Inflammation evaluation through low-cost 3D scanning of human body parts

Author: Guillermo Girona
Director: Carlos Andújar
GEP Tutor: Jasmina Berbegal

Specialization: Computer Engineering

This project is one of the best reports from a previous edition of GEP. The work is of great quality; therefore it can be taken as an example when writing your own project. Nevertheless, it is important to highlight some aspects that came out in the final presentation and that could be further developed in order to improve the report. Find them listed below:

- Task description:
  - In order to have a more comprehensive picture of the project, prior to start with task description, a list indicating which tasks are included is needed.
  - Resources should be explicitly linked to each of the tasks, so that it is possible to figure out resource consumption at each stage of the project.
- Costs: Relevant costs are clearly identified and discussed. However, there are two points that deserve further attention:
  - Human resources costs are not realistic (too cheap).
  - Not all potential costs are considered (you have to imagine that someone has hired you for doing this project). Have a look at the document uploaded in ATENEA (Mòdul 2.4.1 - Costos i sostenibilitat del projecte informatic.pdf) for general expenses.
- Link planning-budget: It would be great if the planning (deliverable 2) could be somehow reflected in the budget. In the ATENEA you will find an example on how to do this (Mòdul 2.4.2 - Ejemplo presupuesto a nivel de Actividades en el Gantt.pdf).
- State of the art: An additional paragraph at the end of this section should be added describing the main gaps that are still to be covered in the literature which justify the need for carrying out your. Link this discussion with the main features that your solution is going to include and that are not currently addressed by existing platforms.

Inflammation evaluation through low-cost 3D scanning of human body parts
Final deliverable

Guillermo Girona San Miguel

Director:	Carlos Andújar
Specialisation:	Computing

October 2014

Contents

1 Introduction

The use of computer applications is becoming a very common practice in the modern society. Mobile devices like smartphones or tablets have become very popular, and that has lead to the creation of applications that solve any kind of problem that one can imagine. As it could be expected, there are applications that help a person to know if he or she is in good health. There are applications that measure the heart rate, applications that help the user to keep a healthy diet and applications that give indications of how to act in a medical emergency.

In this project we aim to develop a tool that given data of the same body part in different states of inflammation compares them in order to give the user information about the evolution of the inflammation. Using our application a patient will be able to compare a joint with its symmetrical one, or with the same joint in a past state.

The data used will be a 3D model of the body part that suffers from inflammation, thus allowing us to make use of 3D graphics and virtual reality techniques in order to give the user the feedback that he or she needs.

This document aims to give the reader a detailed description of the project. It is divided in several sections, giving details about the scope, the context, the temporal planning and the economic planning, respectively.

2 Scope of the project

This section aims to make a description of the scope of our project. We are going to start with the formulation of the problem that we want to solve. Then, the scope of the project is going to be defined, describing all the possible obstacles that we may face during its development. Finally, we are going to describe how are we going to work to solve the problem in the *Methodology and rigour* section.

2.1 Objectives

The main objective of our project is to develop a tool that given data of the same body part in different states of inflammation compares them in order to give the user information about the evolution of the inflammation. We aim to design the tool in a way the user will find easy to track the evolution of his or her inflammation, providing him or her enough information in a way it is intuitive to read and understand.

A secondary objective is to implement algorithms capable of comparing the data representing the different states of the same body part in order to be able to give the user useful feedback.

The data used will be a 3D model of the body part that suffers from inflammation. In order to do that, we define another secondary objective of the project, which consists in using different 3D scanning techniques to obtain the 3D model of a body part.

2.2 Scope

In order to be able to solve the problem that defines our project we need a program capable of loading two or more 3D models of the same body part and comparing them, showing the user the differences caused by swelling, which is a direct effect of inflammation. So, we are going to design a 3D viewer with more features than a the ones that a common viewer would have, but since it is not the most important part of the project, we are not going to explore advanced 3D features. We are going to try to create a program that is able to show the information the user needs in an intuitive way, but anything else.

Algorithms to compare different 3D models that represent the same physical object that may have a slightly different shape are going to be needed. Our project requires at least a working algorithm, but we are going to explore different algorithms in order to compare them and choose the one that offers the best results. These algorithms have to detect the variation of the inflammation level, but it would be good if they could also detect the level of inflammation of each zone of the body part. Furthermore, it would be desirable that they could match automatically the different models that represent the same body part.

Finally, we need a way to obtain the 3D model of a given body part. In order to do that, we are going to explore different low-cost techniques of acquiring a 3D model of a physical object. We already have a technique that works, which is the use of a light-based 3D scanner. We consider it being a low-cost technique because even if the price of the scanner is high, it can be used plenty of times, so the price per use is low. It has a problem, though, since not everyone can have access to this kind of scanner. That is a reason to explore techniques that are available for everyone. We propose using a *Kinect*, which is a device that a great amount of the people that owns a *Xbox* console has, and which has got extensive support for software development. Another proposal is the use of multi-view stereo, since everyone can have access to a camera. We plan do not plan to explore more techniques.

We have explained what we have planned to include in the project. We also have mentioned some things that we are not going to include in the project, but we may include them if our planning is too pessimistic and we actually have more time than we expected.

2.3 Methodology and rigour

In order to develop the project we have planned a rough development schedule that is going to help us in its creation. This is what is going to be described in the following lines.

First of all, we wanted to assure that the project was something that we were going to be able to do. Before deciding on choosing this project some basic tests were carried in order to decide if the project would be possible. These basic tests consisted in doing some measures on a 3D model of a body part, these measures were closely related to the ones that we are going to need in this project. The tests were successful.

The development of the project itself is going to start with the creation of a first version of the viewer. The reason to start with it is the fact that we need the viewer in order to check any results, so it is the first we have to do, even if it has only basic features. It has to be able to load 3D models and do basic navigation tasks like pan, rotate or zoom.

