Three-dimensional Visualisation of Bacterial Interactions with Nano-structured Surfaces

Master of Science
by
Veselin Boshkovikj

Faculty of Life and Social Sciences
Swinburne University of Technology
2013
Abstract

In recent years there has been a growing interest in understanding the ways in which bacteria interact with nano-structured surfaces. Bacterial interaction with surfaces has become a problem in that it can be both an economic and health risk. As a result, there is a need for innovative approaches to enable researchers to understand the biological processes taking place when bacterial cells approach a substrate surface, despite the fact that these processes are not possible to directly observe. One promising avenue is to take advantages of advances in computer technology and computer graphics to enable high-quality 3D visualisation of the processes taking place.

The objective of this research was to develop a novel methodological approach for the 3D visualisation of nano-structured surfaces and their interaction with bacterial cells. The approach would seek a balance between the artistic and scientific representation of bacterial interactions. Researchers would be encouraged to be involved in the 3D visualisation process as a way of enhancing their own insight and understanding of the processes taking place. The desired outcome would be the development of a simplified method for producing high-quality 3D movies and images of the research data to support both scientific publications and public presentations.

The foundation of the visualisation process is the software package Autodesk Maya, which was chosen as a suitable candidate to be used as a tool for 3D graphical representation and simulation of research data. Maya’s powerful 3D modelling and programming capabilities where used to develop 3D models of *Staphylococcus aureus* and *Pseudomonas aeruginosa* cell shapes, titanium substrate surfaces, and the interaction that takes place between them. The dynamic menu sets available in Maya were used for the development of dynamic interactions between cells and surfaces models. The nano-structured surfaces were scanned using atomic force microscopy (AFM), and the data obtained from these scan were used in the production of graphic images and animations.
A tool was also developed with the Python programing language, allowing the direct importation of AFM data, and 3D reconstruction of surfaces from that data, into Maya, making this software more independent and friendly from a scientific perspective. The resulting 3D movies, depicting the interactions between the two modelled bacteria, *S. aureus* and *P. aeruginosa*, with different types of titanium surfaces that are used in the production of biomedical devices, gave an increased insight and understanding of the research data, both to the researcher and a wider public audience. The 3D animations resulting from this study have already been included on the nature.com website, and in the production of a story for the ABC television program ‘Catalyst’. This research has opened new avenues for developing further tools that will allow the visualisation of complex 3D dynamic interactions taking place between bacterial cells and nano-structured surfaces.
Acknowledgment

First and foremost I would like to acknowledge and thank my coordinating supervisor Professor Elena Ivanova for her guidance and support. Her experience, enthusiasm and motivation made this project possible to achieve. I am extremely grateful for her patience and understanding, as she would always spear a moment of her busy schedule to provide critical assessment and feedback of my work.

Similarly, I would also like to acknowledge and thank Associate Processor Christopher Fluke and Professor Russell Crawford for co-supervising the project. Their experience and knowledge in their respected fields has made equal outstanding contribution towards the completion of the project.

To Dr. Hayden Webb, Dr. Khanh Truong, Dr. Jafar Hasan, Vy Pham, Ha Nguyen and Peter Nguyen, thank you for you efforts for performing the experiments and providing all the necessary data and much needed assistance. Also, thank you for the friendship and wonderful time spent at Swinburne. No doubt, we make a pretty good team.

I would also like to acknowledge the whole Swinburne staff, thank you for your outstanding work. To all my friends and colleagues, thank you for the great time spent in the office during the course. Special thanks to the ladies from the Faculty of Life and Social Sciences research administration office, for their exceptional work and professional assistance and for making the life of a student more entertaining and enjoyable.

And last but not least, thank you to my whole family, especially to my father Dragan and my sister Ana Boskovic, to my uncle Momcil and my aunt Ilunka Ilkov, my cousins Igor and Marjan Ilkov and Anna Ilkov. Without you guys none of this would have happened. Your support in good and bad moments made the life easier to comprehend.

Autodesk, Maya are registered trademarks or trademarks of Autodesk, Inc., and/or its subsidiaries and/or affiliates in the USA and/or other countries.
Declaration

I, Veselin Boshkovikj, declare that this thesis is my original work and contains no material that has been accepted for the award of Master of Science, or any other or any other degree or diploma, except where due the reference is made.

I declare that to the best of my knowledge this thesis contains no material previously published or written by any other person except where the reference is made. Wherever contributions of others were involved every effort has been made to acknowledge the contributions of the respective workers or authors.

Signature _______________________________________________
List of Publications

Peer-review articles


Submitted articles


Table of Contents

Abstract ........................................................................................................................................... 2
Acknowledgment ............................................................................................................................. 4
Declaration ........................................................................................................................................ 5
List of Publications .......................................................................................................................... 6
List of Figures ..................................................................................................................................... 10
List of Tables ..................................................................................................................................... 15
Chapter 1. Introduction ..................................................................................................................... 17
  1.1. Overview ................................................................................................................................... 18
  1.2. Aims and scope .......................................................................................................................... 20
Chapter 2. Literature Review ......................................................................................................... 23
  2.1 Scientific visualisation ................................................................................................................ 24
  2.2. Three-dimensional visualisation in nanotechnology ................................................................. 24
    2.2.1. Molecular and atomic visualisation ...................................................................................... 25
    2.2.2. Cellular visualisation ........................................................................................................... 27
    2.2.3. Surface visualisation ........................................................................................................... 28
  2.3. Problems .................................................................................................................................. 30
  2.4. Possible solutions ...................................................................................................................... 31
  2.5. 3D applications for visualisation of bacterial cell-surface interactions................................. 32
  2.6. Autodesk Maya ........................................................................................................................ 35
    2.6.1. 3D geometry in Maya .......................................................................................................... 35
    2.6.2. Animations and simulations in Maya ................................................................................... 36
    2.6.3. Maya in scientific research ................................................................................................ 37
Chapter 3. Materials and Methods ................................................................................................ 41
  3.1. Visualisation of Scientific Data: ‘Scenario-based visualisation’ .............................................. 42
  3.2. Scenario-based visualisation application for the 3D visualisation of cell-surface interactions ................................................................. 50
    3.2.1. Semi-automated step .......................................................................................................... 51
    3.2.2. Creative step ....................................................................................................................... 60
Chapter 4. Visualisation of the Bacterial Interaction with Smooth Titanium Substrata

4.1. Overview
4.2. Data collection
4.3. Production workflow
4.3.1. S. aureus interaction with titanium surfaces
4.3.2. P. aeruginosa interaction with titanium surfaces
4.3.3. Image and movie production
4.4. Summary

Chapter 5. Visualisation of P. aeruginosa Cell Interaction with Cicada Wing Surface

5.1. Overview
5.2. Data collection
5.3. Production workflow
5.4. Summary

Chapter 6. Visualisation of the Interaction of S. aureus with the Lotus-like Titanium Surface

6.1. Overview
6.2. Data collection
6.3. Production workflow
6.3.1. Bacterial cells simulation
6.3.2. EPS simulation
6.3.3. Movie post-production
6.4. Summary

Chapter 7. Novel Surface Visualisation Solutions

7.1. Overview
7.2 Data collection
7.3. Reconstruction of titanium surface via cross section profiling
7.4. Transition of titanium surface roughness: received titanium into ECAP-modified
7.5. 3D construction of nano-structured surfaces using a single SEM image
7.5.1 Data collection ..............................................................................................................115
7.5.2 Three-dimensional visualisation of SEM images .....................................................116
  7.5.2.1 Construction of displacement maps ....................................................................116
  7.5.2.2 Conversion of displacement maps into 3D polygonal geometry ....................116
7.5.3 Visualisation of titanium surfaces ...........................................................................119
7.5. Summary .......................................................................................................................123
  7.5.1. Advantages and limitations of displacement map technique .........................124

Chapter 8. Discussion .........................................................................................................129
  8.1. Overview .......................................................................................................................130
  8.2. Maya simulation versus scientific simulation .........................................................134
  8.3. Limitations ..................................................................................................................135
  8.4. Impact of 3D visualisation on public audience .......................................................138

9. Conclusions and future directions ...............................................................................141
  9.1. Summary ......................................................................................................................142
  9.3. Conclusion ..................................................................................................................144

Bibliography ......................................................................................................................145

Appendices .........................................................................................................................160
# List of Figures

**Figure 2.1.** AFM scan of a surface. When the scan is performed, the AFM tip physically interacts with the surface topography producing numerical value of the surface height ................................................................. 29

**Figure 3.1.** Stages for visualisation of data for subsequent analysis and inspection.... 43

**Figure 3.2.** Stages for visualisation of data and final production for publication....... 44

**Figure 3.3.** Scenario-based visualisation. The stages of the Scenario-based visualisation workflow were described in the following section................................. 46

**Figure 3.4.** Three-dimensional models of the surface topography. The converted AFM data visualised in Avizo (a) and in Maya with the newly developed tool (b)........... 52

**Figure 3.5.** User interface of the tool. The tool developed in Python allows 3D surface reconstruction from a .csv file, creation of basic bacterial models and applying colour maps and colour map values............................................................... 54

**Figure 3.6.** Import data sub-menu........................................................................ 55

**Figure 3.7.** Create 3D objects sub-menu.............................................................. 56

**Figure 3.8.** 3D reconstruction of surface topography in Maya. The AFM data files, in .csv format, were imported and a set of equally spaced curves is created based on the surface height values form the file (a). Each curve consists of control vertices (CVs) to which values for each dimension (X, Y and Z) can be applied (b). A polygonal surface is created from the curves (c). .................................................................................. 57

**Figure 3.9.** Surface manipulation sub-menu. ......................................................... 58

**Figure 3.10.** Surface height values. The height value of a single vertex is displayed in the 3D view port (a). Five colour map values displayed in a separate window based on the minimum and maximum values. ................................................. 59

**Figure 3.11.** Three-dimensional modelling of bacterial cell shapes. The bacteria S. aureus (round shaped) and P. aeruginosa (rod shaped) were made with Maya’s modelling tools. A cube is used as a base mesh for developing the models (a). The smooth and extrude tool were used as well for further modelling of the bacteria cells shape (b and c). A 2D bump texture was assigned in order to create more “realistic” bacteria (d) and at the same time to preserve a lower quality mesh. ....................... 62
Figure 3.12. Colour mapping on the 3D models of the surfaces. The ramp 2D texture with layers of colours representing terrain map (a) applied to a 3D surface model (b).

