

OptiLift

Tietotekniikan Sovellusprojekti

Lauri Laasala
Olli Lukkarinen
Ville Räisänen
Vesa Tanhua-Tyrkkö

Sovellusraportti
Versio 0.8
31.5.2004

Jyväskylän yliopisto
Tietotekniikan laitos

- Tekijät:** Laasala Lauri
Lukkarinen Olli
Räisänen Ville
Tanhua-Tyrkkö Vesa
- Projektin tiedot:** OptiLift-projekti
Kotisivu: <http://sovellusprojektit.it.jyu.fi/optilift/>
- Työn nimi:** OptiLift-projektin sovellusraportti.
- Työn kuvaus:** Sovellusraportti tietotekniikan Sovellusprojektista.
- Tilaja:** Kilpa- ja huippu-urheilun tutkimuskeskus KIHU.
- Teettäjä:** Jyväskylän yliopisto, tietotekniikan laitos.
Vastaava ohjaaja: Markus Inkeroinen.
Tekninen ohjaaja: Ville Tirronen.
- Tiivistelmä:** Tämä dokumentti on Jyväskylän yliopiston keväällä 2004 toteutetun OptiLift-Sovellusprojektin sovellusraportti. Dokumentissa kuvataan kuinka vaatimusmäärittelyssä esitetyt vaatimukset toteutettiin sovelluksessa.
- Avainsanat:** KIHU, kilpa- ja huippu-urheilun tutkimuskeskus, käyttöliittymä, luokkakaavio, olio, OptiLift, sovellusprojekti, sovellusraportti, jatkokehitys, ylläpito.

Versiohistoria

Versio	Päiväys	Tehnyt	Muutokset
0.1	11.5.2004	Lauri Laasala	Ensimmäinen versio.
0.2	12.5.2004	Lauri Laasala	Lisätty kuvat käyttöliittymästä sekä luokkakaavio.
0.3	14.5.2004	Lauri Laasala ja Ville Räisänen	Korjattu virheitä, lisätty muuttujien laskemiseen käytetyt kaavat ja ledin seurantaan käytetty algoritmi. Lisäksi muutettu Jatkokehitys-lukua.
0.4	19.5.2004	Lauri Laasala	Muokattu kuvia ja luokkakaaviota sekä korjattu kirjoitusvirheitä.
0.5	24.5.2004	Lauri Laasala	Korjattu virheitä, vaihdettu kuva 4.1 ja muokattu lukua 7.
0.6	24.5.2004	Lauri Laasala	Lisätty asennusohjeet.
0.7	27.5.2004	Lauri Laasala	Korjattu ulkoasua ja kappalejakoja, tarkennettu vaatimusten täyttymistä, muokattu luokkakaaviota ja -kuvauksia sekä lisätty tiedostonnimien käsittelystä kertova kappale lukuun 5.
0.8	31.5.2004	Ville Räisänen	Korjattu viimeiset kirjoitusvirheet, lisätty yksi jatkokehitysidea.

Sisältö

1	Johdanto	1
2	Termistöä	2
3	Yleisarkkitehtuuri	3
3.1	Käyttöliittymä	3
3.2	Sisäinen toteutus	3
3.3	Tiedostot	4
4	Käyttöliittymäkuvaus	5
4.1	Kalibrointi	7
4.2	Suoritusryhmän luominen ja hallinta	9
4.3	Suorituksen läpivienti	10
4.4	Videon analysointi	11
4.5	Analyysitulosten esittäminen	12
4.6	Suoritusten vertailu	13
5	Sovelluksen rakenne	14
5.1	Sovelluksen luokkakaavio	14
5.2	Käyttöliittymän luokat	16
5.2.1	TResultViewer	16
5.2.2	TVideoAndResultView	17
5.2.3	TFormVideo	19
5.2.4	TExerciseGroupView	20
5.3	Sovelluksen sisäiset luokat	21
5.3.1	TDetectorThread	21

5.3.2	TAnalyzer	22
5.3.3	TController.....	23
5.3.4	TLifterRegister.....	24
5.3.5	TLifter	24
5.3.6	TVideo	25
5.3.7	TLiftsContainer.....	26
5.3.8	TLift	26
5.3.9	TVarList.....	27
5.3.10	TVariable	27
5.3.11	TGrabber	28
5.3.12	FilenameHandler.pas	29
5.4	Ulkoiset komponentit.....	30
5.4.1	VideoGrabber.....	30
5.4.2	ResizerPanel.....	30
5.4.3	ExtendedListBox.....	30
5.5	Muuttujien laskenta.....	31
5.6	Ledin seurantaan käytetty algoritmi.....	36
6	Tiedostot	37
6.1	Tiedostojen formaatit	37
6.1.1	Käyttäjärekisteritiedosto	37
6.1.2	Liikeratatiedosto	38
6.1.3	Analysointitulostiedosto	39
6.2	Videokoodekit.....	39
7	Sovelluksen asentaminen	40

8	Vaatimusten täytyminen ja sovelluksen jatkokehitys	41
8.1	Vaatimusten täytyminen	41
8.2	Tunnetut virheet	47
8.3	Jatkokehitysehdotuksia	47
9	Yhteenveto	49
10	Lähteet.....	50

1 Johdanto

OptiLift-projekti suunnitteli ja toteutti Kilpa- ja huippu-urheilun tutkimuskeskukselle painonnoston levytankoharjoitteluun nostotekniikan automaattisen mittaus- ja analysointisovelluksen. Järjestelmän avulla analysoidaan urheilijan levytankoharjoittelua, joka on olennainen osa monien eri urheilulajien voimaharjoittelua. Oikean tekniikan hallitseminen on tärkeää niin vammattoman kuin mahdollisimman tehokkaankin harjoittelun kannalta. Monilla valmentajilla ei kuitenkaan ole riittävästi tietoa oikeista nostotekniikoista, jotta mahdollisimman hyödyllisen palautteen antaminen olisi mahdollista [8].

Projekti toteutettiin Jyväskylän yliopiston tietotekniikan laitoksen sovellusprojektina kevään 2004 aikana. Projektiin liittyvistä käytännöistä ja toimintatavoista sekä projektiin kuuluvista henkilöistä on kerrottu tarkemmin projektiraportissa [5].

Sovellukselle asetetut vaatimukset asetettiin tärkeysjärjestykseen, jotta ainakin tärkeimmät ominaisuudet saataisiin toteutettua projektin puitteissa. Tilaaaja toivoi projektin kuluessa toimivaa sovelluksen prototyyppiä sen sijaan, että yritettäisiin saada kaikki matalamman prioriteetin vaatimukset täytettyä. Tilaaaja tulee jatkokehittämään projektin tuottamaa sovelluksen prototyyppiä.

Tässä dokumentissa kuvataan sitä, miten vaatimusmäärittelyssä [7] esitetyt vaatimukset toteutettiin. Luvussa 2 esitellään sovelluksen toteutukseen liittyviä keskeisiä termejä. Luvussa 3 käydään läpi sovelluksen yleinen arkkitehtuuri. Sovelluksen käyttöliittymää esitellään luvussa 4. Luvussa 5 on kuvattu sovelluksen rakenne luokkajakojen ja -kaavioiden avulla sekä esitelty muuttujien laskentaan käytetyt kaavat ja ledin seurantaan käytetty algoritmi. Luvussa 6 kerrotaan sovelluksen datan tallennukseen käytettävistä tiedostoista. Luvussa 7 annetaan sovelluksen asennusohjeet. Luvussa 8 kerrotaan sovelluksen jatkokehitykseen ja ylläpitoon liittyvistä asioista. Luku 9 on yhteenveto sovelluksen toteuttamisesta.

2 Termistöä

Tässä luvussa selitetään yleisimmät sovelluksen toteutukseen liittyvät termit.

AVI	Microsoftin tiedostomuoto äänelle ja liikkuvalla kuvalla. Tiedostot ovat yhteensopivia sekä PC:lle että Applen Macintoshille. AVIin sisältyy CODEC-tiedosto, jota tarvitaan videotiedostoja käyttävissä ohjelmissa tiedon pakkaukseen ja purkuun.
Delphi	Ohjelmistotalo Borlandin olioperustaiselle Pascal-ohjelmointikielelle suunnittelema ohjelmistokehitin, jolla tuotetaan valmiiksi käännettyjä ohjelmia.
Dialogi	Keskusteluikkuna, valintaikkuna. Graafisissa käyttöliittymissä ikkuna, johon täydennetään ohjelman tarvitsemat tiedot.
Form(i)	Lomake, sivu, graafisessa ohjelmoinnissa visuaalinen tapa esittää yksi ohjelman ikkuna [9].
Frame	Videoleikkeen tai animaation yksi kuvaruutu.
Freeware	Ilmaisojelmat. Tietokoneohjelmia, joita voi vapaasti jakaa. Oikeuden omistaja ei vaadi ohjelman käyttämisestä mitään korvausta. Freeware-ohjelmat ovat kuitenkin tekijänoikeudellisesti suojattuja, vaikka korvausta ei vaadita. Käyttöoikeuksia koskevat ehdot esitetään tavallisesti ohjelman käynnistyksen yhteydessä [4].
Koodekki	Ohjelma tai laite, jota käytetään informaation pakkaamiseen ja purkamiseen. (engl. <i>code & decode= codec</i>).
MDI	Suom. moniasiakirjaliittymä. Tekniikka sovelluksen ikkunoinnin ja asiakirjojen hallintaan [9]. (engl. <i>Multiple Document Interface</i>).
Olio	Luokan ilmentymä.
Säiliöluokka	Jonkin luokan olioiden tallettamiseen tarkoitettu luokka.

3 Yleisarkkitehtuuri

Kuvassa 3.1 esitetään järjestelmän yleinen rakenne. Järjestelmä koostuu tietokoneeseen asennetusta sovelluksesta ja sen käyttämistä tiedostoista sekä tietokoneeseen ja sovellukseen liitetystä digitaalisesta videokamerasta. Käyttäjä käyttää sovellusta käyttöliittymän kautta. Sisäinen toteutus ja käyttöliittymän toteutus on selkeästi erotettu toisistaan.

3.1 Käyttöliittymä

Käyttöliittymässä on erilaisia näkymiä käyttötapauksen mukaan.

Videonäkymässä on videokuvanäyttö, jolla näytetään kerrallaan yhtä tai kahta videota sekä näiden liikeratakäyriä.

Analysointituloksenäkymässä näkyvät nostosuorituksen analysoidut muuttujat graafisina kuvaajina ja numeerisina arvoina taulukossa.

Kerrallaan voidaan näyttää kahta suoritusta, jolloin sekä videokuvanäyttöä että graafisia kuvaajia on kaksi kappaletta.

Suoritusryhmänäkymästä hallitaan suoritusryhmää, käyttäjärekisteriä sekä nauhoituksen ja vertailun käynnistämistä ja lopettamista.

Käyttöliittymän toteutusta on kuvattu tarkemmin luvussa 4.

3.2 Sisäinen toteutus

Sisäisen toteutuksen loogisia osia ovat videon- ja kuvankaappaus, analysaattori ja käyttäjärekisteri.

Videon- ja kuvankaappausosan tehtäviä ovat videonpätkien ja yksittäisten kuvien kaappaaminen videokameran välittämästä kuvasta, liikeratakäyrän tunnistaminen sekä videoiden ja käyrän tallentaminen tiedostoon.

Analysaattori laskee analysoidut biomekaaniset muuttujat liikeratakäyrän ja noston parametrien perusteella.