The next step is going to be exploring ways of comparing different states of the same body part. That involves the research and testing of different algorithms that give different measures that can be used for comparison reasons. As it could be expected, in order to perform the tests needed, we are going to need at least a 3D model from a body part.

Finally, we are going to evaluate other methods to obtain 3D models, apart from the light-based 3D scanner. That involves the use of the *Kinect* and the exploration of multi-view stereo techniques.

After having a basic version of the program which solves our problem, using at least a working comparison algorithm and a working scanning technique, we will try to make improvements to the program. It is going to be an iterative process that is going to involve the main program, the comparison algorithms and the scanning techniques.

2.3.1 Development tools

The development of the main program of our project is going to use *Qt* and *OpenGL*. *Qt* is going to be used for the creation of the user interface and *OpenGL* is going to be used for rendering 3D graphics. The main advantages of using them is that they are both open-source and can be used in different platforms like *Windows*, *Linux* or *Mac*. Other tools like *Blender* or *MeshLab* may be used, since they offer many tools that transform 3D models. We might also consider using external libraries, since a lot of code that is related to 3D graphics has already been developed. Everything is going to be be programmed in *C++*, since it is the native language of the *Qt* package, with the exception of the code that involves *Blender*, that has to be written in *Python*.

Git and *BitBucket* are going to be the tools used to track the development of the project. Using them allows us to document all the changes that are done to the code.

2.3.2 Validation of results

To check if the program is working as we want we only have to obtain a 3D model of a body part. There are 3D model libraries that contain models of body parts, so we can obtain a model from there. Then, we can use a 3D model modification program like *Blender* or *MeshLab* to modify the model, simulating the effect of inflammation, then use the program and the information it gives to decide if it is working properly.

What we stated in the last paragraph would validate the program, but we also need a way to check it the 3D scanning techniques that we have explored are useful for our purposes. Doing that is easy, since we just have to do the same than before, but using the scanned model instead of the model from a library.

Finally, it would be good to use the program in a real case. We will try to find a person that has inflammation on one of his joints and use the program to track the evolution of inflammation. We can also compare the joint with its symmetrical one if it has no inflammation, since even if both joints are always different, it is a good reference.

2.4 Obstacles and risks of the project

The problem that we want to solve is well defined, and there are parts of the project like the creation of a 3D viewer or 3D scanning techniques that have been extensively developed. However, the development of the project might not be that easy, since we may face different problems. The main ones have been listed and explained in the following sections.

2.4.1 Main program

The main program is going to be a 3D viewer with the features that this kind of application needs, that is, apart from navigating tools, which is something that every viewer has, there is going to be a way to manage the different joints evaluated and the results of the comparison between joints is going to be shown to the user as graphically as possible. There should not be any problem when creating the main program since there are plenty of 3D viewers created, some of them being open source. Also, we have created similar programs in past courses. The main challenge that we are going to face when creating the viewer is to create a program that is intuitive enough for someone that may not have any experience with 3D applications.

2.4.2 Algorithms used

Although it seems easy to compare two states of the same body part and decide if the inflammation level has changed, it could be a difficult task to do. Having two models of the same body part, it is relatively easy to compare them if they are aligned and they have the same scale. But if the scale is different or they are not aligned, it could be more difficult. In the last case, we would need a way to align both models. Ideally, this would be done automatically, but it could be the case that it would have to be done manually. Moreover, not all the measures used to compare two models could work for our purposes. There are measures that look easy to calculate, like the volume, but it may not give us the results desired. We are going to need extensive testing too since we may use measures that make it difficult to tell if the algorithm is working as we expected.

2.4.3 Scanning techniques

As we said, we already have some 3D scanning techniques that work. However, it would be useful to find techniques that are easier for the user. Microsoft, creators of the *Xbox*, provide a SDK for the *Kinect*, and one of the features that it includes is the generation of a 3D model of a physical object. That means that there should not be any problem when using the *Kinect* to obtain a 3D model.

Multi-view stereo is more complicated. Since skin has got a homogeneous appearance, it could be difficult to match points in different images, hence making the reconstruction an impossible task. Different ways to solve that, such as manual identification of key features of the body part should be explored if this technique is used.

Finally, having a 3D model is not enough to assure that it will work with our program. 3D scanning often generate models with undesirable properties caused by precision errors, noise or by the algorithm used in the construction of the model from the data gathered by the scanner. These models need to be fixed in order to allow the comparison algorithms to work properly.

3 Contextualisation and bibliography

This section is about the context of the project and its bibliography. It features two different sections. The first one aims to describe the context of the project, that is, a brief description of the area of interest that the project is about and the actors affected, while the second one gives information about the state of the art of the areas that we are going to explore in the project.

3.1 Context

In this section we are going to do a description of the areas of interest that this project is about, and we are going to explain which are the actors that are going to be affected by the development of this project.

3.1.1 Areas of interest

In this project we have two clearly different areas of interest. The first one is inflammation, which is what we are trying to help to heal with the application resulting of the project. The second one is the one which involves technical aspects of the project like the creation of the program or the obtaining of 3D models.

In the next lines we are going to detail both areas.

Inflammation In this project we are going to make a program that is going to be used to track the evolution of inflammation. To be able to create a program that satisfies our needs, we have to know what inflammation is, taking into account its main characteristics.

Inflammation is the biological response of vascular tissues to harmful stimuli, such as pathogens or damaged cells. It is a protective attempt of the organism to remove the injurious stimuli and to initiate the healing process. Inflammation can be caused by burns, infection by pathogens, physical injuries or by some diseases like asthma, allergies, inflammatory bowel diseases or rheumatoid arthritis. The five signs that characterise inflammation are pain, heat, redness, swelling and loss of function [1].

