

Een collaboratief platform voor het efficiënt reconstrueren van fresco's

Nicolas Hillegeer

Thesis voorgedragen tot het
behalen van de graad van Master
in de ingenieurswetenschappen:
computerwetenschappen

Promotor:

Prof. dr. ir. P. Dutré

Assessoren:

Ir. N/A

N/A

Begeleider:

Dr. ir. B.J. Brown

© Copyright K.U.Leuven

Zonder voorafgaande schriftelijke toestemming van zowel de promotor als de auteur is overnemen, kopiëren, gebruiken of realiseren van deze uitgave of gedeelten ervan verboden. Voor aanvragen tot of informatie i.v.m. het overnemen en/of gebruik en/of realisatie van gedeelten uit deze publicatie, wend u tot het Departement Computerwetenschappen, Celestijnenlaan 200A bus 2402, B-3001 Heverlee, +32-16-327700 of via e-mail info@cs.kuleuven.be.

Voorafgaande schriftelijke toestemming van de promotor is eveneens vereist voor het aanwenden van de in deze masterproef beschreven (originele) methoden, producten, schakelingen en programma's voor industrieel of commercieel nut en voor de inzending van deze publicatie ter deelname aan wetenschappelijke prijzen of wedstrijden.

Voorwoord

Dit is mijn dankwoord om iedereen te danken die mij bezig gehouden heeft. Hierbij dank ik mijn promotor, mijn begeleider en de voltallige jury. Ook mijn familie heeft mij erg gesteund natuurlijk.

Nicolas Hillegeer

Inhoudsopgave

Voorwoord	i
Samenvatting	iv
Lijst van figuren en tabellen	v
1 Inleiding	1
2 Overzicht van het thera project	3
2.1 Het thera project	3
2.2 Situatie van de thesis in het project	4
3 Doelen van het project	5
3.1 Collaboratie	5
3.2 Gebruiksvriendelijkheid	5
3.3 Snelheid	6
3.4 Uitbreidbaarheid	6
4 Ontwerp van het project	9
4.1 Data, de basis van alles	9
4.2 Visualisatie, een manier om met de data te werken	10
4.3 Integratie in thera project	10
4.4 Uitbreidbaarheid	10
5 Bespreking van het database ontwerp	13
5.1 Vereisten	13
5.2 Externe database traag, interne database snel	13
5.3 Slimme client of slimme server?	14
6 Modules	15
6.1 MatchTileView	15
6.2 Proof of concept: GraphView	15
7 Tests & Vergelijkingen	17
7.1 Use cases	17
7.2 Snelheid (objectieve tests)	18
7.3 Gebruiksgemak (subjectieve tests)	19
8 Besluit	21
A Numerieke resultaten	25

Bibliografie

27

Samenvatting

Deze **abstract** moet nog geschreven worden.

Lijst van figuren en tabellen

Lijst van figuren

Lijst van tabellen

Hoofdstuk 1

Inleiding

Het reconstrueren van fresco's waarvan in opgravingen fragmenten gevonden worden is een moeilijke taak. Men kan het vergelijken met het oplossen van een enorme puzzel waarvan de stukken arbitraire vormen hebben, de meesten hun originele kleur zijn verloren en er vele anderen ontbreken. Daarbovenop komt nog eens dat er heel wat stukken over de eeuwen heen een zekere vorm van slijtage hebben ondervonden, waardoor ze niet meer 100% aan elkaar passen en dus confirmatie nog moeilijker maken.

[afbeelding van enkele opgegraven stukken]

De stukken van de rand van het fresco met recht afgelijnde kanten of die nog een voldoende zichtbaar geometrisch patroon bevatten, zijn in vergelijking met de anderen eenvoudig met elkaar te verbinden. De overige fragmenten die minder informatie bevatten zijn echter een nachtmerrie om aan elkaar te koppelen, gezien het menselijke visuele systeem niet gemaakt is om dat soort grillige randen te vergelijken en aan elkaar te zetten. Om deze reden is het normaalgezien noodzakelijk om vele mogelijke kandidaten visueel te onderscheiden en ze over elkaar te laten glijden om te zien of het past en waar precies.