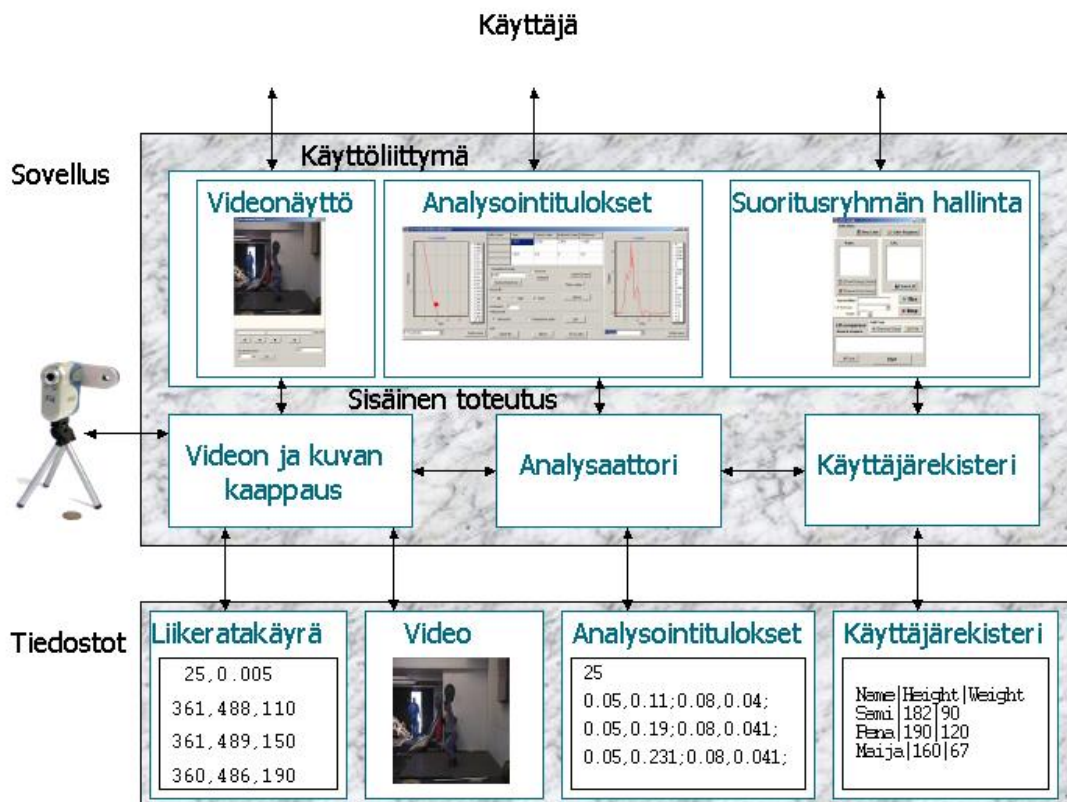
Käyttäjärekisteri lukee ja tallettaa käyttäjiä rekisteritiedostoihin sekä tallettaa ajonaikaiset tiedot suoritusryhmästä.

Sovelluksen sisäiseen toteutukseen perehdytään tarkemmin luvussa 5

3.3 Tiedostot

Kuvan 3.1 alalaidassa on esitelty tiedostot, joita sovellus käyttää. Niitä ovat liikeratakäyrä, video, analysointitulokset ja käyttäjärekisteri.

Tiedostoista on kerrottu enemmän luvussa 6.



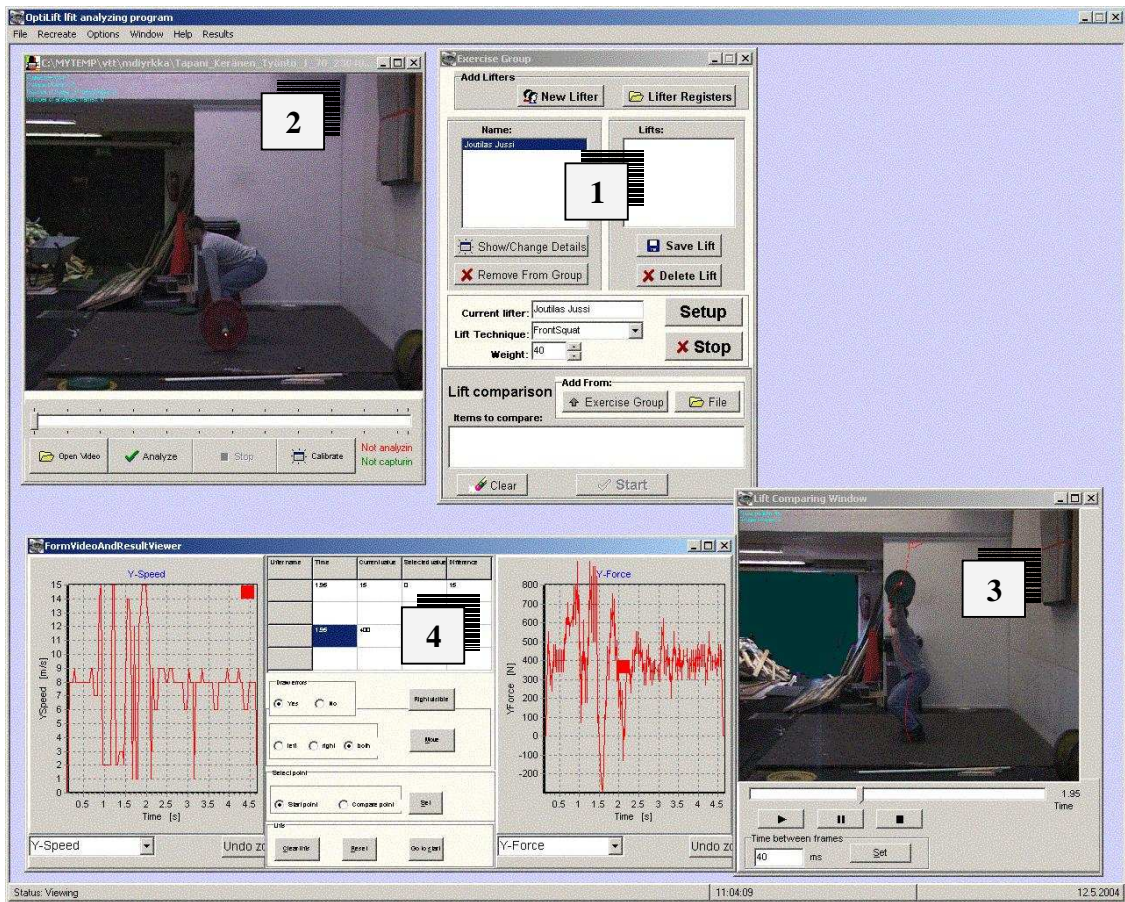
Kuva 3.1: Sovelluksen arkkitehtuuri.

4 Käyttöliittymäkuvaus

Tässä luvussa kuvataan sovelluksen käyttöliittymän toteutusta havainnollistavien kuvien ja niiden selitysten avulla. Kussakin aliluvussa on esitetty yhden käyttötapauksen toteuttaminen käyttöliittymän avulla. Käyttöliittymän suunnittelun lähtökohtana oli helppokäyttöisyys ja havainnollisuus. Käyttöliittymä toteutettiin kokonaisuudessaan englanniksi.

Käyttöliittymä toteutettiin käyttäen MDI-tekniikkaa. Siinä yksi sovelluksen ikkunoista on pääikkuna, johon sijoitetaan kaikki menut ja joka toimii kehyksenä ali-ikkunoille. Ali-ikkunat ovat pääikkunalle alisteisia. Tämä tarkoittaa sitä, ettei niitä voi liikuttaa pääikkunan asettamien aluerajojen ulkopuolelle. Ikkunoita pystytään näin hallitsemaan kootummin ja niiden katoamisilta toisten ikkunoiden alle vältytään. MDI-tekniikkaa käytetään tavallisesti moniasiakirjasovelluksissa, joissa tarvitaan usean asiakirjan samanaikaista käsittelyä. Tätä ominaisuutta ei sovelluksessa tarvittu, mutta MDI-tekniikka oli käyttökelpoinen myös sovelluksen ikkunointiin.

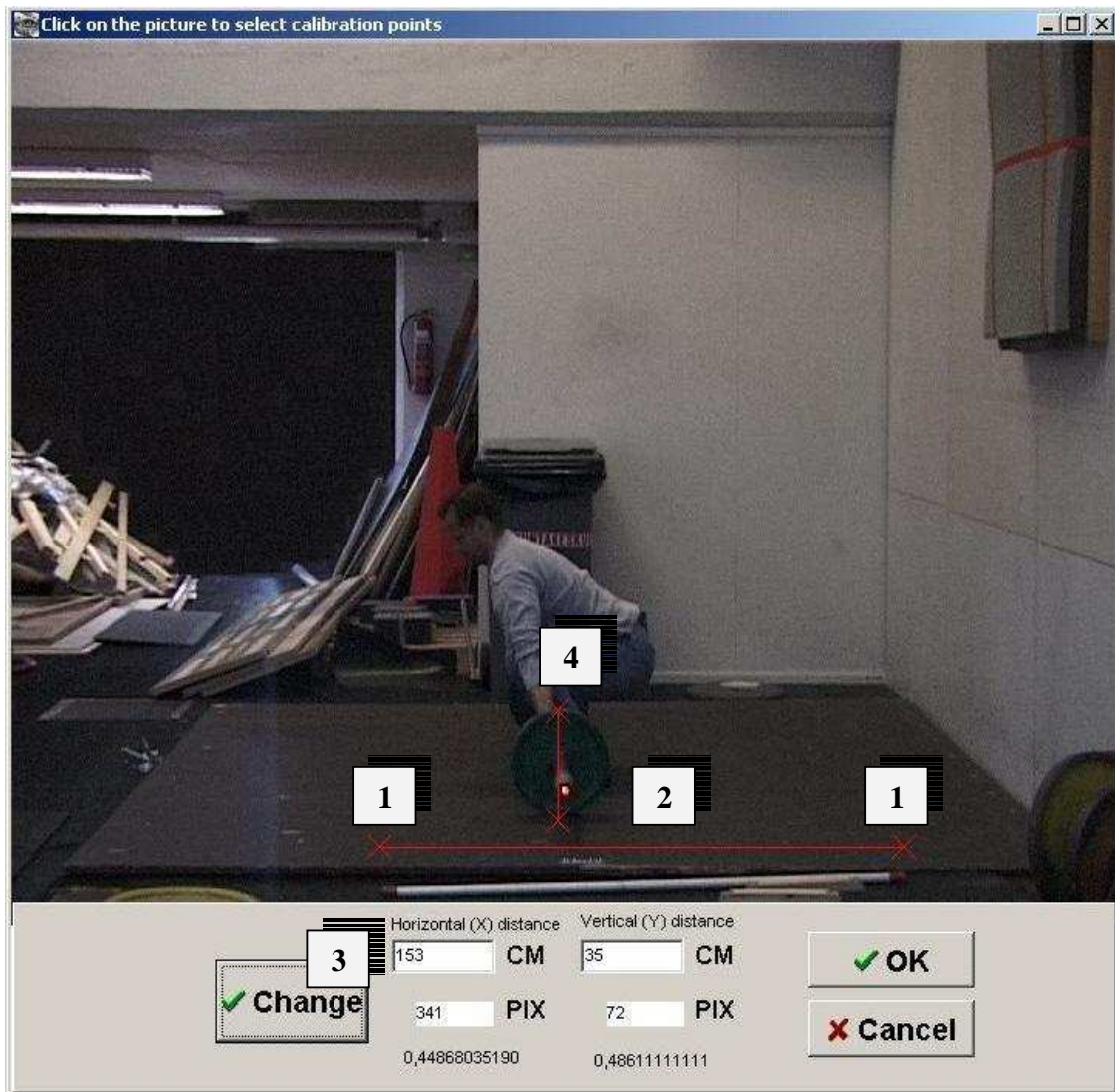
Kuvassa 4.1 on esitetty kuva sovelluksen käyttöliittymästä. Sen osat ovat suoritusryhmänäkymä (1), videonkaappausnäky (2), videonäkymä (3) ja analyysitulospäätelmä (4).



Kuva 4.1: Sovelluksen käyttöliittymä.

4.1 Kalibrointi

Kalibroinnissa käyttäjä asettaa ensin mitoiltaan tunnetun esineen kuvattavalle alueelle. Esine tulisi asettaa mahdollisimman keskelle kuvattavaa aluetta, jotta kalibrointi olisi tarkka. Tämän jälkeen käyttäjä valitsee kalibroinnin kaappaus- tai suoritusryhmäikkunan *Calibrate*-painikkeesta, jolloin kuvattava alue tulee näkyviin sovelluksen kalibrointi-ikkunaan, joka on esitetty kuvassa 4.2. Kalibrointi suoritetaan erikseen x- ja y-suunnassa. Ensimmäinen käyttäjä valitsee kuvasta esineen laidoilta vaakatasossa eli x-suunnassa kaksi pistettä (1). Sovellus piirtää kuvaan viivan pisteiden välille (2) ja antaa käyttäjälle mahdollisuuden hienosäätää pisteiden paikkaa. Kun pisteet on valittu käyttäjää tyydyttävällä tavalla, syöttää käyttäjä sovellukselle pisteiden välisen todellisen etäisyyden (3). Seuraavaksi sama suoritetaan pystytasossa eli y-suunnassa (4). Näiden tietojen perusteella sovellus saa tietoonsa kuvan mittasuhteet, joita tarvitaan analysoitaessa suorituksista haluttuja muuttujia.