Figure 3.13. Dynamic objects. The S. aureus cells were made as rigid dynamic bodies (a), meaning that the cell shape will not be deformed when interact with the surface. The membrane flexibility and fluctuation of P. aeruginosa cells was achieved by assigning soft dynamic body to the 3D polygonal mesh (b). When a soft body is created, a particle system (b, right) is assigned to every vertex of the polygonal mesh (b, left). The changes made to the particle positions by a force field, or another object, affects the corresponding vertex of the 3D model, resulting in a flexible object. This flexibility can be control by adjusting (“painting”) the weights of the particles (c). Where the black value (0) is assigned, the object will be mostly flexible, while the white value (1) makes the object totally rigid. For the P. aeruginosa cells the value of 0.75 was assigned in order to represent the bacteria’s membrane fluctuations, but this number is arbitrary.

Figure 3.14. Different types of lighting options in Maya. A poor choice of light will affect the colour map texture. Ambient light was found to be the most appropriate choice.

Figure 3.15. Parent and unparent camera/light setup. If the ambient light doesn't follow the camera in the 3D space it will affect the colour intensity, resulting in the misinterpretation of the variance between the colour layers (a). If the light is parented to the camera, the light will follow the camera’s animation path, resulting in the constant intensity of the colour map (b).

Figure 4.1. Three-dimensional representations of the surface topographies of three sub-nanometrically smooth Ti samples, as imaged by atomic force microscopy on 1 µm × 1 µm (left) and 10 µm × 10 µm (right) scanning areas.

Figure 4.2. Bake simulation application. A key (red line) is stored for each frame containing the dynamic simulation values (a). By setting apart the last few keys the motion of the 3D object will become slower (b).

Figure 4.3. S. aureus and P. aeruginosa cells interacting with three different sub-nanometrically smooth Ti surfaces. Images are screenshots extracted from the Videos presented in this chapter. In these animations, S. aureus and P. aeruginosa cells first fall onto a 10 µm × 10 µm surface, and upon contact can be seen to roll/slide slightly for
several seconds before settling on the surface. The camera then rotates 360° and zooms in to show a higher resolution, 1 μm × 1 μm section of the same surface. The surfaces were produced based on real AFM data. The cells are also briefly made transparent and the surface features are extruded to better demonstrate the surface topography.

**Figure 5.1.** Time frame of *P. aeruginosa* sinking on cicada wing nanopillars. Adopted from (Ivanova et al., 2012).

**Figure 5.2.** Three-dimensional surface models of cicada wing nano-pillars. An AFM scan of the wing’s nano-pillars did not produce accurate data of the pillars structure (a). A “perfect model” was developed and visualised (b) based on the calculation of the dimensions of the pillars.

**Figure 5.3.** Three-dimensional representation of the modeled interactions between a rod-shaped cell and the wing surface. As the cell comes into contact and adsorbs onto the nanopillars (a), the outer layer begins to rupture in the regions between the pillars (b) and collapses onto the surface (c).

**Figure 5.4.** Trax editor. Two nCache files representing the two simulations of the cell-surface interaction were imported into the Trax editor, where the time of the simulations was adjusted.

**Figure 6.1.** 3D surface representation of Lotus-like titanium.

**Figure 6.2.** *S. aureus* cell attachment on Lotus-like Ti. The animation of the *S. aureus* interaction with the Lotus-like surface was developed with the nDynamic tools and objects representing the first minute (a) and the thirtieth minute (b) of the bacterial interaction.

**Figure 6.3.** Three-dimensional EPS visualisation. The exo-polymeric substances (EPS) were created using an nParticle system (a), which was converted to a polygonal mesh (b).

**Figure 7.1.** Surface sections being sequentially displayed. During the animation process, each section (4 nm per section) appears every 5 frames until the whole surface is presented.

**Figure 7.2.** Graphical representation of the connections of the attributes. Two 2D texture nodes (ramp) were created (left), one for the colour map (five layers of colours representing the terrain map) and the other for the visibility of the surface sections (two colour layers, black and white). The out colour attribute of the ramp representing the colour map was connected to the lambert colour attribute (right). The out colour of the
second ramp, representing the “visibility” of the surface sections, was connected to the transparency attribute of the lambert material.

**Figure 7.3.** Screenshots of the Video 7.1. Cross-sectional profiling of as received (AR) and ECAP titanium in grade 2 (G2) and grade 4 (G4) (a). Whole 3D models after the profiling (b).

**Figure 7.4.** Application of displacement maps. Each polygonal object (e.g. a plane) consists of three-dimensional points (vertices) that are connected with lines (edges) (a). Each pixel is assigned a relative height value based on their grey-scale values; black is assigned the lowest value of 0 and white the highest value of 1. (b). During the displacement map geometry conversion each vertex is translated in 3D space according to its assigned height value, thus transforming the original, planar 2D image into a new 3D geometry (c).

**Figure 7.5.** Three-dimensional representation of a 150 nm-thick thin titanium film surface. A section of an original scanning electron micrograph, 512 × 512 pixels in size was selected for conversion into a 3D object (a). The image was assigned to a 2D polygonal plane as a displacement map, and a terrain colour map was applied (b) before depth translation according to the pixel height values. Note that the colour map values presented in (b) has been calibrated through the use of AFM roughness data. The 3D representation of the surface (c).

**Figure 7.6.** Example of image filtering. A resulting 3D model without filter (left), and with Gaussian filter (right).

**Figure 7.7.** Reconstruction of surfaces using input micrographs with various magnifications. Three-dimensional surfaces have been generated based on electron micrographs of 150 nm-thick titanium thin films at 30000 × (a), 70000 × (b), 90000 × (c) and 150000 × (d) magnifications. Colour scales have been calibrated via AFM roughness data. Three-dimensional surfaces generated based on AFM scans of corresponding surface areas are presented in the right column. The surface features in the AFM scans appear to be more regular than in SEM 3D displacement maps, due to the effects of tip convolution on AFM data.

**Figure 7.8.** Visualisation of material surfaces with different compositions. Three-dimensional surfaces of dragonfly (Hemianax papuensis) wings (a), polystyrene Petri plates (b) and silicon wafers (c) can be reconstructed from electron micrographs.
Magnification in all micrographs is 70000×, and colour scales have been calibrated using AFM roughness data. .................................................................122

**Figure 7.9** The accuracy limitation of an SEM scans: the polygon example. ...............125

**Figure 7.10.** SEM noise reduction for surface reconstruction. A high level of noise in an electron micrograph can adversely affect the final 3D geometry, resulting in a highly rough appearance (Fig. 7.10a). In order to minimize the effect of noise, a convolution algorithm can be applied. The surface presented in (Fig. 7.10b) was processed using a 3 × 3 convolution matrix to ‘smooth out’ the effect of noise. Care must be taken however, to ensure that the image is not excessively smoothed, as this will result in the loss of topographical detail. This is demonstrated in surface (Fig. 7.10c), where the same convolution matrix was applied 5 times in succession. .................................................................126

**Figure 8.1.** Different types of shading/texturing results on the *P. aeruginosa* cell models and the cicada wing surface. Visualised objects in Maya with applied default grey texture (a). Scientific representation of the same objects with applied colour map on the surface and a red texture (red is often associated with dying cells) (b). The artistic representation with applied SEM look a like texture. .................................................................138
List of Tables

Table 2.1. 3D software packages used in entertainment industry ................................34
Table 3.1. Default dimensions and resolutions of the bacterial 3D models if made with
the Bacteria set of functions. .................................................................................. 61
Table 8.1. On-line resources for 3D animation tutorials and examples .......................133
## Acronyms

<table>
<thead>
<tr>
<th>Acronym</th>
<th>Full</th>
</tr>
</thead>
<tbody>
<tr>
<td>2D</td>
<td>Two-dimensional</td>
</tr>
<tr>
<td>3D</td>
<td>Three-dimensional</td>
</tr>
<tr>
<td>AFM</td>
<td>Atomic force microscope</td>
</tr>
<tr>
<td>API</td>
<td>Application programming interface</td>
</tr>
<tr>
<td>.csv</td>
<td>Comma-separated values</td>
</tr>
<tr>
<td>CVs</td>
<td>Control vertices</td>
</tr>
<tr>
<td>ECAP</td>
<td>Equal channel angular pressing</td>
</tr>
<tr>
<td>GUI</td>
<td>Graphical user interface</td>
</tr>
<tr>
<td>MEL</td>
<td>Maya embedded language</td>
</tr>
<tr>
<td>SEM</td>
<td>Scanning electron microscope</td>
</tr>
</tbody>
</table>
Chapter 1. Introduction
1.1. Overview

"A picture is worth a thousand words" is a well-known saying, and highlights the advantages and benefits associated with the graphical presentation of data. Substantial advances in the development of graphical designs has been made since the invention of computers, providing an avenue for collecting and storing vast amount of data, processing and analysing the data, and visualising the data with different computer generated graphical formats. Computer graphics can be defined as the manipulation of data with a computer in order to create an image. While a large variety of specialized software packages and hardware is available, there is no single software solution that solves all computer graphics problems. Different types of computer graphics can be used for the visualisation process: two-dimensional (2D vector and pixel graphics), three-dimensional (3D) and 2D/3D animations. The 3D computer graphics and animations are becoming commonly used, enabling the creation of 3D models of data and animation or the simulation of these models.

When it comes to scientific data, a well-chosen visualisation of the data is able to convert a complex table of numbers into a form that better communicates the information. Indeed, by transforming numerical values into geometrical shapes, followed by adding additional information to the geometry such as textures (i.e. colour maps), further insight into the information context of the data set may be obtained. The process of visualisation brings about the ability to innovatively perceive the research data, resulting in a greater understanding of new knowledge that would otherwise not be possible to achieve. Visualisation graphs are used to illustrate papers, reports and theses, as well as providing the basis for most public presentations of the research.

As previously mentioned, advances in computer graphics have revolutionised the exploration and presentation of research data in scientific areas where objects under investigation are impossible to be seen without the assistance of technology. A good example is the field of nanotechnology, as there has been a rapid development of tools that allow exploration of the world at micro− and nano−scales (Lawton et al., 2000). Instruments such as Scanning electron microscope (SEM) (McMullan, 1995) and Atomic force microscope (AFM) (Binnig et al., 1986) provide ability for data gathering and analytical inspection of the natural process that occur on these scales. The computer
graphics allow recreation of geometrical shapes from that data, thus providing greater insight into the data and visual perception that otherwise would be impossible to achieve.

Developments in the methods employed to exert control over nano- and subnano-scale assembly (Ostrikov et al., 2011, Kumar et al., 2012) have allowed refined analytical techniques to be used for the exploration of structures at these length scales, and in conjunction with advanced computing facilities, afforded us the ability to describe and predict structure–function/property relationships (Um et al., 2013). Yet, as we delve deeper into this relatively uncharted territory, we are faced with an increasing degree of complexity in interpreting the processes that occur in dynamic systems over these scales. A prime example of such a process is the interaction of bacterial cells with their natural surroundings.