In deze context situeert zich het **thera**¹ project, dat probeert om het werk van de archeoloog gemakkelijker te maken door middel van een software platform. De redenering achter het project is dat wat voor een mens zeer tijdsabsorberend en saai zou zijn, geautomatiseerd zou kunnen worden m.b.v. een computer. Dergelijk systeem werd in 2007 aan de *Princeton* universiteit in Amerika geconcipieerd. Sindsdien is er door verschillende onderzoekers van over de hele wereld aan gewerkt om alle noodzakelijke delen te ontwerpen, implementeren en integreren. [vermeld hoofdonderzoekers]. De werking van het systeem wordt in het volgende hoofdstuk nader toegelicht.

Deze thesis draait rond het maken van een uitbreiding op één van deze delen. De

¹Thera is de oude naam voor het huidige griekse eiland genaamd Santorini, waar het project voor het eerst in de praktijk werd toegepast.

uitbreiding moet de gebruikers van het systeem in staat stellen om op efficiënte wijze samen te werken aan het samenstellen van eenzelfde fresco.

De gangbare manuele methode van vóór het thera project om een fresco te reconstrueren leende zich wel tot een zekere vorm van samenwerking. De archeologen moesten in ieder geval op dezelfde site zijn om de fragmenten te kunnen zien en te proberen passen. Verder moest alle betreffende data van commentaren tot classificaties in een centrale plek bewaard worden zodat iedereen erbij kon. Dankzij het thera systeem veranderde deze manier van werken en kon men alle relevante kennis elektronisch gaan bijhouden. Er was echter tot op heden nog geen gemakkelijke manier om deze kennis met andere archeologen te delen en al zeker niet om de bevindingen van verschillende onderzoekers te combineren. Dat is het probleem dat deze thesis zal proberen uit de wereld te helpen. Kortom: de nodige aanpassingen en nieuwe implementaties zullen gedaan worden om het oude systeem collaboratief te maken, zodat het beheren, delen en combineren van data gemakkelijk en intuïtief wordt.

Hoofdstuk 2

Overzicht van het thera project

Zoals eerder gezegd bouwt deze thesis verder op een reeds bestaand project. Om de thesis beter te kunnen situeren in het geheel is het handig om eerst even het thera project bekijken en dan te zien waar dit project in past.

2.1 Het thera project

Ruwweg gezien kan het reconstrueren van een fresco met behulp van de computer opgedeeld worden in 3 fasen.

Acquisitie Alle gevonden fragmenten worden ingescand met behulp van 3D en 2D-scanapparatuur om zo een virtueel model van elk stuk te bouwen, zie [1].

Identificatie Vervolgens worden deze virtuele fragmenten aan een zogenaamde *matcher* gegeven, die voor elk fragmentenpaar gaat kijken of ze mogelijk op elkaar passen. Er zijn verschillende types ontwikkeld, waarvan één van de meeste succesvolle de zogenaamde *RibbonMatcher* [1] bleek te zijn. Deze *matcher* kijkt enkel naar de randen van de fragmenten en kan dus zelfs volledig kleurloze fragmentparen identificeren. Maar het onderzoek gaat verder, in 2010 werd er een nieuw type ontwikkeld dat zijn analyse baseert op een combinatie van verschillende eigenschappen, zoals de sporen die een borstel kan nalaten bij het kleuren van een fresco [2].

Classificatie De laatste stap bestaat uit het classificeren van wat de vorige stap produceert. Elk voorgesteld paar moet gecontroleerd worden op validiteit. Verschillende statussen kunnen zo toegekend worden aan een paar, zoals: *geconfirmeerd*, *misschien*, *nee*, *conflict*, et cetera. De *matcher* produceert namelijk zeer veel mogelijk passende paren, waarvan slechts een klein deel correct zal zijn. De drempel voor het beslissen wat een paar kan zijn en wat niet wordt in de **identificatie**-stap zo laag ingesteld omdat men wil vermijden dat twee fragmenten die toch passen genegeerd worden (de kost van zogenaamde “*false negatives*” wordt hoog ingeschat).

2.2 Situatie van de thesis in het project

Het thesisproject besproken in deze tekst tracht de reeds bestaande hulpmiddelen van deze derde en laatste stap (**classificatie**) aan te vullen. Er werden reeds programma's ontwikkeld om de classificatie uit te voeren, namelijk *Griphos* en *Browsematches*. Hierop volgt een korte bespreking van beide programma's, hun voor- en nadelen, en mogelijke uitbreidingen.

