

Architectural Visualisation Toolkit  
for 3D Studio Max Users

Neil Cooper

A thesis submitted for the degree of  
Master of Science  
at the University of East Anglia  
September 2012

Architectural Visualisation Toolkit  
for 3D Studio Max Users

Neil Cooper

© This copy of the thesis has been supplied on condition that anyone who consults it is understood to recognise that its copyright rests with the author and that no quotation from the thesis, nor any information derived therefrom, may be published without the author's prior, written consent.

## **Abstract**

Architectural Visualisation has become a vital part of the design process for architects and engineers. The process of modelling and rendering an architectural visualisation can be complex and time consuming with only a few tools available to assist novice modellers. This paper looks at available solutions for visualisation specialists including AutoCAD, 3D Studio Max and Google SketchUp as well as available solutions which attempt to automate the process including Batzal Roof Designer. This thesis details a new program which has been developed to automate the modelling and rendering of the architectural visualisation process. The tool created for this thesis is written in MAXScript and runs along side 3D Studio Max.

## Table of Contents

1.	Introduction .....	1
2.	Background and Related Work.....	7
3.	Analysis & Requirements.....	18
3.2.	Google SketchUp.....	19
3.3.	AutoCAD .....	23
3.5.	Roof Designer - Real World Example .....	30
3.6.	Conclusions.....	33
4.	Implementation .....	36
4.1.	Program Implementation .....	36
5.	Results & Evaluation.....	64
6.1.	Improvements / Future Work .....	76
7.	Appendix A .....	78
8.	Appendix B - Evaluation Questions using Traditional Methods.....	87
9.	Appendix C – Evaluation Questions using MAXScript ATK.....	88
10.	Glossary .....	92

## List of figures

<i>Number</i>	<i>Page</i>
Figure 2-1 Example of tiled roof using Roof Designer (Batzal Software) .....	16
Figure 3-1 3D model completed in Google SketchUp .....	19
Figure 3-2 Brickwork Texture to be used in the Diffuse element of a material.....	20
Figure 3-3 Brickwork Texture to be used in the Bump or Displacement element of a material.....	20
Figure 3-4 3D model completed in AutoDesk AutoCAD .....	25
Figure 3-5 3D model completed in 3D Studio Max .....	26
Figure 3-6 Example of individual 3D tile geometry.....	27
Figure 3-7 Example of the inset command in 3D Studio Max.....	28
Figure 3-8 Example of extrude command in 3D Studio Max .....	28
Figure 3-9 Completed tiled roof using Batzal Roof Designer in 3D Studio Max.....	32
Figure 3-10 Time Required to Complete Tasks Using Traditional Methods.....	34
Figure 4-1 Use-case diagram illustrating MAXScript ATK.....	37
Figure 4-2 Error handling .....	38
Figure 4-3 Showing how the chamfer tool was used to soften the edges of geometry .....	41
Figure 4-4 Dialog box for window constructions .....	42
Figure 4-5 Dialog box for door construction.....	43
Figure 4-6 Dialog box for garage door construction.....	44
Figure 4-7 Door handles available for standard doors .....	45
Figure 4-8 Door handles available for garage doors .....	47
Figure 4-9 Garage Door geometry created using MAXScript ATK .....	47
Figure 4-10 Material presets available.....	48
Figure 4-11 Gutter construction options .....	49
Figure 4-12 Displacement map used for the grass material .....	50
Figure 4-13 Gutter profiles available by default.....	52
Figure 4-14 Result of tiling function.....	53
Figure 4-15 Roof tile setup .....	54
Figure 4-16 Flow chart to show procedure followed when modelling roof tiles .....	55
Figure 4-17 Roof ridge tile shapes available by default .....	56
Figure 4-18 Lighting setup dialog.....	57
Figure 4-19 Artefacts in renders due to incorrect global illumination setup.....	58
Figure 4-20 SolidRocks 3D Studio Max plug-in.....	60
Figure 4-21 Daylight overcast lighting setup within 3D Studio Max .....	61
Figure 4-22 Daylight sunny lighting setup within 3D Studio Max.....	61
Figure 5-1 Showing Subject C gutter shapes.....	66
Figure 5-2 Showing gutter shapes provided with instructions.....	66
Figure 5-3 Showing the inaccuracies of the gutter model created by Subject C.....	67
Figure 5-4 Graph to show time required to complete tasks using traditional 3D modelling method .....	69
Figure 5-5 Graph to show time required to complete tasks using MAXScript ATK .....	70

Figure 5-6 Graph to show the average time required to complete tasks – Traditional vs. MAXScript ATK.....	70
Figure 5-7 Average user confidence.....	72
Figure 5-8 Graph to compare the time taken to perform modelling tasks using evaluated software and MAXScript ATK .....	74
Figure 7-1 Completed render 01.....	78
Figure 7-2 Completed render 02.....	79
Figure 7-3 Completed render 03.....	80
Figure 7-4 Completed render 04.....	81
Figure 7-5 Completed render 05.....	82
Figure 7-6 Completed render 06.....	83
Figure 7-7 Completed render 07.....	84
Figure 7-8 Completed render 08.....	85
Figure 7-9 Completed render using HDRI lighting option.....	86

## **Acknowledgments**

The author wishes to thank his supervisor, Prof. Andy Day who was extremely helpful and offered valuable assistance, guidance and support during the writing of this research project. The author also wishes to thank the members of the supervisory committee, including Dr. Stephen Laycock without whose knowledge and assistance this study would not have been successful.

# **1. Introduction**

## **1.1. Methodology**

### **1.1.1. Problem Description and Statement.**

Many AutoCAD design engineers now strive to create increasingly sophisticated and visually impressive solutions. However, it has been observed that many specialists attempt to create visualisations of their designs without any prior knowledge of visualisation or 3D modelling packages. It has also been observed by the author that a wide range of sub-standard visualisations have been used for public exhibitions and publications. The civil engineering and construction industry has become increasingly more competitive and this coupled with the recent recession requires all work which is on display to be of the highest standard possible and also value for money.

The software solutions which are most widely used in design offices are not specifically designed for architectural visualisation specialists. For example, 3D Studio Max is designed to be used for character animation, modelling for advertising, vehicle modelling amongst other purposes. Personal experience confirms the tools available within 3D Studio Max are not specifically tailored to create the geometry required for architectural work. For example, to tile a roof in 3D Studio Max is not a simple procedure. There are currently and have been for many years 'quirks' in 3D Studio Max which can make the architectural modelling process difficult for all but advanced users who understand the fine points of local co-ordinate systems. The loft tool, for example, uses a two dimensional (2D) profile which is combined with a path to create a 3D object, a demonstration of this would be a road profile which has 2 lanes and 2 footways which would be lofted along a road alignment (centre line) to create the 3D Geometry for a road.

To learn all of these tools and how they can be effectively used requires a substantial investment in time. With visualisation becoming more necessary and increasing future demand for these products it is considered that a solution to this problem is required.



There are potentially a variety of resolutions to this problem. One solution could be to write a standalone program with specific design tools. A standalone application would not require the 3D Studio Max software therefore the costs involved would be significantly less to the end user. Also, all modelling tools would need to be programmed from scratch, this could have some performance enhancements because all programming would be specific for the task.

Alternatively, a toolkit could be programmed in MAXScript. As MAXScript is a high level programming language running over 3D Studio Max, it is able to incorporate much of the modelling and rendering functionality which is already available within 3D Studio Max, working this way should promote faster application development. By programming a toolkit in MAXScript, 3D Studio Max which is a general purpose tool could be tailored to complete specific tasks required by Architectural Visualisation Specialists.

The research for this project could be conducted in various ways. One option would be to assess the existing solutions to the problem, compile the portions from each which work well and those which work not so well, then build a solution based upon this. This method would involve assessing software packages which could be used to manually achieve the solution, as well as software packages which achieve the solution in an automated way. The problem with working this way would be that the final solution is not likely to be highly original in its operation. The positive aspects of working this way would be that the solution would be based on a set of instructions which may have been developed over many years and have already had an extensive amount of testing. Using this method relies on the knowledge of the author only.

Alternatively, the research could be conducted by sending out questionnaires to visualisation specialists which would gather information on existing products and processes used for modelling and rendering. The questionnaire would gather information on the success of each of the methods and a solution could be based around this feedback. The advantage of this option would be that the final solution would be based on the knowledge of a number of visualisation specialists therefore could result in a more effective solution. The negative aspects of this research would be that it may be difficult to get enough willing participants for the research.

Another research method could be to contact any software developers which are working in a similar field to assess their processes when designing a new toolkit. The research would also assess whether any developers were working on a similar solution and the research could be conducted alongside their development process. The negative aspects of this type of research once again could be that it would be very difficult for a developer to speak to you about a program which is currently in development. Also, it is unlikely that any information gained from the developer research could be used as part of the research project.

Due to time constraints for this thesis, it was decided that the initial suggestion of researching existing software packages and processes would be most appropriate.

### **1.1.2. Aims**

The overall aim of the project is as follows.

To make it possible for relatively inexperienced office engineers to produce high quality visualisations without having to outsource work to visualisation specialist consultants or engineering companies who have visualisation capabilities.

### **1.1.3. Objectives.**

Objective 01.

Evaluate existing methods of creating 3D models that are used for architectural visualisation.

Objective 02.

Evaluate existing 3D modelling packages, and the techniques used in each package to create a high quality 3D render for architectural visualisation. The project will be aimed specifically at producing architectural visualisations of modern British standard construction houses. For example, the type of houses which would currently be constructed on many British housing estates. From personal experience, it is recognised that the techniques which would be used

when creating a visualisation could be broken down as follows. Modelling roof tiles, modelling gutters, modelling windows, modelling doors, lighting the scene, texturing the objects and rendering the scene. These processes should be analysed individually for each piece of software. Also, from personal experience, it has been recognised that V-Ray is currently thought of as the most effective and most widely used rendering engine by the visualisation community, ideally, this should be used for the final rendering.

Objective 03.

Based on the research mentioned in Objective 02, produce a tool which both decreases the time required to produce a 3D render for architectural visualisation and makes the process easier to achieve for both skilled visualisation specialists and those with no visualisation experience.

Objective 04.

Carry out a series of evaluations and interviews with relevant subjects to determine how successful the solution is.

Objective 05.

Compare the tool produced in this study as part of objective 03 to previous methods of creating 3D models for architectural visualisation in order to assess whether the aims of the project have been achieved.

Objective 06.

Explain any appropriate future work which may be relevant.

## **1.2. Motivation**

After successfully completing a BSc in Computer Science at the University of East Anglia in 2003, the Author has been working in a civil engineering office as a 3D architectural

visualisation specialist. This prompted the observation that, in the current working environment, many visualisation specialists are self taught, mainly from tutorials on the internet and text books. These people are rarely taught by other skilled visualisation specialists. As engineering and construction design processes have become increasingly advanced, three dimensional (3D) design has become the norm in addition visualisation has also become a requirement that is necessary to get many projects through the planning application stage.

Therefore with the increased demand for visualisations and also the lack of necessary skills in the workplace, the author saw the need for a tool which could make the production of high quality 3D visualisations accessible to almost anyone with basic computer skills and access to a 3D modelling package. Architectural visualisations can be referred to as 3D Renders, the term 'rendering' is used to describe the creation of an image or movie, which is generated using 3D Data. A 'rendering engine' is a software solution which creates the images or movie, based on the 3D model and any related textures.

### **1.3. Contributions**

This section provides an overview of the scientific contributions from this thesis.

The contribution of this thesis is the design and implementation of a plug-in which assists architectural visualisation specialists and also makes the time consuming and repetitive tasks of 3D modelling accessible to inexperienced users. The solution is called MAXScript ATK (Architectural ToolKit) and is programmed using MAXScript which is the scripting language running within 3D Studio Max. The plug-in utilises the V-Ray rendering engine to create a near photo realistic visualisation using basic tools. The work completed as part of this thesis has not been conducted before as a research project. However, there are a small number of plug-ins which are available to purchase which provide some of the functionality of the program developed. This solution attempts to provide all required functions when constructing a building in 3D. 3D visualisation is commonplace although is still regarded as an expensive optional requirement for construction projects which is completed by highly skilled modellers.

This solution could be used by architectural visualisation firms for time and cost savings and could enable companies to utilise more of their unskilled workforce for visualisation if required.

#### **1.4. Structure**

The structure of this thesis will involve a background and information section which will discuss the broad uses of visualisation and architectural visualisation specifically. Following this, the thesis discusses the various software options which are available to complete architectural visualisation tasks. This section is concluded and requirements of the solution are defined. The thesis leads on to the implementation section which illustrates the processes followed to create the solution. The thesis is concluded by pulling together conclusions based on testing and evaluation of the toolkit.

## **2. Background and Related Work**

### **2.1. Visualisation**

Visualisation is the technique used for creating images, animations or interactive solutions to get information or a message across to a user and provides an alternate means of displaying data. The process of visualisation has been around for thousands of years, a simple paper map is a form of visualisation, as the map is a visual representation of the land and ocean (Harley, 1968). Based on the requirements of the viewer this alternative means of viewing data would need to be more accessible and easily understood than the original data set. Data in a table can be presented in the form of a chart to display relevant messages across to the reader. In the same way for example plans to build a new villa in Bermuda could be visualised to give a photorealistic representation of the new development.

### **2.2. 3D Visualisation Uses**

3D Visualisation can be used for numerous purposes, these include car concept design, civil engineering, medical training, and architectural uses. Car manufacturers have explored the uses of large scale displays with interactive properties during the design process (Buxton et al., 2000).

One of the applications of interactive visualisation is to simulate operations when training surgeons. Visualisation for medical imagery has been taken further and used in combination with Augmented Reality, the term used for technology which combines real-world footage and computer generated images in real time (Macmillan-Dictionary, 2011). In 2004 ARView software was developed at the University of East Anglia which allowed the user to perform the standard procedures for stereoscopic augmented reality surgical navigation (Lapeer et al., 2004). 3D visualisation is often used for medical and scientific research procedures. For example Optical Projection Tomography (OPT) has recently been developed which rapidly generates digital 3D models of intact specimens used to construct a 3D model of the early human brain development. OPT has an advantage over traditional methods such as Magnetic Resonance imaging (MRI) because it can produce a model using fewer samples and smaller subjects.

(Kerwin et al., 2004). An online method for analysing biological interaction data entitled VisANT was investigated by (Hu et al., 2004). This system visualised data including sequence and structure of biological organisms. In a similar way to the brief of this project, programming can be used to speed up the process for 3D modelling for medical purposes. (Alejandro F. Frangi, 2002) discussed an automated process which involved a program which analysed Magnetic Resonance Imaging data and automatically constructed true 3D models from the data.

Recently, 3D display technology has become widely available on the open market. 3D Cinema has been around in various forms for many years, however from personal experience, it has been noted that a recent push for 3D movies seems to have been embraced by the public. This technology has been brought to the home with various types of technology for 3D televisions and more recently 3D technology for PCs and games consoles. The wearing of 3D glasses, which is required for what's known as Stereoscopic TV is certainly the main barrier to the mass acceptance of 3D TV (Chassaing et al., 1991). There are various methods of no glasses TV, including parallax barrier and lenticular lens technology. These methods are sometimes referred to as auto stereoscopy. These methods are all currently in development however the current solutions cannot match the image quality currently available with 3D TV using glasses (Matusik and Pfister, 2004). Unfortunately the current technology involved in auto stereoscopic TV limits the viewing angle drastically to make the solutions unacceptable for general use. In 2004 a 3D TV prototype system was implemented with real-time acquisition, transmission, and 3D display of dynamic scenes by engineers at Mitsubishi Electric Research Laboratories (Matusik and Pfister, 2004).