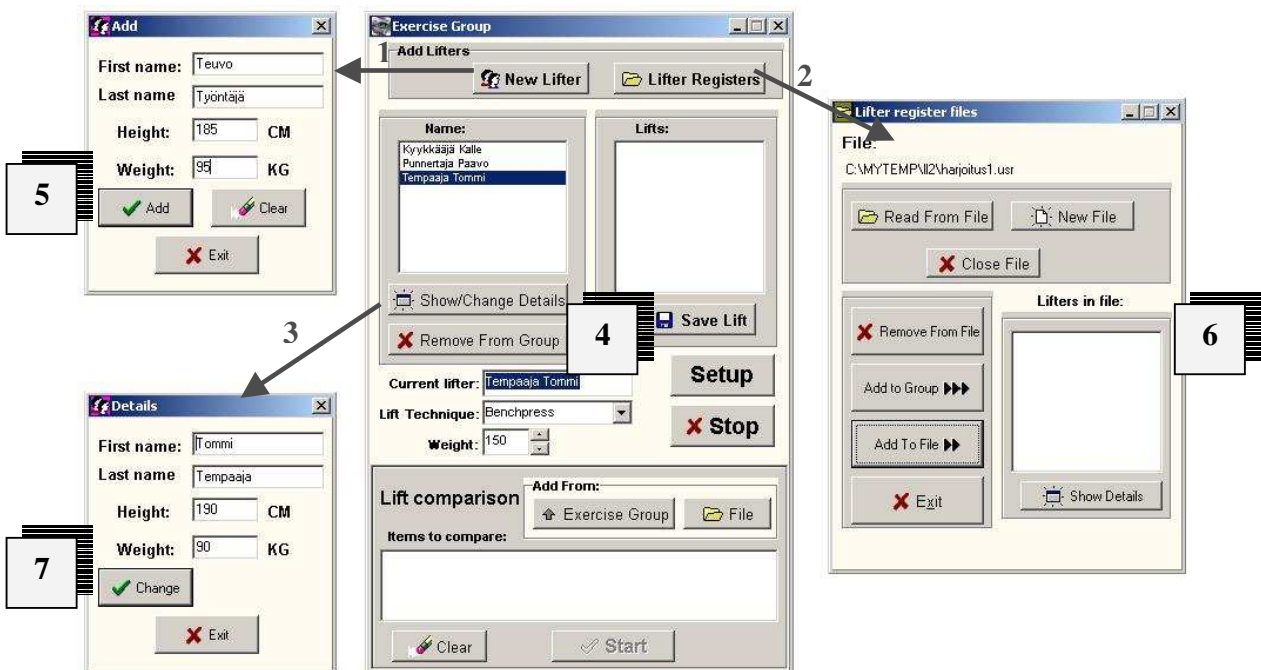


Kuva 4.2: Kalibrointi-ikkuna.

4.2 Suoritusryhmän luominen ja hallinta

Suoritusryhmää hallitaan suoritusryhmänäkymän ja sen ali-ikkunoiden avulla, jotka on esitetty kuvassa 4.3. Suoritusryhmä on ajonaikainen tietorakenne, johon voidaan lisätä uusia nostajia syöttämällä heidän tietonsa (1), tai lisäämällä heitä käyttäjärekisteritiedostosta (2). Suoritusryhmään lisättyjen nostajien tietoja voidaan muuttaa (3), tai heitä voidaan poistaa suoritusryhmästä (4). Suoritusryhmässä olevia nostajia voidaan lisätä käyttäjärekisteritiedostoon, jolloin nostajien tiedot voidaan säilyttää käyttökertojen välillä.

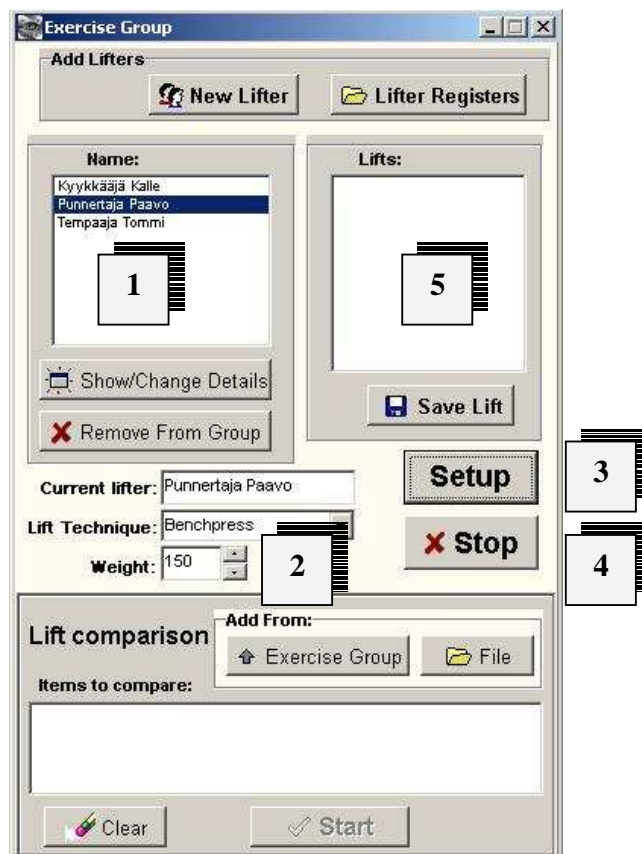
Ali-ikkunoita ovat uuden nostajan lisäämisikkuna (5), käyttäjärekisteritiedostojen hallintaikkuna (6) ja henkilötietoikkuna (7). Uuden nostajan lisäämisikkunassa annetaan henkilötiedot ja lisätään painikkeella nostaja suoritusryhmään. Käyttäjärekisteritiedostojen hallintaikkunassa voidaan lukea entisiä käyttäjärekisteritiedostoja, luoda uusia, lisätä nostajia rekisteritiedostoon ja poistaa heitä sieltä sekä lisätä nostajia rekisteritiedostosta suoritusryhmään. Henkilötietoikkunassa näkyvät valitun nostajan henkilötiedot, joita voidaan muuttaa.



Kuva 4.3: Näkymä suoritusryhmän hallinnasta.

4.3 Suorituksen läpivienti

Nostosuurituksen nauhoittaminen aloitetaan suoritusryhmänäkymästä, joka on esitetty kuvassa 4.4. Ensin valitaan suoritusryhmästä nostaja (1), sitten suoritusparametrit, eli nostotekniikka ja painomäärä (2). Nauhoituksen aloittamiseksi painetaan painiketta (3), jossa lukee tilanteen mukaan joko *Setup*, *Calibrate* tai *Rec*. Jos videonkaappausasetuksia ei ole vielä tässä vaiheessa asetettu, painamalla *Setup*-painiketta avautuu näyttöön ikkuna, jossa voidaan valita käytettävät videon- ja äänentallennusasetukset sekä tallennuspolku. Jos taas asetukset on jo tehty, niin painamalla *Calibrate*-painiketta siirrytään kalibrointi-ikkunaan. Jos molemmat edellä mainitut tehtävät on suoritettu, voidaan nauhoitus aloittaa painamalla *Rec*-painiketta. Nauhoittaminen voidaan katkaista painamalla *Stop*-painiketta (4). Nauhoitettu suoritus lisätään nostajan nostojen listaan (5).



Kuva 4.4: Suoritusryhmänäkymä.

4.4 Videon analysointi

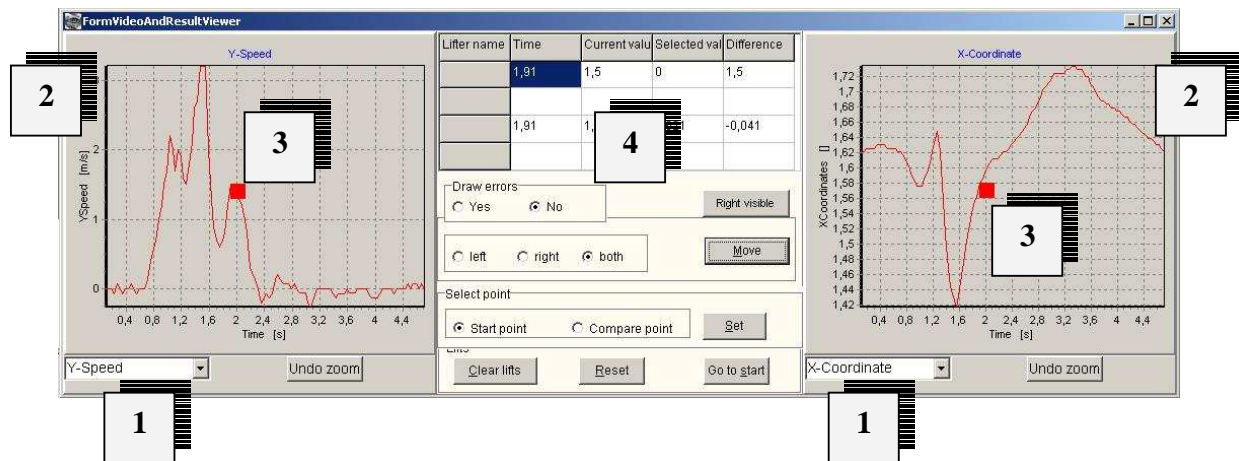
Videon nauhoituksen jälkeen nostosuoritus analysoidaan klikkaamalla kuvasta lediä, jolloin siihen piirretään keltainen risti (1). Tämän jälkeen painetaan *Analyze*-painiketta, jolloin video näytetään ja analysoinnin edetessä piirretään liikeratakäyrää (2) videokuvan päälle.

Aikaisemmin tallennettuja videoita voidaan myös analysoida uudelleen avaamalla videotiedosto *Open Video* -painikkeesta.



Kuva 4.5: Videon analysointi.

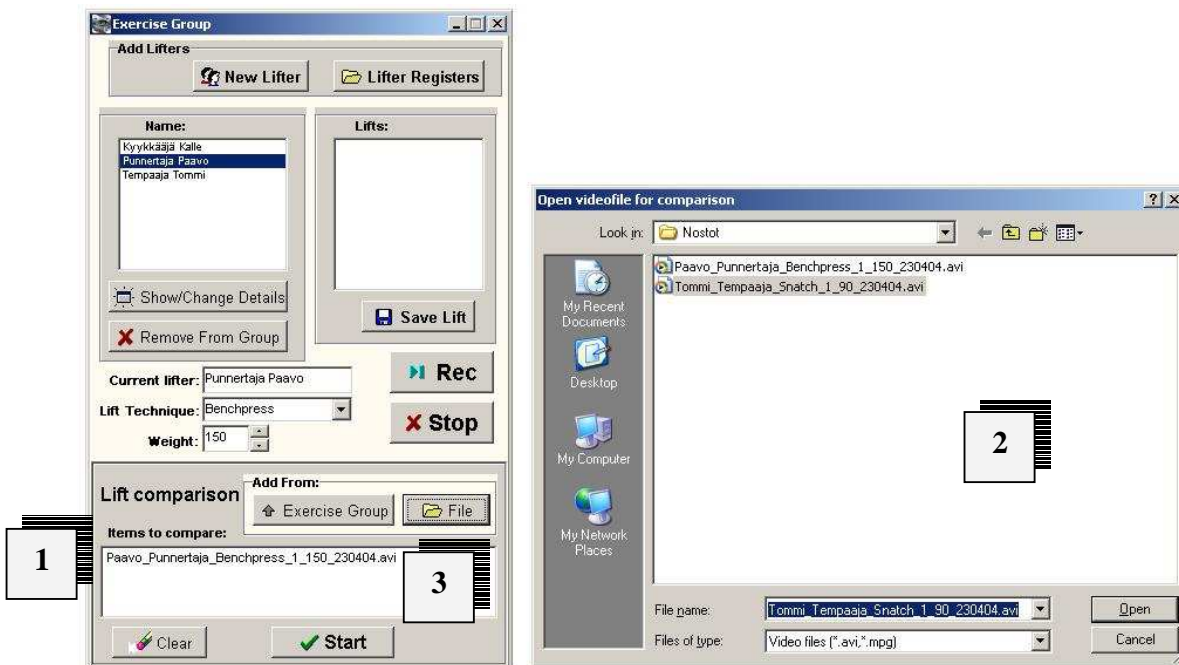
4.5 Analyysitulosten esittäminen



Kuva 4.6: Analyysitulostenäkymä.

Suorituksesta saadut analyysitulokset näytetään kuvan 4.6 kaltaisessa ikkunassa. Käyttäjä voi valita kuvan alareunassa näkyvistä alasvetolistoista (1), mitä muuttujaa halutaan seurata. Tällöin vastaavassa koordinaatistossa (2) näytetään käsiteltävänä olevista suorituksista valitun muuttujan arvot. Jos esimerkiksi kaksi suoritusta on valittuna vertailua varten, näkyy molemmista suorituksista sama muuttuja yhdessä koordinaatistossa. Tällöin suoritusten väliset erot kunkin muuttujan osalta tulevat selkeästi esille. Kursori (3) näyttää käyrältä vastaavan kohdan, kun videokuvaa vieritetään videonäytön palkilla. Graafisen kuvaajan lisäksi tiedot näytetään numeroina näytön keskellä (4). Käyttäjä voi valita *Set*-painikkeella kullakin käyrältä haluamansa vertailukohtaan. Tällöin valitun kohdan ja nykyisen kursorin osoittaman kohdan erotus tulee näkyviin numerotaulukossa (4). Numerotaulukossa näytettäviä tietoja ovat aika, muuttujan nykyinen arvo, vertailuarvo sekä vertailuarvon ja nykyisen arvon erotus.