2.2.1 Griphos

TODO

[afbeelding van griphos]

2.2.2 Browsematches

TODO

[afbeelding van browsematches]

Hoofdstuk 3

Doelen van het project

Het hoofddoel van het hele thera project is om het hele proces van fresco's reconstrueren zo gemakkelijk mogelijk te maken. Daarom moeten ook de doelen van dit thesisproject zoveel mogelijk in functie van dit hoofddoel gekozen worden.

Gezien de huidige stand van zaken besproken in hoofdstuk 2, wordt daarop verdergebouwd om gebruikers in staat te stellen gemakkelijk samen te werken aan eenzelfde fragmentencollectie. Ideaal gezien zouden archeologen dit op elk gegeven moment vanop elke plaats in de wereld moeten kunnen doen. Een mogelijk scenario hierbij is dat een archeoloog in de Verenigde Staten de virtuele collectie kan bezien en classificeren. Deze veranderingen moeten dan gemakkelijk te laten zien zijn aan een archeoloog of assistent die toegang heeft tot de feitelijke brokstukken, zodat die kan nakijken of de paren die de archeoloog in de V.S. heeft geïdentificeerd correct waren.

3.1 Collaboratie

Het belangrijkste implementatie-aspect van de thesis. Alle andere aspecten staan normaalgezien in dienst van het vergemakkelijken van de collaboratie (indien niet staan ze wel in dienst van het hoofddoel van het thera project).

3.2 Gebruiksvriendelijkheid

Archeologen zijn dikwijls niet opgeleid om complexe applicaties zonder meer te gebruiken. Om deze reden moet er moeite gestoken worden in het ontwerp van een visueel aangename en intuïtieve gebruikservaring. Om deze reden wordt in het programma steeds de aandacht gevestigd op hetgeen het belangrijkste is: de fragmenten en de paren. Elke mogelijke operatie moet zo goed mogelijk zichtbaar gemaakt (liefst geen instellingendialoog bnb.). Ook moet er steeds uitleg voorzien worden van wat een bepaalde operatie werkelijk doet.

// hoort mss bij [DESIGN] De gebruikersinterface van de oude programma's was goed en kon efficiënt gebruikt worden mits enige training. Maar voor het ondersteunen van collaboratief werken moeten er natuurlijk allerhande nieuwe operaties toegevoegd worden. Tijdens het implementatieproces werd het duidelijk dat de reeds bestaande

code van het Browsematches niet uitbreidbaar genoeg was en die van Griphos te complex en belangrijk. Daardoor werd de beslissing genomen om de basisinterface van Browsematches over te nemen maar alle onderliggende code te herschrijven zodat die uitbreidbaar zou zijn. Bij het opnieuw construeren van dit alles zijn er een aantal verbeteringen gebeurd die niet meteen te maken hebben met het collaboratie aspect maar wel met de workflow van het classificeren. Dit werd gedaan om het hele programma gebruiksvriendelijker en krachtiger te maken in functie van het hoofddoel van het project.

3.3 Snelheid

Volgens vele studies op het gebied van gebruikersinterfaces [refereer een goed boek] geraken gebruikers gefrustreerd vanaf een operatie een zeker tijd duurt. Dit hangt een af van de aard van de operatie (het resultaat), de frequentie waarmee die uitgevoerd wordt, of er visuele tekenen van voortgang zijn en of er tijdens het wachten (steeds) iets anders kan uitgevoerd worden. Om deze reden is het van het grootste belang dit aspect in acht te nemen bij het ontwikkelen van de applicatie. Een programma kan nog zo goed zijn, een gebruiker die sloomheid ervaart zal het niet vaak gebruiken.

3.4 Uitbreidbaarheid

Bij het ontwikkelen van een platform voor een kennis- en ervaringsintensief domein als de archeologie waarvoor weinig precedentes bestaan, is het reeds van in het begin duidelijk dat het zal moeten vervangen worden als het niet met uitbreidbaarheid in gedachte ontworpen wordt.

(eerder bij design) Ontkoppeling data en visualisatie, beide orthogonaal uitbreidbaar.