Crime scene 3D reconstruction is a common use for visualisation and is being used regularly by police forces across the world. The creation of a crime scene model needs to be accomplished rapidly and solutions to speed up the process have been discussed by (Se and Jasiobedzki, 2005). These involve a software solution named Instant Scene Modeller (iSM), this 3D imaging system automatically creates models using a readily available hand-held stereo camera. The system

requires the user to point the camera at an area of interest and will create a photo-realistic 3D calibrated model automatically within a few minutes.

Visualisation has played a role in astronomy for thousands of years. Ancient Egyptians mapped the location of prominent stars in their tomb decorations, constellations inscribed on the Farnese globe have been produced by Greeks in the 2nd century CE and star maps have been found dating back to 940 CE created by the Chinese (Barnes and Fluke, 2008).

Another form of visualisation are the weather reports we see on television every day. The data gathered using technical monitoring equipment generates forecast models that are displayed in a form which is easily accessible to the general public and in a visually pleasing manner. Weather data can also be an important part of architectural visualisations, historical weather data files can be imported into mainstream software such as 3D Studio Max and information from these files can be used to illuminate a scene. This data could be used by the police to reconstruct a crime scene accurately. For example, if a car accident occurred and the driver suggested that their view was impaired by the reflections of the sun on a nearby building, this situation could be reconstructed in 3D using historical weather data.

The more recent affordability of head mounted displays, data gloves, tracking systems and powerful computer graphics solutions has made the use of computer based 'Virtual Training Systems' very attractive (Doerr et al., 2007). These solutions can be used for many purposes, including flight simulation for aircraft or helicopter pilots or tank training simulations for infantry to gain insight to battlefield practice via virtual reality before being sent into actual battle. Military operations regularly use 3D visualisation, for example jet fighter pilots, Robotic Operated Vehicle (ROV) operators or office based commanders. Predicting where crowds of non-combatants are likely to be is important when planning modern military operations as they can create difficulties for the combatant forces. In 2008 research was conducted to include realistic crowd behaviour into military simulations (McKenzie et al., 2008).



The user interface is the portion of an interactive computer that communicates with the user, and includes any aspect of the system which is visible to the user (Jacob, 1998). User interfaces are important for 3D modelling packages, in the current environment, many computer users are not IT specialists, and the user interface has become much more of an important part of the program. The recent growth of in the popularity of the Apple iPhone has demonstrated the value of a well thought out interface for computer systems. Head mounted displays have become a mainstream device and can be attached to a PC or blue ray player to better immerse the viewer in the scene or simply used to re-create a cinema experience at a relatively low cost.

In modern film production architectural visualisation has become invaluable for many modern blockbusters. Productions including '300' 2006 and the TV series 'Spartacus' 2011 have relied on the skills of 3D modellers to create accurate and impressive 3D environments, the majority of acting in these productions is undertaken in front of a green screen and overlaid onto a rendering of a 3D scene.

Due to the high costs of physically running a Formula 1 car around a test track, 3D training solutions are used to train drivers before races. These systems are built using a highly accurate 3D survey which will generate a model of the surface of a track so that all undulations in the surface are appreciated within the simulator. There have been a number of simulators which have been built in recent years however most of these have been built for commercial purposes and little has been published (Funkhouser et al., 1996).

Another form of visualisation which has become commonly used in the recent years is to illustrate the potential impacts of engineering proposals or natural disasters on the environment. For example, visualisation can be used to illustrate flood mapping showing that a town or sections of the community could be under 5m of water if a 1 in 100 year storm were to strike. The use of visualisation for this purpose can show a house owner the actual effects of a predicted event with far more impact than a 2D map showing flood levels. A recognised means of displaying flood prediction data is Geographic Information Systems (GIS). GIS software can be used to analyse and integrate data from various sources for floodplain management (Nunes

Correia et al., 1998). The spatial data used for GIS is easily transferable to Computer Aided Design (CAD) and visualisation applications which make 3D visualisation a natural workflow progression. Data from GIS mapping can also be used in conjunction with statistical analysis programs to create visuals with data content. In 2008, software named Biosphere3D was developed which evaluates and describes the application of interactive landscape visualisation. The project focused on real time rendering of vegetation and foreground detail. The work followed on from coastal erosion studies conducted in Norfolk (Paar et al., 2008).

Traditionally, the only way to understand the workings of a coded computer program would be by examining its source code with a debugger. Because computer software is one of the most complex human artefacts (Panas et al., 2007), work has been undertaken by (Stasko, 1992) which visualises and animates computational processes to help understand a program. The system is titled Polka-3D which was programmed in C++ and uses the GL graphics library.

### **2.3. Architectural Visualisation**

Architectural visualisation will become even more widely used as BIM (Building Information Modelling) becomes a preferred design solution. Software packages such as Autodesk Revit are commonly used by both architects and engineers across the world. BIM software can be used to create a complete digital model of a building, each piece of geometry in the 3D model holds data detailing what it would eventually be constructed from. This way, a list of materials and respective costs is automatically created by the BIM software, streamlining the design process. When a model is altered, the bill is automatically updated to reflect the alterations. By using BIM software, Architects can build a model without the detailed knowledge which is normally required by a structural engineer (Ibrahim and Krawczyk, 2004). Although the goal of creating a single building information model has existed for over 30 years, useable solutions have only come to the market within the previous 5 years. (Howard and Björk, 2008).

As 3D models are increasingly used at the design stage of building projects, the use of these models can be multi purpose. An accurate 3D model used for BIM or visualisation could then

be used with building simulation tools such as AutoDesk Ecotect which can analyse a design for sustainable building characteristics and predict environmental aspects of the design (Roberts, 2001). Work was carried out by (Caldas and Norford, 2001) which looked at a solution to integrating energy consumption data into the CAD design stage. From personal experience, it has been noted that some architects have used similar systems to attempt to develop micro climates within their designs to minimise heating costs.

3D Virtual reality has potential to substantially improve the visualisation of building construction and design. (Whyte et al., 2000). Methods are becoming more commonplace which allow high level managers without any modelling or Computer Aided Design (CAD) skills to interact with a model in a virtual environment and relocate building objects using gloves or haptic devices which attempt to communicate the feel of a 3D object to the user (Salisbury et al., 2004).

When a high profile engineering proposal is planned and a 3D visualisation is required, an excellent way of representing the visualisation would be to have a 3D tour via a website. Various tools including 'SWIFT 3D' are available to convert 3D models to run in the web environment. These methods are also used regularly for product visualisation and often appear on mobile phone websites. The process of integrating 3D architectural visualisations in the web environment has been discussed by (Yan, 2006), the system involves VRML technology and as well as housing 3D objects, the 3D objects hold information to illustrate their purpose and usage, in a similar way to GIS and BIM.

Visualisation of architecture is a regular occurrence in computer games. Arguably the most popular virtual world platform is Second life (<http://secondlife.com/>) which puts an emphasis on social interaction, however architectural models can be bought and sold for a substantial amount of money (Boulos et al., 2007). This and stock model sites such as TurboSquid (<http://turbosquid.com/>) or image sites similar to IstockPhoto (<http://itockphoto.com/>) are a tool for 3D modelling specialists who wish to sell their creations online.

Visualisation is often used by architects for master planning purposes. Master planning is for planning land use, conservation, and development at larger scales, involving comprehensive areas or units of landscape and incorporates essential infrastructure into the design at the earliest stages (Encyclopedia-Britannica, 2011). The uses of landscape visualisation for planning purposes has been discussed by (Sheppard, 2001) who explains how further knowledge of the visualisation systems and their uses is required.

Architectural visualisation can take many forms; the final output can involve still imagery, movies, 3D movies or a 3D interactive solution. In all the above mentioned uses, the style and visualisation content would also depend on the purpose. For example, at early stages of a construction design project, a conceptual low detail model would be most relevant.

When starting a new project, architectural visualisation specialists must first decide on the appropriate tool for the job, this depends on the required output. The tools available to create visualisations are mainly specialist and require skilled operators. More recently there have been new tools which have entered the market; Google SketchUp for example which aims to simplify the 3D modelling process, however, the final rendered output is generally relatively simple too. Currently, within the architectural visualisation community, it is generally thought that the market leader for the highest quality architectural visualisations would involve modelling with 3D Studio Max and rendering with VRay. Although more recently, VRay has become available to use as a rendering engine in Google SketchUp.

If a required model does not have to be particularly detailed and is of a small scale, Google SketchUp is likely to be chosen due to the basic user interface and relatively simple modelling tools. Google SketchUp would not however be the most appropriate tool when developing a large scale visualisation due to the fact that within the software there are no options to perform any procedural modelling. Procedural modelling is the process used to semi-automatically model geometry based on rules and settings. Procedural modelling and texturing is discussed in detail in (Ebert et al., 2002). For a very large scale project, the most appropriate procedural modelling solution is likely to be 'CityEngine' '<http://www.procedural.com>'. A procedural city

modelling solution may take a road layout as an input and the algorithms within the program would generate a city that is based on this road layout. CityEngine is able to read in GIS or CAD information and use this data to create a model. CityEngine is also able to create reports to analyse the design for BIM models. The construction of buildings can be tailored to specific environments based on rules set out in the Python scripts which can be applied to the model. Modelling of large and complex cities has a great appeal to computer graphics specialists (Parish and Müller, 2001). The final output from CityEngine can be visually impressive, however the software is not designed for low level manipulation of buildings which is required in the brief of this Masters project. With the introduction of BIM products such as Revit, the majority of any design is completed in 3D, this along with the fact that AutoCAD has high quality rendering capabilities, in the future there may not be a requirement for visualisation specialists. A solution was discussed by (Lipp, 2008) which involved a graphical user interface alongside an automated architectural building program which enabled the user to have a higher level of control. The program was designed for users with little computer science background.

From personal experience, including animated characters into an architectural visualisation scene has often proved time consuming and problematic. (Blumberg and Galyean, 1995) looked at behaviour based 3D characters which incorporated a behaviour model to control the characters within a 3D scene.

Including realistic plants and trees into an architectural visualisation is challenging for computer graphics as the number of polygons required can be high. (Deussen et al., 1998) looked at an automated solution which inputs terrain data and covers the terrain with plants and trees based on rules.

Procedural modelling has been used to develop visualisations of historical sites or buildings. Reconstructing large areas of historic cities involves assembling a scene from a combination of historical information from multiple sources often without known points of reference; this uses surviving landscapes and monuments that can still be (at least partially) observed and measured (Flack et al., 2001). Interest is growing in the construction of 3D models of urban

environments which use a combination of digital mapping and rendering techniques (Shiode, 2000). With the significant improvements in computer power and graphics over the recent years, the visualisation industry has benefitted from faster rendering of complex architectural scenes (Aliaga et al., 2007).

A key goal within computer vision has been to create a process for modelling and rendering 3D structures from 2D images (Debevec, 1999). Photogrammetry is structured around the processing of images, with main products: DTMs, DSMs, orthoimages, 2D and 3D reconstruction and classification of objects for mapping or thematic applications, and visualisation maps, 3D views, animation and simulation. (Debevec, 1999)

Recently, if a high quality model is required of an existing area. This can be surveyed using a laser scanner to create a point cloud which can then be used to accurately model architecture within a modelling package. Point cloud data can be analysed and manipulated in recent versions of AutoCAD which make the data collection method readily available. These methods are described in more detail in (Allen et al., 2005).

As far as the author is aware, the brief of this project has not been attempted before, there are however a number of tools available which can accomplish a proportion of the tasks required of this masters project. For example 'Roof Designer' produced by 'Batzal Software' '<http://www.batzal.com>' which is a plug-in for 3D Studio Max and provides a good solution to automated roof tiling in 3D Studio Max.. The results can be realistic, as shown in Figure 2-1, although much of the detail in this case is gained from time spent applying texture maps.

Leading into the future, modelling of 3D objects using 3D modelling software tools may become redundant. Software has been developed which can use 2D images of a building taken with either a standard digital camera or from scanned images and analyse these to generate a 3D model automatically. (Lee et al., 2002) describe a method that uses two or three vanishing points, and three 3D to 2D line correspondences to estimate the rotational and translational parameters of the ground level cameras.

A modelling solution which is relightable (can be lit accurately from within a 3D package) has been discussed by (Melendez et al., 2011). This process involves a LiDAR (Laser) scan used in combination with photographs; these are taken under diffuse lighting conditions to create the model, which is accurately textured with bump maps. A realistic output can be gained using high level automated modelling to construct existing architecture. A system described by (Dick et al., 2000) constructs a 3D model using a number of square 3D objects known as planes based on a small number of photographs. A method has also been discussed by (Glencross et al., 2008) which uses only a standard digital camera and a single view; the results of the modelling were said to be similar to that of a model which was constructed using base data from a laser scan.



Figure 2-1 Example of tiled roof using Roof Designer (Batzal Software)

The work conducted for this thesis leads on from the work taken place by (Glencross et al., 2008) and aims to assist a user in creating a highly detailed visualisation.

Many of the methods within the work investigated as part of this study are relevant to this project. The most obvious is the processes used from Batzal Roof Designer to generate the

roof tile geometry. The easy accessibility of Google SketchUp makes the solution popular and some of the aspects of the interface should be taken into account when programming.



## **3. Analysis & Requirements**

### **3.1. Analysis**

To determine the practicality of producing a well rounded and user friendly toolkit for visualisers, a survey of existing tools was required. The survey required an analysis of the main processes involved when building a 3D architectural model. As set out in the objectives, the processes required for the visualisation are divided up into 6 main categories, modelling roof tiles, modelling gutters, modelling windows and doors, lighting the scene, texturing the objects and rendering the scene. From personal experience, it is thought that all of these processes could be automated as part of this work. It is thought that the only process which cannot be automated is the basic building modelling. This involves creating the walls, and roof shape of the building. Tools already exist in 3D modelling software packages to perform these tasks and therefore they are not required as part of this toolkit.