4.6 Suoritusten vertailu



Kuva 4.6: Suoritusten vertailun aloittaminen.

Suoritusten vertailu aloitetaan kuvassa 4.6 näkyvästä suoritusryhmänäkymästä. Näkymän alalaidassa on suoritusten vertailuosa (1). Suoritus voidaan lisätä vertailuun joko harjoituksessa suoritetuista valitsemalla haluttu suoritus ja painamalla painiketta *Exercise Group*, tai tiedostosta painikkeella *File*, jolloin avautuu tiedostonavausdialogi (2). Vertailuun valitut suoritukset lisätään listaan (3), ja vertailun voi aloittaa *Start*-painikkeella.

Kun vertailu on aloitettu, lisätään videonäyttöön toinen videokuvanäyttö, ja analyysinäkömään toinen kuvaaja.

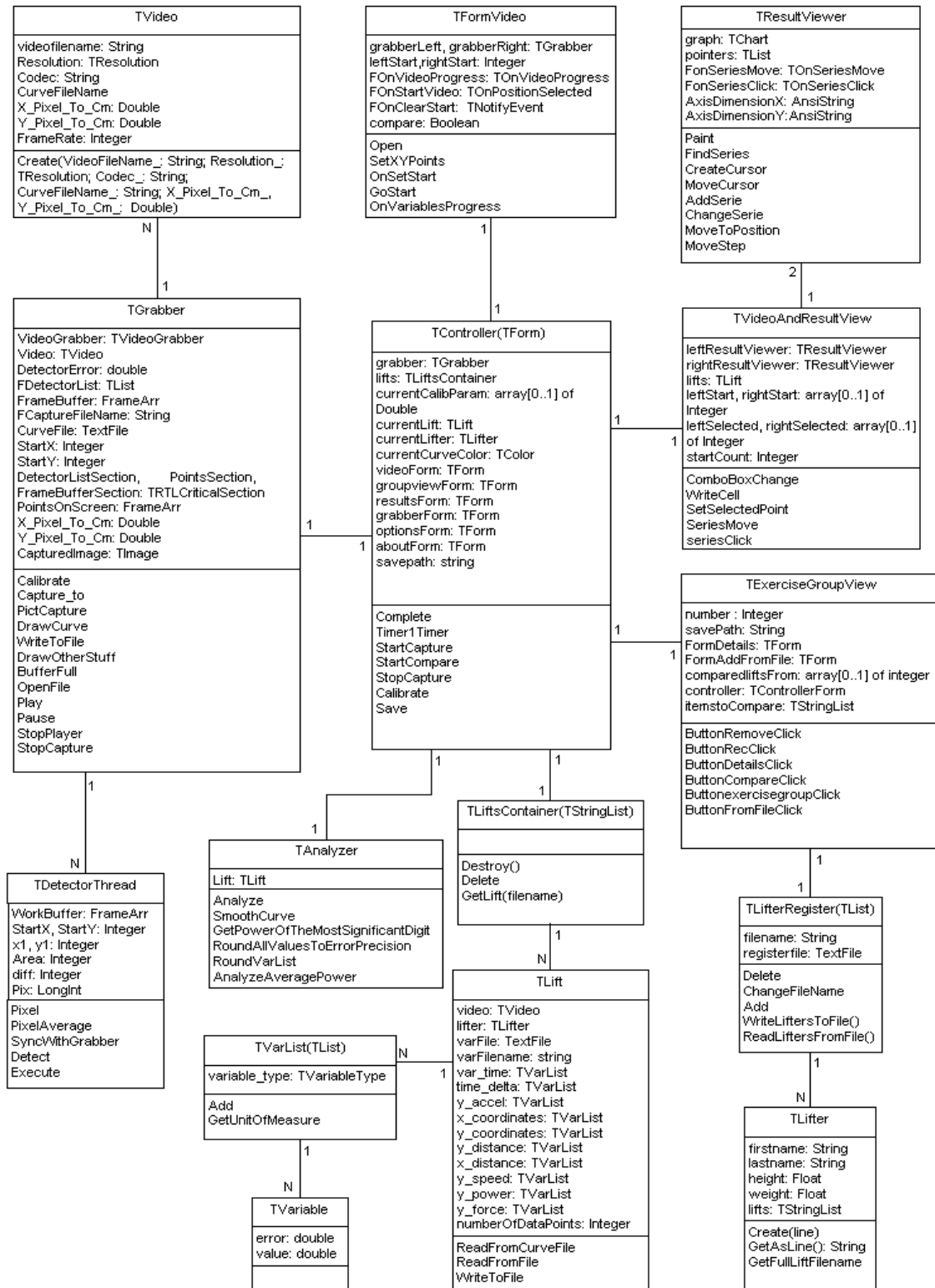
5 Sovelluksen rakenne

Tässä luvussa kuvataan sovelluksen sisäistä rakennetta luokkakaavion ja luokkakuvausten avulla. Tässä on selitetty vain tärkeimmät luokan toimintaan vaikuttavat attribuutit ja metodit, joten esimerkiksi kaikkia asetus- ja saantimetoiteita ei ole lueteltu.

Luvussa 5.1 esitetään sovelluksen luokkakaavio. Luvuissa 5.2 ja 5.3 esitellään sovelluksen luokat luokkakuvausten avulla. Sovelluksen luokat on nimetty hyvän Delphi-ohjelmointitavan mukaisesti T-alkuisina, esim. TController. Luokat on eroteltu käyttöliittymän luokkiin, luku 5.2, ja sisäisen toteutuksen luokkiin, luku 5.3. Luvussa 5.4 kerrotaan sovelluksessa käytettävistä ulkoisista komponenteista. Luvussa 5.5 kerrotaan muuttujien laskemiseen käytetyistä kaavoista sekä luvussa 5.6 ledin seurantaan käytetystä algoritmista.

5.1 Sovelluksen luokkakaavio

Luokkakaaviosta näkyvät sovelluksen luokkien keskinäiset suhteet, niiden attribuutit ja metodit sekä luokkien väliset kardinaalisuudet, eli luokkien olioiden esiintymien määrät. Sovelluksen luokkakaavio on esitetty kuvassa 5.1.



Kuva 5.1: Sovelluksen luokkakaavio.

5.2 Käyttöliittymän luokat

5.2.1 TResultViewer

TResultViewer on yhden noston muuttujien näyttämiseen tarkoitettu visuaalinen komponentti. Niitä voi olla sovelluksessa kaksi yhtäaikaisesti. Se sisältää käyrien piirtämiseen tarkoitetun TChart-komponentin. TResultViewer käyttää hyväkseen TLIftin muuttujataulukoita joten sen ei tarvitse lukea muuttujatiedostoa.

	Nimi:	Kuvaus:
Attribuutti	graph: Tchart	Muuttujan käyrän piirtämiseen käytetty komponentti.
	pointers: TList	Lista kursoreista, jotka osoittavat nykyiset arvot kuvaajassa.
	FonSeriesMove: TOnSeriesMove	Tapahtuma, joka aiheutuu kuvaajaa vastaavan kursorin liikkeessä.
	FonSeriesClick: TOnSeriesClick	Tapahtuma, joka aiheutuu käyttäjän klikatessa kuvaajaa.
	AxisDimensionX: AnsiString	X-akselin mittasuhte.
	AxisDimensionY: AnsiString	Y-akselin mittasuhte.
Metodi	Paint	Piirtää muuttujan käyrän TChart-komponenttiin ja asettaa x- ja y-akseleiden mittasuhteet.
	FindSeries	Palauttaa kuvaajan indeksin.
	CreateCursor	Luo uuden kursorin.
	MoveCursor	Siirtää kursorin haluttuun kohtaan.
	AddSerie	Lisää uuden kuvaajan TChart-komponenttiin
	ChangeSerie	Vaihtaa kuvaajan x- ja y-arvot.
	MoveToPosition	Siirtää kuvaajan osoittimen haluttuun kohtaan.
	MoveStep	Siirtää kuvaajan osoitinta halutun määrän joko eteen- tai taaksepäin.

Taulukko 5.1: TResultViewer.

5.2.2 TVideoAndResultView

TVideoAndResultView ohjaa TResultView-luokan olioita, jotka näyttävät analysointituloksia. TVideoAndResultView päivittää lisäksi taulukkoa, joka näyttää muuttujien arvot numeerisina, TResultView-luokan olioiden välittämällä arvoilla.

	Nimi:	Kuvaus:
Attribuutti	leftResultViewer: TResultViewer	Osoitin vasempaan TResultViewer- komponenttiin.
	rightResultViewer: TResultViewer	Osoitin oikeaan TResultViewer- komponenttiin.
	lifts: TList	Sisäinen lista nostojen tallennukseen.
	leftStart, rightStart: array[0..1] of Integer	Käyrien lähtöpisteiden sijainnit.
	leftSelected, rightSelected: array[0..1] of Integer	Pisteet, joihin nykyistä arvoa verrataan.
	startCount: Integer	Suurin arvo attribuuteista leftStart, rightStart, leftSelected ja rightSelected.
Metodi	ComboBoxChange	Muuttaa TResultViewer- komponentissa näytettävän muuttujan.
	WriteCell	Kirjoittaa TStringGrid- taulukon haluttuun soluun halutun arvon.
	SetSelectedPoint	Asettaa kuvaajan pisteen, johon nykyistä arvoa verrataan.
	SeriesMove	Tapahtumakäsittelijä, jota kutsutaan, kun jokin kuvaaja muuttaa sijaintiaan.
	seriesClick	Tapahtumankäsittelijä, jota kutsutaan kun kuvaajaa klikataan.

	<code>rightResultViewerVisible</code>	Piilottaa tai tuo esiin oikeanpuoleisen <code>TresultViewer</code> -komponentin.
	<code>SetLiftToView</code>	Asettaa uuden noston esitettäväksi.
	<code>Compare</code>	Asettaa kaksi nostoa vertailtavaksi.
	<code>SeriesGotoStart</code>	Asettaa kaikki kuvaajat lähtöpisteisiin.
	<code>moveForfard</code>	Siirtää kaikkia kuvaajia eteenpäin.

Taulukko 5.2: `TVideoAndResultView`.

5.2.3 TFormVideo

TFormVideo on videoiden näyttämiseen tarkoitettu luokka. Siinä voidaan näyttää kerrallaan kaksi videota, jotka ovat synkronoitu keskenään. Lisäksi se toimii yhteistyössä TVideoAndResultView-luokan kanssa siten, että video ja sen vaiheeseen kuuluva kuvaajan arvo ovat aina samassa vaiheessa.

	Nimi:	Kuvaus:
Attribuutti	grabberLeft, grabberRight : TGrabber	Videoiden näyttämiseen tarkoitettut TVideoGrabber-komponentit.
	leftStart, rightStart : Integer	Videoiden alkusijainnit.
	FOnVideoProgress : TOnVideoProgress	Tapahtuma, joka aiheutuu videon edetessä.
	FOnStartVideo : TOnPositionSelected	Tapahtuma, joka aiheutuu videon ollessa alussa.
	FOnClearStart : TNotifyEvent	Tapahtuma, joka lähetetään videoiden alkupistettä asetettaessa.
	compare : Boolean	Kertoo, onko kyseessä vertailu vai yhden noston näyttö.
Metodi	Open	Avaa videon TGrabberiin.
	SetXYPoints	Kopioi käyrän arvot TGrabberiin.
	OnSetStart	Asettaa TrackBarin alkupisteeseen.
	GoStart	Asettaa videot alkupisteisiin.
	OnVariablesProgress	Tapahtuma, jota kutsutaan, kun kuvaajan kursorin sijaintia muutetaan. Siirtää videot vastaavaan kohtaan.

Taulukko 5.2: TFormVideo.

5.2.4 TExerciseGroupView

TExerciseGroupView on näkymä, jolla hallitaan suoritusryhmää ja käyttäjärekisteriä. Sen kautta aloitetaan ja lopetetaan suoritusten nauhoitukset sekä käynnistetään vertailu.

	Nimi:	Kuvaus:
Attribuutti	number : Integer	Harjoituksen nostojen juokseva numero.
	savePath: String	Tallennuspolku.
	FormDetails: TForm	Nostajan yksityiskohtaiset tiedot näytettävä ikkuna.
	FormAddFromFile: TForm	Ikkuna, josta voidaan lisätä nostajia harjoitusryhmään tiedostosta.
	comparedliftsFrom: array[0..1] of integer	Vertailtavien nostojen tiedostonnimet.
	controller: TControllerForm	Osoitin controller-luokkaan.
Metodi	ButtonRemoveClick	Poistaa valitun nostajan harjoitusryhmästä.
	ButtonRecClick	Kysyy kaappausasetuksia ja aloittaa nauhoituksen.
	ButtonDetailsClick	Näyttää nostajan yksityiskohtaiset tiedot.
	ButtonCompareClick	Aloittaa vertailun.
	ButtonexercisegroupClick	Lisää nykyisestä harjoituksesta noston vertailtavaksi.
	ButtonFromFileClick	Lisää tiedostosta noston vertailtavaksi.