3.4.1 Data

Een voorbeeld: een onderzoeker vindt een nieuw algoritme om fragmentparen te rangschikken. Deze rangschikking moet naadloos in het platform kunnen geïntegreerd worden zodat de gebruikers hier op kunnen zoeken (sorteren, filteren). Stel een archeoloog acht het nuttig om van een paar te kunnen opvragen wat het dikteverschil is tussen de twee fragmenten. Dit moet eveneens met een minimum aan moeite in het platform geïntegreerd kunnen worden. Voor zulke toevoegingen is het ook preferibel dat de visuele kant van het platform (de applicatie) zelf niet vervangen moet worden.

Data die complexer is en mogelijk speciale handelingen vereist om in een voorstelbare vorm te krijgen zal echter wel vergezeld moeten worden van een speciale visualisatiemodule om gebruikt te worden. Een vereiste hiervoor is wel dat de aanwezigheid van deze data de rest van de applicatie op geen enkele manier beïnvloed. Op die manier kunnen gebruikers met weinig modules zonder problemen gelijktijdig bewerkingen uitvoeren met gebruikers die er meer hebben (op dezelfde data).

3.4.2 Visualisatie

Er zijn vele manieren van visualisatie denkbaar die elkaar kunnen aanvullen bij het volbrengen van het zoek- en classificeerproces. Zo stelt men zich bij de uitdrukking “fresco’s samenstellen” waarschijnlijk een grote puzzel voor waar je ten alle tijde het overzicht kan behouden en stukken proberen te passen. Deze puzzel visualisatie is visueel aantrekkelijk en biedt het menselijke patroonherkenningsvermogen¹ de mogelijkheid om zich van zijn beste kant te laten zien. Echter, door de grote hoeveelheid aan fragmenten en dus mogelijke paren is het zeer moeilijk om hier aan te beginnen. Wanneer er reeds een deel geconfirmeerde paren zijn, zouden deze bijvoorbeeld op de puzzel kunnen verschijnen om zo een globaal overzicht te krijgen van de voortgang van het proces en om gericht te zoeken naar stukken. Dit is een macro-perspectief en een top-down manier om te reconstrueren.

Even naar het stuk waar de computer wél goed in is: fragmenten aan elkaar passen en rangschikken naar kans/overeenkomst/et cetera. Door een gemakkelijk navigeerbare lijst op te stellen van alle voorstellen die de reconstructie algoritmes hebben gedaan, kunnen er snel op elkaar passende fragmenten geïdentificeerd worden. Dit is een micro-perspectief en een bottom-up manier om fresco’s te reconstrueren. De puzzel visualisatie heeft na deze stap reeds een paar beginpunten om te tonen en is nu veel informatiever. Als er bijvoorbeeld duidelijke “gaten” ontstaan in een resem goede fragmentparen kan er gericht gezocht worden naar een fragment dat erin past (indien het gevonden werd bij de opgraving).

¹Iets waar computers de mens nog altijd niet in hebben geëvenaard. Een ander Thera subproject onderzoekt wel de mogelijkheid om zogenaamde ‘clusters’ te identificeren en te gebruiken voor reconstructie. [CITATIE]

Hoofdstuk 4

Ontwerp van het project

In dit hoofdstuk komt het werkelijke ontwerp van de applicatie aan bod. Met de doelen besproken in hoofdstuk 3 in het achterhoofd.

Afgezien van de zaken die puur op het collaboratieve aspect van de bestaande software verbeteren, zijn er natuurlijk nog andere zaken veranderd.

Browsematches werd vanaf de grond terug opgebouwd om een uitbreidbaar platform te worden dat zich leent tot gemakkelijke uitbreiding en ondersteuning biedt tot het selectief af- en aanzetten van modules. (om zo hun stabiliteit en inmpact op het programma te kunnen meten én om ze te kunnen vervangen indien nodig zonder de volledige applicatie te moeten herschrijven.)

4.1 Data, de basis van alles

TODO: verhuizen naar ontwerp van database

4.1.1 Het oude systeem: XML-bestanden

Eén van de belangrijkste stappen op weg naar een collaboratieve applicatie, is het omvormen van

Het ontwerp van het database model en de implementatie details zijn groot genoeg om hun eigen hoofdstuk te verdienen, zij worden later in hoofdstuk [...] besproken.