## 3.2. Google SketchUp

### 3.2.1. Tiling

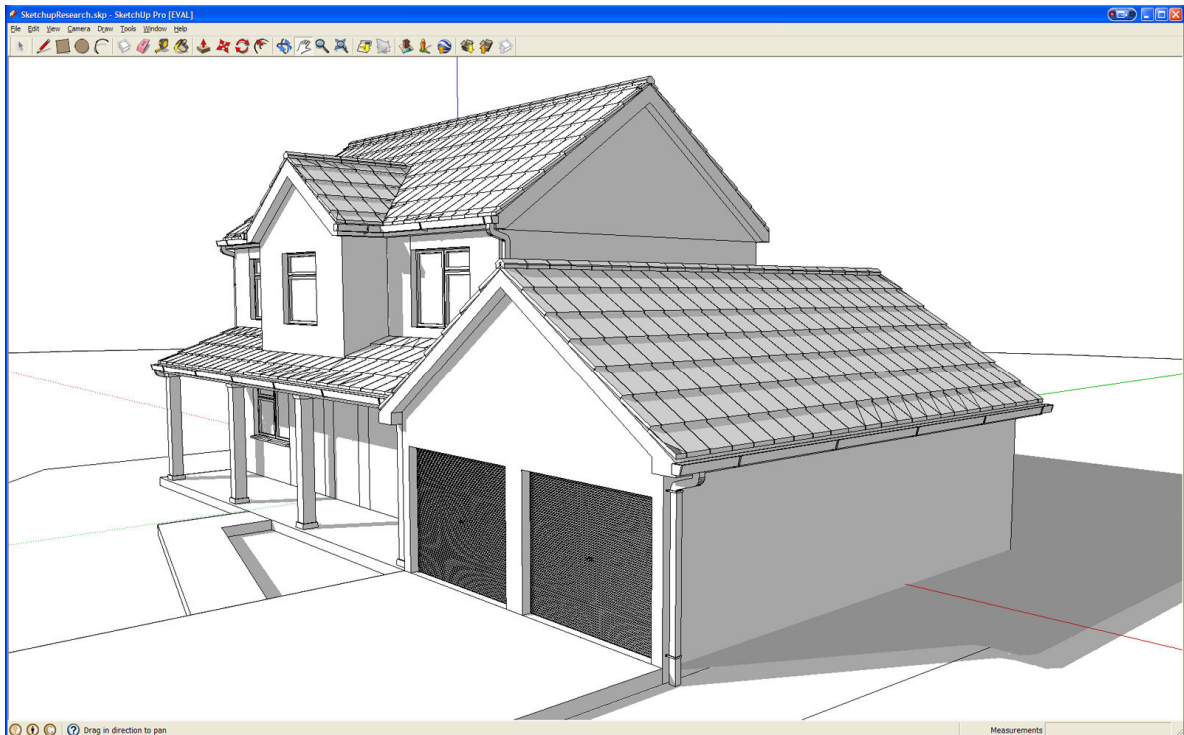


Figure 3-1 3D model completed in Google SketchUp

Tiling a roof in Google SketchUp can be accomplished when using V-Ray as renderer through using the displace option on texture maps. The displacement method is often used in 3D Studio Max and V-Ray, this can be particularly useful for brickwork textures. A texture is used in the diffuse slot which gives the material colour, and an additional texture is used in the displace slot as well as sometimes the bump slot. The textures used for displace and bump are generally greyscale and have a light tone where the desire is for the object to extend away from the geometry, and a dark spot where the object does not extend. For example, if displacement were used with brickwork, the grout area on the displacement map would be dark shades of grey and the bricks would be light shades of grey. See Figure 3-2 for an example of a brickwork diffuse texture and Figure 3-3 for an example of a brickwork bump or displacement map.



Figure 3-2 Brickwork Texture to be used in the Diffuse element of a material

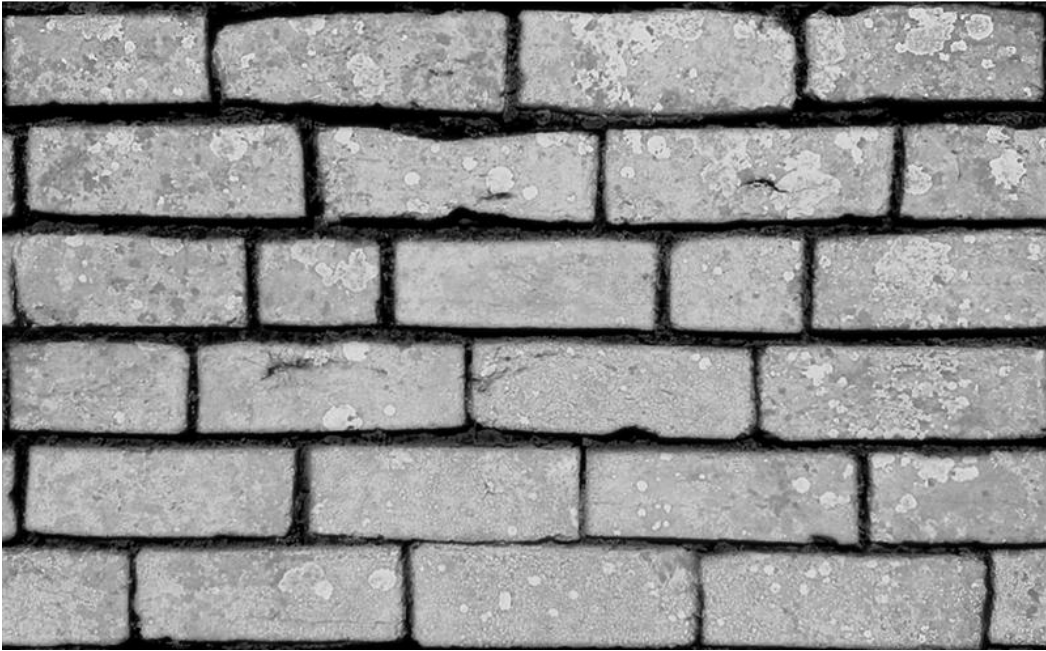


Figure 3-3 Brickwork Texture to be used in the Bump or Displacement element of a material

Unfortunately there is no native align tool for creating roof tile geometry in Google SketchUp. A plug-in can be downloaded which will mimic the align tool in 3D Studio Max which is an essential part of the modelling process when creating roof tiles. In Google SketchUp the object (roof tile geometry) must be rotated to the correct angle, before being moved to the surface. The rotate tool is however intuitive and easy to use, it comes in the form of a protractor and once the reference angle has been set up, the object can be rotated to a given point. In Google SketchUp the automated option for snapping to other objects is turned on by default and seems to be well implemented.

### **3.2.2. Gutters**

The 'Follow Me' tool can be used to create the gutter geometry. This tool can be compared to the loft tool which appears in 3D Studio Max. The object which is extruded needs to be touching the path for the extrusion, this is perceived as an unnecessary step, in 3D Studio Max, the object to be extruded can be located elsewhere in the scene.

### **3.2.3. Windows / Doors**

To create window geometry in SketchUp, the 'Offset' and 'Push/Pull' tool are used. These tools work similarly to the 'Inset' and 'Extrude' tool in 3D Studio Max. Although both these tools are apparently friendly, there are no options available to set a specific 'Offset' value, so when modelling windows or doors, the user would need to snap to a template 2D drawing to get the exact width of window frame.

### **3.2.4. Lighting**

VRay is currently thought of as the industry standard for visualisation specialists. In recent years, VRay has become available as a rendering engine plug-in for SketchUp therefore VRay materials and lights can be used within SketchUp. In SketchUp, VRay options mimic those available in 3D Studio Max, therefore the options are just as complex as those in 3D Studio Max. With this in mind, the process to get an accurate and rapid rendering scene requires substantial knowledge of the VRay rendering engine.

### **3.2.5. Texturing**

Texturing in Google SketchUp has fewer steps than required in 3D Studio Max. Once again, the options for texturing in 3D Studio Max are extensive because the software is targeted at multiple disciplines. One important point in Google SketchUp is that when importing a texture, the texture can instantly be scaled and adjusted to fit the object as necessary. In 3D Studio Max, a 'UVW Map' modifier is required to perform this function; this adds an extra step and complexity to the operation.

### **3.2.6. Rendering**

Rendering can use the V-Ray rendering engine, so once the V-Ray settings have been created for a scene, as with 3D Studio Max, these settings can be saved to a preset file and used to save time in the future.

### **3.2.7. Overview**

SketchUp has proved over recent years to be an invaluable tool for architects. Where the software excels is in concept modelling. The aim of Google SketchUp is to be easy to use for both the construction industry and general public, and it accomplishes this. Google SketchUp does have options to write and run scripts which can be written in a language called Ruby, therefore Google SketchUp could be used for this project. However, the aim of this Masters by Research project is to focus on detailing models, rather than high speed, low polygon 3D modelling. Google SketchUp does not give the user the option to input exact co-ordinates when creating geometry and drawing profiles which are generally required for architectural visualisations, with this in mind, Google SketchUp was not considered the ideal tool to base the project around.

### **3.3. AutoCAD**

#### **3.3.1. Tiling**

AutoCAD has developed over many years from a 2D design tool which was operated using command line codes, to a modern day 3D modelling tool. However, this transition seems to have evolved into a 3D modelling process which is not straight forward. The 'extrude' option within CAD is a good example of this. A problem many first time users encounter is that of aligning an object before extruding it, unfortunately this is required to create a roof tile from a profile.

There is currently not an option to apply a displacement material to a flat face in AutoCAD, therefore a displacement map for roof tiles would not be possible.

#### **3.3.2. Gutters**

The extrude command is used in AutoCAD to effectively 'Lof' a profile. The function lets you select a path to extrude your profile along, but as mentioned previously the user must ensure that the shape is aligned to the path before lofting.

#### **3.3.3. Windows/Doors**

The process of modelling in AutoCAD differs to that used in Google SketchUp. There are no tools available to offset a face so the geometry for the windows would need to be created using a number of boxes and Boolean commands i.e. subtract, union, etc. The Boolean commands tend to create unclean geometry in 3D Studio Max and therefore are generally the last resort option for 3D modellers. However, in AutoCAD, the geometry created using Boolean commands are used frequently without issues.

#### **3.3.4. Lighting**

Several years ago 3D Studio Max was purchased by AutoDesk from Discreet. In recent years, rendering in AutoCAD has become similar to rendering in 3D Studio Max. As 3D Studio Max

opted to default for the Mental Ray rendering engine, Mental Ray was included in AutoCAD as the only rendering option. Knowledge of rendering using Mental Ray makes it easier for a visualisation specialist to work between these items of software which is required in the working environment. The Mental Ray daylight system is generally a solution which can create simple and accurate daylight renders without making substantial changes to the settings. A common problem when using this lighting setup on default settings is that a scene rendered using Mental Ray tends to de-saturate the colours of textures and materials. This is also a common occurrence in 3D Studio Max, however this can be fixed by setting image gamma to 2.2.

### **3.3.5. Texturing**

In recent versions of AutoCAD, the material browser is in a similar style to 3D Studio Max. To adjust the position of a texture, the user must alter the UVW coordinates. Adjusting the UVW co-ordinates in AutoCAD is considerably more complex than with Google SketchUp and 3D Studio Max. The coordinates of the textures cannot be adjusted with the mouse in the viewport, they need to be adjusted using a menu which can make it difficult to gain the desired effect.

### **3.3.6. Rendering**

Rendering in AutoCAD is a simple process as the settings for Mental Ray generally give a good, 'out of the box' daylight render, see Figure 3-4.

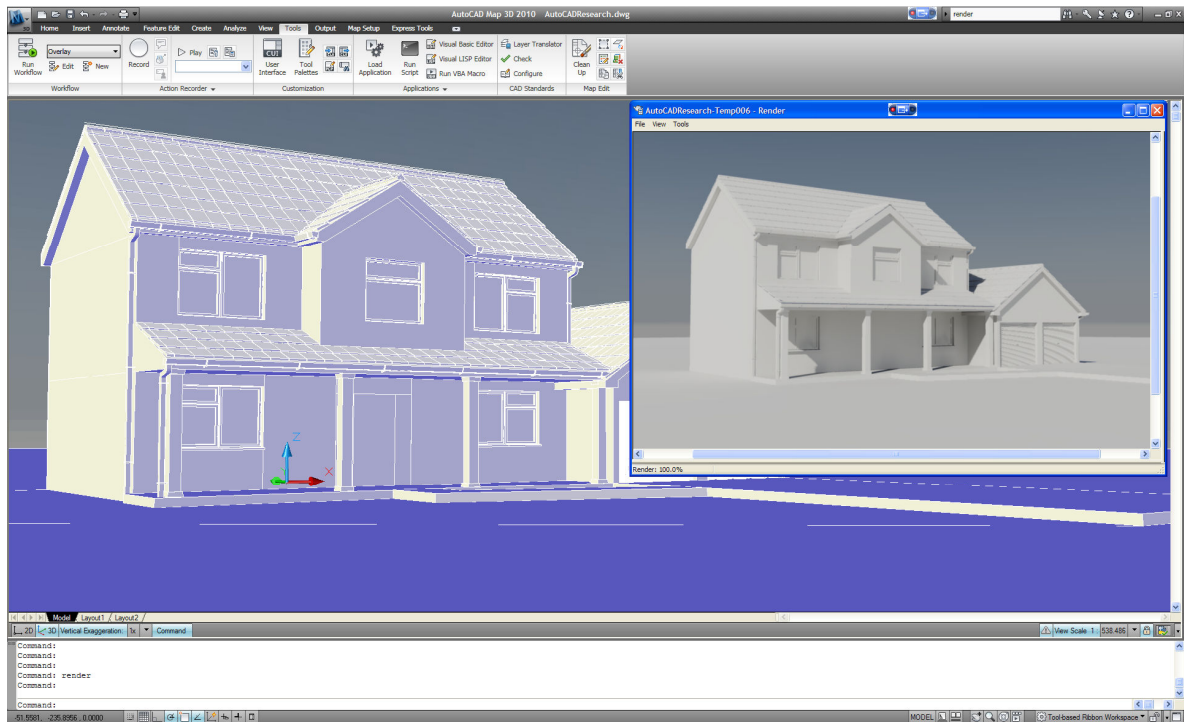


Figure 3-4 3D model completed in AutoDesk AutoCAD

### 3.3.7. Overview

AutoCAD is designed to be an extremely accurate design tool. Unlike 3D Studio Max and Google SketchUp, AutoCAD does not store coordinates using single precision floating point numbers. This means that the model can be far from the origin (0,0) without incurring rounding off errors problems. When working on scenes in 3D Studio Max, it is advised to move the model to a local coordinate system before starting work. The rounding off errors which occur in 3D Studio Max and SketchUp are well documented and the result is the inability to snap onto objects if they are located far away from the origin.

Where Google SketchUp has created an effective user interface with basic options, AutoCAD is the opposite, some of the options are hidden within settings which need to be called using the 'command line' and are not readily available.



In recent years Autodesk have incorporated the 'ribbon' into AutoCAD and 3D Studio Max, the ribbon has been included so that these products are in line with current Microsoft products including Excel and Word. The ribbon provides tool buttons in a thick bar along the top of the screen with all functions in one place and negates the need for traditional menu items. Unfortunately, there are many options available in AutoCAD and often a ribbon can become extremely complicated and cluttered. Many experienced users who have attempted to adopt the 'ribbon' have found themselves using some commands on the ribbon and some using traditional command prompt commands. Many AutoCAD professionals have not embraced the ribbon and continue to work in the traditional interface.

### 3.4. 3D Studio Max

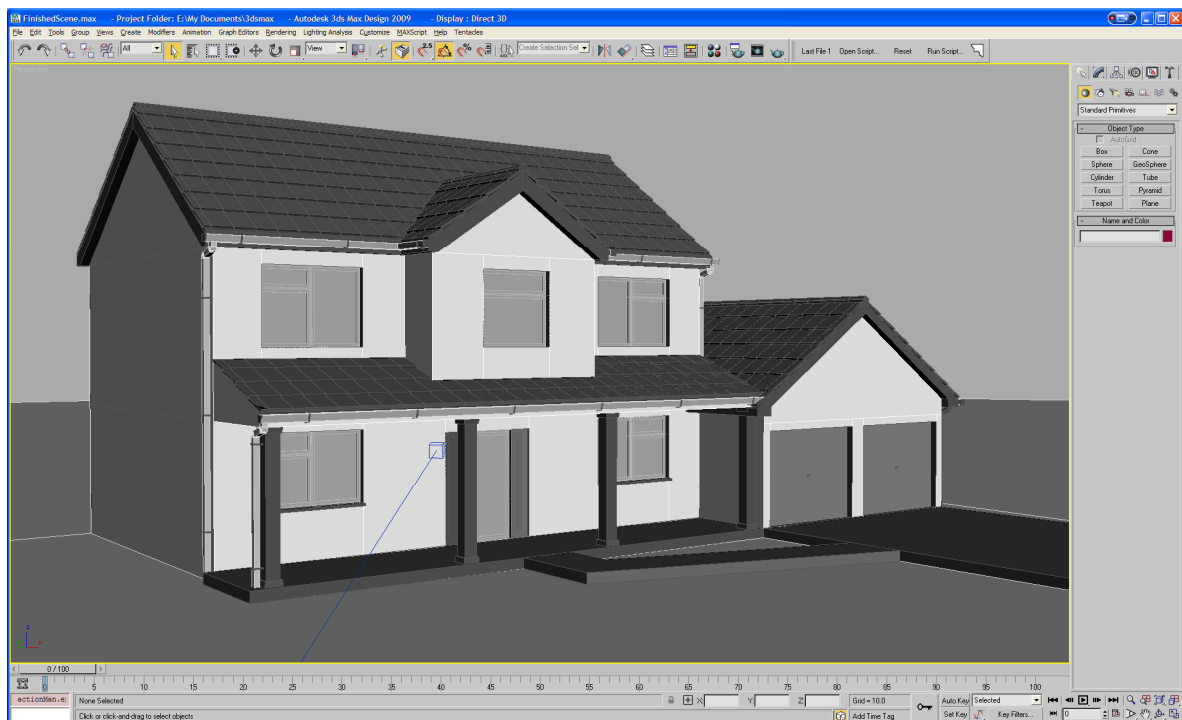


Figure 3-5 3D model completed in 3D Studio Max

### 3.4.1. Tiling

Creating roof tiles in 3D Studio is a laborious task and would benefit from some other features which could aid the process. Generally, the geometry for the individual roof tile would be created by first modelling using 2D lines (spline modelling) then the lines would be extruded a given distance to create the tile object see Figure 3-6. The pivot point of the individual tile would be set to the extents of the tile and the tile would be rotated (tilted slightly) so that the individual tiles can rest on top each other as they would on a roof. An array would next be used to create the instances of the tile with an offset supplied by the user.