In this project we are going to focus on the inflammation that appears in joints. It is a very common kind of inflammation, since it can appear as a result of an injury like a sprain or a luxation, which are both injuries that most of the people that practise any kind of sport will easily suffer. Also, there is a disease that causes it, rheumatoid arthritis, which affects between 0.5 and 1% of adults of the developed world [2].

We have already said that there are five signs that characterise inflammation. Its control is the way to track the evolution of inflammation. Our project is going to take into account swelling only, since it is the sign that involves the geometry of the body part.

Technical aspects In order to develop the project, we have to take into account a few technical aspects.

First of all, we need a program capable of representing a 3D model. That is, we need a 3D viewer. We have planned to include the common features a 3D viewer has, like rotate, pan and zoom, plus other features that can be useful for the kind of program that we are developing.

Also, we need a way to obtain 3D models from physical objects. That is, we need a 3D scanner. A 3D scanner is a device that analyses a real-world object or environment to collect data on its shape and possibly its appearance (i.e. colour). The collected data can then be used to construct digital three-dimensional models [3].

Finally, we need methods to work with the 3D models so that we can get the data needed in order to give the user information about the evolution of his or her inflammation. That is, we need algorithms that are able to process 3D models in order to obtain information from them. When trying to find algorithms that are able to process 3D models in order to give some data we enter a field called computational geometry. Computational geometry is a branch of computer science devoted to the study of algorithms which can be stated in terms of geometry [4].

In the *State of the art* section, we are going to give more details about the items mentioned in this section, giving then information about specific techniques that have already been developed.

3.1.2 Actors

The development of our project involves several actors, which are listed and described in the following lines.

Project developer This project has an only developer, myself. I am going to work on everything which is needed, that means that I am who is going to work on the project plan, the documentation, the information research, the code and the tests.

Project director The director of this project is Carlos Andújar, associate professor from the Computer Science department of Universitat Politècnica de Catalunya. His role is to guide the developer of the project so that I can carry it out when having difficulties.

Users Our aim is make a program such that everyone can be considered a potential user of it, so, if someone suffers from an injury that causes inflammation he or she could use our program to track its evolution. There is a problem, though, since depending on the scanning techniques that we manage to use not everyone is going to be able to obtain the data needed for our program to do an analysis. Nevertheless, we still consider everyone as a potential user of the project.

Benefited actors Apart from the users, we can consider that doctors and physiotherapists are going to be benefited by the creation of our project, since using it their patients are going to be able to tell them about the evolution of their inflammation with precision, helping them to know if the treatment they are applying is effective. Still, we have talked with doctors about this project and they do not seem to like the idea of its creation, but we still think that it could be useful for them.

3.2 State of the art

The goal of this section is to give information about the state of the art of the different elements that form our project. In the following lines we have listed the different areas that our project is going to explore, giving details of each of them.

3.2.1 Computing in medicine

Since computers started to became popular in the last century almost every field has been benefited by their evolution. Medicine has not been the exception, and that is the reason why doctors have been taking profit from the development of medical applications. Also, since smartphones have become popular, applications that help people to be healthy have been developed. In this section we are going to focus on the imaging techniques that have been developed in the last years to be used by doctors and we are going to show examples of applications that help people to be healthy.

Medical imaging Medical imaging is the technique, process and art of creating visual representations of the interior of a body for clinical analysis and medical intervention. Medical imaging seeks to reveal internal structures hidden by the skin and bones, as well as to diagnose and treat disease. Medical imaging also establishes a database of normal anatomy and physiology to make it possible to identify abnormalities [5].

Medical image computing (MIC) is an interdisciplinary field at the intersection of computer science, data science, electrical engineering, physics, mathematics and medicine. This field develops computational and mathematical methods for solving problems pertaining to medical images and their use for biomedical research and clinical care.

The main goal of MIC is to extract clinically relevant information or knowledge from medical images. While closely related to the field of medical imaging, MIC focuses on the computational analysis of the images, not their acquisition [6].

An example of the use of medical imaging combined with the use of a computer is computed tomography. A computed tomography (CT) scan is an imaging method that uses x-rays to create pictures of cross-sections of the body. A CT scan creates detailed pictures of the body, including the brain, chest, spine, and abdomen. The test may be used to diagnose an infection, guide a surgeon to the right area during a biopsy, identify masses and tumours, including cancer and study blood vessels [7].

Medical applications for patients Although there are many tests that a patient can not do without having appropriate tools and knowledge, there are others that anyone can do. Here we are going to present an application that can be used to check a certain constant of the body.

An example of a medical application is *Instant Heart Rate*. This application, which is freely available for all the *iPhone* users, allows the user to find out his or her heart rate in a few seconds just placing the tip of one of his or her fingers on the *iPhone* camera. It is continuously rated as the world's best mobile heart rate measurement app and is trusted by Stanford's leading cardiologists for use in clinical trials [8].

3.2.2 3D viewers

We can define a 3D viewer as a program capable of loading 3D models in order to show them to the user. With that definition, every program with a 3D window includes a 3D viewer, so, all the programs used in 3D modelling and all the 3D-based video games are examples of 3D viewers. Knowing the huge amount of modellers and video games that exist, we can say that this field has been deeply explored.

In this section we are going to describe two different viewers. The first one is closer to the one we want to create. We aim to create a 3D viewer using *Qt*, and an example of what it can be achieved using this platform is *libQGLViewer*. The second one is a tool that is provided by the authors of a video game allowing people who wants to add custom content to the game to get a preview of what they have created. The video game is a racing simulation called *rFactor2*.

libQGLViewer *libQGLViewer* is a C++ library based on *Qt* that eases the creation of OpenGL 3D viewers.

It provides some of the typical 3D viewer functionalities, such as the possibility to move the camera using the mouse, which lacks in most of the other APIs. Other features include mouse manipulated frames, interpolated key frames, object selection, stereo display, screen shot saving and much more. It can be used by *OpenGL* beginners as well as to create complex applications, being fully customisable and easy to extend [9].