Taulukko 5.3: TExerciseGroupView.

5.3 Sovelluksen sisäiset luokat

5.3.1 TDetectorThread

TDetectorThread-luokka etsii TGrabber-luokan kaappaamista videon jokaisesta frameista ledin paikan ja tallentaa nämä tiedot TGrabber-luokan PointsOnScreen-muuttujaan.

	Nimi:	Kuvaus:
Attribuutti	WorkBuffer: FrameArr	Kaapatut framet.
	StartX, StartY: Integer	Ledin etsinnän aloituspiste.
	x1, y1: Integer	Viimeisen analysoidun framen ledin sijainti.
	Area: Integer	Ledin etsintään käytettävän alueen koko.
	diff: Integer	Arvo, jonka verran etsittävän pisteen rgb-arvo voi poiketa annetusta pisteestä (Pix).
	Pix: LongInt	Pikselin arvo, jota halutaan seurata.
Metodi	Pixel	Palauttaa kuvasta keskiarvon pisteistä, joiden rgb-arvo poikkeaa arvosta Pix vähemmän kuin diff.
	PixelAverage	Palauttaa kuvasta niiden pikseleiden värin keskiarvon, joiden rgb-arvot poikkeavat arvosta Pix vähemmän kuin diff.
	SyncWithGrabber	Palauttaa analysointitulokset ja herättää seuraavan säikeen.
	Detect	Etsii ledin sijainnin.
	Execute	Säikeen päämetodi.

Taulukko 5.4: TDetectorThread.

5.3.2 TAnalyzer

TAnalyzer-luokka analysoi nostosuorituksen biomekaaniset muuttujat tangon liikeradan ja käyttäjän antamien tietojen perusteella. Se ei siis käsittele varsinaista videota.

	Nimi:	Kuvaus:
Attribuutti	Lift: TLift	Analysoitava nosto.
Metodi	Analyze	Analysoi noston muuttujat ja tallentaa ne lifttiin.
	SmoothCurve	"Pehmentää" saadut koordinaattien arvot eli laskee liikkuvan keskiarvon kolmen peräkkäisen mittauspisteen koordinaateista.
	GetPowerOfTheMostSignificantDigit	Etsii sopivan pyöristystarkkuuden.
	RoundAllValuesToErrorPrecision	Pyöristää kaikki analysoidut tulokset.
	RoundVarList	Pyöristää yhden muuttujalistan.
	AnalyzeAveragePower	Laskee kahden framen välisen keskitehon.

Taulukko 5.5: TAnalyzer.

5.3.3 TController

TController on ohjausluokka, joka toimii muiden luokkien välissä viestien välittäjänä. Käytännössä se on sovelluksen pääformi.

	Nimi:	Kuvaus:
Attribuutti	grabber: TGrabber	Videonkaappauskomponentti.
	lifts: TLiftsContainer	Nostoja sisältävä attribuutti.
	currentCalibParam: array[0..1] of Double	Kalibrointitiedot.
	currentLift: Tlift	Käsiteltävä nosto.
	currentLifter: TLifter	Käsiteltävän noston nostaja.
	currentCurveColor: TColor	Liikeratakäyrän väri.
	videoForm: TForm	Videoikkuna.
	groupviewForm: TForm	Harjoitusryhmäikkuna.
	resultsForm: TForm	Ikkuna tulosten näyttämiseen.
	grabberForm: TForm	Videonkaappausikkuna.
	optionsForm: TForm	Asetusikkuna.
	aboutForm: TForm	About-ikkuna.
	savepath: string	Hakemistopolku tiedostojen tallennukseen.
Metodi	Complete	Tekee TVideosta Tliftin liittämällä siihen nostajan tiedot, painomäärän, noston tekniikan ja päivittämällä tiedostonnimit.
	Timer1Timer	Kirjoittaa kerran sekunnissa ajan ikkunan alareunaan.
	StartCapture	Aloittaa videonkaappauksen.
	StartCompare	Aloittaa vertailun.
	StopCapture	Lopettaa videonkaappauksen.
	Calibrate	Avaa kalibrointi-ikkunan.
	Save	Tallentaa valitun noston.

Taulukko 5.6: TController.

5.3.4 TLifterRegister

TLifterRegister-luokka lukee ja tallettaa käyttäjärekisterin tiedostoon. Se on peritty TList-luokasta, ja sitä voidaan käyttää myös suoritusryhmän tallentamiseen ajonaikaisesti. Objektien lisäämiseen ja poistamiseen tarvittavat metodit tulevat suoraan perittyinä.

Attribuutti	filename: String	Tiedostonnimi, josta käyttäjärekisteri luetaan / johon se tallennetaan.
	registerfile: TextFile	Käyttäjärekisteritiedosto.
Metodi	Delete	Poistaa nostajan.
	ChangeFileName	Muuttaa tiedostonnimen.
	Add	Lisää nostajan rekisteriin.
	WriteLiftersToFile()	Tallettaa käyttäjät tiedostoon.
	ReadLiftersFromFile()	Lukee käyttäjien tiedot rekisteritiedostosta.

Taulukko 5.7: TLifterRegister.

5.3.5 TLifter

TLifter sisältää sovelluksen yhden käyttäjän eli nostajan tiedot.

	Nimi:	Kuvaus:
Attribuutti	firstname: String	Nostajan etunimi.
	lastname: String	Nostajan sukunimi.
	height: Float	Nostajan pituus.
	weight: Float	Nostajan paino.
	lifts: TStringList	Ajonaikainen tieto suorituksista, jotka nostaja on tehnyt. Suoritukset tallennetaan nostovideon tiedostonnimen perusteella.
Metodi	Create(line)	Luo uuden nostajan tiedoston rivin tiedoista.
	GetAsLine(): String	Palauttaa nostajan tiedot merkkijonona, joka voidaan suoraan kirjoittaa tiedostoon.
	GetFullLiftFilename	Palauttaa kokonaisen tiedostonnimen, jossa on noston tiedot ja juokseva numerointi.

Taulukko 5.8: TLifter.

5.3.6 TVideo

TVideo-luokka sisältää videonpätkän, liikeratatiedoston sekä tiedot videon framemäärästä ja käytetystä koodekista.

	Nimi:	Kuvaus:
Attribuutti	videofilename: String	Videotiedoston nimi.
	Resolution: TResolution	Videon resoluutio.
	Codec: String	Videossa käytetty koodekki.
	CurveFileName	Levytangon liikeradan tallennustiedosto.
	X_Pixel_To_Cm: Double	Kalibrointitieto pikselin koosta sentteinä x-akselin suunnassa.
	Y_Pixel_To_Cm: Double	Kalibrointitieto pikselin koosta sentteinä y-akselin suunnassa.
	FrameRate: Integer	Videon kuvataajuus.
Metodi	Create(VideoFileName_: String; Resolution_: TResolution; Codec_: String; CurveFileName_: String; X_Pixel_To_Cm_, Y_Pixel_To_Cm_: Double)	Luo uuden TVideoon annetuista parametreista.

Taulukko 5.9: TVideo.

5.3.7 TLiftsContainer

TStringList-luokasta peritty säiliöluokka, joka tallettaa nostovideoita ajonaikaisesti. TLiftsContainer hoitaa myös videoiden lopulliset tallennukset kiintolevylle.

	Nimi:	Kuvaus:
Attribuutti		
Metodi	Destroy()	Vapauttaa kaikki nostojen käyttämät resurssit.
	Delete	Vapauttaa valitun noston varaamat resurssit.
	GetLift(filename)	Palauttaa filename mukaisen noston.

Taulukko 5.10: TLiftsContainer.

5.3.8 TLift

TLift sisältää yksittäiseen nostosuoritukseen kuuluvat tiedot kuten videon, nostajan ja nostosta lasketut muuttujat.

	Nimi:	Kuvaus:
Attribuutti	video: TVideo	Noston video.
	lifter: TLifter	Noston nostaja.
	varFile: TextFile	Muuttujatiedosto.
	varFilename: string	Muuttujatiedoston nimi.
	var_time: TVarList	Mittauspisteiden ajat.
	time_delta: TVarList	Ajan muutokset mittauspisteiden välillä.
	y_accel: TVarList	Kiihtyvyydet.
	x_coordinates: TVarList	X-koordinaatit.
	y_coordinates: TVarList	Y-koordinaatit.
	y_distance: TVarList	Matkat pystysuunnassa mittauspisteiden välillä.
	x_distance: TVarList	Matkat vaakasuunnassa mittauspisteiden välillä.
	y_speed: TVarList	Nopeudet pystysuunnassa.
	y_power: TVarList	Voimat pystysuunnassa.
	y_force: TVarList	Tehot pystysuunnassa.
	numberOfDataPoints: Integer	Mittauspisteiden lukumäärä.

Metodi	ReadFromCurveFile	Lukee liikeratatiedostosta mittauspisteiden ajat sekä x- ja y-koordinaatit.
	ReadFromFile	Lukee muuttujien arvot tiedostosta.
	WriteToFile	Kirjoittaa muuttujien arvot tiedostoon.

Taulukko 5.11: TLIft.

5.3.9 TVarList

TVarList on TList-luokasta peritty muuttujien tallentamiseen tarkoitettu luokka. Se sisältää tiedot muuttujan tyypistä ja niiden mittayksiköistä.

	Nimi:	Kuvaus:
Attribuutti	variable_type: TVariableType	Muuttujan tyyppi.
Metodi	Add	Lisää listaan uuden muuttujan arvon ja virheen.
	GetUnitOfMeasure	Palauttaa muuttujan mittayksikön.

Taulukko 5.12: TVarList.

5.3.10 TVariable

TVariable on muuttujaa kuvaava luokka, joka sisältää tiedot muuttujan arvosta sekä sen virheestä.

	Nimi:	Kuvaus:
Attribuutti	error: double	Muuttujan virhearvo.
	value: double	Muuttujan arvo.
Metodi		

Taulukko 5.13: TVariable.

5.3.11 TGrabber

TGrabber-luokka sisältää TVideoGrabber-komponentin. TGrabber-luokan parametrina olevaa TVideoGrabberia käytetään videonpätjän kaappaamiseen sekä videon näyttämiseen suoritusten vertailun ja kaappauksen sekä analysoinnin aikana. Videokamera on kytkettyä juuri tämän luokan olioon.

	Nimi:	Kuvaus:
Attribuutti	VideoGrabber: TVideoGrabber	Kuvankaappauskomponentti.
	Video: TVideo	Kaapattu tai avattu video.
	DetectorError: double	DetectorThread:n virhemarginaali.
	FDetectorList: TList	Lista suoritettavista säikeistä.
	FrameBuffer: FrameArr	Väliaikainen puskuri kaapatuille frameille.
	FCaptureFileName: String	Tiedostonnimi, johon videokuvaa kaapataan.
	CurveFile: TextFile	Liikeratatiedosto.
	StartX: Integer	Analysoinnin aloituspisteen x-koordinaatti.
	StartY: Integer	Analysoinnin aloituspisteen y-koordinaatti.
	DetectorListSection, PointsSection, FrameBufferSection: TRTLCriticalSection	Säikeiden synkronointiin liittyvät oliot.
	PointsOnScreen: FrameArr	Pisteet, jotka piirretään videokuvan päälle.
	X_Pixel_To_Cm: Double	Kalibroitivakio vaakasuunnassa.
	Y_Pixel_To_Cm: Double	Kalibroitivakio pystysuunnassa.
	CapturedImage: TImage	Videokuvasta kaapattu kuva.
Metodi	Calibrate	Suorittaa videon kalibroinnin.
	Capture_to	Aloittaa videon kaappauksen tiedostoon.
	PictCapture	Kaappaa yksittäisen framen kuvaksi.
	DrawCurve	Piirtää videokuvan päälle viivan PointsOnScreen-muuttujassa olevien pisteiden välille.

	WriteToFile	Kirjoittaa ledin sijainnin ja kalibrointitiedot tiedostoon.
	DrawOtherStuff	Piirtää videokuvan päälle haluttuja objekteja.
	BufferFull	Tyhjentää framepuskurin ja lähettää sen sisällön TDetectorThread-luokalle.
	OpenFile	Avaa videotiedoston.
	Play	Aloittaa videon toiston.
	Pause	Keskeyttää videon toiston.
	StopPlayer	Lopettaa videon toiston.
	StopCapture	Lopettaa videon kaappauksen.

Taulukko 5.14: TGrabber.

5.3.12FilenameHandler.pas

Tähän tiedostoon on koottu kaikki tiedostonnimen käsittelyyn liittyvät aliohjelmat. Niissä käsitellään tiedostoja, jotka ovat muotoa:

(tallennuspolku)Matti_Meikäläinen_Squat_1_120_190504.avi.