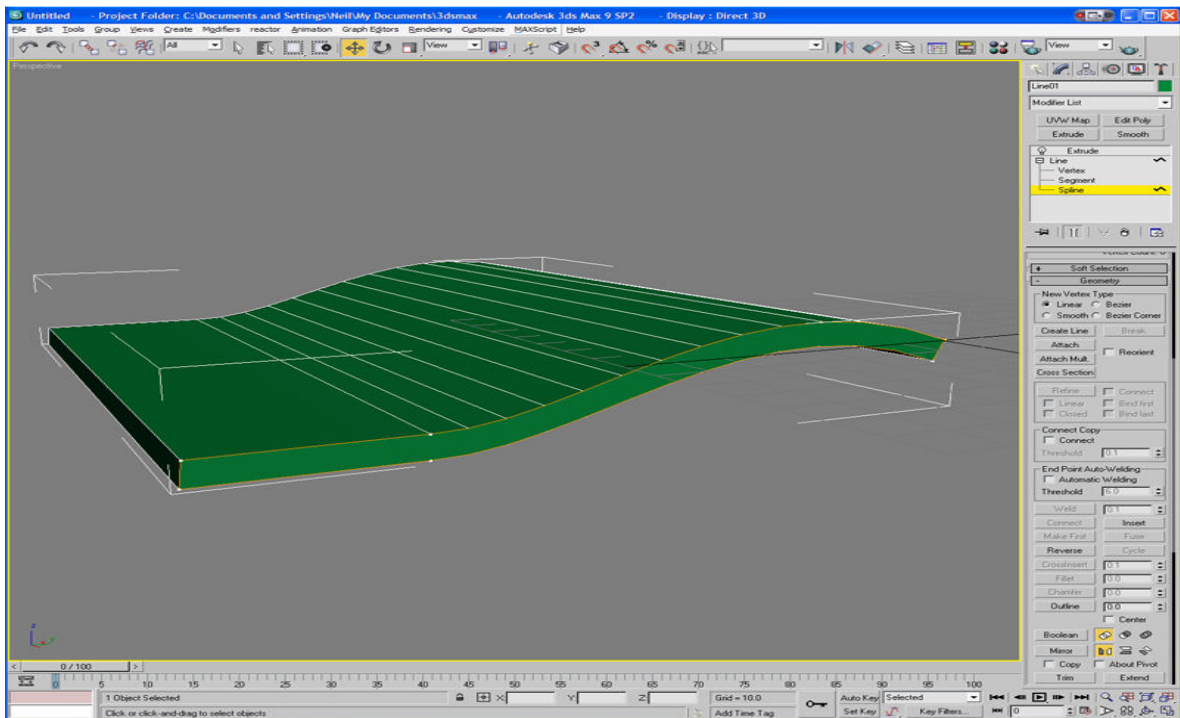


Figure 3-6 Example of individual 3D tile geometry

The tiles would then need to be joined to make one object, aligned to the surface using the align tool, and then trimmed to fit the exact area of the roof using the slice modifier. When creating a tiled roof this way, if the size or geometry of the tile needs to be altered, this laborious process must be repeated.

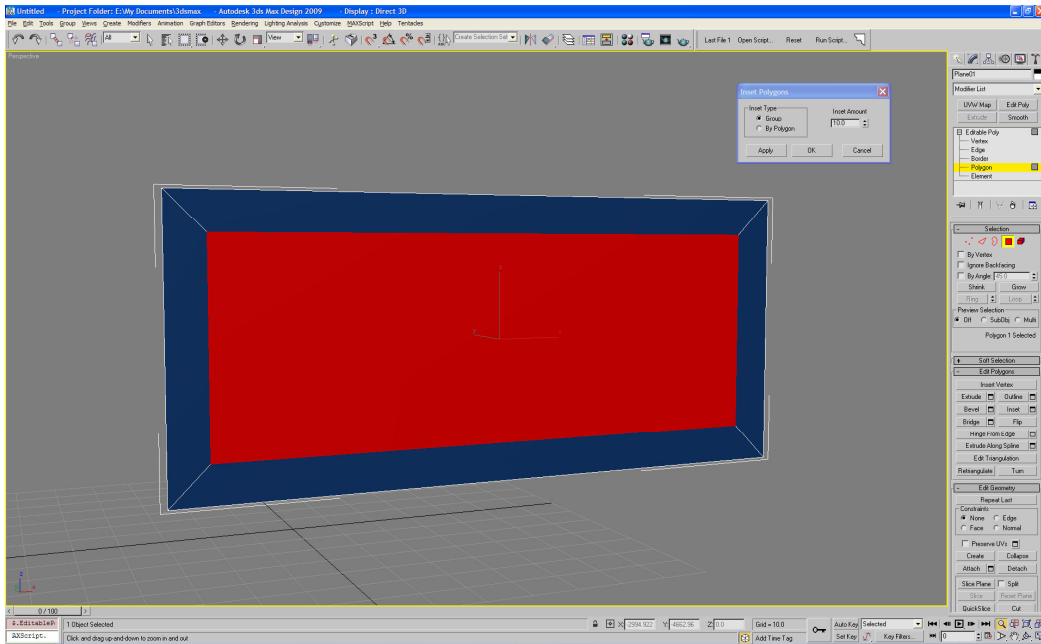


Figure 3-7 Example of the inset command in 3D Studio Max

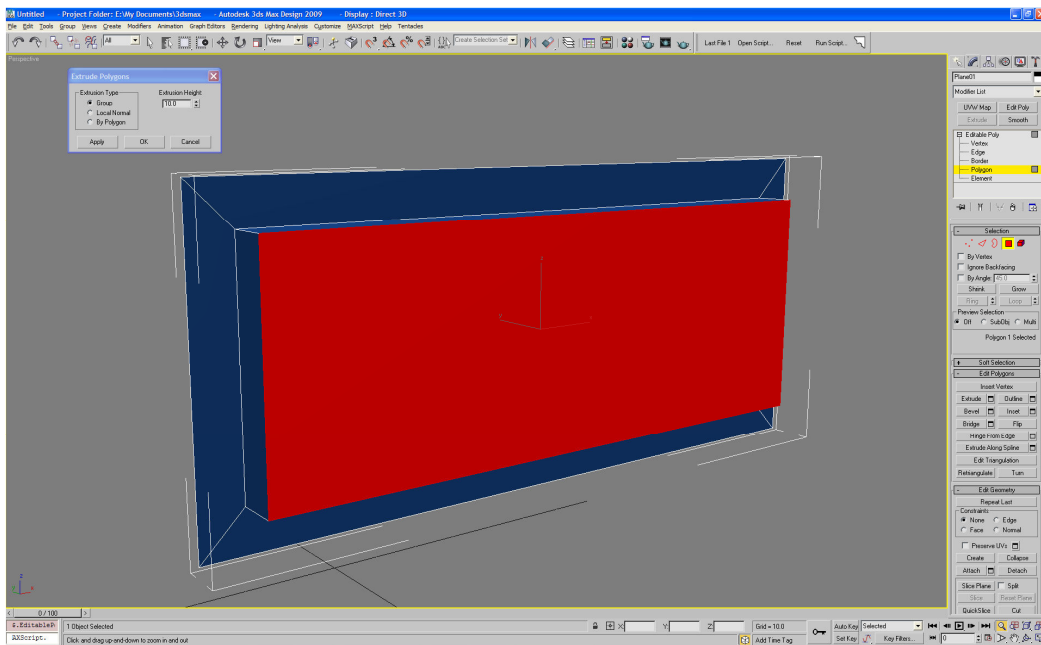


Figure 3-8 Example of extrude command in 3D Studio Max

### **3.4.2. Gutters**

In 3D Studio Max, the process of modelling the gutters involves using a profile shape and the loft tool. Unlike AutoCAD, the profile does not need to be aligned to the end of the spline in order to enable the loft. One other useful feature is that when using the loft command, the resulting geometry is still linked to the profile. For example, if once the gutters had been modelled, then the client requested a slight amendment to the gutter profile, the user could adjust the profile using the editable spline commands and all geometry linked to that profile would then be updated. The spacing tool would be used to align the support brackets to the gutters and the gutter downpipes would be modelled in the same way as the main gutters, by lofting a profile and spacing the brackets.

### **3.4.3. Windows/Doors**

To model the windows and doors in 3D Studio Max, the user is required to use the inset and extrude command. The inset command creates a smaller polygon within the face of an existing polygon, the new polygon is highlighted in Red in Figure 3-7. The extrude command will create a new polygon which is pushed or pulled from the existing polygon as shown in Figure 3-8.

### **3.4.4. Lighting**

The lighting setup process can be somewhat involved when using V-Ray and 3D Studio Max. Global Illumination is used to create a more realistic lighting setup in a 3D environment. The term Global Illumination covers a set of algorithms which are used to calculate the light intensity and colour in different parts of the scene. The calculation will generally take into account the light emitted from the light source as well as the colour of objects from within the scene to give a more realistic lighting solution.

### **3.4.5. Texturing**

Texturing in 3D Studio Max for architectural visualisations has become easier in recent years as the software has adopted real world mapping as default. This function allows the user to set real

world co-ordinates to their textures. This results in the user only needing to set the correct size of their texture once and any new geometry created in their scene which has this texture applied will show the texture at the correct size. Previous to this introduction, a 'UVW Map' modifier needed to be added to the geometry and adjusted using a scaling control until the map reached the correct size.

#### **3.4.6. Rendering**

There are various settings which can make rendering take significantly longer in V-Ray and 3D Studio Max. Anti-aliasing is the process which minimises distortion of pixels in the final render by analysing and interpolating data from nearby pixels. Various options for anti-aliasing are available in V-Ray and altering the settings to give a high quality final render can increase render times exponentially.

#### **3.4.7. Overview**

The process involved when creating high quality architectural visualisations using 3D Studio Max and V-Ray requires expert knowledge and is time consuming. As detailed previously, the elements which require most knowledge are the lighting and rendering functions. The modelling tasks can be learnt by following short tutorials which are easily accessible on the internet. However, mastering lighting and rendering in V-Ray can take many years of experience.

### **3.5. Roof Designer - Real World Example**

This section will analyse a real world example of a program which is currently available and has aimed at solving part of the brief of this project.

Roof designer is a plug-in which assists when creating roofs in 3D Studio Max. The plug-in creates 3D Geometry for the tiles and also provides tools to assist in texturing the tiles.

### 3.5.1. Process

Standard construction using Roof Designer by Batzal Software requires a flat polygon as a starting point.

It is suggested in the instruction manual for the software, that a plane should be drawn over the proposed pitch and outline of the roof in plan view. Following this, the plane should be converted to an editable poly in 3D Studio Max. An editable poly applies a set of tools to the geometry and these tools can be used to edit the geometry. The polygons for each face of the roof should be cut out of the plane using the 'cut' tool. After deleting the polygon which is left, the user would be in possession of a geometry defining a roof shape, however it would be flat. At this stage the Batzal Roof Designer Modifier would be added.

The slopes and angles required for the roof are edited from within the Roof Designer Modifier. In a working environment, this could cause problems when attempting to get the final model to relate perfectly back to the AutoCAD drawings provided by the architects. For example, the angles required for the slope of the roof will need to be calculated from within AutoCAD. There is a more effective option provided by Batzal which enables the user to use roof geometry which has already been modelled.

There are options to add a random material ID to the tiles which can be used to reduce the 'texture tiling' effect. In order to produce photorealistic images, this process is almost always a requirement.

Preset tiles are included with the software; however it is also possible for the user to include their own custom tile geometry.

There is a method included with the software which semi-automatically creates a spline to use for 'lofting' the guttering. Elements of this method would be useful to include as part of this project to save time for the modeller.

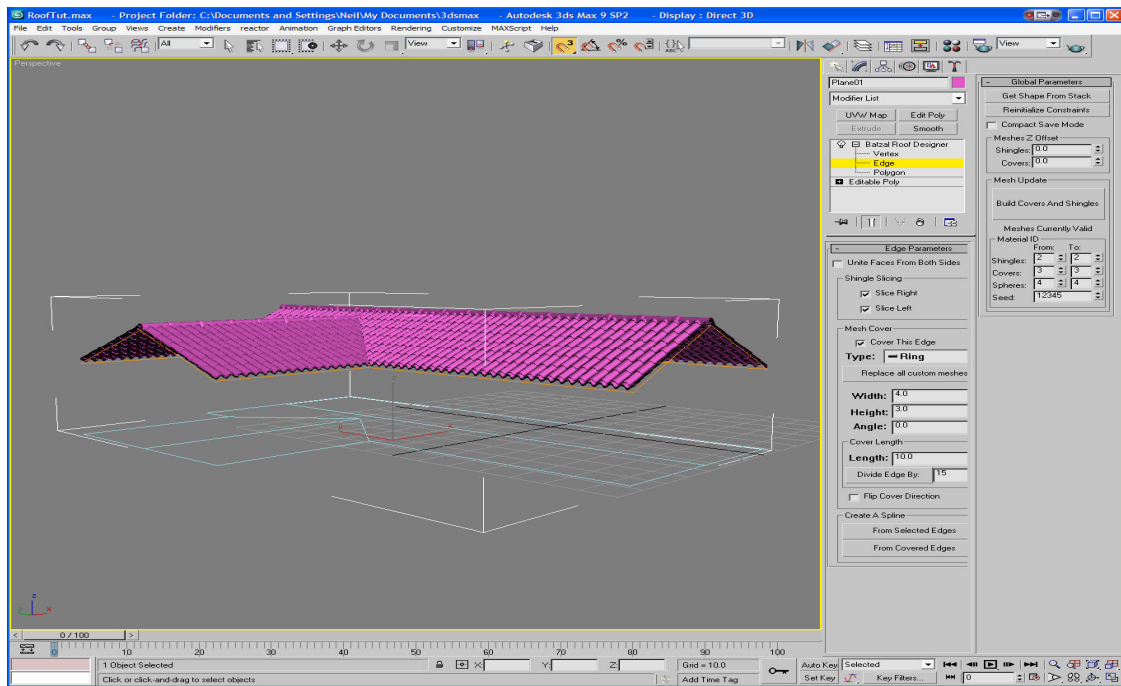


Figure 3-9 Completed tiled roof using Batzal Roof Designer in 3D Studio Max

### **3.6. Conclusions**

The software solutions which are available for producing the architectural visualisations analysed as part of this study showed that there is certainly scope for streamlining these processes. The methods used by 'Batzal Software' have been well implemented, and elements of the functions within 'Roof Designer' should be provided in the software produced for this project.