4.1.2 Ontleding van de data

...relationeel van aard => SQL database

Paren

Attributen van paren

Complexe informatie die niet in een simpel attribuut past

4.1.3 De vereisten van een database ontwerp

Zoals in sectie [doelen] omschreven moet de applicatie in staat zijn om data op een uitbreidbare manier uit te lezen en te veranderen, liefst zonder

4.1.4 Complexe informatie voorstellen als een attribuut

Meta-attributen, VIEWS

4.1.5 Verschillende iteraties

4.1.6 Databases samenstellen

Geschiedenis bijhouden

4.2 Visualisatie, een manier om met de data te werken

4.2.1 Model-View-Controller

4.2.2 UML diagramma

[maak UML diagramma]

4.2.3 GUI

Met behulp van de Qt toolkit [link]...

4.3 Integratie in theria project

Gebruik van bibliotheken, ... Extensie door refactoring: FragmentConf -> IFragmentConf ==> SQLFragmentConf | FragmentConf Hierdoor is het nodig om de reeds bestaande code van het theria project om te zetten naar het gebruik van IFragmentConf waar mogelijk maar het aanmaken van FragmentConf anderzijds of FragmentConf ==> SQLFragmentConf Alternatieve oplossing, geen veranderingen in theria code maar SQLFragmentConf sleept dan veel onnodige ballast mee van FragmentConf

4.4 Uitbreidbaarheid

Het programma is ontworpen met uitbreidbaarheid in gedachten. Dit heeft niet enkele betrekking op het klassen-ontwerp maar ook op het dynamisch en statisch

verwijderen en toevoegen van modules. Het basissysteem zonder modules bestaat uit een manier om een fragmenten en paren-database te laden en te kiezen uit welke module op te starten. Het ziet er zo uit:

[afbeeldingen invoegen]

Een module die ingeladen wordt weet niets over de hoofdapplicatie of andere modules behalve de huidige database. Deze informatie zit vervat in een object dat een model van een aantal paren voorstelt. Dit werd mogelijk gemaakt door een zogenaamde Model-View(-Controller)¹ [citatie] aanpak. Een module kan vragen om een applicatie-wijd of een nieuw model te krijgen. Met een applicatie-wijd model kan invloed uitgeoefend worden op andere modules die hetzelfde model gebruiken. Dit is handig bijvoorbeeld als men in één module een verzameling paren filtert en selecteert en men een andere module deze selectie wil visualiseren. applicatie-modellen zijn dus bijzonder nuttig voor visualizatie plugins. Om de werkbaarheid van het systeem uit te testen werd een voorbeeldmodule ontwikkeld die alle huidige paren in een grafe plaatst en deze met een ontwarringsalgoritme probeert te plaatsen, zodat men een globaler beeld kan krijgen van de huidige selectie.

[afbeelding invoegen, selectie naar grafe transit!]

¹De controller functionaliteit zit in de applicatie verdeeld bij het model en de view

Hoofdstuk 5

Bespreking van het database ontwerp

5.1 Vereisten

Gebruikers moeten een externe database server kunnen aanspreken om zo te werken op de meest actuele data. Het moet ook steeds mogelijk zijn om op een eigen kopie van de data te werken

5.2 Externe database traag, interne database snel

Het grote probleem dat duidelijk werd na het rechtstreeks werken met externe databases, is dat zelfs op een lokaal netwerk de verzoeken een zeer grote vertraging opleverden.

De redenen hiervoor waren natuurlijk de vele queries die elk object kan versturen om zijn attributen op te halen of te veranderen.

Nog een groot voordeel bij het maken van dergelijk systeem is dat er gebruik kan gemaakt worden van niet-destructieve veranderingen. Op het einde van een werksessie kan men de volledige lokale database ofwel in de hoofd database invoegen ofwel gewoon verwijderen.

Het design moest transparant zijn, zowel een rechtstreekse connectie als een gebufferde connectie zouden voor de eindgebruiker en de ontwikkelaar aan de buitenkant hetzelfde lijken.

Idee: Vervang object ID met object hash voor snellere match-to-match identificatie zonder een globale id nodig te hebben

5.3 Slimme client of slimme server?

Het programma bevat alle nodige functionaliteit en de server is in dat opzicht gewoon de specifieke database waarnaar het een verwijzing heeft.

