MAP);

Size largest = Collections.max(
    Arrays.asList(map.getOutputSizes(ImageFormat.JPEG))
    , new CompareSizesByArea());
previewSize = chooseOptimalSize(
    map.getOutputSizes(SurfaceTexture.class),
    width, height, largest);
preview.setAspectRatio(previewSize.getWidth(),
    previewSize.getHeight());
manager.openCamera("0", stateCallback, backgroundHandler);
```

Kuvio 10. Kameran avaus

Uudessa kamerarajapinnassa esikatselukuva suoritetaan tausta-ajossa. Kameran avauksen jälkeen määritellään esikatselu.

```

SurfaceTexture texture = preview.getSurfaceTexture();
assert texture != null;
texture.setDefaultBufferSize(
    previewSize.getWidth(), previewSize.getHeight());
Surface surface = new Surface(texture);
previewBuilder.addTarget(surface);
cameraDevice.createCaptureSession(surface, stateCallback, backgroundHandler);

```

Kuvio 11. Esikatselun asetusten määrittely

Onnistuneen esikatselun asetusten määrittämisen jälkeen voidaan käynnistää esikatselu. Esikatselun käynnistäminen tapahtuu CameraCaptureSession-luokalla. CameraCaptureSession-luokalle annetaan setRepeatingRequest-komento.

```

cameraCaptureSession.setRepeatingRequest(previewBuilder,
    captureCallback, backgroundHandler);

```

Kuvio 12. Esikatselun käynnistäminen

Kameran sulkeminen tapahtuu sulkemalla CameraCaptureSession- ja CameraDevice-luokka.