The study has shown that the processes involved are relatively similar between the software packages. The roof tiling procedure consists of a repetitive and not always straightforward task. The alignment of the roof tile geometry to the roof could cause problems to an inexperienced user, automating this process would be beneficial.

The processes involved to model the gutters is also similar between the software packages, there are elements of this which would always require some user input, for example, the user would need to set the spacing between the gutter brackets.

The method for creating windows across all packages is repetitive and time consuming. The user would find themselves repeating tasks on each individual window frame, this would certainly benefit from a semi-automatic procedure to aid the modelling process.

The complexity of lighting a scene differed considerably between the software packages evaluated. When V-Ray is used, a user without prior knowledge would generally be overcome by the complexity of the settings, if an experienced user were to create preset settings for different lighting conditions, this would be beneficial to an inexperienced user. When performing the study, it was noted that even though the actual processes are slightly different, the time taken to complete the processes was similar. With this in mind, it would highlight that a productivity improvement tool is likely to be well received for any of the software packages involved. A short evaluation was completed to assess the time required to complete the tasks in each software package, the Results are shown in Table 3.1. These results show as expected, that Batzal Roof Designer performed the roof tiling task faster than traditional methods. The time



difference required between Google SketchUp, AutoCAD and 3D Studio Max was comparable for all results. The general results showed that Google SketchUp was faster, with AutoCAD requiring the longest time.

	Google SketchUp	AutoCAD	3D Studio Max	Batzal Roof Designer
Tiling	05:04	07:16	05:21	00:47
Gutters	04:40	06:12	05:04	N/A
Windows	03:06	06:39	05:11	N/A
Doors	04:28	05:28	04:58	N/A
Lighting	05:58	05:16	05:48	N/A
<b>Total</b>	<b>23:16</b>	<b>30:11</b>	<b>26:22</b>	<b>00:47</b>

Table 3.1 Table to show the time taken to perform modelling tasks using evaluated software

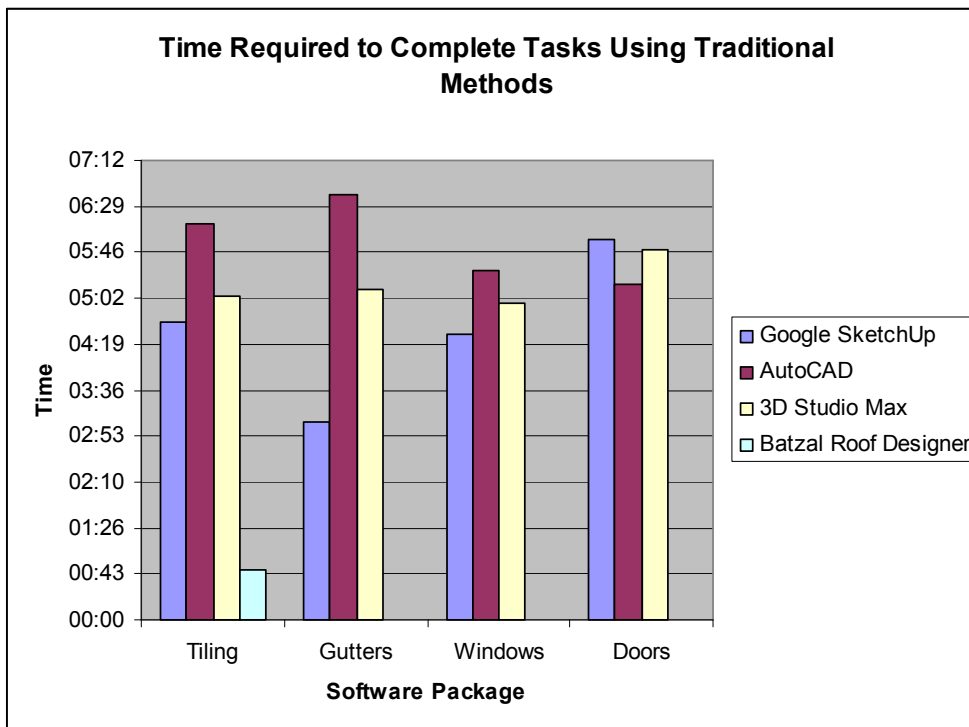


Figure 3-10 Time Required to Complete Tasks Using Traditional Methods

The research conducted on these methods has highlighted that the most important part of this process is lighting and rendering. Because these elements take the longest time to learn, if the time taken to complete these tasks is reduced, the goals of this project have been largely achieved.

### **3.7. Requirements**

As many design offices already have access to 3D Studio Max software, and this software is currently the industry standard for architectural visualisation, it is considered that the most effective solution would be to create a plug-in for 3D Studio Max. The plug-in is to be created using MAXScript which is a high level programming language which runs aside 3D Studio Max.

Based on the findings of the research, the program will have the following requirements.

1. A set of functions which will provide tools to assist the modelling and texturing of windows, doors, gutters roof tiles.
2. The tools provided should include semi-automated methods for constructing geometry and adjusting material or rendering settings.
3. The program should make the lighting and rendering operation considerably easier to complete.
4. The program should have a graphical user interface which is user friendly.
5. The tools built into the program should not have unnecessary options or user input requirements. This is especially important for the V-Ray settings

## **4. Implementation**

### **4.1. Program Implementation**

As detailed previously, MAXScript is a high level programming language which runs over 3D Studio Max. MAXScript is object-oriented, event-driven and expression based (Lama et al., 2007). MAXScript will allow the user to interact with a 3D Studio Max scene, control the actions which are performed automatically with script statements and operations (Lama et al., 2007). The language is able to call functions from within 3D Studio Max and parse parameters to these functions. It is also able to define it's own functions which it can use from within the MAXScript or from another MAXScript.

A MAXScript editor is provided from within 3D Studio Max, however the scripts can also be opened and edited from notepad or any text editor.

The most common format for a MAXScript is a .ms file, other files which are common are .mce files which are encrypted scripts.

A Macro recorder is available which will record the MAXScript code which relates to any modelling or rendering functions which the user performs in a traditional manner. This recording can assist the user with the programming. A MAXScript listener window is also available which shows a command line interface, this can be used in real time by allowing the user to query variables or perform functions on objects or values.

Because MAXScript is able to call existing functionality from within 3D Studio Max, it provides a platform to streamline specific tasks, increase productivity and decrease complexity for the user. Some but not all existing tools are available to call from within MAXScript, for example, the spacing tool is not available, this is discussed further in 4.1.1 Approach and Methods.

#### 4.1.1. Approach and Methods

The Use-case diagram shown in Figure 4-1 illustrates the user interaction with the functions which have been included with the MAXScript ATK. As shown, the functions are not dependent on each other and can be accessed individually, in a different order or some may not be used at all.

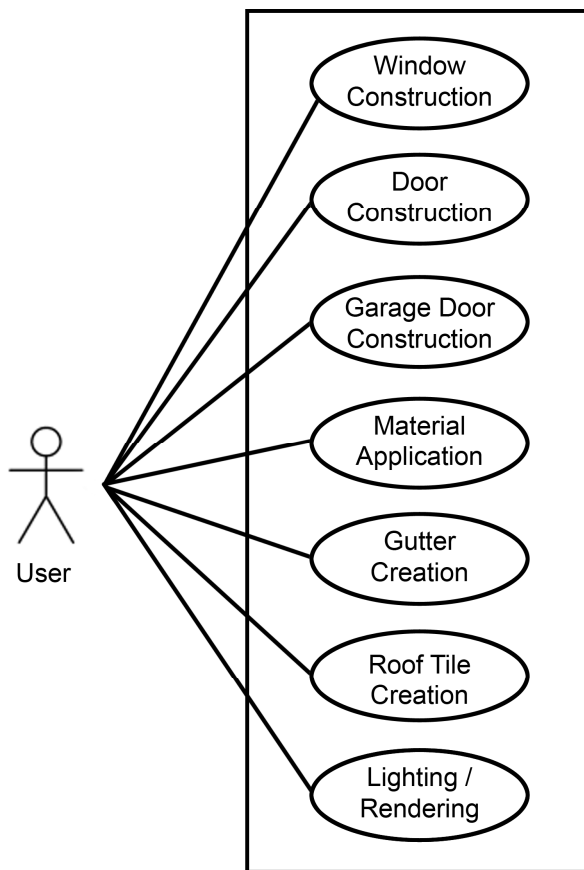


Figure 4-1 Use-case diagram illustrating MAXScript ATK

A manual process is required before running the script which involves modelling basic shapes for the building. Many drawings created within AutoCAD are badly organised i.e. the objects are incorrectly assigned to Layers. This hinders the modelling process in 3D Studio Max so it is good practice to ensure a clean up step is completed on AutoCAD drawings before bringing them in to 3D Studio Max.

The common process for creating the basic building shape required for MAXScript ATK in a visualisation office is as follows, this work would be completed in AutoCAD.

1. Draw new lines around key features which would be required to create the basic model of the building. If the object is to be a surface, i.e. a driveway, this line would need to be closed i.e. the first and last

vertex on the line would touch each other exactly. The user must also ensure the line work is on a layer which has a descriptive title, for example '00\_WindowLinework' would be an appropriate name for any lines which represent the windows.

2. Import only the relevant line work into 3D Studio Max.
3. Use the line work and extrude and manipulate to create 3D shapes.

#### 4.1.2. Windows

When modelling windows, MAXScript ATK requires all the windows to be one entity. During the building modelling process of 3D Studio Max, it is a simple process to ensure this is the case.

Once the 3D shapes are created, the 'connect' edges tool is used to create the polygons which would represent the windows in the construction. At this stage, all the windows are to be selected, and the detach button is pressed which will separate the window polygons from the

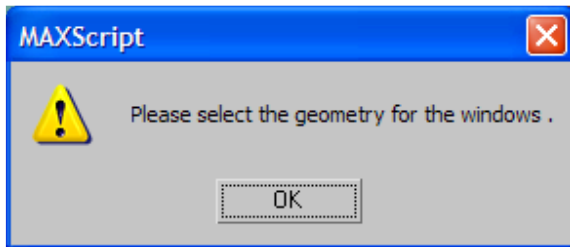


Figure 4-2 Error handling

rest of the building, it is this shape which is the starting point for windows in MAXScript ATK. At this stage, the settings for the windows are tailored by the user.

When the 'Model Windows' button is pressed, some error handling code is written in, consequently the program first checks to see if there is any geometry selected. If there is no geometry selected, a popup window is displayed, see Figure 4-2 and Source Code 4.1.

Once the program has confirmed that the user has the geometry selected, MAXScript ATK reads in the values which are set by the user, values are entered into what are referred to as 'spinners' in 3D Studio Max.

The script now performs an operation which splits the windows into individual polygons, each polygon is assigned to an array and the program performs a number of 'Inset' and 'Extrude' functions based on the user inputs to create the window geometry. Whilst the 'Inset' and 'Extrude' functions are taking place, an ID is applied to the currently selected polygon which is used to ensure that the correct material is applied to each piece of geometry, Source Code 4.2 shows the loop routine used to model the windows.

```
if selection.count == 0 -- is anything selected?  
then messagebox "Please select the geometry for the windows ." -- display message box
```

Source Code 4.1 Error Handling

```

for i = 1 to ATKWindowArray.count do -- perform the loop on all windows within the array
(
  select ATKWindowArray [i] -- select the current window
  convertToPoly $ -- convert to an editable poly
  max modify mode -- ensure modify mode is enabled
  subobjectLevel = 4 -- polygon sub object mode
  actionMan.executeAction 0 "40021" -- select all
  $.EditablePoly.setMaterialIndex 11 1 -- set the material ID
  $.faceExtrudeHeight = offset2 -- set the offset value 2
  $.EditablePoly.buttonOp #Extrude -- perform the extrusion
  $.EditablePoly.setMaterialIndex 1 1 -- set faces to ID 1
  setcommandPanelTaskmode #modify -- move to modify mode
  subobjectLevel = 4 -- go to polygon sub object mode
  actionMan.executeAction 0 "40021" -- Select All
  $.insetAmount = spn_FrameWidth.value -- inset the frame by the amount spinner
  $.insetType = 1 -- set inset mode to individual faces
  $.EditablePoly.buttonOp #Inset -- apply the inset
  $.faceExtrudeHeight = spn_Foffset.value -- set frame thickness
  $.extrusionType = 1 -- set extrusion type to face normals
  $.EditablePoly.buttonOp #Extrude -- apply the extrusion
  $.insetAmount = spn_FrameWidth2.value -- frame inset
  $.insetType = 1 -- set extrusion type to individual faces
  $.EditablePoly.buttonOp #Inset -- apply the inset
  $.faceExtrudeHeight = -(spn_FrameReturn.value) -- extrude the faces back into the window
  $.extrusionType = 1 -- set extrusion type to individual faces
  $.EditablePoly.buttonOp #Extrude -- apply the extrude
  $.EditablePoly.setMaterialIndex 3 1 -- give all the faces currently selected material index of 3
  subobjectLevel = 0 -- return to top-level edit mode
  $.material = meditMaterials[1] -- apply the material from slot one to the window
  $.name = uniquename "ATK_Window_01" -- rename
  clearselection() -- clear the selection
) -- end window loop

```

Source Code 4.2 Window Construction Code

A common problem when modelling for photo realistic visualisations is that the edges of objects tend to look very sharp. There are options within materials to round edges, however this is generally not a practical solution as architectural visualisation specialists would only require certain edges to be rounded, i.e. the vertical edge of a building would need to be rounded

slightly, however, the edge of the building where the wall meets the soffit would not need to be rounded. As the material makes all edges rounded, this option is not appropriate.

It was considered that it would be appropriate that an option be built into the window function which would select only the edges to be rounded and apply the chamfer tool to these edges, see Figure 4-3. The 'Soften Edges' tool works by selecting only the relevant edges based on assigned IDs which were set as the windows were being modelled. The chamfer tool is then performed only on these edges.