Voor het gebruik op kleine apparaten (zoals tablets) zijn de mogelijkheden voor computationeel zware activiteiten niet zo uitgebreid. Daarenboven moet ervoor gezorgd worden dat de batterijduur van dit apparaat niet te hard beknot wordt door het gebruik van de applicatie. Daarom zal de transitie naar zeer mobiele platformen enkel mogelijk worden indien er een simpele client applicatie kan ontwikkeld worden die op de server vertrouwd om de juiste berekeningen te maken.

In de thesisperiode is geen tijd gevonden om dit concept uit te werken maar er zijn wel plannen gemaakt waardoor dergelijke applicatie op een efficiënte manier zou kunnen ontwikkeld worden. De gebruikersinterface Tangerine voert alle functies met betrekking tot de achterliggende data uit met behulp van een grote bibliotheek van klassen die op zich niets met de interface te maken hebben. Er zou dus een alternatief programma gebouwd kunnen worden die in plaats van met een grafische interface kan bediend worden via een zelfgemaakt protocol. Dit soort ontwerp wordt vaak aangetroffen in de UNIX wereld, waar er een enkele server-variant van een programma bestaat en meerdere mogelijke clients die ervan gebruik maken. Het bekendste is misschien wel de X server.

Hoofdstuk 6

Modules

Hier komen de modules!!!!

6.1 MatchTileView

6.2 Proof of concept: GraphView

Hoofdstuk 7

Tests & Vergelijkingen

Een goede manier om te weten te komen of het project zoals dit nut heeft gehad is om tests uit te voeren. Er zal worden vergeleken tussen de reeds bestaande oplossingen voor het classificeren van fragmentparen en de nieuwe applicatie waar mogelijk. In sommige gevallen zijn er nieuwe zaken toegevoegd die vroeger op geen of slechts manuele wijze mogelijk waren, hier zal worden vergeleken met (evt. fictieve) alternatieve implementaties. Voor het geval er helemaal geen vergelijking tussen alternatieven mogelijk is, zal een zo goed mogelijke maatstaf bedacht worden zodat er toch een beeld ontstaat van hoe de applicatie het doet.

7.1 Use cases

Elke gebruiker wenst met een applicatie een bepaald doel te bereiken, in dit geval is dat het classificeren van fragmentparen. Hiervoor zullen de mogelijkheden die het programma aanbiedt hopelijk zowel voldoende als gemakkelijk in gebruik zijn.

- 7.1.1 Vind uit bak ONGESORTEERD het paar dat het meeste ruimte inneemt
- 7.1.2 Vind alle mogelijke paren die conflicteren met een waarschijnlijk paar
- 7.1.3 Vind alle mogelijke paren die grenzen aan het geselecteerde paar maar niet conflicteren ermee
- 7.1.4 Voeg een andere database bij de huidige, prefereer de veranderingen van de andere als er conflicten zijn
- 7.1.5 Gebruik conflict detectie & 3D voorstelling om correcte paren te identificeren
- 7.1.6 Duid aan dat een verzameling mogelijke paren eigenlijk hetzelfde paar zijn (duplicaten)
- 7.1.7 Bekijk alle duplicaten van een paar
- 7.1.8 Vind alle paren die geclassificeerd staan als 'misschien'/'maybe', schrijf bij een paar je mening ('comment')
- 7.1.9 Exporteer een database naar een leesbaar formaat (XML)
- 7.1.10 Importeer een database van een leesbaar formaat (XML)
- 7.1.11 Kopieer een aantal paren naar Griphos
- 7.1.12 Selecteer een aantal paren om te groeperen en in 3D weer te geven
- 7.1.13 Vindt wat je weet zijn: gebruik de zoek- en sorteerfuncties om snel een gewenst deel van de verzameling paren te zien
- 7.1.14 Vindt wat je niet weet zijn: gebruik de zoek- en sorteerfuncties om snel veelbelovende paren te vinden
- 7.1.15 Werken op een externe database (gebruiksgemak)
- 7.1.16 Databases samenstellen: conflicten oplossen

7.2 Snelheid (objectieve tests)

- 7.2.1 Opstartsnelheid
- 7.2.2 Inladen van fragmenten

Meet het inladen van fragmenten, verschil tussen browsematches en tangerine (intern en extern)

7.2.3 Navigeren tussen schermen

Meet het navigeren tussen schermen, sorteren, filteren

Teruggaan naar eerder schermen

Door een caching mechanisme...