Among its features we can find the ability to modify the camera, object selection, display of world axis and frame rates, animation of the scene, screen shot saving in both raster and vector formats, use of textures and more things that are much more than we are going to need for the development of the project.

rFactor2 Viewer *rFactor2* is a racing simulation. There are plenty of users who modify the game, creating new content like cars and tracks. To let people see the result of their work before adding it to the game they provide tools that are available at their website [10].

One of the tools is the *rFactor2 Viewer*, which is a program capable of loading the same scene files that the game can open, showing the scene with the same quality than it would be shown in the game, with the advantage that it lets the user navigate through the scene and it avoids going through all the menus that the video game has.

As it happens with the other example, it has much more features than we need, but if our project was about previewing 3D scenes where lighting and materials were important, this would be the kind of viewer that we would have developed.

3.2.3 Algorithms to compare 3D models

In this section we are going to present different algorithms that we have planned to use in order to fulfil the requirements of our project.

The first we have to do in order to compare two 3D models that represent the same physical object is to align them. In order to do this, we present an algorithm called Iterative Closest Point (ICP).

The ICP algorithm has become the dominant method for aligning three dimensional models based purely on the geometry, and sometimes colour, of the meshes. ICP starts with two meshes and an initial guess for their relative rigid-body transform, and iteratively refines the transform by repeatedly generating pairs of corresponding points on the meshes and minimising an error metric [11]. In our case, a rigid-body transform is not going to match all points, since the 3D models used are not going to have exactly the same shape, but it should be working doing slight modifications to the algorithm if needed.

Once we have the different meshes aligned we have to use a way to measure its differences.

Basic ideas include measuring areas or volumes, having algorithms that we consider too basic to be included in this document. A more complex idea consists on using a structure called reeb graph. A reeb graph is an interesting graph to describe topology structure that encodes the connectivity of its level sets based on the critical points of a scalar function [12]. We can think about the reeb graph as a graph that represents a “skeleton” of the mesh, thus containing information about its shape.

Another structure that it can be used to summarise the information about the geometry of the mesh in a simplified way is the medial axis. The medial axis of a solid is defined as the set of centres of maximal balls contained in the solid. It has been proposed as a tool for shape analysis, surface reconstruction, motion planning, and many other applications. It is useful because it provides a local lower-dimensional characterisation of the solid. In particular, for a solid in 3D the medial axis consists of a union of surfaces that provide information about the shape and topology of the solid. If the distance to the boundary is also stored for each medial axis point, the resulting structure is known as the medial axis transform (MAT) and the entire boundary representation can be reconstructed from it [13].

Finally, we can measure distances with the calculation of a distance field in the object. Given a set of objects, a distance field in 3D is defined at each point by the smallest distance from the point to the objects. Distance fields are frequently used in computer graphics, geometric modelling, robotics and scientific visualisation [14]. Computation of distance fields is closely related to the medial axis since some implementations use a distance field to calculate the medial axis [13].

3.2.4 3D scanning techniques

In this section we are going to describe the 3D scanning techniques that we aim to explore in order to know what are the results that have been achieved using them.

Light-based 3D scanning A structured-light 3D scanner is a 3D scanning device for measuring the three-dimensional shape of an object using projected light patterns and a camera system [15].

Universitat Politècnica de Catalunya has an associate center, Centre de Realitat Virtual, which has a light-based 3D scanner. It is a scanner from *Artec 3D Scanners*, and its resolution is up to 0.1 mm [16], which gives us more than enough precision to solve our problem. *Artec 3D Scanners* provide software capable of processing the data obtained by the scanner in order to create a 3D mesh that we can use for our purposes.

We have already done a test with this scanner, scanning both knees of a person. The result was a high-resolution mesh that fulfils all the requirements that we need in order to use it as a test case for our project.

Kinect When most of the people think about a *Kinect*, it gets associated with gaming. It is partially true, since it has become popular thanks to it being sold with the *Xbox*, but what people does not know is that there is a version for *Windows*. This version enables the development of applications that allow people to interact naturally with computers by simply gesturing and speaking. There are examples of *Kinect* for *Windows* being used in health care or in education [17].

Microsoft provides a SDK to let developers design programs to work with the *Windows* version of the *Kinect*. One of the tools included in the SDK is *Kinect Fusion*, which is a program that provides 3D object scanning and model creation using a *Kinect* for *Windows* sensor. The user can paint a scene with the *Kinect* camera and simultaneously see, and interact with, a detailed 3D model of the scene [18].

While the results might not be as good as what it could be obtained with a light-based 3D scanner, the models obtained could be used in our program, and taking into account the difference in price between the devices, it is worth exploring the possibility of using it in the project.

Multi-view stereo The goal of multi-view stereo is to reconstruct a complete 3D object model from a collection of images taken from known camera viewpoints.

The most important part in the scanning through this technique are the algorithms used to reconstruct the mesh. There are many algorithms for this purpose, some of them have been analysed and we can find algorithms that are able to reconstruct a 3D model with great accuracy [19].

A problem arises, though, since doing a reconstruction using images can be done when there are enough features to allow the algorithms to match the different images. In our problem, we are going to try to reconstruct a body part, which usually has a homogeneous colouring, so it may be difficult to get any multi-view stereo technique working, because it may be difficult to create a 3D mesh even for a human.

Autodesk have created an application called *123D Catch* that allows users to load photographs of an object in order to create a 3D model, which could be printed using a 3D printer. In their website we can find examples of captured objects using this technique, and if the photographs are taken following the guidelines they provide, the result is pretty accurate [20].

3.3 Use of previous results

In this section we are going to discuss whether we should use the solutions that we have presented in the *State of the art* section or not in each of the areas explored.