It is well known that bacteria are the main cause of life-threatening infections (Hahn and Sohnle, 2013). Materials such as titanium and titanium alloys (Liu et al., 2004), which are used widely in biomedical applications (Valiev et al., 2008, Norowski Jr and Bumgardner, 2009, Krishna Alla et al., 2011, Crawford et al., 2012, Kim and Park, 2013), are substrates that are very attractive for bacterial attachment; bacteria can attach, colonise and multiply on biomaterial surfaces in significant numbers, increasing from an initial few bacteria within minutes, to tens of millions of bacteria within a few days.

There has been much recent interest in modifying these substrate materials at the nano-scale (Faghihi et al., 2006, Valiev et al., 2008, Lamolle et al., 2009, Okawa and Watanabe, 2009, Bjursten et al., 2010, Subramani, 2010, Dreaden and El-Sayed, 2012), with the goal of making them resistant to bacterial attachment. A number of naturally occurring surfaces have been identified that exhibit a natural resistance to bacterial attachment. As a result, research has focused on mimicking and recreating the nano-structured surface architecture that is thought to be responsible for the bacterial resistance when designing surfaces for use in industrial and biomedical applications where bacterial resistance is desirable (Fadeeva et al., 2011, Webb et al., 2011a). Understanding the exact processes taking place in bacterial interactions with nano-structured surfaces is a challenging task, mainly because of the small scale of the objects under investigation.
1.2. Aims and scope

Since all of the processes comprising the bacterial attachment mechanism, cannot be viewed continuously in real-time, as there is no available technique to record the interaction. Therefore, we have to rely on the knowledge that is available and interpret the “snapshot” data that is obtained using surface characterization and imaging techniques. 3D visualisations and computer-generated animations that make certain equated assumptions regarding the attachment process may, in part, fill this gap in knowledge and provide researchers with the ability to visually perceive what previously was not possible, facilitating both the communication and comprehension of complex scientific findings. Considering the importance of understand such processes with regards to the long term health of biomedical implant recipients, a new approach is needed to improve understanding of bacterial attachment process. The aim of this research was to develop and evaluate a practical methodological approach for the 3D visualisation of micro- and nano-scopic objects and their interaction with different types of substrata surface interaction.

The first objective was to develop solutions to effectively display surface topographical data. This required raw data pertaining to the surface topography, which was obtained using specialised instruments, which was later transformed into 3D geometric images and animations.

The second objective was to develop plausible animated scenarios of surface-cell interactions based on scientific theories and assumptions. Here, it was necessary to represent the dynamic interactions between bacterial cells (basic 3D models) and the 3D models of the surfaces.

In the upcoming chapters, the state-of-the-art techniques for 3D scientific visualisation and 3D visualisation in nanotechnology will be reviewed, followed by an explanation of the methodological approach and the tools involved in the 3D visualisation process. This study builds on pre-developed applications, in particular the Autodesk Maya software (Chapter 2). It was important to adapt the chosen software to be capable to perform all the necessary tasks (e.g. visualise data into 3D graphical
format) (Chapter 3). Following this, the development of resulting 3D movies will be presented, where the methodology that was applied in order to depict different situations of cells attachment on different types of non-modified and modified titanium and natural occurring surface (e.g. insect wing.). This study extends the existing ability to present cell-surface interactions in 2 dimensions, providing tools and options for researchers to visualise, inspect and present their data into 3D graphical format.

The DVD contains the 3D video files described in chapters 4-7. The movies can be accessed from the attached HTML file (Video legend). The new python tool is also included in the DVD along with a demonstration of the tools applicability.
Chapter 2. Literature Review
2.1 Scientific visualisation

Scientific visualisation encompasses the exploration and graphical representation of research data as a means of gaining insight and discovering new knowledge. It enables understanding of the complex relationships that may exit between individual components within a system from which data is acquired (Mohler, 1999). Whether employed to assist in interpretation of experimental efforts or to graphically represent theoretical works, the importance of scientific visualisation is undeniable. Indeed, by displaying multi-dimensional data in an easy-to-understand form, scientific visualisation provides a means for in-depth interrogation of 3D and higher dimensional data and data sets at a new multidisciplinary level (Mura et al., 2010), a previously impossible undertaking. On the other hand, inadequate or inappropriate visualisation may result in misguided interpretation of the data, limited understanding of the true meaning of data, and reduced communicative efficacy of the graphical representation.

Scientific visualisation continues to grow in interest to the scientific community (Lipşa et al., 2012). Even though its development has already had a huge impact on exploring and analysing scientific data, it brings new challenges and problems (i.e. the ability to visualise increasing quantity and complexity of data) (Johnson, 2004). Given the aptitude of scientific visualisation to advance our ability to investigate and interpret scientific data, new methods and applications are being developed to keep pace with the rapidly advancing data acquisition tools (Gehlenborg et al., 2010).

In the remaining of this chapter the advancement and applicability of scientific 3D visualisation in nanotechnology is explored. An overview of the achievements and progress in the past two decades will be presented, depicting influence of 3D computer graphics over the visualisation of atomic, molecular, cellular and surface experimental data.

2.2 Three-dimensional visualisation in nanotechnology

Nanotechnology relies on the development of tools for imaging, measuring and manipulating living organisms and abiotic matter at increasingly smaller scales. The
extremely low size of objects under investigation renders them too small to be perceived naked eye. Yet, adequate interpretation and comprehensive understanding of the characteristics and the processes at nano- and molecular scales can further develop our understanding regarding the processes that occur in nature.

While data capturing techniques can provide researchers with data describing nano-scale properties and processes, scientific 3D visualisation is an essential tool for interpretation of these data and comparison of the collected data against the predicted model (Long et al., 2012). It is not surprising that 3D visualisation of data is becoming an increasingly attractive and popular tool employed by scientists working within the nanotechnology field. The growing interest in 3D visualisation is further supported by the increasing availability to the researchers of powerful instruments for collecting data (Binnig et al., 1986, Danilatos, 1991) and highly sophisticated software and hardware for visualizing that data. With the advances in affordable computer technology and nanotechnology, researchers now have the ability to routinely visualize the basic building blocks of organisms and non-living matter.

2.2.1. Molecular and atomic visualisation

The early 1970s saw the dawn of a new era in scientific visualisation with increasing accessibility of computer technology to research institutions. This led to the development of the first computer-generated animations of molecular models (Graver et al., 1972). Much like physical models, these animations were capable of providing an attractive visual representation of molecules and structures, together with their global and local properties (Eufri and Sironi, 1989). Indeed, the ‘ball and stick’ model where balls and sticks are used to represent atoms and bonds within a molecular skeleton, and colouring is employed to distinguish between the different elements, remains the most common 3D visual representation of the molecular and atomic models in physical and computer-generated approaches alike (Huijsmans et al., 1987). The ability to interactively inspect molecular models was an important step in the ability to examine data, and as a result, this functionality has become an integral part of almost every visualisation application that has been developed (Bajaj et al., 2004). This interactivity
allowed users to manipulate the 3D models in real time using some of the standard interactive commands such as translation, rotation and zooming of 3D objects.

The fundamentally important distinction between computer-generated visualisations and a physical model was the potential ability of these animations to depict molecular processes and chemical reactions. The analytical and predictive potential of molecular and atomic visualisation attracted significant interest from both academia and industry, with a number of data visualisation tools emerging over the years (Ihlenfeldt, 1997).

The rapidly advancing computer technology that occurred during the 1980s and 1990s resulted in an increase in the development of new software applications that were specifically designed to enable the examination and presentation of molecular and atomic data (Weaver et al., 1994). Indeed, many of the applications developed during that period not only supported the presentation of 3D models of molecular and atomic data, but also gave researchers the opportunity to analyse and display the molecular dynamics taking place (MD) (Laaksonen, 1992, Oldfield, 1992, Ewart et al., 1994, Callahan et al., 1996, Greenberg, 1996, Frank et al., 1999).

By the beginning of the 21st century, many more powerful and sophisticated applications emerged. These, together with the ever-increasing volume of information and consequent difficulties in data processing, demanded further improvements in the ability to perform high-speed calculations and analyses (Pavlopoulos et al., 2008, Stone et al., 2010, Stone et al., 2011). As such, more advanced applications, capable of supporting high quality imaging, 3D animations and inspection of atomic and molecular models, were developed (Huitema and Van Liere, 2000, Adler et al., 2002, Wako et al., 2004, Hodis et al., 2007).

The Internet provided researchers with the option to create web-based applications to allow bimolecular visualisation, where online servers enabled the user to import data and create high quality pictures and animations (Autin and Tuffery, 2007) or calculate and analyse molecular motion (Hollup et al., 2005, Porollo and Meller, 2010). Furthermore, Internet portals have been created for the visualisation and analysis of molecular dynamic simulations (Frank et al., 2003).
The advantage of such on-line services is their ability to function without the need for connection to specific software applications, yet they still provide the ability to develop high-quality images and animations (Autin and Tuffery, 2007).

2.2.2. Cellular visualisation

The need to visualize objects at the cellular level brought new challenges to 3D scientific visualisation. In addition to providing high resolution images of cellular structures, cellular visualisation aims to provide researchers with the means to visually perceive the metabolic activity and behaviour of the cell in response to the application of specific stimuli (Lidke and Lidke, 2012). The task of recreating complex processes in cells as a graphical representation, however, is difficult. Notable advancements in sample preservation, staining and labelling, and data acquisition have brought researchers closer to their goal of being able to visualize how cellular organization is related to its function. Indeed, there has been significant progress made in the context of cellular imaging and visualisation (Leis et al., 2009, Hannig et al., 2010), particularly with regard to the ways by which proteins interact within a cell (Li et al., 2011).

Different methods have been developed for the 3D visualisation of cellular structure (Teo et al., 2010, Medeiros et al., 2012) and cellular simulations (Hoehme and Drasdo, 2010). Another study presented a method for visualisation of cell shapes (in this case red blood cell) from a SEM image (Wang, 2013), based on mapping the intensity level to get the height filed of each pixel and boundary contour tracing to separate cells.

Approaches that rely on two or more distinct applications in order to develop different parts of the final 3D visualisation have also been investigated (Czech et al., 2009). These can involve a combination of scientific software and software from entertainment industry, (e.g. composing and adding special effects to the pre-visualised data) with the latter often used for post-production. However, there are also examples for direct visualisation of data in these applications (Section 2.6.3). In the study present by Czech, the visualisation process of cells has rather different approach. A 3D model
of cell was developed in a software package manly used in entertainment industry (Blender). The geometry was exported for the simulation in external software (MCell to preform simulation and DReAMM to visualised or animate the simulation results).

2.2.3. Surface visualisation

While the need for high-resolution imaging is obvious when investigating micro- and nano-scale objects such as bacterial cells, it may not be immediately apparent in the case of macroscopic objects. For instance, when researchers observe the surface of common biomaterials such as titanium and titanium alloys with the naked eye they can make a judgment regarding the apparent smoothness or roughness of the surface. It has been shown that macroscopically rough surfaces are likely to be more susceptible to bacterial colonization, as surface grooves and undulations can afford bacterial cells with a protective habitat against flow detachment (Truong et al., 2010). Despite these observations, there are numerous reports that have detailed the preferential bacterial colonization of macroscopically smooth surfaces (Estrin et al., 2009, Busscher et al., 2010).