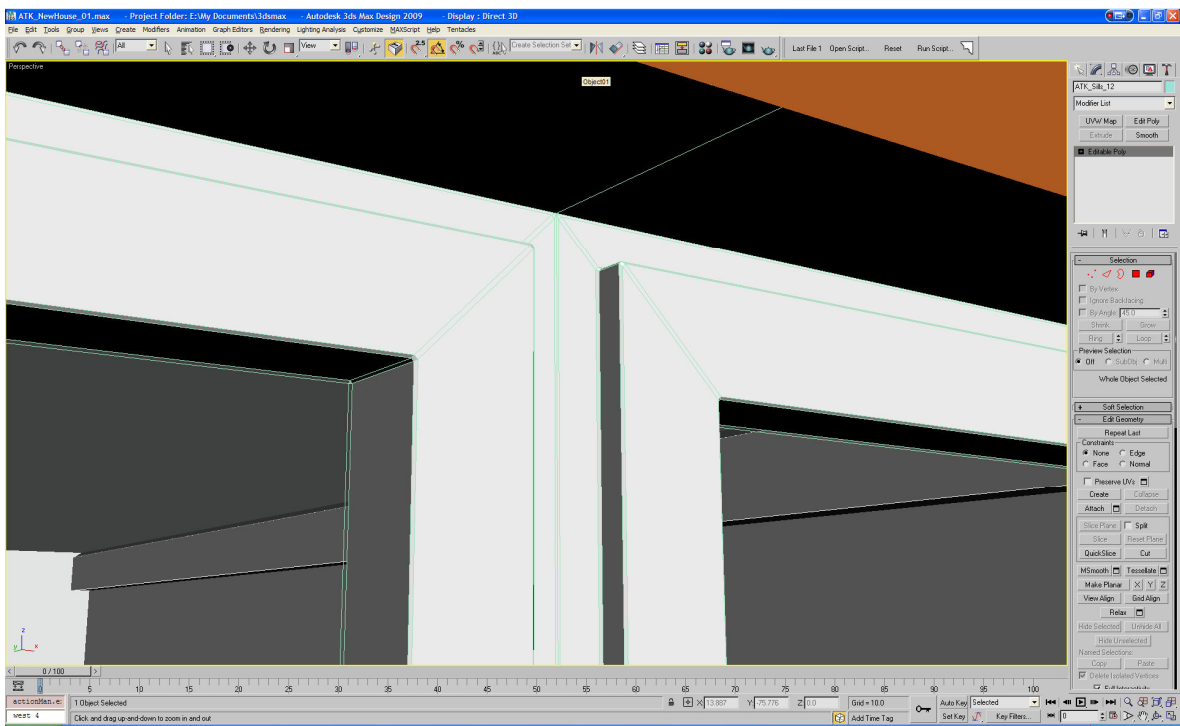


Figure 4-3 Showing how the chamfer tool was used to soften the edges of geometry

When the MAXScript ATK script is initially run, a set of Multi-Sub object materials are created automatically. A multi-sub object material is a material which is applied to one entity in 3D Studio Max but can contain many different materials. Each individual face or polygon within the object can be assigned an ID, and the IDs are matched to the corresponding ID within the Multi-sub object material. This type of material is perfect for producing windows as the window



geometry can stay as one entity which helps keep the scene organised and easier to find objects in the future. In this case only 2 different materials were required, a V-Ray material to represent the glass, and a white plastic material to represent the frame of the window. The program could easily be adjusted so that a material with a wooden texture could be used for the window frames.

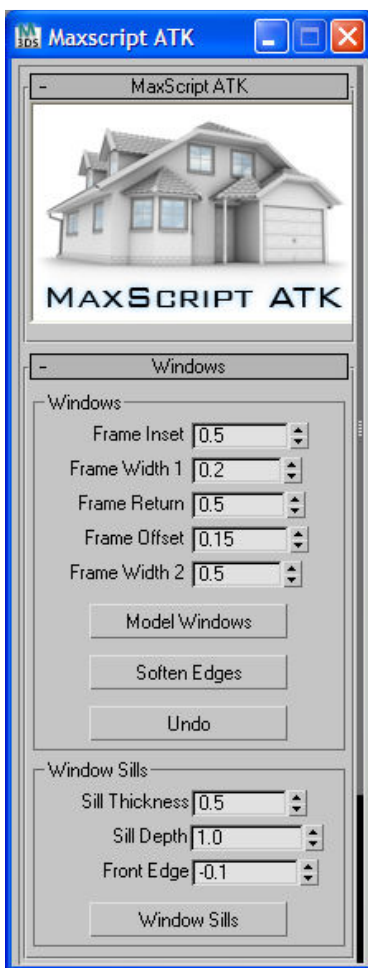


Figure 4-4 Dialog box for window constructions

the windows are modelled (the hold function will save a temporary file). An Undo button has been included so the user can go back and alter settings i.e. Increase the frame return (wall thickness) before pressing the 'Model Windows' button again.

The material used to represent glass on the windows is generally required to be more reflective than it would generally be in real life, Fresnel reflections are commonly used on glass in 3D Studio Max. Fresnel reflections are used to determine the amount of reflection based on the viewing angle. For example, when looking at a snooker ball in real life the viewer would see more reflections on the side of the ball rather than on the part of the ball which is closest to the human eye. After many years of experience as a visualisation specialist, it has been found that to get the desired effect on exterior architectural visualisations, Fresnel reflections will need to be turned off for the glass windows. This can be partly due to the fact that the majority of the time, although the exterior of a structure has been designed to a standard suitable for visualisation, the interior has not. Modelling the interior of buildings could double or triple the modelling time required for the visualisation, so more prominent reflections on exterior windows can help screen this.

As part of the program, the 3D Studio Max 'hold' function is incorporated which records the state of the scene before

A tool has also been included which models the window sill. This is a simple tool which has options for sill thickness, depth and front edge. Altering the value for the front edge will define the angle of slope for the window sill.

### 4.1.3. Doors

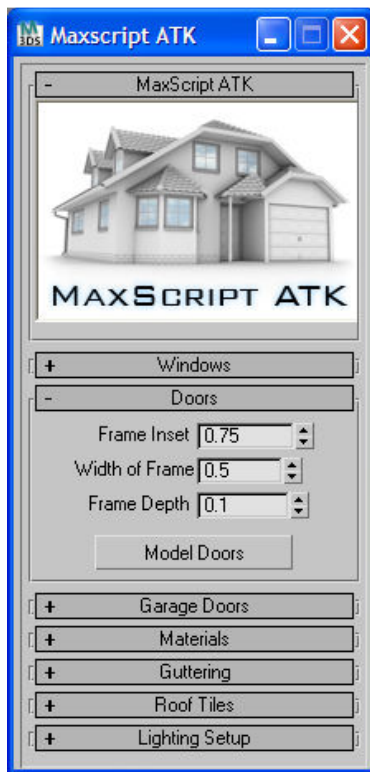


Figure 4-5 Dialog box for door construction

is pressed.

Many modern roller doors are powered by electric motors therefore a door handle is not always required, a separate option is included for if the user requires a door handle.

The function included in the program which creates the doors is similar to the function used to create the windows, see Figure 4-5. User inputs are required and the 'Inset' and 'Extrude' commands are called from within the MAXScript to create the geometry.

Door handles are also available to add to the door geometry, see Figure 4-7.

### 4.1.4. Garage Doors

A function for detailing garage doors has been included in the program. A separate set of user inputs is requested, although default options are already entered. To save the user time, a set of default options for different types of garage doors has been included, a roller door for a garage often has a large number of horizontal sections, so the 'Horizontal No.' reflects this when the 'Roller Door' button

The door handle section of the program imports the relevant door handle from an external file, see Figure 4-8. The handles which are housed in this external 3D Studio Max file are named in a way so that they can be imported into the main working file which is being used for the visualisation. An example of the naming convention used is as follows. '02\_DH\_03' – This reference refers to geometry reference 02, door handle DH and this particular piece of geometry is object number 03. The program is written so that if a user would like to add to the external Max file 'ATK\_DoorHandles.max' they can do this providing their new geometry for the door handle is named sequentially i.e. '02\_DH\_04'. This door handle would then be available to use in the MAXScript ATK program. In this case the new door handle could be included in the scene by adjusting the 'Door Handle Ref:' spinner then clicking 'Add Door Handle' As with other functions, an 'Undo' button is included if refinements are required.

The code shown in Source Code 4.3 was used to action the door handle element of the program. An align tool is not available to call within MAXScript therefore a new function was required to align objects, the function is detailed in Source Code 4.4 and works by creating a vector based on the normal of the source polygon, and aligning the vector to the normal of the target polygon. This function is also used to align the roof tiles to the roof geometry.

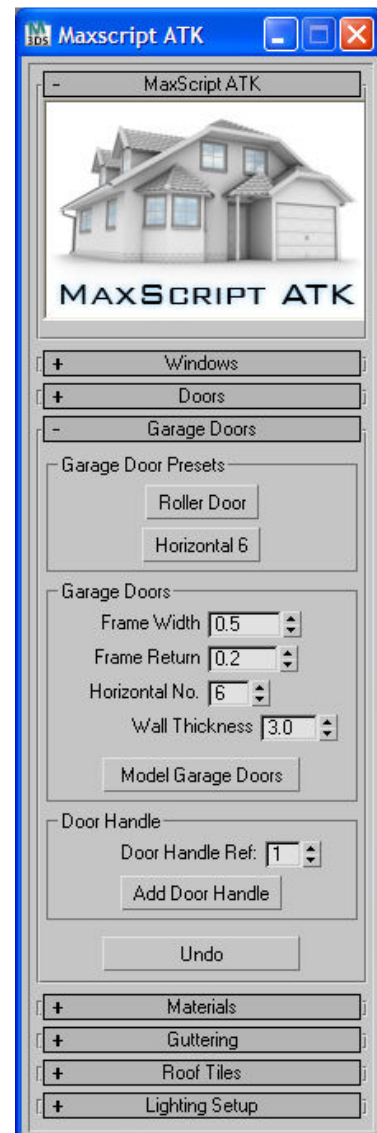


Figure 4-6 Dialog box for garage door construction

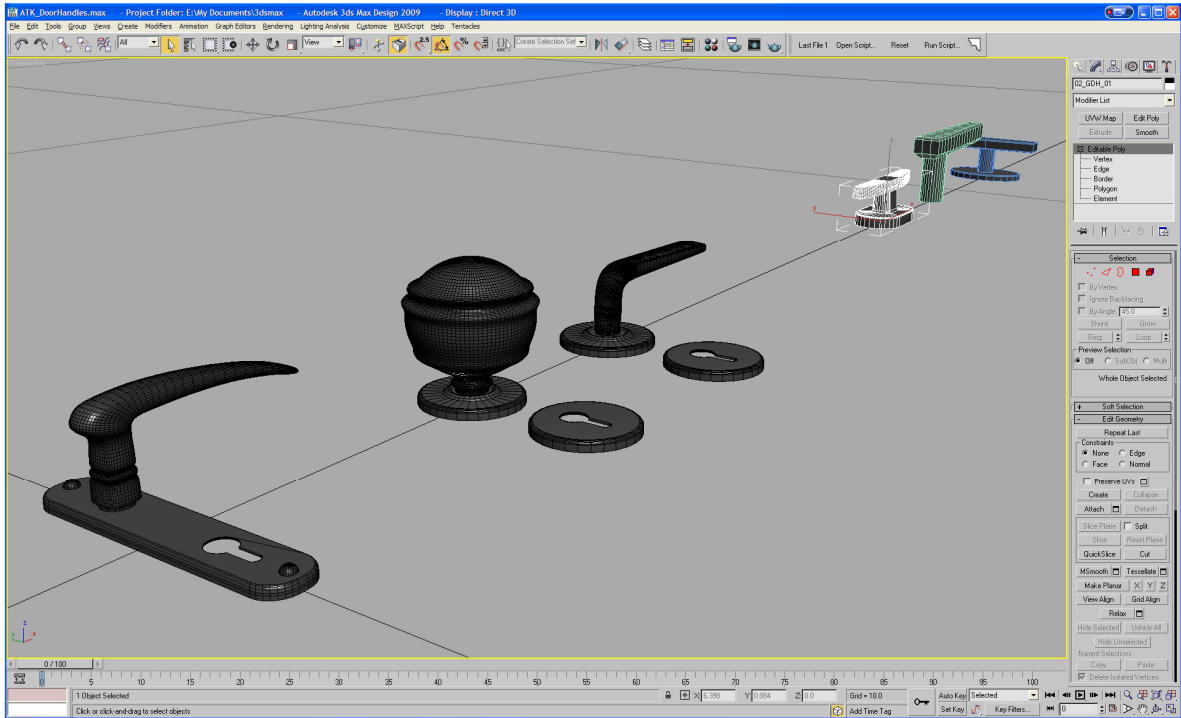


Figure 4-7 Door handles available for standard doors

on button\_doorHandle pressed do

```
(
select $ATK_GarageDoor_* --select all garage door entities
garageDoorArray = selection as array -- puts selected objects into an array
currentHandleRef = spinner_handleRef.value --get the handle selection
currentString = currentHandleRef as string --concatenate current with the object name
currentShapeName = "02_GDH_0" + currentString --set current shape string back to object
in scene
for i = 1 to garageDoorArray.count do -- loop for all garage doors
(
currentShape = xrefs.addNewXRefObject "L:\9.0
UEA\ATK_Files\ATK_DoorHandles.max" currentShapeName --#quiet--#merge --
import correct door handle
garageDoorHandle = garageDoorArray [i] --set the alignment object to first garage door
garageDoorHandleGeom = currentShape --set the alignment to object to the tiles
```

```

-- insert the door handle above...
garageDoorHandleGeom.transform = (transMatrix garageDoorHandle)
)

```

Source Code 4.3 Attach door handle to Garage Door

```

fn transMatrix obj = -- function to align objects (transformation matrix)
(
  local FaceCenter=polyOp.findFaceCenter obj 1 -- get face center
  local normalVal=polyop.findFaceNormal obj 1 -- get normal of the face
  local vectorUp = [0,0,1] -- define a vector pointing up (+z)
  local XVal = if acos (dot vectorUp normalVal) < 0.005 then -- if normalUp is close to
pointing up.....
[1,0,0] --assume the X value to be the world coordinate system
else
normalize (cross normalVal vectorUp) -- otherwise the X is perpendicular to the normal & up
vector
local YVal = normalize (cross normalVal XVal) -- Y is perpendicular to the Normal & X
matrix3 XVal YVal normalVal theFaceCenter -- matrix uses 3 axes and the face center
)

```

Source Code 4.4 Function to align objects (Door Handle to Door)