	Nimi:	Kuvaus:
Funktio:	ExtractWeightFromfilename	Eroottaa painojen määrän tiedostonnimestä.
	Separate	Eroottaa nostotekniikan ja juoksevan suoritusnumeron tiedostonnimestä.
	RemoveSavePathfromFilename	Poistaa tallennuspolun koko tiedostonnimestä.
	ChangeRunningNumber	Muuttaa tiedostonnimen juoksevan suoritusnumeron.
	CheckExisting	Tarkistaa, onko annettu tiedostonnimi jo olemassa. Jos on, muuttaa juoksevaa suoritusnumeroa.
	ExtractNameFromFilename	Eroottaa etu- ja sukunimen tiedostonnimestä.
	ChangeFileExtension	Vaihtaa tiedoston päätettä.

5.4 Ulkoiset komponentit

Tässä luvussa esitellään sovelluksen toteutukseen käytettyjä ulkoisia komponentteja, jotka eivät sisälly Delphin mukana tuleviin vakiokomponentteihin.

5.4.1 VideoGrabber

VideoGrabber-komponenttia käytetään videon kaappaamiseen ulkoiselta videolähteeltä eli digitaaliselta videokameralta. Sovelluksessa sitä käytetään myös videoiden näyttämiseen. TVideoGrabber-luokka on peritty Delphin TCustomPanelista. Se on tavallinen graafinen komponentti, jonka voi asentaa Delphin komponenttipalettiin ja lisätä lomakeikkunaan hiirellä vetämällä. Sen attribuuttien arvoja voi muuttaa graafisesti Delphin ominaisuuseditorin avulla. VideoGrabberissa ei ole videoiden näyttämisen hallintaan tarvittavaa painikeriviä valmiina, vaan se täytyy tehdä itse. KIHULLa on kyseiseen kaupalliseen komponenttiin lisenssi, joka on annettu projektiryhmän käyttöön. Lisätietoa komponentista löytyy sen tekijän verkkosivuilta [3].

5.4.2 ResizerPanel

ResizerPanel on Delphin vakiokomponenttia Panelia toiminnaltaan ja TPanel-luokkaa rajapinnaltaan vastaava komponentti. ResizerPanel säilyttää sillä olevien näkyvien komponenttien mittasuhteet, kun ikkunan kokoa tai resoluutiota muutetaan. Tämä helpottaa sovelluksen suunnittelua käytettäväksi erilaisilla resoluutioilla. Myös ResizerPanel voidaan asentaa Delphin komponenttipalettiin.

Paneeleita käytetään graafisissa sovelluksissa ryhmittelemään muita kontrolleja sekä niiden keskitettyyn hallitsemiseen. Komponentti on freewarea, ja tekijä antaa komponentilleen täyden käyttöoikeuden sovelluksissa. Komponentin ja lisätietoa siitä saa tekijän verkkosivuilta [1].

5.4.3 ExtendedListBox

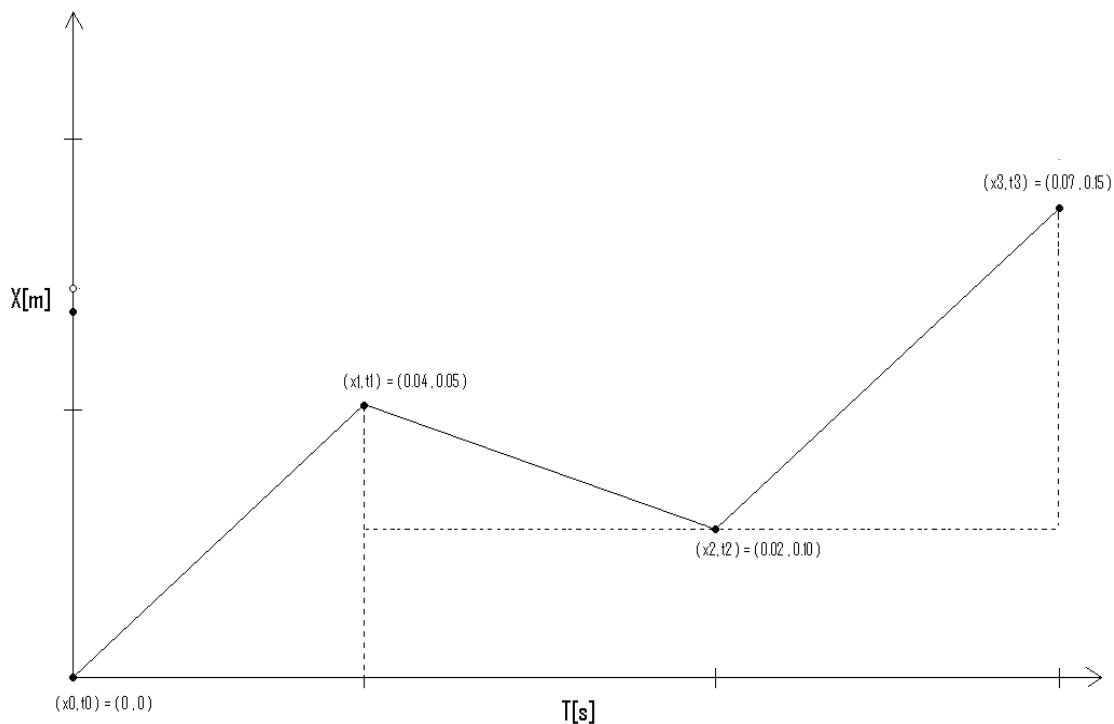
Projektiryhmän jäsenen Vesa Tanhua-Tyrkön yliopiston kurssilla tekemä komponentti, joka lisää Delphin tavalliseen ListBox-valintalistakomponenttiin lisäominaisuuksia. Rajapinta on samanlainen kuin TListBox-luokassa, täydennettynä esimerkiksi

tapahtumalla, jota kutsutaan kun valintalistan valinta muuttuu. Tekijä on luovuttanut komponenttiinsa tilaajalle täydet rinnakkaiset muuntelu-, käyttö- ja edelleenluovutusosoikeudet allekirjoittamalla oikeuksiensiihtosopimuksen. Komponentista voi kysyä lisätietoja Vesa Tanhua-Tyrköltä sähköpostiosoitteesta vtanhua@cc.jyu.fi.

5.5 Muuttujien laskenta

Tässä luvussa kerrotaan sovelluksessa biomekaanisten muuttujien laskemiseen käytetyistä kaavoista.

Kuvan 5.2 vaaka-akselin ajanhetket $t_0 - t_3$ vastaavat nauhoitetun videon joitakin neljää peräkkäistä framea. Näistä frameista mitattujen ledin paikkojen avulla saadaan laskettua halutut biomekaaniset muuttujat.



Kuva 5.2: Esimerkki mittaussarjasta, jonka tuloksena on saatu levytangon paikka ajan funktiona

Kaikki virhekaavat on johdettu käyttäen yleistä virheen etenemiskaavaa [10]

$$\delta f = \sqrt{\left(\frac{\partial f}{\partial q} \delta q\right)^2 + \left(\frac{\partial f}{\partial r} \delta r\right)^2 + \dots + \left(\frac{\partial f}{\partial w} \delta w\right)^2}, \quad (5.1)$$

missä $f = f(q, r, \dots, w)$.

Laskenta suoritetaan erikseen pysty- ja vaaka-akselilla. Kaikki vektorisuureet on esitetty skalaarimuotoisina. Laskenta on jäljempänä mainituin poikkeuksin samanlainen kummallakin akselilla, ts. kaavoissa (5.2) – (5.10) esiintyvien koordinaattien x_n tilalle voidaan tarvittaessa vaihtaa koordinaatit y_n .

Kuljettu matka:

$$s_n = x_n - x_{n-1}, \quad (5.2)$$

missä paikka x_n on jokin tuntematon funktio $x_n = f(t_n)$ ja vastaavasti $x_{n-1} = f(t_{n-1})$.

Matkan virhe:

$$\delta s_n = \sqrt{(\delta x_n)^2 + (\delta x_{n-1})^2}. \quad (5.3)$$

Virheitä δx_n ja δx_{n-1} ei voida johtaa virheen etenemiskaavalla, vaan ne saadaan videokameran kuvan ja käytetyn led-seuranta-algoritmin tarkkuuksista.

Matkaan s_n käytetty aika ja sen virhe:

$$\Delta t_n = t_n - t_{n-1}, \quad (5.4)$$

$$\delta \Delta t_n = \sqrt{(\delta t_n)^2 + (\delta t_{n-1})^2}. \quad (5.5)$$

Käytetään hetkellisen nopeuden $v(t_n) = \frac{dx_n}{dt} = \frac{df(t_n)}{dt} = f'(t_n)$ numeeriseen approksimointiin kolmipistemenetelmää [2], jonka mukaan

$$f'(t_n) = \frac{1}{2h} [f(t_n + h) - f(t_n - h)] + \frac{h^2}{6} f'''(\zeta_1), \quad (5.6)$$

missä $t_n - h < \zeta_1 < t_n + h$.

Koska funktiota $f(x_n)$ ei tunneta, virhetermin suuruutta ei voida luotettavasti arvioida. Sen sijaan käytetään yleisestä virheen etenemiskaavasta (5.1) saatavaa maksimivirheen arvoa.

Kolmipistemenetelmällä ei voida arvioida nopeutta mittausalueen päätepisteissä, ja nämä pisteet jätetään laskennasta pois (arvojen tuntemisesta päätepisteissä ei ole erityistä hyötyä).

Hetkellinen nopeus ja sen virhe:

$$v(t_n) = f'(t_n) = \frac{1}{t_{n+1} - t_{n-1}} [f(t_{n+1}) - f(t_{n-1})] = \frac{x_{n+1} - x_{n-1}}{t_{n+1} - t_{n-1}}, \quad (5.7)$$

$$\delta v(t_n) = \sqrt{\left(\frac{\delta x_{n+1}}{t_{n+1} - t_{n-1}}\right)^2 + \left(\frac{\delta x_{n-1}}{t_{n+1} - t_{n-1}}\right)^2 + \left(\frac{x_{n+1} - x_{n-1}}{(t_{n+1} - t_{n-1})^2} \delta t_{n+1}\right)^2 + \left(\frac{-x_{n+1} + x_{n-1}}{(t_{n+1} - t_{n-1})^2} \delta t_{n-1}\right)^2}. \quad (5.8)$$

Toinen derivaatta (kiihtyvyys) ja sen virhe saadaan vastaavasti:

$$a(t_n) = v'(t_n) = f''(t_n) = \frac{1}{t_{n+1} - t_{n-1}} [f'(t_{n+1}) - f'(t_{n-1})] = \frac{v(t_{n+1}) - v(t_{n-1})}{t_{n+1} - t_{n-1}}, \quad (5.9)$$

$$\delta a(t_n) = \sqrt{\left(\frac{\delta v(t_{n+1})}{t_{n+1} - t_{n-1}}\right)^2 + \left(\frac{\delta v(t_{n-1})}{t_{n+1} - t_{n-1}}\right)^2 + \left(\frac{v(t_{n+1}) - v(t_{n-1})}{(t_{n+1} - t_{n-1})^2} \delta t_{n+1}\right)^2 + \left(\frac{-v(t_{n+1}) + v(t_{n-1})}{(t_{n+1} - t_{n-1})^2} \delta t_{n-1}\right)^2}. \quad (5.10)$$

Hetkellinen voima (vaaka- tai pystyakselilla):

$$F(t_n) = m_t a(t_n), \quad (5.11)$$

missä m_t on käytetyn levytangon massa.

Hetkellisen voiman virhe:

$$\delta F(t_n) = \sqrt{(a(t_n) \delta m_t)^2 + (m_t \delta a(t_n))^2}. \quad (5.12)$$

Painonostajan aikaansaama voima F_p^x on ainoa levytankoon vaakasuunnassa (merkitsevästi) vaikuttava voima, ja se siis vastaa mittauspisteissä kaavalla (5.11) vaakakselin suunnassa laskettua voimaa: $F_p^x(t_n) = F^x(t_n)$. Vastaavasti levytangon kaavalla (5.9) vaakakselin suunnassa laskettu kiihtyvyys on sama kuin nostajan aikaansaama kiihtyvyys a_p^x , eli $a_p^x(t_n) = a^x(t_n)$.

Pystysuunnassa levytankoon vaikuttava kokonaisvoima on painonostajan aikaansaaman voiman F_p^y ja painovoiman F_g summa. Ts. mittauspisteissä kaavalla (5.9) pystysuunnassa laskettu kiihtyvyys $a^y(t_n) = a_p^y(t_n) + a_g$. Kun levytangon pystysuuntainen kokonaiskiihtyvyys eroaa vapaata pudotusta vastaavasta arvosta $a_g = -9,81 \frac{m}{s^2}$ (etumerkki viittaa kiihtyvyyden suuntaan alaspäin), erotus on nostajan aikaansaama (poislukien tietysti tilanne, jolloin tanko on maassa).