```

cameraCaptureSession.close();
cameraDevice.close();

```

Kuvio 13. Kameran sulkeminen

3.3 Kamera HAL v3

HAL yhdistää korkeatasoisen kamerarajapinnan kameran ajureihin ja laitteistoon. Ohjelman kehys jakaa pyyntöjä kameran alijärjestelmälle. Alijärjestelmä palauttaa yhtä pyyntöä kohden yhdet tulokset. Kehyspyyntö kiteyttää kaikki asetukset kaappausta varten ja näiden tulosten käsittelyn. (Android Devices [Viitattu 30.3.2015].)

Kameran alijärjestelmä sisältää toteutuksen kameran komponenteille kuten 3A-algoritmeille ja ohjauksen käsittelyn. Kamera HAL tarjoaa rajapinnan omalle toteutukselle. Eri laitevalmistajien ajureiden välillä on suuria eroja, jolloin HAL-järjestelmän tarjoama rajapinta toimii kaikilla valmistajilla. (Android Devices [Viitattu 30.3.2015].)

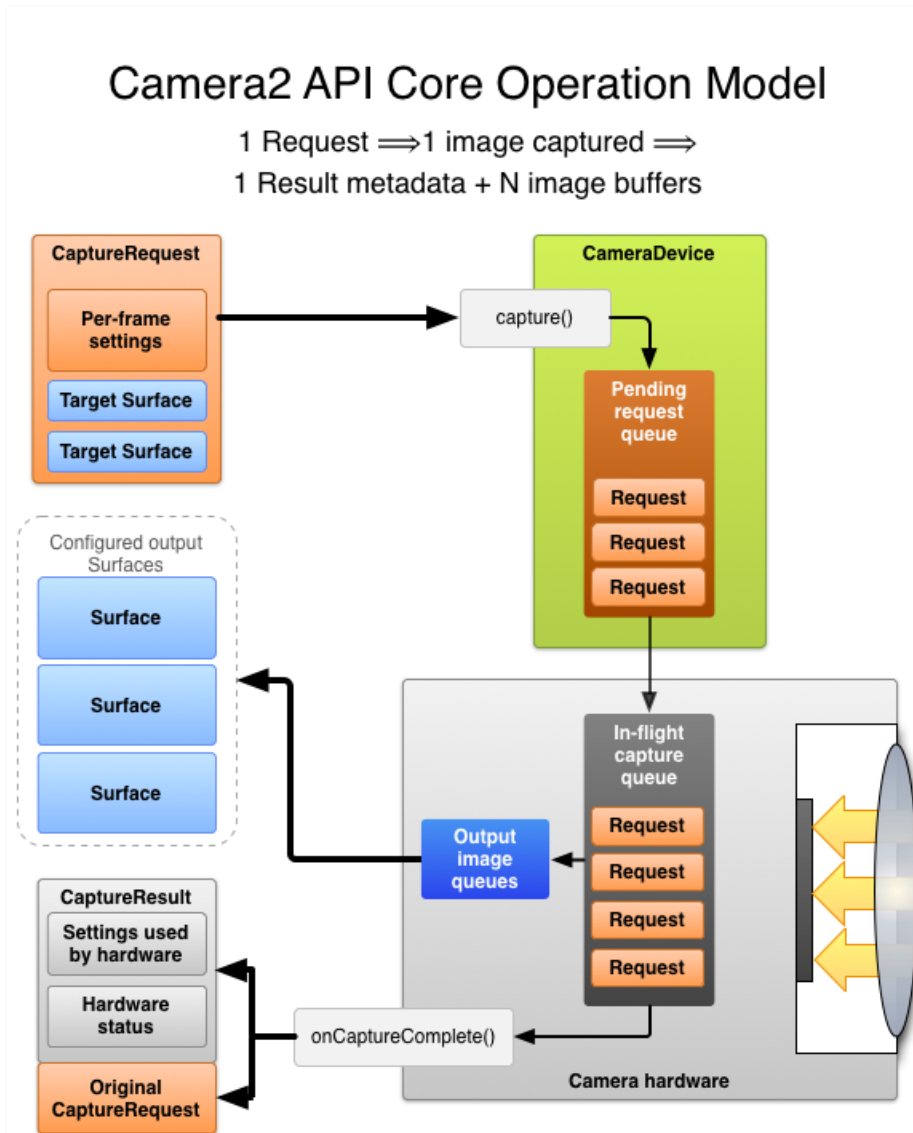
Kamera HAL -järjestelmään toteutetulla laitteella on toimintamalli joko rajoitettu tai täydellinen. HAL ilmoittaa tukitason numeroin, jolloin 0 tarkoittaa rajoitettua tukea ja 1 tarkoittaa täydellistä tukea. Rajoitetussa tuessa laitteen ei tarvitse toteuttaa tarkkoja synkronointeja kaappauksen pyynnöissä. (Android Devices [Viitattu 30.3.2015].)

Android-käyttöjärjestelmän päivityksen mukana tuli uusi kamerarajapinta. Kamerarajapinta oli uudelleensuunniteltu, jonka tavoitteena oli lisätä merkittävästi sovelluksien kykyä ohjata kameraa. Uudelleenjärjestely tekee kamerarajapinnasta tehokkaamman ja ylläpidettävämmän. (Android Devices [Viitattu 13.2.2015].)

Lisäohjaus helpottaa laadukkaan kameran sovelluksen rakentamista Android-laitteisiin. Lisäohjaus tuo luotettavuutta laitekohtaisiin algoritmeihin sekä suorituskyvyn ja laadun maksimoimiseksi. (Android Devices [Viitattu 13.2.2015].)

Versiossa 3 on muokattu kameran alijärjestelmän rakenteiden toimintatilat yhdeksi yhtenäiseksi näkymäksi, jota voidaan käyttää toteuttamaan edelliset tilat ja useita muita, kuten sarjakuvaus. Tämä johtaa käyttäjän parempaan ohjaukseen tarkennuksessa, valotuksessa ja jälkikäsittelyssä, kuten kohinan vähentämisessä, kontrastissa ja terävöinnissä. Yksinkertaistettu näkymä helpottaa sovelluksen kehittäjiä käyttämään kameran toimintoja. (Android Devices [Viitattu 13.2.2015].)

Kuviossa 14 (Android Devices [Viitattu 13.2.2015]) on kuvattu uuden kamerarajapinnan ytimen toimintamalli. Rajapinnan ytimen toimintamalli on yksinkertaistettuna seuraava: sovelluskehys pyytää kehyksen kameran alijärjestelmältä ja kameran alijärjestelmä palauttaa tulokset ulostulossa. (Android Devices [Viitattu 13.2.2015].)



Kuvio 14. Kameran ytimen toimintamalli
(Android Devices [Viitattu 13.2.2015].)

4 SoMA AUTOMAATTINEN KAMERATESTAUSJÄRJESTELMÄ

SoMA on automaattinen sulautettujen kameralaitteidentestausjärjestelmä. SoMA-järjestelmän tarkoitus on löytää tuotekehitysvaiheen ongelmat ja virheet mahdollisimman aikaisessa vaiheessa. SoMA-järjestelmä nopeuttaa tuotteen valmistumista sekä parantaa sen laatua. SoMA-järjestelmä on täysin automatisoitu, jolloin ihmisen vuorovaikutuksesta syntyvää virhettä ei pääse syntymään. (Sofica 2014a.)



Kuvio 15. SoMA automaattinen kameratestaajärjestelmä (Sofica 2014a.)

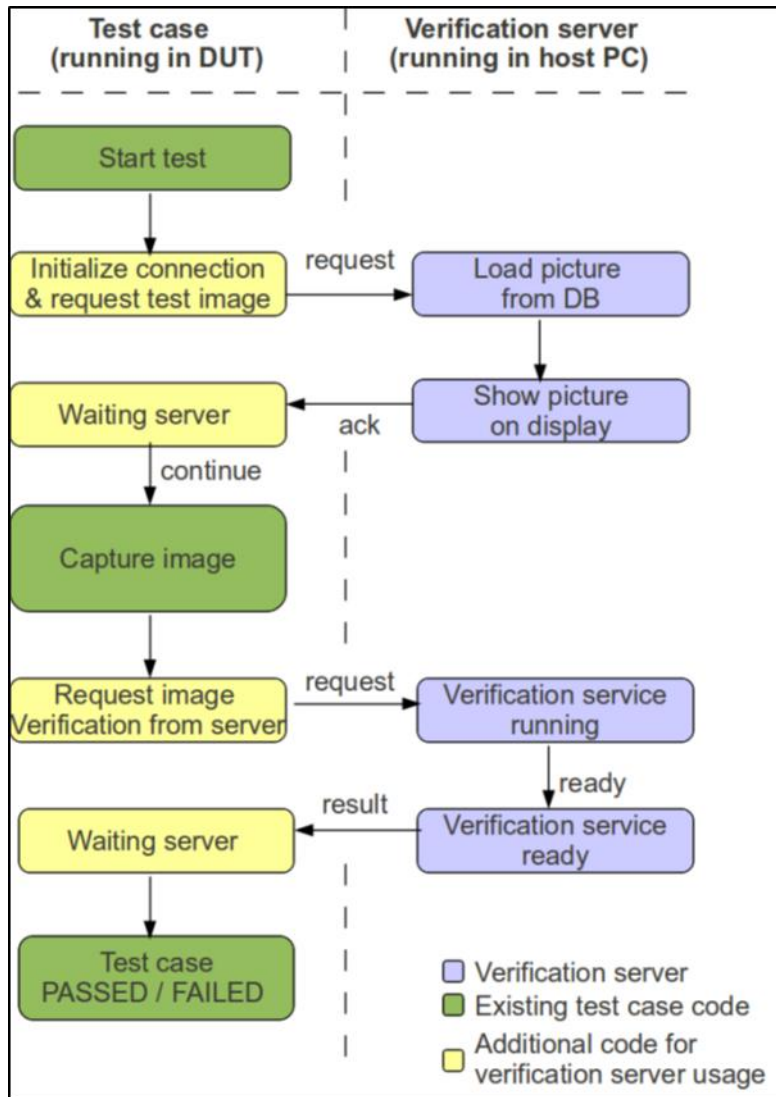
Järjestelmä sisältää useita erikseen valittavissa olevia testaustasoja. Testaukseen voidaan valita eri yhdistelmiä tasoista, jolloin saadaan nykyiseen testirakenteeseen sopiva testausmenetelmä. (Sofica 2014b.)