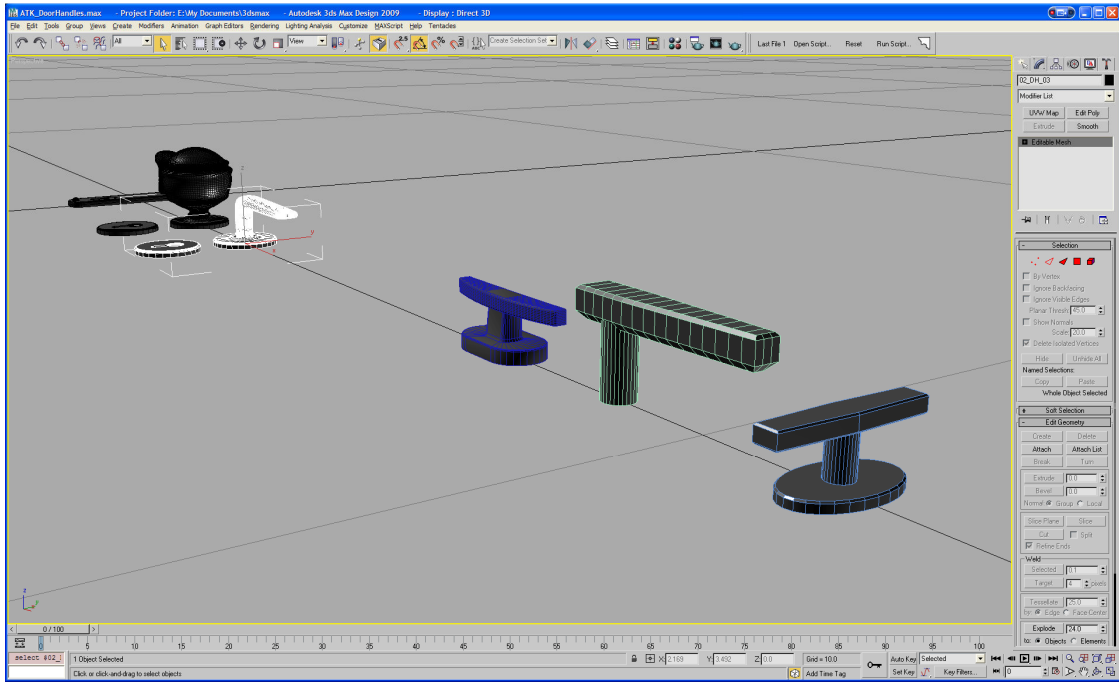


Figure 4-8 Door handles available for garage doors

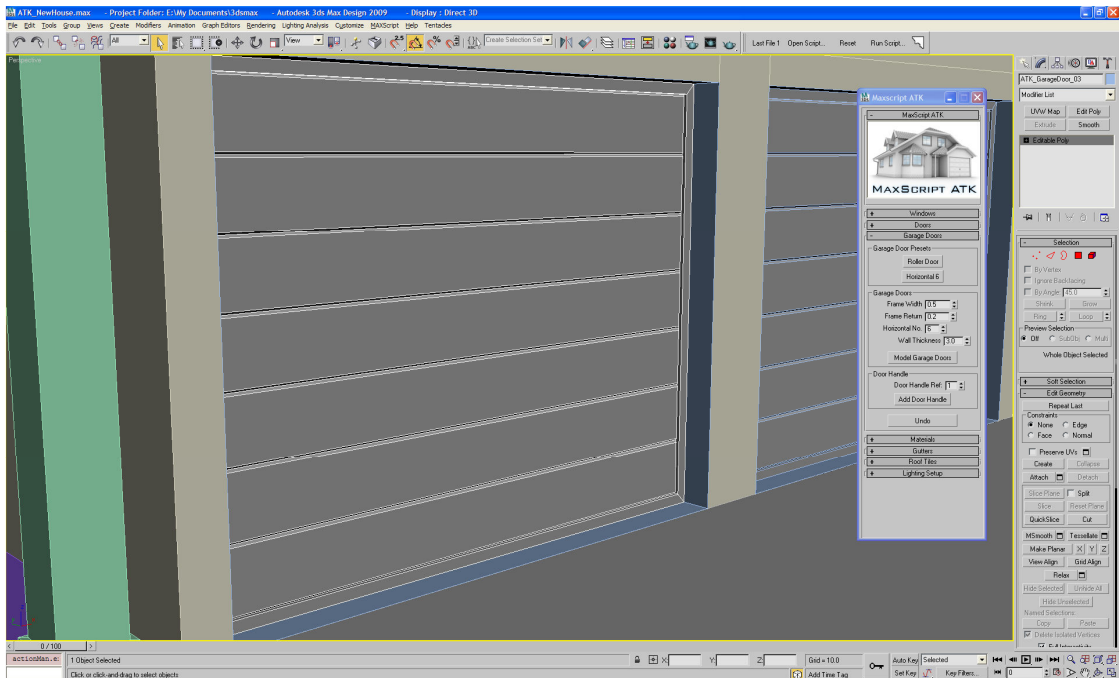


Figure 4-9 Garage Door geometry created using MAXScript ATK

#### 4.1.5. Materials

It was thought that it would be useful to include a set of predefined V-Ray materials as part of the MAXScript. This can be especially useful as V-Ray material settings can be complicated for beginners. One option which has been included is a displacement grass material, the displacement map used is shown in Figure 4-12. When using displacement materials, the 3D geometry is only realised at render time so working within the scene is not processor intensive. This method of 3D grass creation can cause problems in V-Ray if it is used over a large area, V-Ray is well known to crash without any notification because of memory issues, this is another reason why the rendering engine is not suitable for beginner modellers. The displacement grass material is generally only used in areas which are close to the camera, and in areas which will definitely be seen by the camera. This effect can be seen more clearly on the finished renders at the back of the paper, see Figure 7-3.

Presets for Aluminium, Chrome and Glass are also included within the program. There is the potential for thousands of materials to be added, however, it was thought that these basic few would be a functional start for an architectural visualisation.



Figure 4-10 Material presets available

#### 4.1.6. Guttering

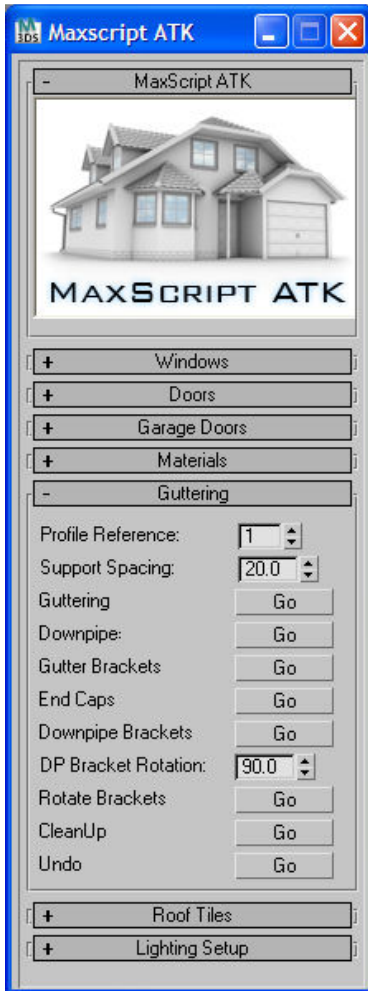


Figure 4-11 Gutter construction options

The guttering section of the program imports a set of profiles from an external file in a similar way that is used to import the door handles. These profiles can be edited easily in the file 'ATK\_GutterProfiles.max', or additional profiles could be added so that more options are available to the user.

There were a large number of options required to make the guttering function usable and adaptable for architectural visualisations. The supports for the gutters are imported, as you can see from Figure 4-13 a different gutter support is required for the horizontal and vertical (downpipe) sections of the guttering, The row of profiles at the top of Figure 4-13 are for the horizontal sections, and the bottom row are for the vertical sections. You can see an example of the final output at the back of the report in Figure 7-3.

It was initially suggested that a process for semi-automatically creating the spline for 'lofting' guttering could be created in a similar way to that provided by Batzal Software in Roof Designer. It was decided that this would not be incorporated into the program as the route of guttering required often doesn't follow each roof face.

One of the problems encountered when copying the vertical gutter supports into the scene is that there is a requirement for the supports to be facing towards the building. A way of fully automating this could not be found, when the program spaced the supports for the vertical sections of guttering, each vertical section was assigned to an array. This in effect grouped the supports for each section so that if manual adjustment was required, this was completed easily



with the 'Rotate Brackets' button. This button rotates the currently selected vertical guttering pipe bracket 90 degrees, along with all other brackets on that vertical pipe.

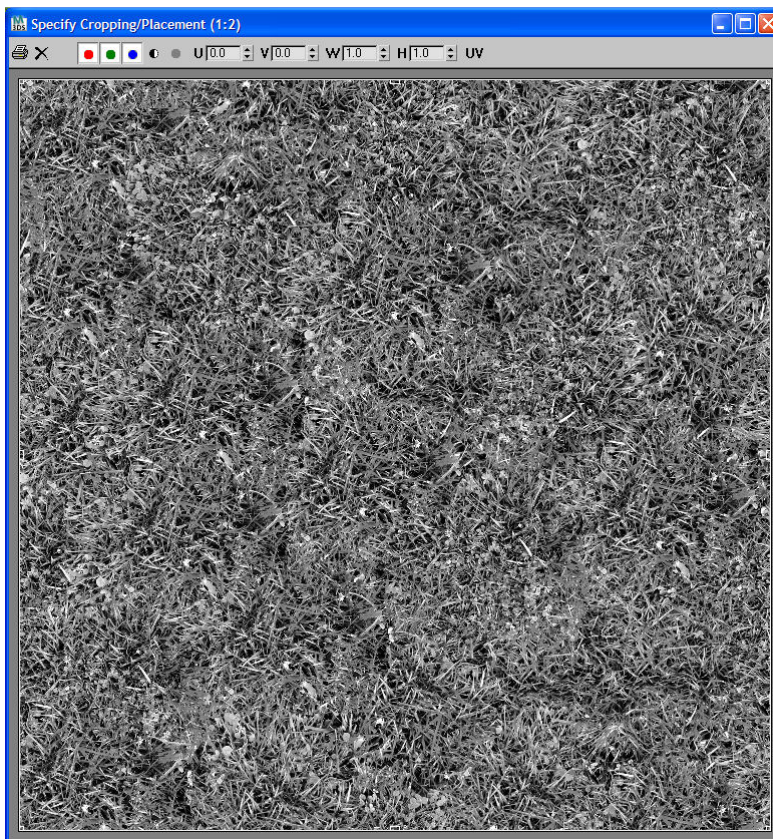


Figure 4-12 Displacement map used for the grass material

in 3D Studio Max, a 'loft' is used along the path of the gutter using one of the imported profiles, then a separate profile is used to cap the ends of the gutter to ensure that there is a hollow area within the gutter where the water would flow.

There is a cleanup option incorporated into the gutter function which removes any unwanted geometry imported from the 'ATK\_GutterProfiles.max' file from the scene. This ensures that the scene has a minimum file size and that finding objects within the scene is relatively easy.

The end caps are added to the gutters using a separate button, because there is no function to create the gutter shape automatically

```

for i = 1 to gutterPathArray.count do -- perform this loop for each gutter path
(
  gutterPath = gutterPathArray[i]
  gutterLength = gutterPath -- set the current length of guttering
  lengthVar = curvelength gutterLength -- find the length of guttering
  step = lengthVar / spacing -- calculate the steps for the spacing
  step = 100 / step -- step as percentage ie. 100 frames
  obj = currentShape --set the profile for the gutter brackets
  maxOps.cloneNodes currentShape cloneType:#copy newNodes:&backup --clone the shape
  backup.name = "backup" --rename
  spl = gutterPath --set the spline to use as path
  disableSceneRedraw() --disable viewport redraw for speed
  old_pos = obj.pos.controller --get the original position controller
  obj.pos.controller = path follow:true path:spl constantVel:true --assign a Path controller to
  follow
  with animate on -- code to create multiple copies of the brackets
  (
    at time 0f obj.pos.controller.percent = 0 -- set the percentage for frame 0 to 0
    at time 100f obj.pos.controller.percent = 100 --set the percentage for frame 100 to 100
  )
  for t = startPercentage to endPercentage by step do --Loop through time with a Step of +1
  (
    at time t -- at the current time
  (
    new_copy = instance obj -- create an instance of the bracket
    new_copy.name = uniqueness (obj.name+"_01") --rename the bracket
    new_copy.pos.controller = Bezier_Position() -- assign a Bezier controller
    new_copy.transform = obj.transform --copy the transform Matrix to orient the bracket
    correctly
  )
  )
  obj.pos.controller = old_pos --assign the controller back to the source object
  enableSceneRedraw() --enable viewports to redraw
  redrawViews() --redraw them
  delete currentShape -- delete the template shape
  backup.name = currentShapeName -- change backup name
  currentShape = execute ("$" +currentShapeName + "") -- get the shape in the scene which
  corresponds to the string
  )
)

```

Source Code 4.5 Code to Space and Align the Gutter Brackets

The code used to space and align the gutter brackets is shown in Source Code 4.5. The process involves assigning a path constraint to each of the gutter brackets. A path constraint is generally used in animation to make one object follow the path of a path object (a line). In this case the path object is the route of the guttering. Once the path constraint is assigned, the gutter brackets are copied at set intervals as assigned by the user.

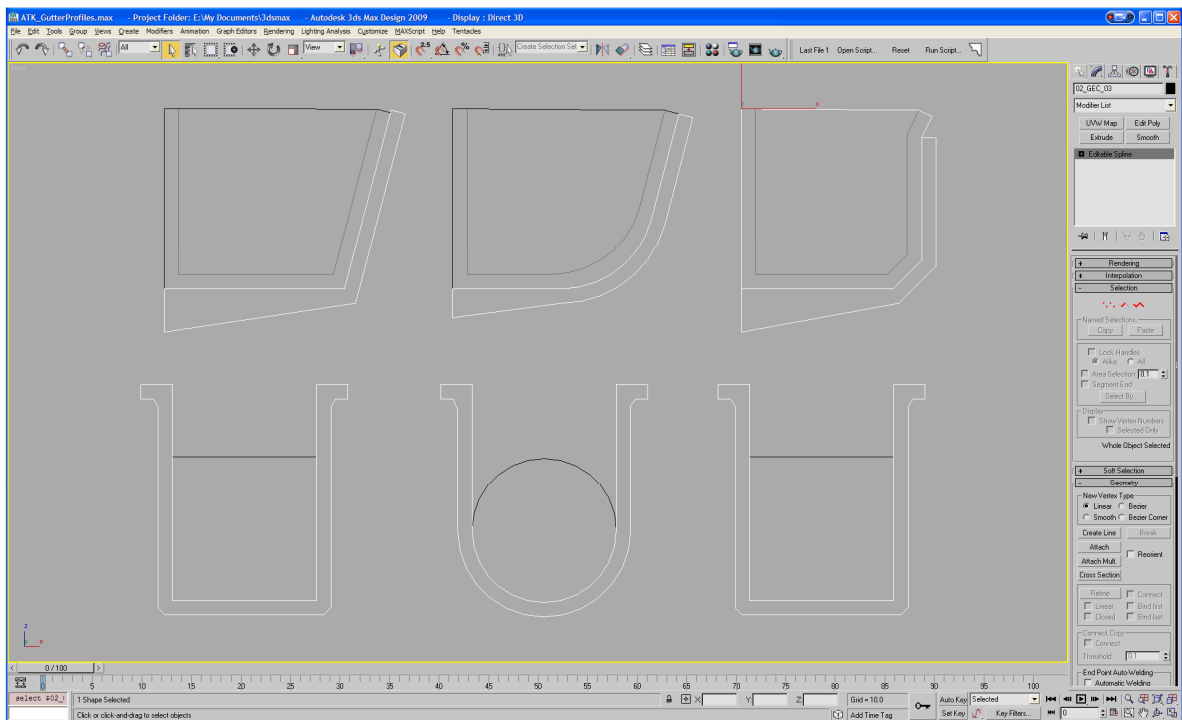


Figure 4-13 Gutter profiles available by default

### 4.1.7. Tiling Function

For added realism, a multi sub-object material has been applied to the geometry which represents the tiles. It is generally thought that the most effective way of texturing a tiled roof is

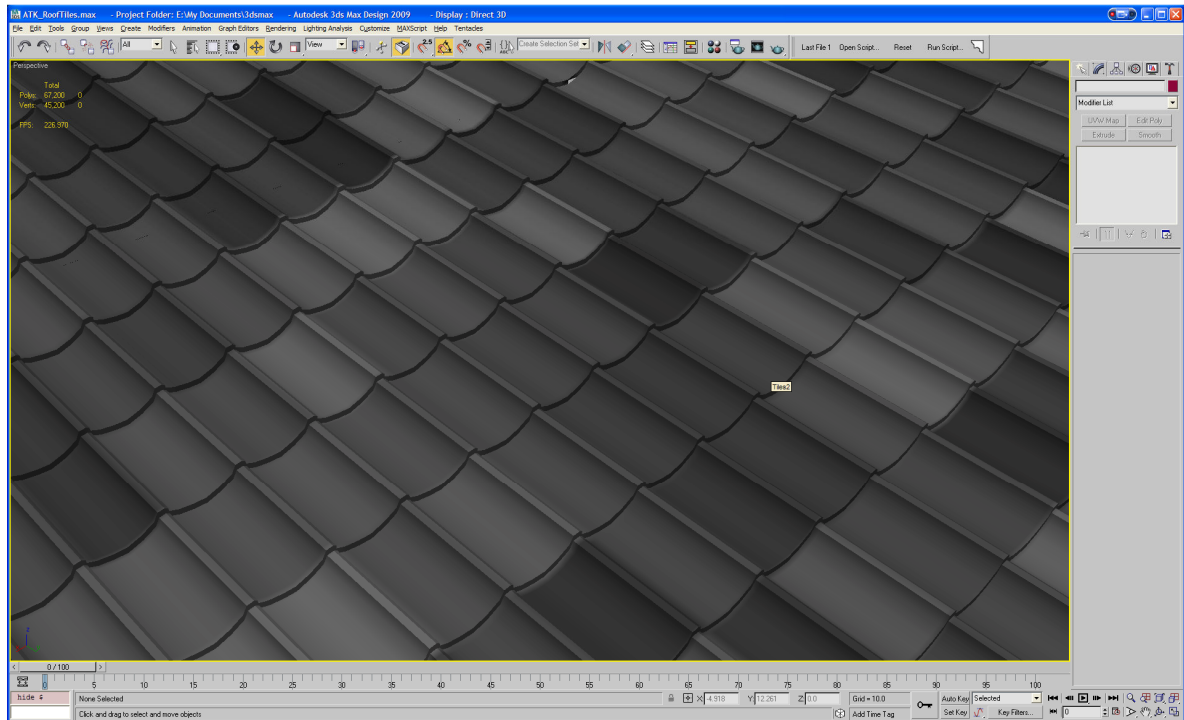


Figure 4-14 Result of tiling function

to use the 'tiles' option in the material editor with a positive value in the 'Color Variance' option and to increase the horizontal and vertical count so that no 'texture tiling' is visible.