Painonnostajan (hetkellisesti) pystysuunnassa levytankoon kohdistama voima F_p^y on siis:

$$F_p^y(t_n) = m_t(a^y(t_n) - a_g), \quad (5.13)$$

missä $a^y(t_n)$ on kaavalla (5.9) pystysuunnassa laskettu kiihtyvyyys.

Voiman $F_p^y(t_n)$ virhe:

$$\delta F_p^y(t_n) = \sqrt{\left((a^y(t_n) - a_g)\delta m_t\right)^2 + \left(m_t\delta a^y(t_n)\right)^2 + \left(m_t\delta a_g\right)^2}. \quad (5.14)$$

Painonnostajan (pystysuunnassa) tekemän työn hetkellinen teho:

$$P_p^y(t_n) = F_p^y(t_n) \cdot v^y(t_n), \quad (5.15)$$

missä $v^y(t_n)$ on kaavalla (5.7) pystysuunnassa laskettu nopeus.

Tehon $P_p^y(t_n)$ virhe:

$$\delta P_p^y(t_n) = \sqrt{\left(v^y(t_n)\delta F_p^y(t_n)\right)^2 + \left(F_p^y(t_n)\delta v^y(t_n)\right)^2}. \quad (5.16)$$

Kun levytanko, jonka massa on m_t kohotetaan korkeudelle $h = y_n - y_0$ ajassa $t = t_n - t_0$, tehdään painovoimaa vastaan työ

$$W_g = F_g h = m_t a_g h, \quad (5.17)$$

jonka keskiteho on

$$P_{avg} = \frac{W_g}{t} = \frac{m_t a_g h}{t}. \quad (5.18)$$

Keskitehon virhe:

$$\delta P_{avg} = \sqrt{\left(\frac{a_g h}{t}\delta m_t\right)^2 + \left(\frac{m_t h}{t}\delta a_g\right)^2 + \left(\frac{m_t a_g}{t}\delta h\right)^2 + \left(\frac{m_t a_g h}{t^2}\delta t\right)^2}. \quad (5.19)$$

5.6 Ledin seurantaan käytetty algoritmi

Ledin seuranta videokuvassa perustuu sen väriarvojen erottumiseen ympäristöstään. Alue, jolta lediä etsitään, on kooltaan 30 kertaa 30 kuvapistettä. Ledin paikan löytämiseksi uudesta videokuvasta tutkitaan ledin paikasta edellisessä videokuvassa 15 kuvapistettä jokaiseen suuntaan. Jokaisesta videokuvasta etsitään pisteet, joiden väriarvot poikkeavat merkitystä ledin lähtöpisteen väriarvoista korkeintaan asetetun raja-arvon verran. Näiden pisteiden koordinaateista lasketaan keskiarvo, ja tämä arvo tulkitaan ledin kunkin hetkiseksi sijainniksi.

Tämä algoritmi mahdollistaa sen, että seurattavan pisteen ei välttämättä tarvitse olla valoa lähettävä ledi, vaan seurantaan riittää myös kohde, joka väriltään eroaa ympäristöstään selvästi. Kaikkein paras tulos kuitenkin saavutetaan, kun nostosuoritus kuvataan hämärissä olosuhteissa tummaa taustaa vasten ja seurattava kohde on ledi.

6 Tiedostot

Sovelluksessa käytetään tiedostoja datan tallennukseen. Liikeratakäyrän pisteet sekä analysointitulokset tallennetaan tiedostoihin, josta ne voidaan lukea uudelleenpiirtämistä tai suoritusten vertailua varten. Käyttäjärekisterit tallennetaan tiedostoihin, jolloin käyttäjien tietoja ei tarvitse joka kerta syöttää uudestaan. Videopätkät tallennetaan suoraan tiedostoihin, sillä ne ovat liian suuria suoraan keskusmuistissa käsiteltäväksi.

6.1 Tiedostojen formaatit

Liikeratakäyrät ja käyttäjien tiedot tallennetaan tavallisiin tekstitiedostoihin. Videotiedostot tallennetaan joko AVI : na tai MPG : nä.

Kaikki nostoon liittyvät tiedostot tallennetaan saman nimisinä erilaisin tiedostopäättein. Nimeämiskäytäntö on toteutettu yhdenmukaiseksi jo aiemmin KIHUn käytössä olevien järjestelmien kanssa. Tiedostonnimet ovat seuraavaa muotoa:

```
Etunimi_Sukunimi_Nostotekniikka_suorituksenro_painot_pvm.  
tiedostopääte,  
esimerkiksi Matti_Meikäläinen_Squat_4_150_160304.bal .
```

Esimerkki 7.1: Tiedoston nimi.

6.1.1 Käyttäjärekisteritiedosto

Käyttäjärekisteri tallennetaan tavalliseen tekstitiedostoon, jonka tiedostonimen päätteenä on `.usr`. Käyttäjistä tallennettavat kentät erotellaan `'|'`-merkin avulla. Ensimmäinen rivi on otsikkorivi, jossa selitetään kenttien sisällöt. Alla on esimerkki käyttäjärekisteritiedoston sisällöstä.

```
First name|Last name|Height|Weight  
Kalle|Ahola|190|123  
Maija|Meikäläinen|167|50  
Sami|Pajula|190|100  
Heikki|Hynninen|190|87  
Marika|Kinnunen|150|62
```

Esimerkki 7.2: Käyttäjärekisteritiedoston sisältö.

6.1.2 Liikeratatiedosto

Liikeratatiedoston tiedostopäätte on .BAL. Ensimmäisellä rivillä on pilkulla eroteltuina tiedot käytetystä kuvataajuudesta ja virhemarginaalista sekä kalibrointitiedot x- ja y-suunnissa. Kenttinä, jotka erotellaan pilkuin, x-koordinaatti, y-koordinaatti ja aika millisekunneina. Desimaalierottimena, kuten myös analysointitulostiedostossa, on piste.

```
25,0.005,0.43965,0.5049  
360,487,30  
360,488,70  
361,487,110  
361,488,150  
361,488,190  
361,487,230  
362,488,270  
362,488,310  
362,488,350  
362,488,390  
361,487,430  
361,488,470  
361,488,510  
361,488,550  
360,488,590  
360,488,630  
359,486,670  
357,482,710  
356,479,750  
354,470,790  
352,464,830  
350,452,870  
350,442,910  
350,426,950  
352,408,990  
354,396,1030
```

Esimerkki 7.3: Liikeratatiedoston sisältö.

6.1.3 Analysointitulostiedosto

Analysointitulostiedoston tiedostopääte on .CSV. Ensimmäisellä rivillä on tieto käytetystä kuvataajuudesta. Muilla riveillä ensimmäinen kenttä on mittausaika. Sen jälkeen on puolipiste sekä muuttujan virhemarginaali ja muuttuja pilkulla eroteltuina. Seuraavaksi on taas puolipiste, jonka jälkeen samalla tavalla virhemarginaali ja muuttuja pilkulla eroteltuina.

```
25
0.005,0.15;0,0;0,0;0,0;0,0;0,0;0,0;0,0;0,0;0,0;
0.005,0.19;0.008,0.041;0.008,-0.005;0.00707106781186548,0;0.09,-0.06;0,0;0,0;0,0;
0.005,0.231;0.008,0.041;0.007071068,0;0.008,-0.005;0.09,-0.06;2,1;40,30;70,420;
0.005,0.271;0.008,0.041;0.008,-0.005;0.00701,0;0.088385,0;2,1;36.9713227436,0;70,420;
0.005,0.31;0.008,0.04;0.00707106548,0;0.0070786548,0;0.0888835,0;2,1;36.9722736,0;70,420;
0.005,0.35;0.008,0.04;0.00707106781186548,0;0.00707106,0;0.09,0.06;1.5625,0;40,30;70,400;
0.005,0.391;0.008,0.041;0.007071548,0;0.008,0.005;0.0883883184,0;2,-2;27.88697,0;70,320;
0.005,0.43;0.008,0.04;0.0070710681186548,0;0.008,-0.005;0.09,-0.11;1.5625,0;40,50;70,400;
0.005,0.47;0.008,0.04;0.00707548,0;0.008,-0.005;0.0883883184,0;2,2;41.547076738,0;70,480;
0.005,0.51;0.008,0.041;0.008,0.01;0.008,0.005;0.09,0.06;2,-2;40,20;70,350;
```

Esimerkki 7.4: Analysointitulostiedoston sisältö.

6.2 Videokoodekit

Videon pakkaamiseen valitun koodekin valinta on erityisen tärkeää. Koodekkien välillä on suuria eroja niin kuvanlaadussa, vaaditussa tallennustilassa kuin nopeudessaakin. Koska analyysi tehdään pakatusta kuvasta, vaikuttaa kuvanlaatu myös analyysin lopputulokseen. Hyvin voimakkaasti pakkaavien koodekkien käyttöä tulee tästä syystä välttää. Lopullinen koodekin valinta jätetään kuitenkin käyttäjän tehtäväksi, mutta pelkkiä käyrien muotoja ja muuttujien arvoja tarkastelemalla ei voi tehdä päätelmiä analyysin oikeellisuudesta. Arvot täytyy jollain tapaa tarkistaa, jolloin tiedetään koodekin valinnan todellinen vaikutus analyysin tarkkuuteen Video pakataan valittua koodekkia käyttäen joko samanaikaisesti kaappauksen kanssa tai kaappauksen jälkeen. Edellinen vaihtoehto vaatii enemmän laskentatehoa ja jälkimmäinen hieman vähemmän. Videokuva on mahdollista myös jättää pakkaamatta, jolloin se vie suuren määrän levytilaa, mutta kuvan laatu säilyy alkuperäisenä.

7 Sovelluksen asentaminen

Sovelluksen asennukseen riittää projektin tuottaman CD-levyn sisältämän `OptiLift.exe`-tiedoston kopioiminen haluttuun hakemistoon. Tämän jälkeen sovellus on valmiina käytettäväksi. Uusia nostosuorituksia kuvattaessa tietokoneeseen on lisäksi liitettävä videokamera.

8 Vaatimusten täyttyminen ja sovelluksen jatkokehitys

Projektin aikana sovelluksesta saatiin toteutettua toimiva prototyyppi. Sen testaamisesta on kerrottu tarkemmin testausraportissa [6]. Sovelluksen jatkokehitys ja ylläpito siirtyvät tilaajalle. Tässä luvussa kerrotaan, miten sovellukselle asetetut vaatimukset ovat täyttyneet ja miten sovellusta voisi edelleen kehittää.

8.1 Vaatimusten täyttyminen

Kaikki sovellukselta vaaditut ominaisuudet saatiin toteutettua projektin puitteissa, mutta vaatimusmäärittelyyn kirjattiin muitakin ominaisuuksia, joita sovellukseen voisi toteuttaa.

Järjestelmän vaatimuksena oli, että urheilijan nostosuoritus pystytään analysoimaan välittömästi suorituksen jälkeen. Tämä vaatimus on toteutunut.

Nostosuuritukset, joihin järjestelmän tulee soveltua, ovat tempaus, työntö, etu- ja takakyykky sekä penkkipunnerrus.

Seuraavassa jokaisen vaatimuksen jälkeen on kerrottu sen toteutumisesta sovelluksessa.

Sovelluksen on täytettävä ainakin seuraavat vaatimukset:

1. Sovellus selvittää yksittäisestä nostosuorituksesta levytangon liikeradan.

Sovellus osaa selvittää yksittäisestä nostosuorituksesta levytangon liikeradan sillä edellytyksellä, että tangon päähän asetettu ledi erottuu riittävästi ympäristöstään.

2. Liikeradasta lasketaan kalibroinnin jälkeen aikaan perustuen seuraavat biomekaaniset muuttujat:

- matka:

- a) x-akselin suunnassa sivupoikkeaman laskemiseen

Matka lasketaan X-akselin suunnassa ledin alkupisteeseen nähden. X-akselin suuntainen matka on siis sivupoikkeama alkupisteestä. Negatiivinen

poikkeama tarkoittaa aloituspisteestä nostajaan päin (oikealle) kuljettua matkaa ja positiivinen nostajasta poispäin (vasemmalle) kuljettua matkaa.

b) y-akselin suunnassa tangon korkeuden laskemiseen

Matka lasketaan Y-akselin suunnassa ledin alkupisteeseen nähden. Y-akselin suuntainen matka on tangon korkeus alkupisteeseen nähden.

- nopeus y-akselin suunnassa

Toteutettu.

- voima y-akselin suunnassa

Toteutettu.

- teho y-akselin suunnassa

Toteutettu.

Muuttujille lasketaan lisäksi virhemarginaalit virhekaavoilla, jotka on johdettu käyttäen yleistä virheen etenemiskaavaa

3. Laskennan on tapahduttava 30:n sekunnin kuluessa nostosuorituksesta.

Sovelluksen laskentaosuus kestää tehokkaalla tietokoneella korkeintaan muutamia sekunteja. Myös hieman heikkotehoisemmat koneet suoriutunevat siitä reilusti alle vaaditun rajan.