Kuviossa 16 (Sofica 2014b) on esitetty SoMA-järjestelmän eri testaustasot. Eri testaustasot tutkivat puhelimen järjestelmän heikkouksia eri testein (Sofica 2014b).



Kuvio 16. SoMA:n testaustasot
(Sofica 2014b.)

Kuviossa 17 (Sofica 2012b) on esitetty suoritusajon sekvenssi. Testin alkaessa ohjelma alustaa yhteyden palvelimeen ja pyytää referenssikuvan näytölle. Referenssikuvan avautuessa näytölle palvelin antaa testiohjelmalle luvan jatkaa testausta. Testiohjelma suorittaa halutut testaukset. Suoritusten loputtua ohjelma ilmoittaa palvelimelle testin suorituksesta. Palvelin suorittaa tarvittavat tarkistukset tuloksille ja ilmoittaa testin tulokset. (Sofica 2012b.)



Kuvio 17. Suoritusajon sekvenssi (Sofica 2012b.)

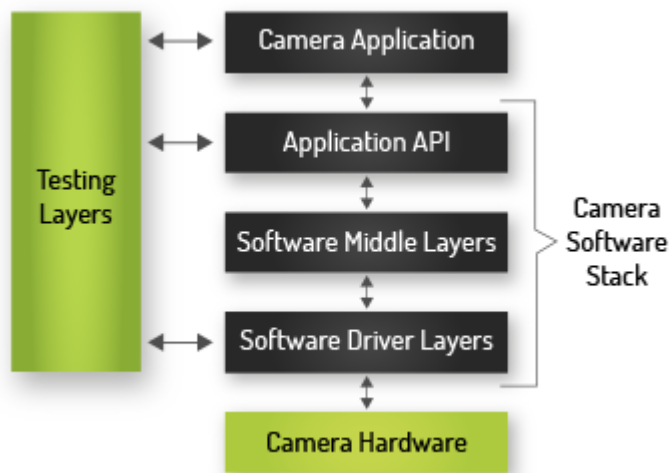
4.1 API-testaus

API-testaus sisältää noin 200 erilaista testiä ja se voidaan suorittaa kaikkia rajapintoja vastaan kaikilla alustoilla. API-testaus sisältää kolme erilaista testaustapaa. (Sofica 2014b.)

Parametritestauksessa syötetään valinnan pienin ja suurin mahdollinen arvo sekä mahdollisesti satunnaisia arvoja. Lisäksi negatiivinen testaus tehdään syöttämällä laiton tulo parametrina, jolloin tarkistetaan, että oikea virhekoodi palautetaan. (Sofica 2014b.)

Seuraavassa testauksessa tarkastellaan tulojen välistä vastaavuutta. Testaus varmistaa, että syötetyt arvot palautuvat oikeana. Myös oikea palautus virhetilanteessa tarkistetaan. (Sofica 2014b.)

Viimeisessä vaiheessa testaus osoittaa järjestelmän vakauden, kun laittomat menetelmät suoritetaan (Sofica 2014b).



Kuvio 18. API-testaus
(Sofica 2014b.)

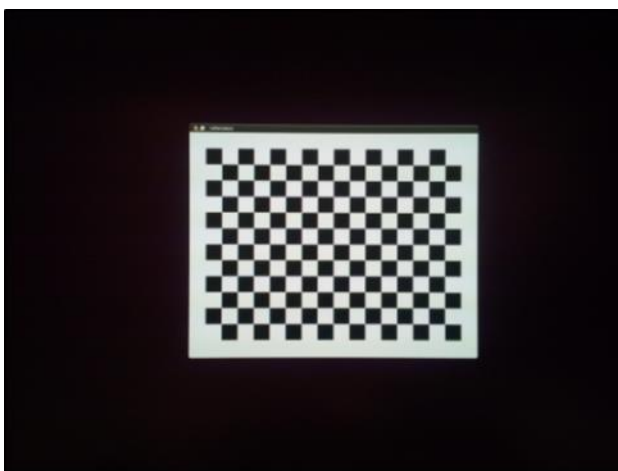
4.2 Toiminnallinen testaus (functional testing)

Toiminnallisen testauksen algoritmi vaatii kuvan talteenoton todellisessa kuvasta. Testausmenetelmä käyttää matemaattista analyysia havaitakseen erot alkuperäisen kuvan ja talteen otetun kuvan välillä. Toiminnallisen testauksen päätarkoitus on tarkistaa oikea toiminnollisuus ja mahdollistaa toistettavia testauksia ja mittauksia eri ohjelmistoversioiden välillä. (Sofica 2014b.) Kuviossa 19 (Sofica 2014b) on kuvattu laatikkonäkymin toiminnallisen testauksen eri testitapoja.

Image Resolution	Auto Focus	Auto Exposure Compensation
Video Resolution	Auto Exposure	
Dynamic Range	Auto White Balance	Image Content Verification
Color Effects	Face Detection	
IR Hotspot	Smile Detection	Audio / Video Synchronization
Flash	Zoom	Blink Detection
Image EXIF Data Correctness	Video Content Verification	

Kuvio 19. Toiminnallinen testaus
(Sofica 2014b.)

Laitteiston kalibrointi suoritetaan ennen haluttuja testauksia. Kalibroinnin tarkoituksena on saada seuraavien testien referenssikuvat oikean kokoisiksi ja oikeaan paikkaan näytölle. Kalibrointitestissä ei ole mahdollista muuttaa asetuksia. Kalibrointitesti tuo näytölle shakkilaudan, josta laitteella otetaan kuva. Kuvasta etsitään shakkilaudan reunat, josta palvelin laskee oikean koon kameran reunoille. Laskujen jälkeen näytetään seuraavan testin referenssikuva oikeassa koossa laitteiston kameran näkymäkenttään nähden. Jos palvelin ei pystynyt määrittämään shakkilaudan paikkaa, ilmoitetaan käyttäjälle virheestä ja lopetetaan testien suorittaminen. (Sofica 2012a.)



Kuvio 20. Kalibrointitestin shakkilautakuva
(Sofica 2012a.)

4.3 Suorituskykytestaus (performance testing)