Indeed, recent research (Crawford et al., 2012) has demonstrated that bacterial attachment depends strongly on the surface topographical features at the micro–, nano– and subnano–metric scale, suggesting the possibility of an alternative (non–chemical) way of limiting the extent of bacterial adhesion and proliferation that can take place on a surface. This breakthrough in cell–surface dynamics has been afforded to some extent by the advancement in microscopic technology (SEM and AFM), which has afforded researchers the ability to inspect surface topographies and to perhaps use this data in efforts to visualize the nature of cell–surface interactions at these length scales.
Figure 2.1. AFM scan of a surface. When the scan is performed, the AFM tip physically interacts with the surface topography producing numerical value of the surface height

AFM is perhaps the most advanced of these techniques, as it can be employed to generate topographical maps of a surface with sensitive and accurate measurement in the depth dimension (Schift et al., 2009). AFM surface scans are of a physical nature, in that the AFM tip physically interacts with a sample surface (Fig. 2.1).

Since the data obtained from these microscopes are in numerical form the development of applications and methods that allow the 3D visual representation of microscopic data is paramount. Avizo Standard is one such application; it allows the inspection of surface topography and provides an analysis of the characteristics of a surface and bacterial adhesion (Webb et al., 2011b). In order to perform an in–depth characterization of the features of substrate surfaces ranging from novel and modified materials to biological substances, a number of different methods have emerged for the purposes of surface visualisation (Marschall et al., 2000, Ostadi et al., 2011).

One approach developed by Samak (2007), facilitates the 3D visualisation of the surface microstructure from the 2D images obtained using a stereophotogrammetry
technique. Briefly, the 3D reconstruction of the surface is achieved by matching the respective points in two separate images of a surface based on a stereo correlation algorithm, followed by texturing process. Another method was developed to recreate sharp 3D images of the micro-scale areas of the surface of bloodstains, described by Hortolà (2010). A micro-scale area was selected on an uneven surface of a bloodstain, where the SEM was used to capture the images of the surface in order to determine its morphology. Vertical stacks of images were obtained with INCAEnergy system, while Helicon Focus (version 4.60.3) software was used to generate a single image and subsequent animations from the image stacks.

2.3. Problems

New methods for advanced 3D visualisation provide scientists with an opportunity to look deeper into the nano–world. Among the many biological processes that occur at this scale accurate 3D visualisation of different types of bacterial cells and their interaction with the immediate environment provides a challenge. Bacterial colonization is the cause of many infections, especially with regard to infection associated with biomedical implants (Kim and Park, 2013, Krishna Alla et al., 2011, Valiev et al., 2008). As a result, obtaining an increased understanding of cell–surface dynamics is paramount. Indeed, the ability to visualise and interpret specific interactions between bacterial cells and implantable surfaces is fundamental in order to be able to control such interactions, e.g. by tailoring surface properties to target specific aspects of bacterial attachment or biofilm formation (group of microorganisms such as cells stick to each other on a surface) (Rogers et al., 2008).

Very often, the extremely small size of objects under investigation undermines the quality of the data that can be collected (e.g. S. aureus has a body diameter of 1000 nm). This in turn affects the ability to develop an accurate and representative 3D model of the actual dynamic interactions, as the accuracy of the visualisation strongly depends on the gathered data sets. Further complexity stems from the nature of the attachment process itself, where numerous factors (chemical and physical structure) influence the ability of bacteria to attach and remain on the surface (Truong et al., 2010). As such, these factors and their respective effects need to be adequately accounted for in the
model. One example is the need for the visualisation of the behaviour of the bacterial membrane before, during, and after it comes into contact with a surface.

2.4. Possible solutions

In science, the demand for new visualisation applications is often driven by the need for more efficient approaches, algorithms or to use strategies to deal with of increasingly large research data sets (Dziubecki et al., 2012). While some of these applications need to be developed, there are also opportunities to use existing applications, such as those currently used in entertainment industry. These can be adopted for scientific visualisation, provided they can be adapted to suit the specific requirements of the area.

Using 3D animation packages that are typically employed for the creation of 3D animation and special effects in movies and games is not new to the scientific community. Unlike some of their counterparts that tend to oversimplify the system or lack discrete representations of various system components, these applications have highly sophisticated tools capable of producing more accurate animations and dynamic simulations. In spite of their attractive features, these applications have not been specifically designed to work with scientific data, and, as such, require additional tools to support importing and processing of scientific information. As many of these applications support different application programming interfaces (APIs), such as C, C++ and Python, these APIs can be used for the development of additional tools that will perform specific task pertaining to the scientific data visualisation.

In terms of overall usability, there are several benefits these software packages provide, including the ability to produce high quality rendered images, and the availability of many online resources (scripts, plug−ins, tutorials) to assist the user. On the other hand, there are several potential downsides to employing these packages, namely the time and effort required to learn the workflow, and potential high financial cost associated with purchasing the package. Regarding the latter point, there is a number of software packages for which an open access/free version is offered when used for educational purposes.
2.5. 3D applications for visualisation of bacterial cell-surface interactions

Consideration was made towards selecting an appropriate modelling/animation application for development new approaches to the 3D visualisation of bacterial cell-surface interactions. As there are different aspects that could affect the appropriate selection of software (e.g. the availability, the performance, the range of built-in tools, the cost etc.), the choice of specific application depended primarily on three factors:

- Ability to support and process scientific data.
- Options for creation of dynamic interaction (motions) between objects created from that data.
- Production of high quality images and movie sequences.

The first logical option was to evaluate the scientific applications that have already been used for the visualisation of data. One such piece of software is Gwyddion (http://gwyddion.net/), which is free and open-source software for data visualisation. It provides options and tools to enable the analysis of height fields obtained using scanning probe microscopy techniques (e.g. AFM). It has ability to visualise data as 2D or 3D models. It is a powerful tool for data analysis, however its main drawback is its lack of ability to process dynamic simulations and interactions between multiple 3D objects.

As mentioned in section 2.2.3, Avizo has been employed for scientific visualisation, the 3D surface visualisation in particular. Avizo is commonly used for 3D scientific data analysis, processing and presentations. This software can be implemented for scientific, mesh and surface and volume visualisations as it supports a large variety of standard file formats (e.g. 3D geometry, 2D/3D images, microscopy and molecular formats. Additional details are available from the following website: see http://www.vsg3d.com/sites/default/files/related/VSG_Avizo_FileFormats.pdf).

Avizo system components are presented as modules and data objects that are connected with lines that indicate dependencies among the components. These modules can be used for data visualisation or to perform specific operations on these objects. The models of object data are created automatically for file input data or as the output
calculation of a module. For some objects, there are purpose-built interactive editors that allow modifications to be made to the attributes. A Tcl command interface can be used for controlling all components in Avizo. Avizo is an extremely powerful tool for 3D visualisation and representation of data, offering different approaches for the visualisation (e.g. scalar, vector or tensor visualisation, etc.). Complex computation could be executed using the data with Matlab software. A disadvantage of Avizo, however, is the lack of available options to create more complex dynamic animations that would involve interactions between two or more 3D objects.

Another option was to use the program ‘s2plot’ for the 3D visualisation process (Barnes et al., 2006). S2plot is a 3D plotting library written in C programming language, that allows visualisation of data in a 3D space, provides interactive controls for exploring 3D data sets, The development of these tools was initially intended for use by scientist in the field of astronomy, however it has been utilised for the 3D visualisation of nano-structured surfaces (Truong et al., 2010). Its architecture supports dynamic geometry, and can be used for the simulation of dynamic interactions, but to do this it would require a simulation code.

Despite the powerful capabilities of these scientific applications for creating visualisations, performing analyses and inspections of experimental data, they did not fulfil all the requirements need to perform this research. Primarily, their inability to simulate and animate dynamic interactions (i.e. cell-surface interactions) between 3D objects was the main reason for them to be excluded as an option. The next target for consideration was the software packages used widely in entertainment industry for the development of animations and special effects for games and movies. Table 2.1 presents a summary of the capabilities of several 3D animation packages.
Table 2.1. 3D software packages used in entertainment industry.
The tools and options provided for the creation of dynamic animations in these applications afforded them an advantage over their scientific counterparts for 3D visualisation processes. Options for the 3D representation of native scientific data formats were not, however, included in these applications. This was considered less of a challenge, as it was determined to be more effective and less time consuming to develop a tool to import and visualize the data, and continue working with the existing tools for dynamic animation and simulation, rather than to develop new tool sets for the dynamic motion requirements.

2.6. Autodesk Maya

Maya is a powerful 3D software package, which offers extensive tools for 3D modelling, animation, simulation and rendering (Stam, 2007). The objects and dynamic simulations that can be developed in Maya are based on calculations of object interactions and dynamics in 3D coordinate system.

Maya was designed as a database for storing graphical information, which is deposited in objects called nodes. The nodes have properties (or attributes) that store information regarding their changeable characteristics, and data can flow between nodes. Maya’s graphical user interface (GUI) consists of over 900 commands, allowing creation, modification and manipulation of nodes. Behind every GUI-accessible command is a script written in the Maya Embedded Language (MEL). MEL supports customization of existing commands or the development of new ones to perform specific tasks that are not already part of Maya’s default menu set. Maya also supports APIs for C++ and the Python programming language (http://www.python.org).

2.6.1. 3D geometry in Maya

Maya is very versatile software when it comes to 3D modelling. It offers the following types of geometry for modelling and animation: Polygons, NURBS (Piegl, 1991) and Subdivisions. Polygons are the most often used types of geometry for the 3D modelling of objects. A Polygon object comprises three or more three-dimensional
points (vertices), with lines (edges). NURBS allows construction of curves and 3D meshes and are often used for organic modelling. The great advantage of NURBS is the level of deformation that can be applied without the need for an object to have high-resolution components. The NURBS curves can be used for different purposes, i.e. as controls, guidelines from which a 3D surface can be reconstructed, or as motion paths. The Subdivision surfaces are a hybrid surface type that has characteristics of both NURBS and polygonal surfaces.

2.6.2. Animations and simulations in Maya

As is found in many commonly used 3D animation software packages, that animation of an object’s motion in 3D space can be achieved in different ways. One is to manually use the transformation tools (translation, rotation and scale) to move/deform the object’s shape and set key-frames (usually referred to simple as keys) relating to changes in position, rotation and size. The frames and their range (number of frames) are located in the time slider, which is part of Maya’s GUI.