7.3 Gebruiksgemak (subjectieve tests)

Subjectieve waardeschattingen van gebruikers

7.3.1 Intuïtiviteit

Het doelpubliek van de applicatie bestaat in de nabije toekomst uit archeologen en eventueel later uit amateurs die helpen met de classificatie. Qua gebruikersprofiel passen archeologen in het plaatje van een gebruiker met veel domeinkennis (classificeren), maar weinig kennis of ervaring met het gebruik van geavanceerde computerapplicaties. Anders gezegd: de gebruikers weten wat ze willen en moeten doen, maar niet noodzakelijk hoe ze het moeten doen in de applicatie.

Voor elk programma kan training voorzien worden zodat er uiteindelijk optimaal gebruik van kan gemaakt worden, maar het doel is om deze training tot een minimum te beperken. Dit houdt in dat de functies ontdekbaar

Informatie-popups (eenmalig/meermalig)

7.3.2 Responsiviteit

Voor het dagdagelijkse gebruik van een applicatie is het van groot belang dat de gebruiker het niet hinderlijk vindt om ermee te werken. In enkele werken over gebruikersinterface ontwerp [aan elkaar?] [citatie] wordt erop gewezen dat de snelheid waarmee een applicatie reageert een sleutelfactor is voor het gebruiksgemak. Een algemene vaststelling: de tijd die een operatie mag innemen is omgekeerd evenredig met de frequentie waarmee deze operatie moet uitgevoerd worden. Deze regel in acht nemend is het duidelijk dat bijvoorbeeld het inladen van een scherm, het veranderen van een attribuut, het filteren en sorteren en dergelijke meer acties zijn die met de grootst mogelijke snelheid moeten worden uitgevoerd.

Voor al deze operaties is sinds het eerste ontwerp van de applicatie rekening gehouden met de implicaties van elke beslissing op de snelheid. Hierdoor is de snelheid van de applicatie op elke vlak verbeterd tegenover zijn voorgangers, soms op dramatische wijze. Om dit objectief vast te stellen zijn er een paar tests uitgevoerd:

Hoofdstuk 8

Besluit

Nog geen besluit

Bijlagen

Bijlage A

Numerieke resultaten

Nog niets hier

Bibliografie

- [1] B. J. Brown, C. Toler-Franklin, D. Nehab, M. Burns, D. Dobkin, A. Vlachopoulos, C. Doulas, S. Rusinkiewicz, and T. Weyrich. A system for high-volume acquisition and matching of fresco fragments: Reassembling Thera wall paintings. *ACM Transactions on Graphics (Proc. SIGGRAPH)*, 27(3), Aug. 2008.
- [2] C. Toler-Franklin, B. Brown, T. Weyrich, T. Funkhouser, and S. Rusinkiewicz. Multi-feature matching of fresco fragments. In *ACM Transactions on Graphics (Proc. SIGGRAPH Asia)*, Dec. 2010.

Fiche masterproef

Student: Nicolas Hillegeer

Titel: Een collaboratief platform voor het efficiënt reconstrueren van fresco's

Engelse titel: Een collaboratief platform voor het efficiënt reconstrueren van fresco's

UDC: 621.3

Korte inhoud:

Hier komt een heel bondig abstract van hooguit 500 woorden. \LaTeX commando's mogen hier gebruikt worden. Blanco lijnen (of het commando `\par`) zijn wel niet toegelaten!

Nam dui ligula, fringilla a, euismod sodales, sollicitudin vel, wisi. Morbi auctor lorem non justo. Nam lacus libero, pretium at, lobortis vitae, ultricies et, tellus. Donec aliquet, tortor sed accumsan bibendum, erat ligula aliquet magna, vitae ornare odio metus a mi. Morbi ac orci et nisl hendrerit mollis. Suspendisse ut massa. Cras nec ante. Pellentesque a nulla. Cum sociis natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus. Aliquam tincidunt urna. Nulla ullamcorper vestibulum turpis. Pellentesque cursus luctus mauris.

Thesis voorgedragen tot het behalen van de graad van Master in de ingenieurswetenschappen: computerwetenschappen

Promotor: Prof. dr. ir. P. Dutré

Assessoren: Ir. N/A
N/A

Begeleider: Dr. ir. B.J. Brown