Suorituskykytestaus paljastaa pullonkaulat kamerajärjestelmässä mittaamalla suoritusajoja kameran eri toiminnoista ja toimintojen yhdistelmistä. Testi luo aikaleiman, kun kameran rajapintaa kutsutaan ja kun vastaus on saatu rajapinnasta. Suoritus aika on näiden kahden ajan erotus. Testi myös mittaa jokaisen kameran toiminnon suoritusajan. Suorituskykytesti ei vaadi alkuperäisiä kuvia, mutta ympäristö vaikuttaa tuloksiin. Alla on muutamia esimerkkejä tuetuista testeistä:

- Kameran avausaika, mittaa kuinka kauan menee, kun kamera on alustettu.
- Automaattitarkennus, mittaa kuinka kauan menee, kun kamera on tarkennettu kohteeseen. (Sofica 2014b.)

4.4 Rasiustestaus (stress testing)

Rasiustestin tavoitteena on tunnistaa kameran suorituskyky testaamalla luotettavuutta ja järjestelmän vakautta. Lisäksi se paljastaa kumulatiivisia virheitä järjestelmästä ja analysoi muistin käyttöä. Satunnainen testaus on yksi tapa etsiä odottamattomia vikoja. Erilaiset testit suoritetaan satunnaisesti kamerajärjestelmässä ja saadut testitulokset tallennetaan. Jos vika havaitaan, tallennettu testi voidaan suorittaa uudelleen, jolloin vika voidaan analysoida tarkemmin. Jos järjestelmä kaatuu, testi tallentaa kaatumisen tiedot automaattisesti. (Sofica 2014b.)

4.5 Laadun regressiotestaus (quality testing)

Testaus mahdollistaa standardeihin perustuvat kamera- ja videotestausalgoritmit. Laskennallinen laadun analysointi takaa, että kameran ja videoiden käsittelyalgoritmit testataan objektiivisesti, luotettavasti ja ovat toistettavissa jokaisessa vaiheessa. (Sofica 2014b.)

Testauksen algoritmit auttavat hienosäätämään kuvakaappaukset. Standardoitu testikaavio antaa absoluuttisen ja vertailukelpoisen tiedon laadusta. Seuraavassa on muutamia esimerkkejä tuetuista testeistä:

- Kohina, testaa kuinka paljon kohinaa on kamerajärjestelmässä.
- Värien tarkkuus, mittaa kamerajärjestelmän värintoiston.
- Resoluutio, mittaa kamerajärjestelmän todellisen resoluution. (Sofica 2014b.)

Videotesteillä on mahdollista löytää poikkeavaa käyttäytymistä enkoodereista ja dekodeereista (Sofica 2014b).

4.6 Testausympäristö

Kuviossa 21 (Sofica 2013) on SoMA-testilaatikko. SoMA-testilaatikko mahdollistaa kameran testauksen eristetyssä ympäristössä. Laatikko on kestävä ja helppo siirtää. Sähköliittimet sijaitsevat takana ja laatikon runko on alumiinia ja seinät PVC-muovia. (Sofica 2013.)

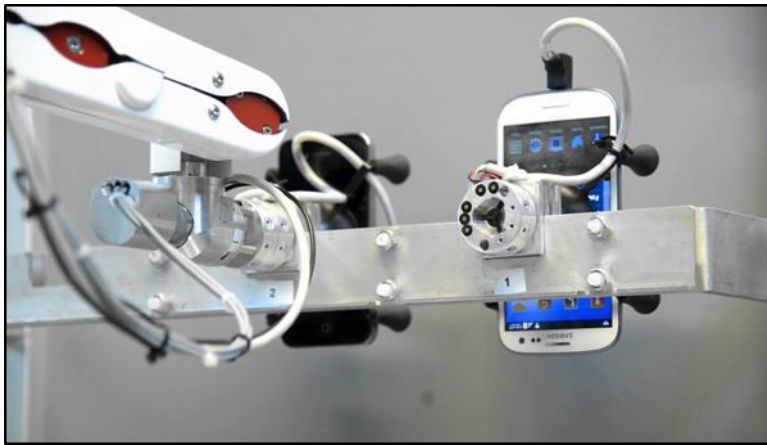


Kuvio 21. SoMA-testilaatikko
(Sofica 2013.)

Kuviossa 22 (Sofica 2013) on testilaatikon sisällä oleva robottikäsi. Robottikäden suurin ulottuvuus on 500 mm joka suuntaan, ja suurin hyötykuorma varrelle on 1kg.

Kun testit eivät ole käynnissä, robottikäsi siirretään valmiustilapidikkeeseen. Valmiustilapidikkeessä on punainen ja vihreä LED-valo, joka osoittaa, että robotti on siirretty asianmukaisesti valmiustila-asentoon. Kun testit aloitetaan, SoMA tarkistaa aina että robotti on valmiusasennossa. (Sofica 2013.)

SoMA-testilaatikko voidaan varustaa tukemaan usean laitteen testausta. Laatikko sisältää laitteille kiinnikkeen, josta robotti noutaa laitteen testaukseen. Enimmäismäärä riippuu testattavien laitteiden koosta. Kiinnike toimii myös latausasemana. (Sofica 2013.)



Kuvio 22. Laitteiden kiinnitysteline
(Sofica 2013.)