3.3.1 3D viewers

We have presented two different viewers, *libQGLViewer* and *rFactor2* viewer. It is not possible to use *rFactor2* viewer since it is not open source, but it is a good example of what it can be achieved in a viewer if advanced 3D graphics are used. It would be possible to use *libQGLViewer* since it is open source, but we have decided to create our own viewer using the *Qt* platform because we think that it is interesting to develop a full application by ourselves and also it has a key advantage, knowing everything about the code that we are using.

3.3.2 Algorithms to compare 3D models

We have presented several algorithms that can help us to compare different 3D models. Since we do not know if they are going to be useful for our purposes, all that we can say is that we are probably going to use all of them in order to check whether they are useful or not.

3.3.3 3D scanning techniques

We have presented three different 3D scanning techniques. We have planned to use the light-based 3D scanning technique that the 3D scanner from Centre de Realitat Virtual provides. Also, we have planned to use *Kinect Fusion* in order to use a *Kinect* in order to do 3D scanning. About multi-view stereo, we can say that we are going to use an algorithm which is already developed, but we do not know which one we are going to use.

4 Temporal planning

This section is about temporal planning, and it aims to describe the tasks that are going to be executed in order to do the project, giving an action plan that summarises the actions that have to be taken in order to finish the project in the desired time frame. However, we have to take into account that the planning described in this project is subject to modifications depending on the development of the project.

The project was started on July 1st, 2014 and its deadline is January 23th, 2015. Therefore we have to develop our schedule given these time constraints.

4.1 Description of tasks

In this section we are going to describe the tasks that we have planned to do in order to make our project.

4.1.1 Viability tests

This was the first task that we had to do. It was done before deciding on this project, since it was used to know if the project was even possible.

It consisted in developing an algorithm which had a 3D model as input, using it to do a voxelisation of the space to calculate the distance of each voxel to the boundary of the object represented by the model. We already had some parts of the algorithm and some 3D related code so what we had to do was to generate a code that could be used for our purposes.

The tests were successful, that is the reason that led us to choose this project.

This task did not take much time if we compare it with the following tasks. After completing it, a long break was taken, reason being holidays and having a job which lasted for nearly one and a half months.

4.1.2 Project planning

This is the task which we are currently doing. It is essentially about everything that is covered by the GEP course. It can be divided in the following four stages:

- Scope of the project.

- Temporal planning.
- Economic management and sustainability.
- State of the art.

It is the first task that has to be done after the viability tests since it defines what is going to be done in the following tasks and the way it is going to be done.

4.1.3 Initial system set up

Before starting the development of the project itself, we need to set up the tools required to work on it. It may seem a simple operation, but it takes enough time to be listed as an important task in our schedule.

To develop the project, we need the *Qt* platform to create the main program, *MeshLab* and *Blender* as tools used to modify 3D models in the case of needing to do it and a platform able to compile LaTeX code to generate documentation. We want to test the program in *Windows* and *Linux*, so we need to install both operating systems and the needed software in both of them.

After installing all these software, configuring it and checking that everything works as we expected, we can start working on the most important phase of our project.

4.1.4 Main development

This is the main task of the project. It covers all the tasks related with implementation and testing of the program that solves the problem stated in the *Scope of the project* section. It can be divided in the following stages:

- Viewer programming: The viewer is the first thing that we have to implement since it will be the program that we will use to test the algorithms that we develop in the following stages. We are going to code a *Qt* application that is going to be able to load 3D models and navigate through them. It is going to be designed so it becomes an easy task to use the algorithms found in the following stages.
- Algorithms programming: This phase can be divided in three stages: research, programming and testing. The first stage is about doing some research in order to find algorithms that can be used to compare 3D models that represent the same physical object. We know that these kind of techniques already exist, so we have to look through the different algorithms to select the most useful ones. In the second stage we are going to implement the algorithms found in the previous stage so that

they work with our program. That involves coding the algorithm and modifying the viewer so it can be used to check the results of its execution. After having the algorithms implemented, we enter the third stage, in order to test that they work for our purposes. In order to do it, we will use a 3D model of a body part and with *MeshLab* or *Blender* we will modify it, simulating the effects of inflammation in the geometry of the body part.

- *Kinect* scanning: This phase consists in doing some research in order to be able to use the *Kinect* device to get a 3D model from a physical object. The research is going to start the resources that *Microsoft*, creators of the *Kinect*, provide, but we are going to try that it becomes more extensive.

- Multi-view stereo scanning: This seems to be the most difficult stage of the project and the one that we will avoid doing in the case that the other stages take more time than we expect. It consists in doing research among the multi-view stereo related bibliography in order to be able to implement a multi-view stereo method that can be used to generate a 3D model from a body part using several pictures of it.

- Scanning tests: In this stage we are going to use the 3D models generated by the light-based scanner, the *Kinect* scanning and the multi-view stereo methods in our program. If the results are not positive, we will need to do more research.

The dependencies between the stages mentioned are easy to describe. First of all, the scanning research stages have no dependencies. We already have a light-based scanner that works and we know that we can use any 3D model from a body part, even the ones which are contained in on-line model repositories, so there is no problem in doing the scanning research related stages at the end or at the beginning. Apart from that there are only two restrictions: the viewer has to be done before testing algorithms and the scanning tests have to be done after having a working algorithm.

Also, we have to take into account that the program and any of the components that are part of it can be improved before the delivery of the project, so any of the tasks described before could be done more than once if needed.

Finally, it is important to note that for each stage we are going to write its documentation since we think that it is better to document the project while it is on development.

4.1.5 Final task

In this task we are going to check that everything works as expected and we are going to prepare the delivery of the project, assuring that the documentation is correct and preparing the final presentation.

4.2 Time table

The table 1 summarises the time spent in each of the tasks described in the previous section.