By setting a key for a given frame, Maya can store the values of the transformations that have been made to that particular frame. In practice, a key frame should not be created for every frame: Maya was able to calculate a transformation between two different keys. For example, if an object’s x-axis translation is defined on the first frame to be 1 unit and 100 on the tenth frame, Maya will automatically move the object between the two keys, regardless of the nine frames in between these limits that were not defined. However, this motion is linear and adding more transformations in between these limits will make the motion more complex.

The actual speed and time of the animation within a scene will depend on how many frames per second (fps) are present, rather then the total number of frames comprising the scene. For example, 24 fps, according to cinema standards, represents the real time speed, but this can be altered to any desired specification. When using simulations preformed in Maya, especially when working with dynamic objects (see below), it is most appropriate to set the frame speed to ‘frame-by-frame’. The rationale is that as Maya calculates changes in a dynamic object’s behaviour on a frame by frame basis, the sudden skipping of frames that can occur when using a constant frame speed,
will not allow Maya to do all the necessary calculations. If the animation was created with the key framing option, the frame speed will not make any significant difference.

Setting key frames is a practical method of animating objects in Maya, but it does have its limits. If there is a need to animate, for example, 1000 objects, each of them performing different tasks, then manual key framing would be extremely tedious and in some cases not possible.

A more practical solution is to use Maya’s in-built dynamic capabilities, which allows the simulation of natural motion and object transformation. The dynamic simulations can be stored as keys (Bake simulation option) or saved as nCache files, which will be discussed in later chapters (Chapters 3-7). An additional method for animating and simulating objects is to use Maya Embedded Language (MEL), expressions or the Python programming languages. This is often the best approach, because it is “clean” (for example, a well designed mathematical model implemented as a code to perform the actual simulation or animation without any interference by a user over the animation process) and allows far greater control over an object’s attributes, but it does require programming skills. A combination of all three types of animation strategies (keys, in-built dynamics, custom scripting and expressions) leads to extremely powerful, accurate and high-quality simulations/animation of diverse data formats.

In order to play back an animation containing dynamic motions in real time, the preferred approaches are to save the actual simulation on the hard drive in a cache file (.xml) or to store the simulation as keys on the time slider. The pre-calculated simulation can be played in real time without the need for Maya to do any unnecessary calculations on frame-by-frame basis. The speed of the object in motion, rather then the speed of the animation, depends on how far apart are the keys set. Introducing more frames between the keys would make the motion slower, while removing frames does the opposite.

2.6.3. Maya in scientific research

Autodesk Maya’s applicability for 3D visualisation of research data has been previously recognised, although it has not yet seen much use in the field of nanotechnology. In 2011 an open-source plug-in was presented for supporting
molecular and atomic visualisation directly in Maya and other commercial 3D applications from the entertainment industry (Johnson et al., 2011). The embedded Python Molecular Viewer (ePMV) is a free molecular modelling plug-in, that allows researchers to display molecular data in protein data bank (PDB) format and perform computational experiments. Another study presented a work, showing the potential of Maya’s Dynamic options to simulate molecular interaction (Zhao and Olguin, 2011). These authors developed a simulation of the self-assembly of diblock copolymers (a chain molecule composed of two different block of molecules) utilising Maya’s particle systems.

Researchers from other fields have relied on the powerful 3D visualisation capabilities of Maya, especially in the field of medical research, where the detailed 3D graphical representation of human body parts is required. The ability to inspect the human anatomy in detail as a 3D model allows a greater understanding of the anatomy, providing the possibility to be of great advantage in the solution of medical problems.

An example of the 3D visualisation of muscle and tendon architecture has been presented in one study (Fung et al., 2009), where Maya was used to visualise the muscle structure. This was achieved by extracting 3D coordinates from points that had been placed on dissected muscle fibres. The data of the tendon footprint and the fibre bundles were used for the 3D reconstruction of the muscle. A custom plugin for Maya allowed quantification of the length of the fibre bundle, the angle of the pennation through the volume of the muscle, and the dimensions of the tendon. Its complexity makes the description of the muscle difficult, so this has implications for the 3D reconstruction of the musculo-tendinous morphology and quantification of parameters associated with the individual fibre bundles.

Another study was performed that considered the importance of modelled geometry for the accuracy of biomechanical simulations (Wu et al., 2007). The simplicity of biomedical applications could lead to a lack of ability to provide sufficient tools for the development of complex anatomical geometrics and simulations. Based on a previous study (Agur et al., 2003), where a software bundle (Anatomy 3D) was presented, this information regarding muscle structure was extended using the more
complex modelling capabilities within Maya. With help of the MEL scripting in Maya, two algorithms were developed using the NURBS modelling functionality for visualisation of aponeuroses (layers of flat broad tendons).

The quality of the 3D computer generated imagery (CGI) of data can play a vital role towards the understanding the meaning of large data sets. Visual effects such as the lighting, texture, and even the final composition of the animated sequence can produce a video that appears realistic. The process of finalising the CGI often requires a great deal of effort and time to create a final product that is effectively able to communicate a concept to an audience. Indeed, applications such as Maya were designed for this purpose, and provide wide range of tools for creating and composing special effects.

Even if it is not intended to directly visualise the data, Maya is able to serve as an effective tool for the final 3D composition and animation. This can involve importing the native data formats into an external application, and subsequent conversion and exportation of the 3D geometry in a format that is compatible with Maya. Such a process of data visualisation (referred to as a hybrid approach) was used by McGee (McGhee, 2010), where the pre-visualised geometry of clinical data was composed using CGI methods. 2D cross-sectional images, derived from MRI scans, were reconstructed and processed into 3D digital data that was subsequently imported into Maya. Following this, the effects and animation were developed, delivering a visual driven approach for communicating anatomical data to patience.

Numerous reports from fields other than medicine have demonstrated the applicability of Maya software for the visualisation of research data. For example, in forensic science (Bolliger et al., 2012), when investigating crime/accident events, the interpretation of crime scene by the forensic scientists based on their experience, is as important as collecting the actual forensic evidence. This is known as secondary interpretation. Here, Maya was used for the 3D reconstruction of models that allowed the virtual reconstruction of the crime scene. Elyan and Ugail (2009) presented a plug-in that was developed using Maya’s API (C++), which allowed the construction of 3D models using partial differential equations (PDE) for designing purposes.

The existing wide range of tools and commands for 3D modelling, animation and rendering, as well the support for different programing languages makes Maya a
highly suitable candidate for the development of 3D visualisation of surface-cell interactions. Several studies (Wu et al., 2007, Elyan and Ugail, 2009, Fung et al., 2009) have demonstrated the usefulness of the NURBS geometry. In Maya, NURBS are well defined and organised, in terms of tools and options for creation and modification of NURBS objects. Indeed, the commands are simplified to such level, that no programing or even mathematical skills are required for the 3D visualisation of NURBS objects. Bearing this in mind, the core function of a new tool, developed for the visualisation of nano-structured surfaces in Maya, allows the surfaces to be constructed using NURBS geometry (curves), the final visualisation of which is based on the describing the surface topography. This will be described in the following chapter. In this work Autodesk Maya version 2013 was used.
Chapter 3. Materials and Methods
3.1. Visualisation of Scientific Data: ‘Scenario-based visualisation’

The methodological design for the visualisation of scientific data can, and does, vary from research problem to research problem. Different scientific data requires different methods and tools for effective data visualisation. Whether it is for analysis of the experimental data or for effectively depicting data for publication purposes (i.e. the data needs to be exported in a 2D/3D graphical format), the methodological development, planning and applicability are of critical importance. In some cases, a research story is best represented in an artistic form, where the actual research data may not play a leading role and in these instances, the “artist” is responsible for most of the visualisation process, without the significant involvement by a researcher.

If the main goal is to visualise the data for scientific analysis, the workflow usually involves two main stages (Fig. 3.1). The first is the data collection stage, where the instruments used for obtaining the data are different, depending on the field. The data collected is then visualised for relevant scientific applications. These applications provide a wide spectrum of tools and commands for the analytical inspection of different data formats. This step of the whole scientific visualisation process can be referred to as the ‘semi-automated step’.

The semi-automated step is a process of data visualisation where most of visualisation tasks are performed mainly by "pre-made" commands in the applications of the software being used. The researcher's involvement is limited to simple tasks, i.e. the selection of the data file and selection of a command within the software that will allow most of the data visualisation to be achieved. Other pre-defined commands are then available within the software package for the subsequent data analysis and inspection. Visualisation for subsequent analysis often has the goal to allow researchers obtain a greater understanding of the meaning and trends present in the dataset of interest. In these cases, there may not be a requirement to create a visualisation that is of high quality, such as that required for publication or presentation.
When there is a requirement for a data set to be exported into a specific graphical format, e.g. an image or movie sequence that will be accessible to a public audience, this adds an additional stage in the scientific visualisation workflow process (Fig 3.2). For the researcher, this means that additional tasks need to be carried out. This can involve the modification of an image (i.e. applying colour maps, adjusting the image format and size, etc.) or if there is a need for a movie to be made, a task such as definition of camera motion may be required. Options for image/movie export may be provided in the scientific software packages associated with scientific instrumentation and are simple enough to be exported without the requirement for additional processing.
Figure 3.2. Stages for visualisation of data and final production for publication.

However, there are times when researchers are limited by the amount or type of data available, and the only option available to present the available data in a visual format is to perform additional operations that require the researcher to provide an interpretation as to, for example, the mechanism taking place when one object approaches another. When this interpretation stage is required, researchers may present their ideas to artists (e.g. from the entertainment industry) to realize their “mental image” of what might be taking place in the process of acquiring the data. The artist's task would then be to recreate the researcher’s idea (in the form of, perhaps, a 3D animated movie) that may require domain-specific scientific knowledge that the artist is unlikely to possess.
With the diverse needs that researchers have in order to effectively visualise their data, a question arises: Is there a single solution that would allow researchers to effectively visualise their data and perform analytical tasks, while also producing high quality graphical representation of their data that take the visualisation to the artistic level? While scientific visualisation depends strongly on the tools (i.e. software applications) available, a well-established methodology could be considered as a more important aspect of the scientific visualization process.

This has led to the development of a visualisation process referred to as “Scenario-based visualisation” (Fig. 3.3). The purpose of this development is threefold:

1. To engage researchers in exploring and analysing their data in an interactive virtual environment.

2. To allow the resulting visualisation to be raised to the artistic level, where any lack of data limits the data’s accurate visualisation. Here, researchers need to have the ability to continue the visualisation process with high reliance on their knowledge of science.

3. To present the data, and possible mechanisms taking place in the form of quality 3D movies that will be accessible to other domain experts and to the wider public.