Testilaatikko sisältää näytön. Näyttöä käytetään näyttämään vertailutiedot. SoMA-järjestelmässä käytetään Asus 4K -mallia. (Sofica 2013.)

Testilaatikosta löytyy Arduino Uno -mikrokontrolleri. Mikrokontrolleri ei näy testilaatikossa, koska se sijaitsee näytön takana. Arduinoa käytetään useisiin tarkoituksiin:

- moottorin ohjaukseen
- kaiuttimen ja ledin ohjauksiin
- turvakytkimien seurantaan
- robotin virhetilojen seurantaan ja tarkistuksiin. (Sofica 2013.)

5 TESTIOHJELMA

5.1 Laitteisto

Ohjelmointi toteutettiin Ubuntu 14.04 -käyttöjärjestelmällä. Ubuntu-käyttöjärjestelmään oli asennettua Android SDK ja Eclipse-editori. Ohjelman testaus toteutettiin LG Nexus 5 -puhelimella. Nexus 5 -puhelin valittiin työhön sen valmiin Android 5.0 -julkaisun takia (Motorola Nexus 6 -puhelimta ei ollut vielä julkaistu). Taulukossa 2 (PhoneArena 2015) on kerrottu LG Nexus 5 -puhelimien ominaisuudet.

Taulukko 2. LG Nexus 5 -puhelimien ominaisuudet (PhoneArena 2015.)

LG Nexus 5	
Näytön koko	5 tuumaa
Näytön resoluutio	1080 x 1920
Näytön pikselimäärä	445 ppi
Takakamera	8 mp
Etukamera	1.3 mp
Salama	LED
Automaattitarkennus	Kyllä
Videotallennus	Kyllä
Videotallennuksen resoluutio	1080p
Suorittimen tyyppi	Qualcomm Snapdragon 800
Suorittimen nopeus	2260 MHz
Keskusmuisti	2 Gb
Sisäinen muisti	16 Gb

5.2 Vanha testiohjelma ja SoMA-palvelin

Ennen kuin uuden testiohjelman ohjelmointi voitiin aloittaa, täytyi tutkia vanhaa testiohjelmaa sekä ohjelman ja palvelimen välistä kommunikointia. Toiminnan ja helppouden takia oli tärkeää, että uusi testiohjelma osaisi käyttää samoja

testikomentoja ja muodostaa yhteyden palvelimeen niin kuin vanha testiohjelma. Testiohjelmasta saatiin selville, että muutamia valmiita luokkia voitiin käyttää suoraan uudessa testiohjelmassa. Testiohjelman ja palvelimen välistä kommunikointia ei tarvinnut muuttaa vanhasta testiohjelmasta.

Vanhan testiohjelman ominaisuudet olivat:

- Esikatselun aloitus ja lopetus
- Kuvan resoluution määrittäminen
- Tarkennuksen tilan määrittäminen
- Tarkennuksen aloitus
- Tarkennuksen alueen määrittäminen
- Kuvan talteenotto
- Kasvojentunnistus
- Valkotasapainon eri tilojen määrittäminen
- Salaman käyttö
- Valotuksen määrittäminen
- Kuvien efektien määrittäminen
- Esikatselu kuvan koon ja laadun määrittäminen
- Esikatselu kehyksien talteenotto
- Videon kaappauksen alustus
- Videon asetusten määrittäminen.

SoMA-palvelimella sijaitsee testikonfiguraatitiedostoja, jotka sisältävät testien suoritukset. Testikonfiguraatitiedostoissa on määritelty, mitä toimintoja palvelin testaa puhelimesta. Kuviossa 23 on esimerkki testikonfiguraatitiedoston sisällöstä. Palvelin lukee tiedostoa rivi riviltä. `Init`-komennolla palvelin alustaa ja käynnistää kameran puhelimesta. `Init emvs gen` -komennolla palvelin alustaa puhelimen ja palvelimen yhteyden. `Set parameters` -komennolla voidaan määrittää puhelimeen haluttuja tiloja. Kuviossa on määritelty puhelimeen kuvan koko ja tarkennuksen tila. `Start preview` -komento käynnistää puhelimen esikatselutoiminnon. Tarkennuksen kohdistaminen aloitetaan `autofocus`-komennolla. Kuva kaapataan `take picture` -komennolla. Kun halutut suoritukset on tehty, täytyy kamera- ja palvelinyhteydet sulkea. Esikatselutoiminto suljetaan `stop preview` -komennolla. `Finalize`-komennolla

palvelin sulkee puhelimen kameran sekä tallentaa tulokset ja puhelimen lokin, jos vikoja testin ajossa on ilmennyt.