Task	Time spent (hours)
Viability tests	20
Project planning	80
Initial system set up	5
Viewer programming	60
Algorithms programming	150
Kinect scanning	50
Multi-view stereo scanning	75
Scanning tests	25
Final task	35
Total	500

Table 1: Summary of the time spent in each task.

4.3 Resources used

We can divide the resources that we are going to use in hardware resources and software resources. Here are both lists, with the tasks that each resource is used in included.

4.3.1 Hardware resources

- PC (with the following specifications: Intel Core 2 Quad Q9550 at 2.83 GHz, 4 GB RAM, nVidia GeForce GTX 650 Ti Boost): used in all tasks of the project.
- *Kinect*: Used in *Kinect* scanning and scanning tests
- Light-based scanner: Used in scanning tests.
- Camera: Used in multi-view stereo scanning and scanning tests.

4.3.2 Software resources

- *Windows* 7: used in all tasks.

- *Ubuntu* 13.10: used in all tasks.
- *Qt*: used in viewer programming.
- *Blender*: used in algorithms programming.
- *MeshLab*: used in algorithms programming.
- LaTeX: used in all tasks.

4.4 Gantt chart

Our schedule is represented by the Gantt chart shown in the figure 1. We have taken into account the break for job and holidays mentioned before.

	Nombre	Duración	Inicio	Fin	Predecesoras
1	Viability tests	15d?	01/07/2014	15/07/2014	
2	Job	45d?	16/07/2014	29/08/2014	
3	Holidays	9d?	30/08/2014	07/09/2014	
4	**⊟Project planning**	**25d?**	**08/09/2014**	**03/10/2014**	**1**
5	Scope of the project	4d?	08/09/2014	12/09/2014	
6	Temporal planning	4d?	13/09/2014	17/09/2014	5
7	Economic management and sustainability	4d?	18/09/2014	22/09/2014	6
8	First presentation	5d?	23/09/2014	28/09/2014	7
9	State of the art	4d?	29/09/2014	03/10/2014	8
10	Initial system set up	1d?	04/10/2014	04/10/2014	4
11	Viewer programming	19d?	05/10/2014	23/10/2014	10
12	Algorithms programming	42d?	24/10/2014	04/12/2014	11
13	Kinect scanning	14d?	05/12/2014	18/12/2014	
14	Multi-view stereo scanning	21d?	19/12/2014	08/01/2015	
15	Scanning tests	7d?	09/01/2015	15/01/2015	13,14
16	Final task	8d?	16/01/2015	23/01/2015	12,15

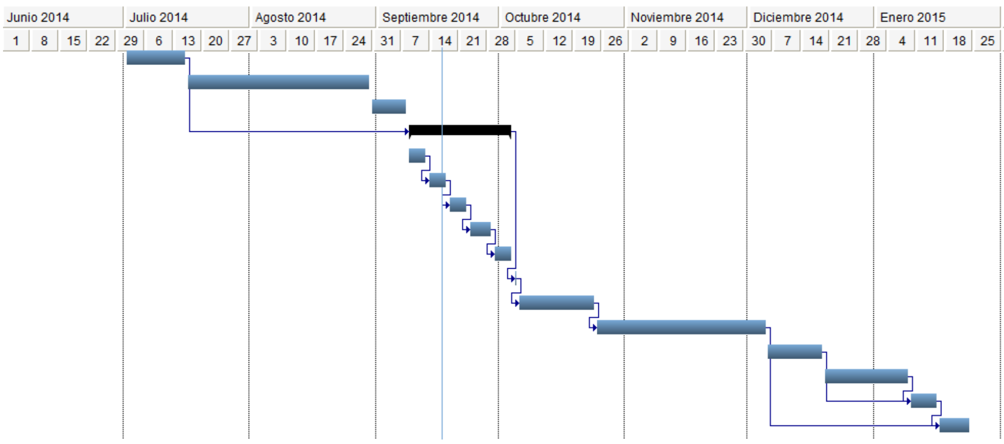

Figure 1: Gantt chart of the project.

4.5 Action plan

In this section we are going to describe how are we going to execute the plan we have created.

Our idea is to work as we have planned, executing the tasks in the order stated in the Gantt chart, but we know that like in every project there may be obstacles that may make it difficult to follow the plan. We have to take into account that the time is limited, and there are things that are a must. We are going to try to do all the tasks stated but if on any of them we run out of time we are going to try to get a basic version only. Of course, there are priorities. We must have a viewer and we must have a working algorithm, and it would be very interesting to have a scanning technique apart from the light-based 3D scanner. We know that multi-view stereo is something that would be very interesting to use but we know that it seems difficult to make it work. So what we are going to do is to make sure that a working viewer and a working algorithm is completed in the time that we expect it to be completed. In case of needing more time, we are going to take it from the time assigned to the scanning techniques, trying to have enough time for the *Kinect* research.

We are going to try to arrange meetings with the project director every time that an important stage of the project is finished.

From the time table it can be deduced that we need a total time of 500 hours. Considering that we have approximately 20 weeks to develop the project, that is 25 hours per week, which makes it possible to finish the project in the given period.

5 Budget and sustainability

This section is about the budget and the sustainability of the project. For this reason, it contains a detailed description of the costs of the project, describing both material and human costs, an analysis of how the different obstacles could affect our budget and an evaluation of the sustainability of the project. Just like in the previous section, the budget described is subject to modifications depending on the development of the project.

5.1 Budget estimation

In this section we are going to do an estimation of the budget needed in order to make our project possible. We are going to divide the budget in three sections, depending on the kind of resources that we are taking into account. They are hardware, software and human resources. At the end of the section, we are going to show the total budget obtained combining the three sections mentioned.

To calculate the amortisation we are going to take into account two factors, the first one being the useful life and the second one being the fact that our project is going to last for approximately six months.

5.1.1 Hardware resources

The table 2 contains the costs of the hardware that we are going to use in the development of the project.