Although the initial target in this work is for the scenario-based visualisation of bacterial attachment onto nano-structured surfaces, its functionality can be applied to the 3D visualisation of scientific data in other fields. The field, the data type or even the software packages do not limit the process of visualisation presented in this chapter.
**Figure. 3.3.** Scenario-based visualisation. The stages of the Scenario-based visualisation workflow were described in the following section.
**Scenario-based visualisation**

- **Stage 1 – Formation of the hypothesis ("mental image")**

  The first stage of visualisation involves the formation of a research hypothesis. A scenario has to be formulated that will transform the hypothesis into a storyline to be used to guide the visualisation of a dynamic process. The 3D visualisation should represent a reflected copy of the researcher's mental image, and thus provide visual support, insight and understanding of the researcher's ideas.

- **Stage 2 – Data collection**

  The second stage of the visualisation process is data collection. This stage involves the researcher's input into the type of data that should be obtained, the way it will be obtained and to what extent the data could be utilised for the 3D visualisation process. Experimental work defines most of this stage. While different fields of science rely on different instruments for collecting data, the goal is usually the same: "gather as much data as possible". The more data available, the more accurate and realistic the 3D representation of that data can be achieved.

- **Stage 3 – 3D visualisation**

  The third stage of the visualisation process is the actual 3D visualisation. The main goal of this stage is to allow researchers to have an ability to directly visualise the data that has been collected. At the same time, if no further data is available, they can proceed with the visualisation process at the artistic level. This is a stage where the hypothesis and the collected data should be realised into a single unifying 3D graphical form. This involves two steps: the ‘semi-automated’ step and the ‘creative step’. The key difference between these two steps is the degree of influence the person undertaking the modelling has on the final visualisation of the process. As explained previously, the semi-automated step is one where most of the data is visualised with tools that do not require a great deal of human interpretation, as it is based on scientific data. On the other hand, the creative step supports the artistic level of visualization. Unfortunately,
the creative step, if not done effectively, can remove the scientific value of the visualization process, mainly because we cannot rely any more on the accuracy of the experimental data. However, it is the stage that could engage researchers to better understand the data. The creative step, as the name suggests, requires “creativity”; visualising research data on an artistic level could lead to ideas and solutions that would slowly transform the creative step into a more accurate and reliable workflow. Additional tools could emerge that are based on the knowledge obtained while working in the creative mode. The choice of an application that will provide options to perform both steps is of paramount importance as the successful completion of the third stage strongly depends on this choice.

• Stage 4 – 3D image/movie production

The final stage of the visualisation process is the image/movie production. The purpose of this stage is to export the 3D visualisation into a format that could be shared and viewed by a large audience. This can involve a production of a single image or a movie sequence.

With all the four main stages accomplished, it does not necessarily mean that the 3D visualization of the data has been completed. If the researcher is absolutely satisfied with the final result, the visualization should be presented to, for example, a research group or scientific community. This could be considered as an additional stage, where the authenticity of the visualisation is tested. If the visualisation is critically reviewed and accepted by a scientific audience, it may be considered a successful visualisation process. However, as long as the creative step is a part of the visualisation the accuracy will always be open to interpretation, meaning that there would be a room for improvement in the way that the data is collected, or in the refinement of the underlying hypothesis. This depends on the level of complexity to which the visualisation was developed. On the other hand, if a scientific audience does not accept the visualisation as an accurate representation of a scientific process, for example, it may not necessarily
be considered a complete failure. The critical feedback should encourage the researcher to go through the stages once more, detect the problems and find the solution to fix and improve the visualisation.

The problems could be of a different nature:

- The scenario was not developed appropriately (the hypothetical assumptions being made should be reconsidered and an alternate scenario be created)
- The amount of scientific data was not sufficient to allow an accurate visualisation to be achieved (here, additional data may be required)
- Inadequate performance of the 3D visualisation (if the hypothesis and the data are correctly developed, the actual visualisation has to be recreated)

The methodological workflow proposed in this section should be considered as a continuous loop through the stages that would need to take place until the visualisation is achieved with the best representation of the research data. It presents an opportunity for researchers to work with their experimental data (semi-automated step) and explore other possibilities to enhance their visualisation (creative step). It is acknowledged that the creative step could give an impression of a flaw in the scenario-based visualisation due to its artistic nature. The approach to visualise objects or interactions is based on an “imagination”, as scientific data should constantly be involved in any type of visualisation. However, there is more to the creative step than meets the eye. It does involve pre-visualised data, and it is an extension of visualisation process where limitations of data do not allow the visualisation to proceed with only the semi-automated step. It encourages researchers to explore the data in different ways, to develop their own ideas and solutions to overcome problems. Based on the work conducted and the artistic options practised in the creative step, new tools could emerge. Another reason for the necessity of the creative step is the amount of data that can be collected, e.g. technology is not yet developed to a point where we can acquire all the necessary information for many scientific processes. The future direction would be to moderately transform the creative step into semi-automated step.

The following sections will describe the application of “scenario-based visualization” for the 3D visualisation of bacterial interactions with nano-structured surfaces. An example of a newly developed tool for use in the software package Maya
will be described, together with an explanation of how the experimental data is visualised and how the creative step is undertaken in this type of visualisation.

3.2. Scenario-based visualisation application for the 3D visualisation of cell-surface interactions

As shown in previous chapter, in nanotechnology, researchers are constantly facing challenges when conducting their experimental work. These challenges are mainly related to the small scale of objects under investigation and the difficulty in controlling their natural behaviour. The same occurs when visualising the data. The extremely small size of objects leads to difficulties in obtaining all the necessary data for the accurate visualisation of the object. It is important to support direct visualisation of experimental data while having the flexibility to proceed with the visualisation when scientific data relating to a mechanism or process is no longer available. The scenario-based visualisation workflow was implemented for the 3D visualisation of bacterial cells and their interactions with surfaces on micro- and nano-scale. The main application used for the 3D visualisation stage was Autodesk Maya.

The first task was to develop a hypothesis regarding a particular naturally occurring process (or interaction) that needs to be recreated (Stage 1). It is the most important stage towards achieving the best (and as accurate as possible) 3D visualisation of plausible bacterial cells-surface interactions.

A few questions about the scenario should be carefully considered, for example:

i. Which species of cells will interact with a particular surface, given that every species has its own specific characteristics and ways to interact with its environment?

ii. How will the cell interact? What is the basic dynamic motion of the cell when it comes in contact with the surface?

iii. What will happen to the cell after the initial interaction the surface? Will it stick to the surface, slide along the surface, will the cell die, etc.?
The second stage of the process is data collection. In the field of nano- and biotechnology researchers rely on different instruments (see examples in Chapter 2) for the collection of data pertaining to objects on the micro- and nano-scale level. Any information regarding the object’s characteristics that can be obtained with these instruments is useful in the development of the 3D models. This data can be in different formats, such as numerical arrays or as an image.

The third stage of the process is the 3D visualisation. The semi-automated step represents a process of transforming the collected data into a 3D graphical representation (i.e. surface topography). The creative step is the development of dynamic interactions and 3D models of cell shapes. For example, in the semi-automated stage, most of the process – apart from selecting the input data set for example – is performed by the software. In the creative step however, hypothetical assumptions about the stages of the bacterial interaction and process of attachment are required, and these are subject to interpretation, albeit influenced by the available theories of interaction, by the animation modeller.

3.2.1. Semi-automated step

For the bacterial cell-surface interactions, the semi-automated step primarily involves the visualisation of the surface topography as a 3D model. The surface topographical data (derived from AFM) scans; for specific details see the Data collection sections, Chapter 4-7) had to be converted into the appropriate format, in this case comma-separated values (.csv) file format. One option is to import the .csv data files into an external application and export the 3D model in an appropriate format that Maya can recognise (Fig. 3.4a), because Maya was not designed to work with common scientific data formats. However, it does require an additional software package, i.e. Avizo, which may not be cost effective and can be time consuming.

Another, more practical approach, was to develop a custom made tool in Maya, that would allow a direct importation of the AFM data and support the 3D reconstruction of the surface models (Fig 3.4b). This can be achieved with several programming languages supported by Maya (refer to Chapter 2 for more details). Both
approaches will be explained in the following sections, as they expand the options for visualising data in the “scenario-based visualisation” workflow.

Figure. 3.4. Three-dimensional models of the surface topography. The converted AFM data visualised in Avizo (a) and in Maya with the newly developed tool (b).

The raw data obtained from AFM was converted to a text file format using the free, open-source application Gwyddion (http://gwyddion.net/). The text files were imported and re-formatted as .csv files utilising Microsoft Excel. One reason for this initial format transformation was Avizo’s lack of ability to import and process text files. Additionally, the text files often contain additional information (e.g. metadata), which needs to be deleted from the file, as only the height values are required for the visualisation process. Regarding the data processing in Maya, it was more convenient to
work with a “clean” data format that will contain decimal values, as it would not require additional coding for reading and converting from the binary value.

3.2.1.1. External application – Avizo Standard version 6

To import the .csv data into Avizo the CSV Uniform Scalar Field was selected from the Format menu. In the Option Window for the CSV Uniform Scalar Field some of the surface attributes were set:

- Under **Column delimiter** a comma was selected.
- Under **Input-Data type** the option **double** was selected.
- The dimension and the resolution of the surface were set in the bounding box option, 1000 for the length and width if the surfaces is 1µm × 1µm or 10 000 for length and width for 10µm × 10µm surfaces. Avizo will create a module, which can be used for extracting a surface mesh for Maya.

To display the surface topography in the 3D view a field must be assigned, in this case the **Height Field**. In order for the surface to be accurate in all dimensions the appropriate value for the surface height had to be calculated This was achieved by dividing one (one represents the highest value perpendicular to the surface in Avizo) by half of the surface dimension, which is 500 nm or 5000 nm depending of the surfaces scanning area, 1000×1000 or 10000×10000. The result was then multiplied by the maximum height value of the surface (the maximum height values can be found in the colour range attribute, which presents the actual minimum and maximum values of the surfaces topography). To get the proper surface, which can be exported as a format that is recognizable by Maya, the **Extract Surface** option was assigned to the Height Field. This allowed for the 3D model to be converted in a module that could be exported as an OBJ file format. The process of data visualisation in Avizo may vary in software’s versions.
3.2.1.2. A new tool – Python script

The other approach to visualise the surfaces was to import the surface topography values from the .csv files directly into Maya, using a custom Python script. Python, an open-source dynamic programming language, became part of Maya’s API in version 8.5. The script was written in Maya’s Script Editor, which simplifies the creation of interface components, such as the user window from which the functions can be accessed (Fig. 3.5).

![User interface of the tool.](image)

**Figure 3.5.** User interface of the tool. The tool developed in Python allows 3D surface reconstruction from a .csv file, creation of basic bacterial models and applying colour maps and colour map values.
As discussed in Chapter 2, Maya does support additional programming languages: MEL (Maya's native programming language) and C++. Python was chosen for the development of the tool, because: it has an easy to understand syntax; it can be written and compiled in the Maya script editor (i.e. C++ cannot); it allows MEL modules to be imported, and most of commands that MEL provides can be called with Python. Apart from that, Python code can be also used in external editors and compilers (i.e. MEL can be only compiled in Maya script editor), as long as the code does not contain MEL syntax.