```

title Resolution_3264x2448
load camtest
camtest init
camtest init_emvs_gen
camtest enable_msg_types CAMERA_MSG_ERROR
camtest set_parameters picture-size 3264x2448
camtest set_parameters focus-mode infinity
camtest start_preview 1
camtest autofocus
camtest take_picture
camtest start_emvs_verif nowait
camtest stop_preview
camtest finalize_emvs
camtest disable_msg_types CAMERA_MSG_ERROR
camtest finalize

```

Kuvio 23. Testikonfiguraatio-tiedosto

5.3 Uusi testiohjelma

Androidin 5.0-version myötä Androidin kamerapuoli koki täydellisen muutoksen. Muutos vaikutti muun muassa kameran ominaisuuksiin, komentoihin, alustamiseen ja käyttämiseen.

Testit suoritetaan käyttämällä luotuja testikonfiguraatiotiedostoja. Palvelin antaa tiedoston avulla halutut asetukset ja toiminnot kameralle.

Uudessa testiohjelmassa on luotu esikatselulle ja kasvojen tunnistukselle `FrameLayout`.

```

FrameLayout mLayout = new FrameLayout(this);
mPreview = new Preview(this);
mLayout.addView(mPreview);
mFaceView = new DrawingFace(this);
mLayout.addView(mFaceView, new FrameLayout.LayoutParams(
    FrameLayout.LayoutParams.MATCH_PARENT,
    FrameLayout.LayoutParams.MATCH_PARENT, Gravity.CENTER));

```

Kuvio 24. Esikatselun ja kasvojentunnistuksen layout

Testiohjelmaan on luotu kuuntelija, joka helpottaa suoritusten seuranta. Kuuntelija ilmoittaa palvelimelle komentojen onnistumisesta tai virheestä. Jos virhe tapahtuu,

kuuntelija ilmoittaa palvelimelle missä komennossa on virhe tapahtunut, jolloin tuloksissa on mahdollista tutkia tarkemmin mistä virhe on johtunut.