Product	Price	Useful life	Amortisation
PC (including all the needed devices)	1000.00 €	4 years	125.00 €
Light-based 3D scanner	15000.00 €	4 years	1875.00 €
Kinect	200.00 €	4 years	25.00 €
Photograph camera	100.00 €	4 years	12.50 €
Total	16300.00 €		2037.50 €

Table 2: Hardware budget.

5.1.2 Software resources

The table 3 summarises the costs of the software that we are going to need to develop our project.

Product	Price	Useful life	Amortisation
Windows 7 Professional	130.00 €	3 years	21.67 €
Ubuntu 13.10	0.00 €	3 years	0.00 €
Qt	0.00 €	3 years	0.00 €
Blender	0.00 €	3 years	0.00 €
MeshLab	0.00 €	3 years	0.00 €
LaTeX	0.00 €	3 years	0.00 €
Total	130.00 €		21.67 €

Table 3: Software budget.

5.1.3 Human resources

The table 4 shows the costs of the human resources needed in the development of our project.

Role	Price per hour	Time	Cost
Project manager	50.00 €	80 h	4000.00 €
Software designer	35.00 €	160 h	5600.00 €
Software programmer	25.00 €	135 h	3375.00 €
Software tester	20.00 €	125 h	2500.00 €
Total		500 h	15475.00 €

Table 4: Human resources budget.

We have to detail how the different tasks are divided, so it is easier to understand the number of hours given to each role stated in table 4.

The project manager is going to work on the project planning only.

The software designer is going to work on everything that involves programming, that is, the viability tests, the viewer programming and the algorithms programming. Also, *Kinect* scanning and multi-view stereo scanning may involve programming. The same can be applied to the software programmer. The software tester is going to take part in the programming tasks, since it is useful to do some tests to check the progress, but his or her main tasks are the scanning tests and the final task, which involves extensive testing. Taking into account that, knowing the number of hours that have to be spent in each task and making a rough calculation using the participation of each role in each task that gives the number of hours stated in table 4.

5.1.4 Total budget

Using data shown in tables 2, 3 and 4, we can use table 5 to describe the total cost of the project.

Concept	Cost
Hardware resources	2037.50 €
Software resources	21.67 €
Human resources	15475.00 €
Total	17534.17 €

Table 5: Total budget.

5.2 Budget control

As we have already said, our budget could need a modification if the development of the project makes it impossible to follow the plan established, which is something that is likely to happen.

We could have difficulties in our project, but it is unlikely that we need any hardware resource apart from the resources that have been already listed in the budget estimations shown in the previous section. We might need more software resources, but nowadays there are plenty of free applications that can act as a substitute of almost any commercial application. So, the budget section that we have to control is the one related to the human resources.

As stated in the *Temporal planning* section, the task that may get longer is the programming of algorithms to compare 3D models. This task involves software designers, programmers and testers, so we have to take into account that the money spent on paying them could grow if the problem becomes more difficult than we have expected when doing our plan.

5.3 Sustainability

In this section we are going to evaluate the sustainability of our project in three different areas: economic area, social area and environmental area.

5.3.1 Economic sustainability

In this document we already can find an assessment of the costs of our project, taking into account both material and human resources.

The cost stated in the *Budget estimation* section of this document could be the only cost spent in the project, since we aim to create a working program, so it will not need maintenance in the case that we do not create any update. However, it is possible that further development is needed. Creating an improved version of the program with more features or adapting it to work in any new platform that is released in the future would add more cost to its development.

It would be difficult to do a similar project with a lower cost. We do not even know if the planned time is going to be enough to finish the project, so it is difficult to say that we can do it in less time, so the money spent in human resources can not be less, same happens with software, everything is free except the *Windows* operating system, and if we want to test it in *Windows*, we need to buy it. We could spend less money on hardware, but that could imply that worse results. For example, we could avoid using the light-based 3D scanner, but then we would lose a reliable tool to get scanned 3D models. What we could do is to rent the scanner every time we need it, so the price would be much lower.

We can not assure that the cost of the project would make it competitive, but we can say that it depends on how it is going to be distributed. We could make it a free application, so it would be very competitive, but then we would need a way (for example, advertisements) to get paid every time that the application is downloaded so we can recover the money invested. In the other hand, we could make it a paid application, but its price should not be too high.

We are going to spend most of the time in the tasks which are more important. As the *Temporal planning* section shows, most of the time is going to be spent in trying to find comparison algorithms, and we consider that being the most important task of our project since the quality of the algorithms is going to determine the quality of our program.

It is not the case, but is important to remark that this project could have been executed with collaboration of medical institutions, since they may be interested in developing an application like that one.

This project is going to be awarded an 8 in the economical viability area, since its price is affordable, even though it could be lower if we had not planned to use devices like the light-based 3D scanner.

5.3.2 Social sustainability

The application that we are going to develop in this project is going to be used in the countries which are developed enough so that everyone can have access to a computer, since our goal is that it can be used by everyone who can have access to a device capable of doing a 3D scan.

Everyone with an injury or suffering a disease that causes inflammation is going to be able to use the program resulting of the development of our project in order to track the evolution of his or her inflammation. The program gives the patient a way to measure inflammation, before its creation he or she had to rely on its memory to determine if the swelling level increased or decreased.

It may seem that doctors are going to be affected negatively by our project, but we think that it is not the case. We want our application to become a tool to help a person to track the evolution of inflammation, but if he or she wants to have good advice about how to heal his or her injury or how to treat any of the inflammatory disorders that he or she is suffering, he or she has to go to a doctor. However, we have talked with doctors about this project but they do not like the idea of developing it.

This project is going to be awarded a 7 in the improvement of the quality of life area, since it can help people to track injuries or diseases causing inflammation so they can have a better recovery, but the change that it is going to make for these people is not that important.