The tool, developed with Python, consists of three main sub-menus for 3D visualisation and manipulation of nano-structured surfaces in Maya: Import data, Create 3D objects and Surface manipulation (Fig 3.5).

![Import data sub-menu.](image)

- “Import data“
  The purpose of sub-menu (Fig. 3.6) is to import and store information, from which the 3D models of surfaces will be created. It consists of three functions: "Dimension", "Resolution" and "CSV file import". The values for the dimensions and resolution must be set. The dimension value depends on the length and width of the surface scanning area (in nm), while the resolution values are the number of rows and columns of the .csv file. The .csv file option allows for a file to be selected from a folder and imported into Maya. The types of files that can be imported were restricted to only .csv formats; therefore other format cannot be chosen. These functions can be
considered as a container for storing information and do not provide any other functionality.

![Create 3D objects sub-menu](image)

**Figure 3.7.** Create 3D objects sub-menu.

- “Create 3D objects”

  This submenu provides functions for the visualisation of the actual 3D objects. It has two sets of options: Surface and Bacteria (Fig. 3.7). The Surface set provides two functions to create the 3D models of the surface topography: “Create Curves” and “Create Surface”. The “Create Curves” command is the core function of the tool for the 3D visualisation of the surfaces. This option transforms and visualizes the numerical values provided in the “Import data” submenu as a graphical representation (curves) in the Maya view port (Fig 3.8a). When the “Create Curves” option is selected, it reads each row of the .csv file, then parses it into a set height values that are equally spaced along the row. A Maya curve object is then created, based on the height values for each line. Every curve consists of control vertices (CVs) to which values for each dimension (X, Y and Z axis) can be assigned, along with an interpolation function. For simplicity, the B-spline is used. The values in each row file are mapped, in Maya’s world coordinates, to the Y-axis values of the curve’s CVs (Fig. 3.8b). The values of the resolution and dimensions of the surface are used to increment the corresponding X and Z values for each Y value. Finally, with the user selection of all the curves, the 3D surface can be constructed by choosing the “Create Surface” option from the script
interface. This function uses the pre-visualised curves CVs as a “blueprint” to
reconstruct a polygonal model of the surface topography (Fig 3.8c). The “Create Surface”
command activates Maya’s loft tool. If this command does not appear to cover every
curve's CV, some adjustments need to be made in the loft tool itself (see Appendix 3.1).

![Figure 3.8](image)

**Figure 3.8.** 3D reconstruction of surface topography in Maya. The AFM data files, in
.csv format, were imported and a set of equally spaced curves is created based on the
surface height values form the file (a). Each curve consists of control vertices (CVs) to
which values for each dimension (X, Y and Z) can be applied (b). A polygonal surface
is created from the curves (c).

The Bacteria set of functions is for creating the basic shapes of the bacteria *S. aureus* and *P. aeruginosa*, which are scaled proportionally, according the surface
dimensions. The options in the script were designed to follow a process of polygonal
modelling of 3D object, which will be discussed in the “Creative step” section (section
3.2.2).
Figure 3.9. Surface manipulation sub-menu.

- “Surface manipulation”

This sub-menu has a set of tools that provide additional functions for scientific inspection, analysis and surface manipulation of the visualized surface (Fig 3.9.). Most of these options require for the 3D model of the surface to be pre-selected, with the selection tool in Maya. The “Magnification” option applies the scale transformation attribute (the transformation attributes such as scale, rotate and translate are default attributes for every 3D object in Maya), to scale the surface along the Y-axis.

The two functions “Create Map” and “Create UVs” allow for a colour map to be applied to the 3D model of the visualized surface. Each of the functions provides two options that can be selected. Before the colour map can be assigned, UV mapping needs to be applied to the surface (the detailed description of the UV mapping and the colour map creation is explained in the “Creative step” section). In order for the colour map to work properly, the height option for the UVs needs to be selected. These functions are a good example of the flexibility of the programming in Maya. Basically both functions preform a single task dedicated to them, however, there are “option boxes” that allows the function to produce multiple results.
The final three functions (“Get Position”, “HUD Position” and "Display Colour Map Values”) allow for the surface height values to be extracted and displayed in the Maya view port. With a selected-surface, “Get Position” provides the height value of each vertex of the polygonal model. With the values stored in Maya’s memory, the other two options can then be activated. The “HUD Position”, displays the height value of a single vertex in the view port (Fig. 3.10a).

The “Display Colour Map” function creates a new window where five values will be presented (Fig. 3.10b), based on the maximum and minimum values of the surface height. The scaling of the surface model will not affect the height values of the surface, unless the surface transformation is reset.

![Figure 3.10. Surface height values. The height value of a single vertex is displayed in the 3D view port (a). Five colour map values displayed in a separate window based on the minimum and maximum values.]
3.2.2. Creative step

With the visualisation of surface topography as 3D models, the semi-automated step is completed. The remained of the visualisation process is realised in the creative step mode. This involves modelling 3D bacterial cell shapes, applying colour maps, developing dynamic interactions and the final rendering setup and image export.

3.2.2.1. 3D models of bacterial cell shapes

The physical interaction among bacterial cells and the natural environment (e.g. nano-structured surface) strongly depends on both the shape and the outer membrane’s molecular composition (Young, 2006). As mention previously, 3D models of two bacterial species were required for the 3D visualisation of cell-surface interactions: *S. aureus* and *P. aeruginosa*. The models were created as polygonal geometry with Maya’s standard modelling tools. It is important to mention that the models are simple 3D objects, where only the size attribute of the cells (diameter and length) was used as a guide for the modelling process. Both species have different body shapes and outer membrane structures. The *S. aureus* cells have a spherical shape with diameter of 1000 nm. The *P. aeruginosa* cells are rod-shaped with round caps. The diameter of the caps is 1000 nm (same as for the *S. aureus*), while the length can vary from 1000 to 5000 nm. Beside the shape, the outer membrane structure of both bacteria is also different, where *S. aureus* as a Gram positive bacteria are of a more rigid outer membrane structure and *P. aeruginosa* as Gram negative bacteria are more flexible. However, the actual molecular structure (i.e. the lipid layers) was not visualised.

The modelling of the bacterial shape began with a basic polygonal primitive – a cube (Fig. 3.11a). The smooth option was applied to the cube in order to transform the object into a sphere and the model can now be used to represent *S. aureus* cells (Fig. 3.11b). Additional smoothing of the spherical object will result with higher resolution. While this resolution allows for better deformation in later dynamic simulations, it does require more processing power. In order to achieve the *P. aeruginosa* cell shape, additional transformations were made to the smoothed cube. Half of the object’s surface was selected and the extrude options was applied, allowing the extruded half to be translated along the x-axis. The result is a 3D model with additional surface components.
(faces) and the required cylindrical shape (Fig 3.11c). In order for the modelling of the cell shapes to become more practical, the steps explained above were combined into a single code command, which can be found in the tool's GUI.

Maya does have a sphere by default object type, however, because of the poles of the sphere this can lead to very bad deformations when used in dynamic simulations. A 2D bump texture is assigned to allow a more detailed representation of bacteria shape to be obtained, whilst maintaining the underlying 3D model at lower resolution (Fig 3.11d). This helps to speed up interactions within Maya by avoiding the need to generate complex geometrical structures until they are rendered in the final frames. Most of the process for creating 3D models of cell shapes, apart from the texturing of the models, was implemented into the Python code in the new tool interface. When a bacterial model is created by selecting the options from the Bacteria set of functions (see Figure 3.7), the model is proportionally scaled to match the surface dimensions. The default dimensions and the resolution for the two bacteria are given in the Table 3.1.

### Table 3.1
Default dimensions and resolutions of the bacterial 3D models if made with the Bacteria set of functions.

<table>
<thead>
<tr>
<th>Dimensions (Maya units representing nanometre scale)</th>
<th>Resolution (polygons)</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>S. aureus</strong></td>
<td></td>
</tr>
<tr>
<td>Diameter -1000 units</td>
<td>348</td>
</tr>
<tr>
<td><strong>P. aeruginosa</strong></td>
<td></td>
</tr>
<tr>
<td>Round cap diameter – 1000 units</td>
<td>256</td>
</tr>
<tr>
<td>Length – 3000 units</td>
<td></td>
</tr>
</tbody>
</table>
Figure 3.11. Three-dimensional modelling of bacterial cell shapes. The bacteria *S. aureus* (round shaped) and *P. aeruginosa* (rod shaped) were made with Maya’s modelling tools. A cube is used as a base mesh for developing the models (a). The smooth and extrude tool were used as well for further modelling of the bacteria cells shape (b and c). A 2D bump texture was assigned in order to create more “realistic” bacteria (d) and at the same time to preserve a lower quality mesh.
3.2.2.2. Colour maps

The colour maps representing the topographical parameters of the surface were created using Maya’s Hypershade function. Hypershade can be used to store a range of information, such as 2D or 3D textures, bump and displacement maps and can be defined as a graph, where nodes can be created and connected with each other. The UV mapping option was used in order to assign a 2D texture onto the 3D model of the surface. The UV map is the one of the most important parts of the visualization. Its function is to create the two-dimensional texture coordinate system of the 3D model, which allows accurate placement of a 2D texture on a 3D model. A single co-ordinate is assigned to each vertex of the UV plane.

A planar mapping was applied to the surfaces along the X-axis. Usually this is not an adequate way to map more general 3D models, as the UV coordinates may overlap, but it met the requirements for the height-based texturing of the surfaces. For the surfaces, a ramp texture containing layers of colours (Fig. 3.12a) was connected to a lambert material. The material was assigned to the surface 3D model (Fig. 3.12b). Maya provides a great deal of flexibility in defining colour maps. Here, a terrain map was used to emphasize features that are above (green/white) or below (blue/black) the average surface height (brown), however alternative methods can be employed to achieve this result.
Figure 3.12. Colour mapping on the 3D models of the surfaces. The ramp 2D texture with layers of colours representing terrain map (a) applied to a 3D surface model (b).

3.2.2.3. Dynamic simulations
Maya’s Dynamics function supports the construction of active/passive rigid and soft bodies or particles and fluid effects, while simulating their interactions within different force fields such as gravity. There are different types of force fields in Maya that allow simulation of “realistic” motions of objects in 3D space. However, these force fields do not support an option to import any real world data so that an accurate representation of dynamic motion can be recreated. Their purpose is to create a simulation based on constant adjustments of the force fields attributes until the desired effects are achieved. Passive bodies in Maya’s environment are bodies that can interact with other objects, but these bodies are not influenced by the interaction. For example, when a sphere (which is, in this instance, an active body) is bouncing on a plane, the plane will be passive and it will not move when the sphere contacts the surface of the plane.