```
public interface CameraListener{
    // Callback for startPreview
    public void PreviewCB(int ok);
    // Callback for autofocus
    public void AutoFocusCB(int ok, long start, long stop);
    // Callback for jpegPictureCB
    public void JpegPictureCB(int ok, byte[] data, String dirname);
    // Callback for zoom
    public void ZoomChangeCB(int ok, int zoomValue, boolean stopped);
    // Callback for shutter
    public void ShutterCB();
    // Callback for face detection
    public void FaceDetectionCB(Face[] faces);
    // Callback for rawPictureCB
    public void RawPictureCB(int ok, byte[] data, String dirname);
}
```

Kuvio 25. Oma kuuntelija kameran toiminnoille

5.3.1 Kameratoiminnot

Esikatselulle on määritelty suurin mahdollinen koko. Kuvasuhde on määritelty puhelimen asennon mukaan. Kameran avaus suoritetaan CameraManagerista käyttäen openCamera-komentoa.

Kuvan resoluutio voidaan määrittellä kolmella eri tyylillä. Jos konfiguraatitiedostossa on määritelty kuvan resoluutio max-wide-komennolla, kuvan resoluutioksi valitaan suurin mahdollinen laajakuvakoko. Max-määrittelyllä valitaan suurin mahdollinen resoluutio. Kolmas vaihtoehto on määrittää kuvan resoluutio numeroin, esimerkiksi 3264x2448.

Tarkennuksen tiloiksi voidaan määrittellä mm. pois, automaattinen tai ääretön. Tarkennuksen manuaaliohjaus vaatii sen, että tarkennuksen tila on pois tilassa. Äärettömyys-komento määrittää tarkennuksen tilan pois-tilaksi ja tarkentaa äärettömyyteen. Tarkennuksen kohdistaminen aloitetaan autofocus-komennolla. Autofocus-komento asettaa tarkennuksen makro-tilaan. Tarkennukselle voidaan antaa myös alue mihin halutaan kohdistaa.

Valkotasapainolle on mahdollista antaa eri tiloja, jolloin valkotasapainoa voidaan testata kuvitteellisissa olosuhteissa. Valkotasapaino pyrkii korostamaan valolle vastakkaisia värejä. Automaattinen-tilassa rajapinta valitsee parhaimmat asetukset automaattisesti. Incandescent-tilassa kamera vahvistaa sinisiä ja vihreitä sävyjä. Fluorescent- ja Warm fluorescent -tiloissa valkotasapainon saavuttamiseksi fluorescent -tilassa korostetaan punaista sävyjä ja warm-tilassa taas sinistä sävyä. Daylight-tila muuntaa valkotasapainoa muistuttamaan päivänvaloa, Cloydy daylight lisää kuvaan punaisuutta valkotasapainon saavuttamiseksi. Twilight- ja Shade-tilat muuntavat valkotasapainoa sopivammaksi varjoissa ja tummassa kuvaamista varten. Pois-toiminto poistaa automaattisen valkotasapainotoiminnon ja mahdollistaa käyttäjän manuaalisen ohjauksen.

Salamalle on mahdollista määrittää eri tiloja. Single-tilassa kamera ainoastaan väläyttää salaman kuvan talteenotossa. Torch-tilassa kamera asettaa salaman päälle, mutta ei sammuta sitä automaattisesti. Automaattisessa tilassa kamerajärjestelmä määrittää itse onko salaman käyttö tarpeellista. Redeye-tilassa kamera ohjaa salamaa tarvittaessa, mutta poistaa automaattisesti punasilmäisyyden kuvasta. Kameralle on mahdollista antaa myös komento always flash, jolloin kamera käyttää aina parhaimmallaan tavalla salamaa kuvantaltteenoton yhteydessä.

```

} else if (param.contains("auto")) {
    isFlashModeTest = true;
    mPreviewBuilder.set(CaptureRequest.CONTROL_AE_MODE,
        CaptureRequest.CONTROL_AE_MODE_ON_AUTO_FLASH);
    mCameraParameters.setFlashMode("auto");
} else if (param.contains("red-eye")) {
    isFlashModeTest = true;
    mPreviewBuilder.set(CaptureRequest.CONTROL_AE_MODE,
        CaptureRequest.CONTROL_AE_MODE_ON_AUTO_FLASH_REDEYE);
    flashMode = CaptureRequest.FLASH_MODE_SINGLE;
    mCameraParameters.setFlashMode("red-eye");

```

Kuvio 26. Salama-tilan määrittäminen

Valotuksen kompensatioissa määritetään kuinka paljon kameraan pääsee valoa. Valotuksen arvo pitää olla -2 ja +2 väliltä. Valotuksen arvot määritetään askeleittain. Valotuksen askeleen arvo on 0.333.

```

public void setExposureCompensation(String param){
    try {
        isExposureCompensation = true;
        exposureCompensation = Integer.parseInt(param);
        mCameraParameters.setExposureCompensation(param);
        mPreviewBuilder.set(CaptureRequest.CONTROL_AE_EXPOSURE_COMPENSATION,
            exposureCompensation);
    } catch (Exception e) {
        isExposureCompensation = false;
        Log.d(LOGTAG, e.getMessage());
    }
}
}

```

Kuvio 27. Valotuksen kompensaaion määrittäminen

Efektejä määrittämällä on kuviin saatavissa erilaista tuntumaa. Mono-tilassa kamera määrittää kuvan näyttämään mustavalkoiselta. Kun negative-tila asetetaan kamera määrittää kuvan värit vastakohtiksi. Solarize-tilassa kamera jäljittelee kuvan altistumista valolle. Sepia-tilassa kuvasta tulee ruskean ja kullan sävyinen.

Kuvien muita eri efektejä ovat:

- Posterize
- Whiteboard
- Blackboard
- Aqua.

Esikatselukuvalla voidaan määrittää resoluutio ja laatu. Esikatselukuvan resoluutio määritetään numerollisesti esimerkiksi 512x288, jolloin testiohjelma ymmärtää tallentaa esikatselukuvan kuvankaappauksen yhteydessä. Esikatselukuvan laatu määritetään antamalla laatu prosentteina.