4. Liikerata ja siihen liittyvät biomekaaniset muuttujat esitetään yhtäaikaaisesti graafisessa näytössä.

Liikerata piirretään videokuvan päälle kun analyysi on valmis, ja biomekaaniset muuttujat esitetään graafisina kuvaajina.

5. Videokuvaa voidaan vierittää alareunan palkilla eri vaiheisiin, jolloin vastaavat kohdat näytetään samanaikaisesti cursorilla graafisilla käyrillä ja taulukossa numeerisina arvoina.

Videokuvan alla on vierityspalkki, jolla kuvaa voidaan vierittää eri vaiheisiin. Kursori siirtyy graafisilla kuvaajilla vastaavaan paikkaan ja taulukon numeroarvot päivittyvät.

6. Sovelluksessa voidaan aluksi valita käyttäjärekisteristä usean urheilijan harjoitusryhmä, jonka suorituksia analysoidaan. Urheilijoista on tallennettu rekisteriin nimi, paino ja pituus.

Käyttäjärekisteritiedoista voidaan lukea urheilijoiden tietoja, ja heistä voidaan muodostaa harjoitusryhmä. Urheilijoista tallennetaan rekisteriin etu- ja sukunimi, paino ja pituus.

7. Urheilijoiden tietoja voidaan muokata ja urheilijoita voidaan sekä lisätä rekisteriin että poistaa sieltä.

Urheilijoiden tietoja voidaan muokata, ja heitä voidaan lisätä käyttäjärekisteritiedostoihin tai poistaa sieltä.

8. Joko saman urheilijan eri suorituksia tai kahden eri urheilijan suorituksia voidaan vertailla keskenään.

Suoritusten vertailu on toteutettu sovellukseen. Siinä ei käytännössä tarkisteta, kenen suorituksia vertaillaan.

9. Vertailussa kaksi suoritusta on pystyttävä manuaalisesti synkronisoimaan.

Vertailun aloituskohta voidaan asettaa manuaalisesti ja videot kulkevat tämän jälkeen synkronissa.

10. Videokuvalle laskuri, jonka avulla manuaalisesti voidaan valita aloitus- ja lopetuskohta sekä niiden välinen erotus tulee näkyviin käyrien numerotaulukossa.

Aloitus- ja lopetuskohdat voidaan valita manuaalisesti. Aloituskohta tarkoittaa kohtaa, josta videon pyörittäminen ja arvojen näyttäminen aloitetaan. *Comparison point* tarkoittaa arvoa, johon videokuvan sen hetkisen kohdan arvoja verrataan. Voidaan siis valita *comparison point*:ksi haluttu aloituskohta, ja vierittää videokuvaa palkilla tai syöttämällä aika numerotaulukoon haluttuun lopetuskohtaan, jolloin valittujen kohtien välinen erotus tulee näkyviin numerotaulukossa.

11. Käyttöliittymän tulee olla mahdollisimman yksinkertainen ja helppokäyttöinen.

Käyttöliittymästä on pyritty tekemään mahdollisimman yksinkertainen ja helppokäyttöinen. Tämä tarkoittaa esimerkiksi sitä, että kaikki turhat painikkeet on karsittu pois, ja painikkeiden, joiden käyttäminen ei sovelluksen ollessa tietyssä tilassa ole mahdollista, painaminen estetään. Käyttöliittymä on pääperiaatteiltaan toteutettu ensimmäisissä palavereissa tehtyjen luonnosten pohjalta.

12. Kaikki kuvaruudussa esitetyt tulokset voidaan tulostaa paperille.

Sovellukseen on toteutettu tulostustoiminto, joka tulostaa kaiken ruudulla näkyvän. Eri ikkunoille ei ole toteutettu omia tulostustoimintoja.

13. Sovellus laskee videokameran kuvataajuuden automaattisesti.

Sovellus laskee videokameran kuvataajuuden automaattisesti.

Sovelluksen toivotaan täyttävän seuraavat vaatimukset:

1. Sovelluksessa toimii virheenkorjaus, ts. suorituksen jälkeinen tietojen muuttaminen on mahdollista, jolloin nostosuorituksen analysointi voidaan suorittaa uudelleen.

Suoritusten uudelleenanalysointi on mahdollista, eli tallennetusta videosta voidaan analysoida liikerata uudelleen. Sen sijaan suorituksen jälkeinen tietojen muuttaminen ei vielä onnistu itse sovelluksessa. Nostajan tietoja ja tangon painoa voidaan muuttaa muokkaamalla tiedostonnimeä käsin, esimerkiksi vaihtamalla tangon painoa, sekä analysoimalla video tämän jälkeen uudelleen.

2. Sovellus laskee koko nostosuorituksen keskitehon.

Sovellus osaa laskea koko nostosuorituksen keskitehon, tai keskitehon kahden valitun pisteen (ajanhetken) välillä. Keskitehon tarkastelua ei kuitenkaan projektin puitteissa toteutettu käyttöliittymään.

3. Kahden suorituksen vertailussa voidaan valita näytettävä alue, joko koko videokuvapituus tai keskimäinen kolmannes.

Sovellus ei täytä tätä vaatimusta.

4. Sovellukseen liitetään työkalupakki, joka sisältää piirtovälineitä ja kulmamittarin.

Sovellus ei täytä tätä vaatimusta.

5. Sovellukseen pyritään saamaan mahdollisimman paljon automatiikkaa, jolloin järjestelmää pystytään käyttämään jopa yksin. Tämä tarkoittaa sitä, että sovellus tunnistaa milloin urheilijan nostosuoritus alkaa ja milloin se päättyy.

Sovellus ei vielä täytä tätä vaatimusta.

6. Sovelluksessa voidaan urheilijan usean toiston sarjasta poimimaan yksittäinen toisto.

Sovellus ei varsinaisesti täytä tätä vaatimusta. Eli jos suoritukset nauhoitetaan yhteen putkeen, ei niistä voida poimia yksittäistä suoritusta. Sen sijaan jos kukin suoritus nauhoitetaan erikseen, ei niitä tarvitse heti analysoida ja uusi nauhoitus voidaan aloittaa välittömästi.

7. Sarjasta voidaan tallentaa toistoja sekä poistaa niitä kesken harjoituksen levytilan säästämiseksi.

Toistojen eli nostovideon ja siihen liittyvien liikeratakäyrä- ja analysointitiedostojen poistaminen kesken harjoituksen on mahdollista.

8. Sovellus näyttää jäljellä olevan kovalevytilan sekä jäljellä olevan kuvausajan.

Sovellus ei vielä täytä tätä vaatimusta.

9. Sovellusta voidaan käyttää tehon testausohjelmana, jolloin usean eri painoilla tapahtuneen noston teho ja voima piirtyy samaan grafiikkaan.

Sovellus täyttää tämän vaatimuksen kahden noston osalta. Useampien nostojen vertailua ei ole toteutettu.

10. Suorituksesta voidaan tulostaa kuvasarjoja. Käyttäjä valitsee palkkia vierittämällä kuvasarjaan haluamansa framet.

Kuvasarjojen tulostamista ei ole toteutettu sovellukseen.

Lisäksi sovellusta toteutettaessa on otettava huomioon jatkokehitystä varten:

1. Urheilijarekistereitä voidaan tulevaisuudessa ladata verkon yli.

Rekisteritiedostot ovat hyvin yksinkertaisia ja niiden lataamiselle verkon yli ei ole mitään esteitä.

2. Sovelluksen kanssa voidaan käyttää erilaisilla kuvataajuuksilla toimivia videokameroita.

Videokameran kuvataajuus on otettu toteutuksessa huomioon.

3. Sovellus voi seurata useita pisteitä.

Liikeradan piirtäminen on pyritty erottamaan mahdollisimman hyvin videon toistosta, jolloin myös useamman pisteen seuraaminen on toteutettavissa oleva vaihtoehto.

8.2 Tunnetut virheet

Jos videonkaappauksessa jää joitakin frameja väliin, niin analyysituloksia näytettäessä videokuva ja muuttujia näyttävä kuvaaja eivät ole keskenään samassa vaiheessa.

Vertailtaessa kahta eri nostosuoritusta, joiden videoiden kuvataajudet poikkeavat toisistaan, muuttujia esittävät kuvaajat eivät näytä oikeisiin vaiheisiin kuuluvia arvoja. Yksittäisiä nostoja tarkasteltaessa muuttujien arvot ja video ovat kuitenkin samassa vaiheessa videon kuvataajuudesta riippumatta.

Juuri nauhoitettua videonpätkää ei kaikissa tapauksissa saatu automaattisesti aukeamaan videonkaappausikkunaan. Ongelma liittyy jollain tapaa videokuvan pakkaamiseen. Jos näin käy, täytyy juuri nauhoitettu video avata *Open video* painikkeella.

Videokameran johdon irtoaminen kesken nostosuorituksen nauhoituksen yleensä kaataa sovelluksen. Tämä virhetilanne tuli esille sovellusta testattaessa, mutta ratkaisukeinoa siihen ei löydetty.

8.3 Jatkokehitysehdotuksia

Sovellukseen voisi toteuttaa piirtovälineitä sisältävän ns. työkalupakin, jolla videokuvan päälle saisi piirrettyä erilaisia viivoja ym. Lisäksi se sisältäisi kulmamittarin, jonka avulla voitaisiin mitata esim. polvikulmia tietyissä noston vaiheissa.

Nostosuurituksen nauhoituksen voisi kehittää alkamaan automaattisesti, kun nostosuoritus alkaa, jolloin sovellusta voisi käyttää yksinkin. Lisäksi videoiden tallennusta voisi parantaa siten, että usein urheilijan molemmilla puolilla olevaa tyhjää tilaa ei tallennettaisi, mikä vähentäisi tallennustilan tarvetta.

Ledin seuranta-algoritmia on varmasti mahdollista kehittää vieläkin paremmaksi, jolloin nostosuoritusten kuvaaminen ei asettaisi kuvauspaikan valaistusolosuhteille niin suuria vaatimuksia.

Sovellukseen voisi toteuttaa videokuvan liikuttelun painikkeella taaksepäin kuva kerrallaan. Videokuvaa voi nykyisessä versiossa selata taaksepäin kuva kerrallaan videokuvan alla olevalla palkilla.

Sovelluksen käyttöliittymää voisi muuttaa käytettävämmäksi ja näyttävämmäksi. Projektin puitteissa käyttöliittymän graafisen ilmeen suurempaan muokkaamiseen ei kuitenkaan ollut aikaa.

9 Yhteenveto

Tässä OptiLift-projektin sovellusraportissa kerrotaan kevään 2004 OptiLift-Sovellusprojektin toteuttamasta sovelluksesta. Kyseessä oli Kilpa- ja huippu-urheilun tutkimuskeskukselle toteutettu painonnoston levytankoharjoittelun nostotekniikan automaattinen mittaus- ja analysointisovellus. Tämä dokumentti käsittelee sovelluksen toteutunutta arkkitehtuuria ja rakennetta vaatimusmäärittelyssä [7] esitettyjen vaatimusten pohjalta.

10 Lähteet

- [1] Barbora Carlos, "Carlos Barbora's homepage", saatavilla WWW-muodossa <URL: <http://www.carlosb.com/>>, viitattu 30.3.2004.
- [2] Burden Richard, Faires Douglas, "Numerical Analysis", Fifth Edition, PWS Publishing Company, Boston, 1993.
- [3] Datastead software, "Datastead software homepage", saatavilla WWW-muodossa <URL: <http://www.datastead.com/vidgrab/index.htm>>, 2004.
- [4] Jaakohuhta, Hannu, IT Ensyklopedia, Sanasto, Edita Oyj, Helsinki, 2001.
- [5] Laasala Lauri, Lukkarinen Olli, Räisänen Ville ja Tanhua-Tyrkkö Vesa, OptiLift-Sovellusprojektin projektiraportti, Jyväskylän yliopisto, Tietotekniikan laitos, 2004.
- [6] Laasala Lauri, Lukkarinen Olli, Räisänen Ville ja Tanhua-Tyrkkö Vesa, OptiLift-Sovellusprojektin testausraportti, Jyväskylän yliopisto, Tietotekniikan laitos, 2004.
- [7] Laasala Lauri, Lukkarinen Olli, Räisänen Ville ja Tanhua-Tyrkkö Vesa, OptiLift-Sovellusprojektin vaatimusmäärittely, Jyväskylän yliopisto, Tietotekniikan laitos, 2004.
- [8] Keränen Tapani, Viitasalo Jukka ym., Nostotekniikan automaattinen mittaus- ja analysointijärjestelmä-esittelymoniste, KIHU, Jyväskylä, 2003.
- [9] Swan, Tom, "Delphi 4", Teknolit Oy, Porvoo, 1999.
- [10] Taylor John, "Introduction to Error Analysis", Second Edition, University Science Books, 1997.