By default, any dynamic geometry (active or passive) is made as a rigid body (Fig. 3.13a). When a soft body is assigned to a 3D geometry, Maya creates a corresponding particle object (Fig. 3.13b). The combination of the particles and the geometry of the object define the soft body. When a force field affects the particle system, the vertices of the 3D object will move in response to the changes in the particle. This affords a level of flexibility to the 3D geometry that would be very difficult to model on a frame-by-frame basis. The amount of flexibility is controlled by adjusting the ‘weighting’ assigned to the particle. The weighting of the particles (vertices) is presented as grey scale values, where white presents maximum weight that affects the particles, which means that those vertices will be most “rigid” and will not deform if influenced, while the black value presents the minimum weight of the particles and will cause the corresponding vertices to deform the object shape (Fig. 3.13c). Each active/passive body contains attributes that can be adjusted and will have high impact on how the 3D object will interact or move.
Figure 3.13. Dynamic objects. The *S. aureus* cells were made as rigid dynamic bodies (a), meaning that the cell shape will not be deformed when interact with the surface. The membrane flexibility and fluctuation of *P. aeruginosa* cells was achieved by assigning soft dynamic body to the 3D polygonal mesh (b). When a soft body is created, a particle system (b, right) is assigned to every vertex of the polygonal mesh (b, left). The changes made to the particle positions by a force field, or another object, affects the corresponding vertex of the 3D model, resulting in a flexible object. This flexibility can be control by adjusting (“painting”) the weights of the particles (c). Where the black value (0) is assigned, the object will be mostly flexible, while the white value (1) makes the object totally rigid. For the *P. aeruginosa* cells the value of 0.75 was assigned in order to represent the bacteria’s membrane fluctuations, but this number is arbitrary.
3.2.2.4. nDynamic simulations

An alternative approach to visualizing the bacterial interactions with the substrate surfaces was to employ Maya’s nDynamics feature. The nDynamics became part of the Maya menu in the 2009 release, whereas in previous versions most of the dynamic simulations were developed via the Dynamic system. This new feature is an advanced dynamic simulation structure that is powered by the Maya Nucleus technology. The Nucleus system consists of objects, such as particle systems (nParticles), nCloth and passive collision objects, dynamic constraints and the Nucleus solver.

The solver calculates the Nucleus simulations and collisions and also controls some of the dynamic forces, i.e. gravity and wind. Other force fields can be added to the simulation process as well. Additional important attributes of the solver are the substeps and scale attributes. The calculations made by the solver work on frame-by-frame basis; increasing the value of the sub-steps also increases the number of calculation per frame. In other words, if a value of 4 was set for the sub-step, the solver will make four calculations of the dynamic motion per frame. More sub-steps means more calculations and more accurate dynamic behaviour. However, when the calculations are increased, it does require more processing power, which can slowdown the visualisation process. The scale attribute has two options: time and space. The solver calculations strongly depend of the values of these two options.

The nCloth object was used to create dynamic objects from the 3D bacterial models. The nCloth is a dynamic polygonal mesh created as a network of connected particles. Attributes of the nCloth (i.e. Collisions, Dynamic properties, Pressure, Time attributes) can be adjusted until the desired bacterial body behaviour is achieved. Compare to the complexity and the number of attributes with the Dynamic objects (rigid and soft bodies), the nCloth allows far greater control of the 3D model’s dynamic behaviour and collisions.
- The collisions attributes control the accuracy of the collisions between two objects.
- The dynamic properties control the 3D model’s behaviour (i.e. how stretchy or rigid it will be, and how the model shape behaves when it is affected by any type of collision or field.)
- The pressure attributes control the internal structure of the nCloth; it behaviour is like pumping or releasing air from a balloon.
- The time attributes sets a time frame from which the nCloth should start its dynamic behaviour.

The nCloth shapes possess other attributes as well, for additional control of the nCloth, however these attributes were not used in the visualisation process and will not be discussed.

At the end of any nDynamic simulation, it is important to save the simulation with the nCache option. The nCache allows every change of the nDynamic objects, their attributes values and deformations to be stored in a specific file (.xml) on a hard-drive. The frame numbers can be set for the start and the end of the nCache. This allows for the simulation to be played without the nDynamic objects to be active, as the calculations step is suppressed, thus speeding up the interactivity in the 3D view, Additional changes made to the nDynamic objects after the nCache is stored, will not update the nCache file, so the cache process must be repeated and a new file must be created. These nCache files can be used to further adjust the simulation (i.e. timing – when the simulation should start, regardless of the original first frame of the simulation or the speed of the simulation). These changes can be achieved in the Trax editor.
3.2.2.5. Lights and cameras

In 3D applications such as Maya, the quality and the style of the rendered images depends strongly on the lighting and camera setup. Both options are represented as additional 3D objects in the scene and can be manipulated like any other 3D object in Maya. By default, when a new scene is created, Maya also creates cameras (perspective, front, side and top panels) and light sets, but it is rare that this default camera-light setup is adequate or appropriate.

The settings of the camera attributes and animation of its path defines the angle of the scene that will be captured throughout the animation timeline. The camera can be set as either a static object, which means that the camera will not move while the animation plays, or can be animated. This can be achieved with the transformation tools, to move and rotate the camera as desired and setting keys on the frames. Other options are also available for the animation of the camera path, for example, movement “along a curve” (the camera is set to travel along a pre-defined curve path) or with scripting. Maya also provides a stereo camera option, which is ideal for exporting stereo images that can be combined in a post-production application for 3D stereo movies.

There are different types of lights that can be created in a Maya scene (Fig. 3.14.). The choice of an appropriate light is of extreme importance, as it will have a strong impact on the final rendering of the objects in the scene. Each light has its own lighting effect, and their purpose is mainly to recreate realistic natural environment illumination. However, when visualising surface in Maya with applied colour maps, the lights can affect the colour’s intensity (Fig. 3.15a). This could lead to confusion of the height of the surface topography, which is represented by the colour maps. The most suitable option is to use the ambient light. This light will illuminate the whole scene in general, however the intensity depends on the position of the light. The solution for constant intensity of the whole object is to match and link the position of the camera and light, thus allowing for the light to follow the camera wherever the camera is located in the 3D space. This is achieved by “parenting” the light to the camera (Fig. 3.15b). It can be achieved manually in Maya every time this camera-light setup is needed - a more practical approach for the parenting process is to develop a simple tool (script command) that will be part of Maya’s menu.
Figure 3.14. Different types of lighting options in Maya. A poor choice of light will affect the colour map texture. Ambient light was found to be the most appropriate choice.
Figure 3.15. Parent and unparent camera/light setup. If the ambient light doesn't follow the camera in the 3D space it will affect the colour intensity, resulting in the misinterpretation of the variance between the colour layers (a). If the light is parented to the camera, the light will follow the camera’s animation path, resulting in the constant intensity of the colour map (b).
3.2.2.6. Image sequence rendering

The final stage of the workflow allows the simulation/animation performed in Maya, to be rendered as sequential images, which then can be combined in a movie sequence. This was achieved with the mental ray plug-in in Maya, however, Maya does support three other types of renderers: Maya Software, Maya Hardware and Maya Vector (additional render plug-ins can be added). They all have their own advantages and disadvantages, and choosing the right render type depends mainly on the type of objects and textures in the scene. Mental ray is a powerful plug-in that can perform highly realistic renders and allows rendering of custom-made mental ray shading materials and nodes, which otherwise cannot be rendered with alternative renders. Mental ray provides great control over variety of attributes, such as ambient occlusion, global illumination, etc. One of the great features of mental ray is the Passes option. This option provides the opportunity to separately render every effect in an image, i.e. reflections, opacity, diffuse, shadow, etc., allowing control and adjustments of each effect of an image in any post-production application. Common settings that needs to be adjusted for any render type are (see Appendix 3.2 for the values):

a) File name prefix - Where the universal name of all images can be assigned;
b) Image format - The file format of an image;
c) Frame/animation ext - This is very important option. It allows rendering a single image, a sequence of images or a movie format.
d) Frame range – the option becomes available if the whole animation is rendered, rather then a single image. It requires the start and the end frame of the animation to be specified.
e) Renderable camera - An option that allows specifying which camera will be used for the rendering of the scene, in case where multiple cameras are present
f) Image size – As in any image editing application, this option allows the final size and resolution of an image to be set.

Regarding the quality of the rendered images, a basics render setup was applied (presets) in the quality menu. The presets are predefined settings of the render quality, where every setting can be adjusted at any given time. These settings control attributes,
for example, raytrace quality, shadow quality, the data type of the image, etc. In his case, a production quality preset was used.

### 3.2.3 3D movie production

As mentioned in the previous section, Maya provides an option for the direct rendering of an animation sequence into a single movie format (e.g. .mov). However, usually this is not an adequate choice for the rendering process. The main reason is that if there is a system failure and Maya crashes in the middle of the movie rendering, the render process has to start from the beginning, because the frames are automatically combined into the movie sequence. If the same thing occurs when each frame is rendered separately (one image per frame), then the rendering can continue from the frame when the rendering was terminated, by accident or by the user.

The final 3D movies were produced using Adobe Premiere CS5.5. However, there are other available applications that can perform same tasks for movie post-production, such as Nuke ([http://www.thefoundry.co.uk/products/nuke/](http://www.thefoundry.co.uk/products/nuke/)), Adobe After Effects ([http://www.adobe.com/au/products/aftereffects.edu.html](http://www.adobe.com/au/products/aftereffects.edu.html)), etc. Premiere is a simple to use and effective software, but has limits in terms of the different types of effects that can be add to the movies, compared to other post-production applications.

The rendered images were imported into a pre-selected sequence as numbered stills. A sequence can be considered as a timeline, where movie clips can be added and assembled in the required order. For the highest quality outcome, the sequence settings, such as resolution, should match those of the rendered images. By importing the images as numbered stills, Premiere will combine the images into a single movie clip that can be added to the sequence. Additional information can be added to the movie clips, for example: movie titles, an overlay of the colour map and its values. The final sequence can be exported in various movie formats, including: H.264, MPEG2, MPEG4, QuickTime, etc.
In the following chapters, the Scenario-based visualisation will be utilised for the 3D visualisation of different scenarios of bacterial cells-surface interactions. The data of different types of titanium surface, along with natural surface structure will be used for the 3D graphical representation of the surface’s topography. The two species of bacterial cells (S. aureus and P. aeruginosa) will be used for the development of the cell-surface interaction as the both species interact differently with the surface. Apart from dynamic interactions, additional solutions for 3D surface reconstruction and visualisation will be presented in order to highlight/compare the surfaces characteristics such as the roughness of the surface architecture. The visualisation process will be demonstrated within Maya software.