```

thumbnailSize = new Size(w, h);
mPreviewBuilder.set(CaptureRequest.JPEG_THUMBNAIL_SIZE,
    thumbnailSize);
mCameraParameters.setThumbnailSize(param);

```

Kuvio 28. Esikatselukuvan resoluution määrittäminen

5.3.2 Video- ja äänitoiminnot

Videon äänikanava määritetään joko mono- tai stereo-tilaan. Äänikooderin määrittäminen on tärkeää, muuten nauhoitettu video ei sisällä ääniraitaa.

Äänikooderille voidaan määrittää tiloihin: AAC, AMR-NB, AMR-WB, HE-AAC, AAC-ELD tai VORBIS. Bittinopeuden määrittäminen tapahtuu antamalla numeroin montako bittiä sekunnissa tallennetaan. Näytteenottotaajuudelle määritetään montako näytettä sekunnissa. Yleinen standardi AAC-äänelle on 8 – 96kHz. AMR-NB-äänelle on 8 kHz ja AMR-WB-äänelle on 16 kHz. Äänilähde määritetään joko default-, camcorder- tai mic-tilaan.

Videon tiedostoformaatti määritetään joko MPEG4-, 3GPP- tai default-formaattiin. Kun käytetään H263-videokooderia ja AMR-ääntä, on suositeltavaa käyttää 3GPP-formaattia, koska muut formaatit voivat sekoittaa joitain toisto-ohjelmia. Videonkooderi määritetään joko default-, H263-, H264-, MPEG4- tai VP8-tilaan. Videokooderin bittinopeus määritetään numeroin bittiä sekunnissa. Videon kehysnopeuden määrittäminen tapahtuu antamalla numeroin haluttu kehysnopeus kehyksien määrällä sekunnissa. Samoin kuin kameran kuvalle täytyy myös videolla määrittää resoluutio. Videon resoluutio määritetään antamalla leveys ja korkeus numeroin. Videonlähde on asetettava Surface-tilaan, jotta videon esikatselu on mahdollista näyttää samassa layoutissa kuin kameran esikatselua. Uusi kamerarajapinta myöskin vaatii sen, että lähde on asetettu Surface-tilaan.

Kaikki videon ja äänen asetukset pitää määrittää ennen kuin kutsutaan prepare-komentoa. Videon tallennus aloitetaan startRecorder-komennolla ja lopetetaan stopRecorder-komennolla.

6 TULOKSET JA YHTEENVETO

Tässä opinnäytetyössä toteutettiin uusi testiohjelma uudelle Android-kameran rajapinnalle. Tavoitteena oli saada toimiva testiohjelma uudelle kamerarajapinnalle. Testiohjelman toteutus onnistui hyvin. Uusi testiohjelma eroaa vanhasta ominaisuuksien käyttöönotossa ja kameran määrittelyssä. Testauksessa uusi ohjelma toimi odotetun mukaisesti ja suoritti halutut testaukset. Lopputuloksena voidaan sanoa, että uusi ohjelma toimii hyvin ja kaikki halutut testiominaisuudet saatiin toimimaan.

Liitteistä löytyy LG Nexus 5 -puhelimella ajettu testiajo. Vaikka kyseessä on tehdasvalmisteinen puhelin, silti puhelimesta löytyi ongelmia. Ongelmia tuotti kamerarajapinta ja sen toiminta puhelimen ajureihin. Ongelmia vielä lisäsi kehitysvaiheilla oleva kamerarajapinta.

Muutamiiin ongelmiin kuitenkin törmättiin testiohjelman teossa. Esikatselun layoutia muokattiin moneen kertaan, että saatiin suurennus toimimaan. Suurennukseen jäi vielä jonkinlainen vika, mutta pienellä tutkimisella selvisi, että uudessa kamerarajapinnassa on vielä vikaa. Tulevien Android-päivitysten myötä toivottavasti viat korjaantuvat. Uusi kamerarajapinta on vielä kehitysvaiheessa, jolloin ominaisuuksien määrittelyt ja kameran toimintojen käyttöönotot voivat vielä muuttua.

Suurimmaksi haasteeksi osoittautui kuitenkin uuden rajapinnan ominaisuuksien käyttöönotto. Juuri ilmestyneen päivityksen takia netistä ei juurikaan löytynyt esimerkkejä tai kunnollisia ohjeita ominaisuuksien käyttöönottoa varten.

Ohjelman teossa oli tärkeää, ettei SoMA-järjestelmän palvelinyhteyksiä tarvitse muokata. Testien määrittelytiedostoja jouduttiin vähän muokkaamaan, että ne saatiin uudelle rajapinnalle sopivaksi. Uudessa kamerarajapinnassa on uusia ominaisuuksia, jotka otetaan käyttöön myöhemmin ohjelmaan.

Tässä työssä tutkittiin myös Android-käyttöjärjestelmää ja verrattiin uutta ja vanhaa kamerarajapintaa koodillisesti. Työssä kerrottiin myös SoMA-järjestelmästä ja sen testauksista ja testausympäristöstä.

Kokonaisuutena työn tekeminen oli mielekästä ja opettavaa. Lisäpäivitysten myötä ohjelman testattavia ominaisuuksia saadaan lisää.

LÄHTEET

- Android API Guides. Ei päiväystä. Camera. [www-lähde]. Android API Guides. [Viitattu 8.2.2015]. Saatavissa: <http://developer.android.com/guide/topics/media/camera.html#>
- Android Developers. 2015. SurfaceHolder. [www-lähde]. Android Developers. [Viitattu 25.4.2015]. Saatavissa: <http://developer.android.com/reference/android/view/SurfaceHolder.html>
- Android Devices. Ei päiväystä. Camera HAL3. [www-lähde]. Android Devices. [Viitattu 13.2.2015]. Saatavissa: <https://source.android.com/devices/camera/camera3.html#supported-version>
- Android Devices. Ei päiväystä. HAL Subsystem. [www-lähde]. Android Devices. [Viitattu 30.3.2015]. Saatavissa: https://source.android.com/devices/camera/camera3_requests_hal.html
- Compile Time Error. 2013. Android History and Android OS Architecture. [www-lähde]. Compile Time Error. [Viitattu 8.2.2015]. Saatavissa: <http://www.compiletimeerror.com/2012/12/blog-post.html#.VNaKjPmsVyU>
- Dolcourt, J. 2010. The hardware guts of your Android phone. [www-lähde]. Jessica Dolcourt. [Viitattu 23.2.2015]. Saatavissa: <http://www.cnet.com/news/the-hardware-guts-of-your-android-phone/>
- Etherington, D. 2014. Google Details Android 5.0 Lollipop's Major Security Improvements. [www-lähde]. Darrell Etherington. [Viitattu 31.3.2015]. Saatavissa: <http://techcrunch.com/2014/10/28/android-5-lollipop-security-features/>
- Harju, J. 2013. Android-ohjelmoinnin perusteet. Helsinki Books on Demand GmbH.
- Hildenbrand, J. 2013. Inside the different Android Versions. [www-lähde]. Jerry Hildenbrand. [Viitattu 16.2.2015]. Saatavissa: <http://www.androidcentral.com/android-versions>
- IDC. 2015. Smartphone OS Market Share, Q3 2014. [www-lähde]. IDC. [Viitattu 16.2.2015]. Saatavissa: <http://www.idc.com/prodserv/smartphone-os-market-share.jsp>
- Laakso, J. 2014. Uusi Android on nimeltään Lollipop. [www-lähde]. Janne Laakso. [Viitattu 30.3.2015]. Saatavissa: <http://mobiili.fi/2014/10/15/uusi-android-on-nimeltaan-lollipop/>

- Miller, J. 2014. Get familiar with Android 5.0 Lollipop. [www-lähde]. Josh Miller. [Viitattu 7.4.2015]. Saatavissa: <http://www.cnet.com/pictures/google-android-lollipop/6/>
- Mitroff, S. 2015. Unified, fresh design makes Android feel whole. [www-lähde]. Sarah Mitroff. [Viitattu 30.3.2015]. Saatavissa: <http://www.cnet.com/products/google-android-5-0-lollipop/>
- Myllyperkiö, I. 2015. Sofica, toimitusjohtaja. Henkilökohtainen tiedonanto 8.4.2015
- PhoneArena. 2015. Google Nexus 5. [www-lähde]. PhoneArena. [Viitattu 16.2.2015]. Saatavissa: http://www.phonearena.com/phones/Google-Nexus-5_id8148
- Sofica. 2012a. SoMA Functional Tests. [www-lähde]. Sofica Ltd. [Viitattu 10.2.2015]. Saatavissa: Yrityksen sisäisessä jakelussa
- Sofica. 2012b. SoMA Verification Server Technical Specification. [www-lähde]. Sofica Ltd. [Viitattu 10.2.2015]. Saatavissa: Yrityksen sisäisessä jakelussa
- Sofica. 2013. SoMA 2.0 Test Box Hardware Specification. [www-lähde]. Sofica Ltd. [Viitattu 10.2.2015]. Saatavissa: Yrityksen sisäisessä jakelussa
- Sofica. 2014a. SoMA: Revolutionizing Camera Testing. [www-lähde]. Sofica Ltd. [Viitattu 21.1.2015]. Saatavissa: <http://www.sofica.fi/index.php?page=soma>
- Sofica. 2014b. Sofica Technologies. [www-lähde]. Sofica Ltd. [Viitattu 21.1.2015]. Saatavissa: <http://www.sofica.fi/index.php?page=overall-tests>
- Toombs, C. 2014. Android "L" Feature Spotlight: New Camera API Enables Burst Mode, Thorough Control Over Photos, And Much More. [www-lähde]. Cody Toombs. [Viitattu 6.4.2015]. Saatavissa: <http://www.androidpolice.com/2014/06/27/android-l-feature-spotlight-new-camera-api-enables-burst-mode-thorough-control-photos-much/>
- Whitwam, R. 2014. Android 5.0 Lollipop: All the details and features of Google's sweetest treat yet. [www-lähde]. Ryan Whitwam. [Viitattu 31.3.2015]. Saatavissa: <http://www.extremetech.com/computing/192112-android-5-0-lollipop-all-details-and-features-of-googles-sweetest-treat-yet/2>

LIITTEET

Liite 1. Testiohjelman raportti

1. Testiohjelman raportti



SoMA Testrun Report

Functional tests for LG5_ADB

Testrun Start Time: 2015-02-25 15:28:45, End Time: 2015-02-25 18:01:30

[Link to results page](#)

Test Results Summary

Total test cases	233
Passed	146
Failed	87
Not testable	0
Crashes	0
Run rate	100 %
Pass rate of total	62.66 %
Pass rate of executed	62.66 %

Test Results by Test Set

Test Set	Total	Passed	Failed	Not testable
Resolution 4 3 JavaLayer	10	9	1	0
Resolution 1.78 JavaLayer	2	2	0	0
Resolution 1.67 JavaLayer	2	2	0	0
Resolution 1.50 JavaLayer	1	1	0	0
Resolution 1.22 JavaLayer	2	2	0	0
Thumbnail JavaLayer	8	2	6	0
White Balance JavaLayer	16	3	13	0
Focus JavaLayer	5	5	0	0
Zoom JavaLayer	100	56	44	0
Flash JavaLayer	6	6	0	0
AE JavaLayer	2	0	2	0
AE Compensation JavaLayer	25	19	6	0
Dynamic range JavaLayer	1	1	0	0
IR HotSpot JavaLayer	1	0	1	0
Color Effects JavaLayer	4	2	2	0
AVSync 1.78 JavaLayer	4	4	0	0
AVSync 4 3 JavaLayer	10	10	0	0
AVSync 1.67 JavaLayer	4	4	0	0
AVSync 1.50 JavaLayer	2	2	0	0
AVSync 1.22 JavaLayer	4	4	0	0
VideoResolution 1.78 JavaLayer	2	2	0	0
VideoResolution 4 3 JavaLayer	5	5	0	0
VideoResolution 1.67 JavaLayer	2	2	0	0
VideoResolution 1.50 JavaLayer	1	1	0	0
VideoResolution 1.22 JavaLayer	2	2	0	0
VideoFrameRate VBR JavaLayer	12	0	12	0