5.3.3 Environmental sustainability

The resources used in the project have been detailed in the *Budget estimation* this document and in the *Temporal planning* section.

During all the development of our project we are going to have a computer running. Also, during some phases we are going to use a light-based 3D scanner, a *Kinect* or a camera. The devices used, the energy spent and the paper used to print the documentation are going to be the only resources used.

Knowing that, we can estimate the energy spent developing the project. We can suppose that our computer consumes an average of 300 W when we are working in the project. Since we need to use the computer in all the development of the project, which is 500 hours long, the energy we need is 150 kWh, which is equivalent to 57.75 kg of CO_2. It is in fact a high amount of energy, but since we need to make extensive use of a computer there is no way to reduce it.

We could have used code from other projects. For example, there are already libraries that offer code to create a viewer or to import 3D models. This would imply spending less time in the project and a consequence is that less energy would be needed, but we preferred to develop most of the features by themselves.

This project is going to be awarded a 9 in the resources analysis area, since the resources it needs are mainly software, meaning that the only thing that is not environmental friendly is the energy that we need to execute the project, and it is not an important amount when compared to the energy which the average person uses nowadays.

6 References

[1] Wikipedia. *Inflammation — Wikipedia, The Free Encyclopedia*. Sept. 12, 2014. URL: `http://en.wikipedia.org/w/index.php?title=Inflammation&oldid=625272396` (visited on Sept. 12, 2014).

[2] Wikipedia. *Rheumatoid arthritis — Wikipedia, The Free Encyclopedia*. Sept. 10, 2014. URL: `http://en.wikipedia.org/w/index.php?title=Rheumatoid_arthritis&oldid=624932825` (visited on Sept. 12, 2014).

[3] Wikipedia. *3D scanner — Wikipedia, The Free Encyclopedia*. Sept. 28, 2014. URL: `http://en.wikipedia.org/w/index.php?title=3D_scanner&oldid=627445734` (visited on Oct. 1, 2014).

[4] Wikipedia. *Computational geometry — Wikipedia, The Free Encyclopedia*. July 7, 2014. URL: `http://en.wikipedia.org/w/index.php?title=Computational_geometry&oldid=615994077` (visited on Oct. 1, 2014).

[5] Wikipedia. *Medical imaging — Wikipedia, The Free Encyclopedia*. Sept. 27, 2014. URL: `http://en.wikipedia.org/w/index.php?title=Medical_imaging&oldid=627239096` (visited on Oct. 1, 2014).

[6] Wikipedia. *Medical image computing — Wikipedia, The Free Encyclopedia*. Sept. 2, 2014. URL: `http://en.wikipedia.org/w/index.php?title=Medical_image_computing&oldid=623794490` (visited on Oct. 1, 2014).

[7] A.D.A.M. Medical Encyclopedia. *CT scan: MedlinePlus Medical Encyclopedia*. Nov. 9, 2012. URL: `http://www.nlm.nih.gov/medlineplus/ency/article/003330.htm` (visited on Oct. 2, 2014).

[8] Apple Inc. *Instant Heart Rate - Heart Rate Monitor by Azumio for Free featuring workout training programs from Fitness Buddy on the App Store on iTunes*. URL: `https://itunes.apple.com/us/app/instant-heart-rate-heart-rate/id409625068?mt=8` (visited on Oct. 1, 2014).

[9] *libQGLViewer*. Sept. 2, 2014. URL: `http://www.libqglviewer.com/` (visited on Oct. 1, 2014).

[10] Image Space Incorporated. *rFactor 2 Developer's Corner —rFactor*. URL: `http://rfactor.net/web/rf2/devscorner/` (visited on Oct. 1, 2014).

[11] Szymon Rusinkiewicz and Marc Levoy. "Efficient variants of the ICP algorithm". In: *3-D Digital Imaging and Modeling, 2001. Proceedings. Third International Conference on*. IEEE. 2001, pp. 145–152.

[12] Rachid El Khoury, Jean-Philippe Vandeborre, and Mohamed Daoudi. "3D mesh Reeb graph computation using commute-time and diffusion distances". In: *IS&T/SPIE Electronic Imaging*. International Society for Optics and Photonics. 2012, 82900H–82900H.

[13] Mark Foskey, Ming C Lin, and Dinesh Manocha. "Efficient computation of a simplified medial axis". In: *Journal of Computing and Information Science in Engineering* 3.4 (2003), pp. 274–284.

[14] Avneesh Sud, Miguel A Otaduy, and Dinesh Manocha. "DiFi: Fast 3D distance field computation using graphics hardware". In: *Computer Graphics Forum*. Vol. 23. 3. Wiley Online Library. 2004, pp. 557–566.

[15] Wikipedia. *Structured-light 3D scanner — Wikipedia, The Free Encyclopedia*. July 24, 2014. URL: `http://en.wikipedia.org/w/index.php?title=Structured-light_3D_scanner&oldid=618287759` (visited on Oct. 2, 2014).

[16] Artec Group. *Artec 3D Scanners*. URL: `http://www.artec3d.com/` (visited on Oct. 2, 2014).

[17] Microsoft. *Kinect for Windows*. URL: `http://www.microsoft.com/en-us/kinectforwindows/` (visited on Oct. 1, 2014).

[18] Microsoft. *Kinect Fusion*. Sept. 11, 2014. URL: `http://msdn.microsoft.com/en-us/library/dn188670.aspx` (visited on Oct. 1, 2014).

[19] Steven M Seitz et al. "A comparison and evaluation of multi-view stereo reconstruction algorithms". In: *Computer vision and pattern recognition, 2006 IEEE Computer Society Conference on*. Vol. 1. IEEE. 2006, pp. 519–528.

[20] Inc. Autodesk. *Autodesk 123D Catch — 3d model from photos*. URL: `http://www.123dapp.com/catch` (visited on Oct. 3, 2014).