The tiling of the roof is performed using the following process.

#### Roof Tiles. (User Input)

- Select Geometry for the roof
- Select tile Profile – Select the required geometry from a drop down list.

#### MAXScript Processing

- Import the required tile geometry from the external file
- Clone the tile geometry so that there is 1 set of tiles for each face of the roof.
- Align each individual set of tile geometry to each of the faces on the roof
- Trim set of tile geometry based to the edges of each face on the roof

#### Roof Ridge Tiles (User Input)

- Select tile Profile – Select the required geometry from a drop down list.

#### MAXScript Processing

- Get the roof geometry which was used for the creation of the roof tiles.
- Locate the edge of each face which is at the highest point of each face.
- Create a spline based on these edges.
- Create an array of the ridge tile geometry following the edges of the spline

The roof tile modelling procedure which is followed by MAXScript ATK is illustrated in a flow chart in Figure 4-16.

The transformation matrix function is used to align the roof tiles to the roof, as shown in Source Code 4.6.



Figure 4-15 Roof tile setup

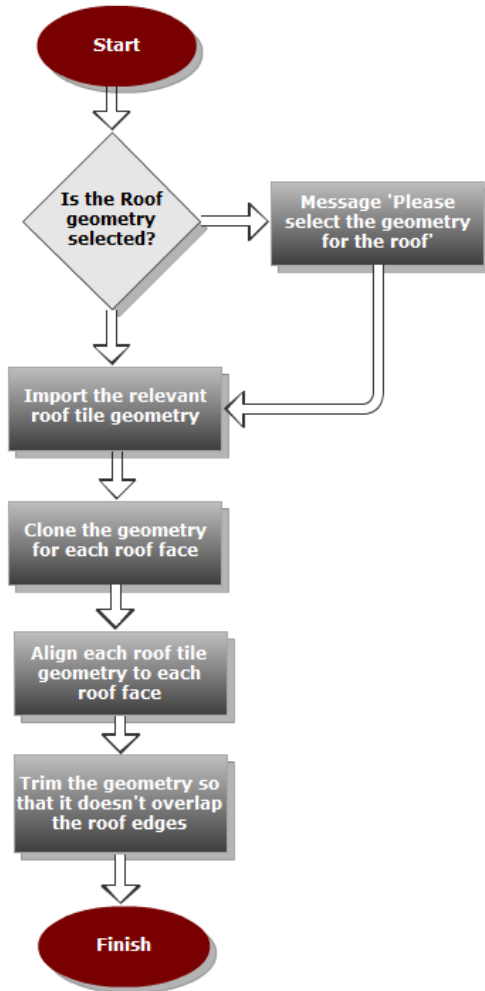


Figure 4-16 Flow chart to show procedure followed when modelling roof tiles.

The ridge tiles are added after the roof tiles have been generated. The program uses the roof tile option selected by the user to automatically select the correct ridge tile.

To select only the edges which are to have roof tiles, the script locates any edges which are attached to 2 polygons where the faces of the polygons have a minimum angle of 181 degrees. These edges are used to space the geometry for the ridge tiles imported from an external file, 'ATK\_RoofRidges.max', see Figure 4-17. The 'spacing tool' which is generally available when manually performing this function in 3D Studio Max is not available within MAXScript so a function was written into the code which completes this operation. The spacing of the ridge tiles is determined automatically by the program based on the dimensions of the ridge tiles. This makes it possible for the user to add

custom modelled ridge tiles in the same way as with the door handle and gutter functions.

`roofTileGeometry.transform = (transMatrix roofGeometry) – align the roof tiles to the roof`

Source Code 4.6 Code to utilise the transformation matrix function to align tiles to the roof

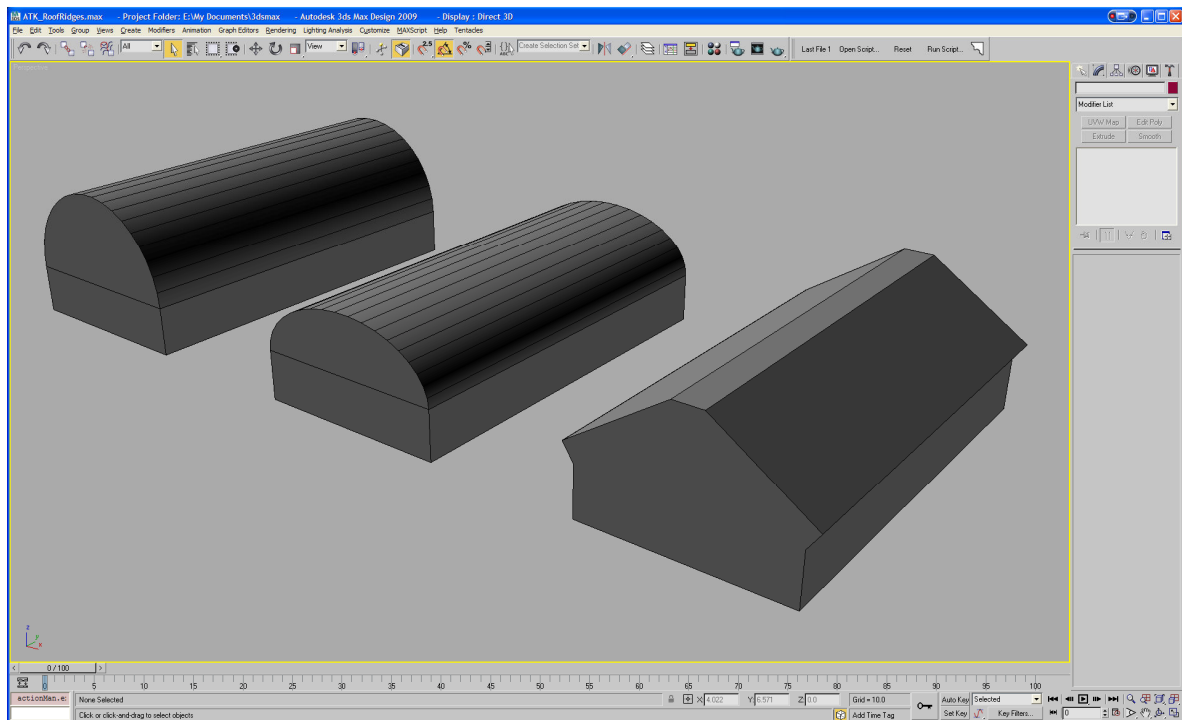


Figure 4-17 Roof ridge tile shapes available by default

The tiling function was found to be the most processor intensive function within MAXScript ATK, therefore it was thought that a progress bar should be included so that the user could see that the function was in progress and that the system was still operating.

When performing the tiling function, the MAXScript ATK ensures that the viewport does not update to show the individual modelling tasks taking place, this was thought to be more professional and also increases the speed of the function as 3D Studio Max is not required to update the viewport.

#### 4.1.8. Lighting and Environment

One of the most time consuming parts of producing 3D renders in 3D Studio Max and VRay can be getting the correct settings for certain lighting conditions. After considerable research into the best methods of Global illumination for architectural visualisations, 3 presets have been included in the program; the presets import settings from a presets file, and geometry from another file to create the environment. A dome with a cloudy sky texture is imported, along with relevant VRay lights.

Following evaluation using various options for global illumination, it was considered that the program would use Irradiance map for primary bounces of light, and light cache for secondary bounces of light. This combination demonstrated the best results in terms of rendering speed and image quality. Depending on the scene and camera location, global illumination settings may need to be altered, if for example, a camera is located in a courtyard where there is a sun shade, with the global illumination settings provided by MAXScript ATK, there may be 'artefacts' on the final rendered image. These artefacts can be fixed by altering settings for the irradiance map and light cache.



Figure 4-18 Lighting setup dialog



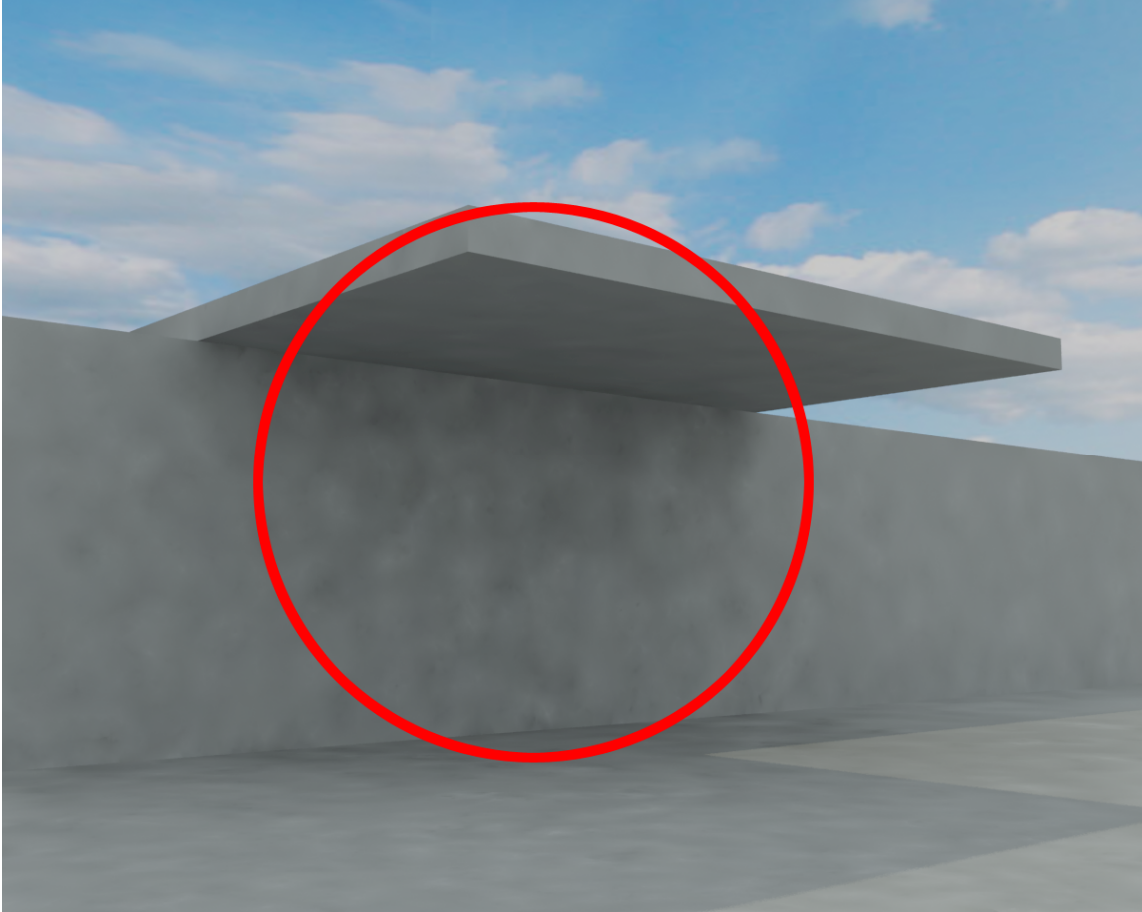


Figure 4-19 Artefacts in renders due to incorrect global illumination setup

To take MAXScript ATK further, a useful addition could be to include a number of different default options for global illumination in V-Ray which could combat this issue. There is a software solution which makes V-Ray rendering settings easier to use. 'SolidRocks' created by 'Subburb' is a MAXScript which has preset rendering options to reduce rendering times, see Figure 4-20 for a screenshot of SolidRocks. It is thought that a preferable option when trying to remove artefacts is to perform a 'clay render' this option will remove all materials from the scene. By performing a test render this way, artefacts can easily be located and corrected before rendering the final scene with all textures turned on. Figure 4-19 illustrates the artefacts which can appear in shaded areas of a scene.

The settings for lighting and rendering are merged into the scene when the user presses the relevant lighting button, the code used to perform this action is shown in Source Code 4.7.

```
on lightingButton pressed do
(
  fobj_names = getmaxfileobjectnames "L:\9.0 UEA\ATK_Files\ATK_DaylightOvercast.max"
  – merge in the environment geometry
  mergemaxfile "L:\9.0 UEA\ATK_Files\ATK_DaylightOvercast.max" fobj_names #select
  renderpresets.Load 0 "L:\9.0 UEA\ATK_Files\ATK_DaylightSunny.rps" – load in the
  respective rendering settings file
)
```

Source Code 4.7 Code to import environment geometry and rendering settings

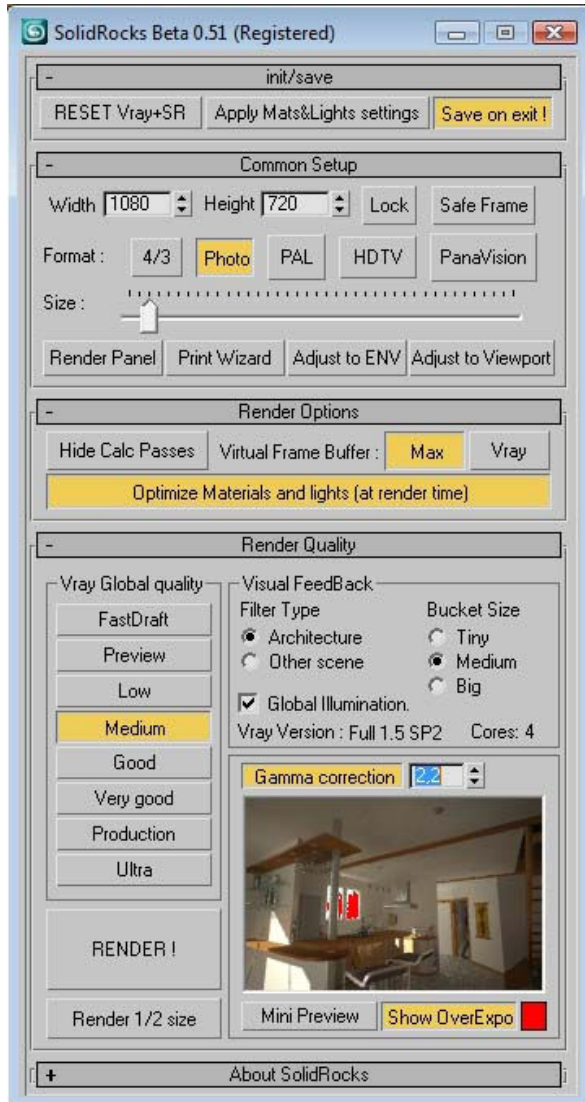


Figure 4-20 SolidRocks 3D Studio Max plug-in

Figure 4-21 shows the 3D Studio Max environment when the 'Daylight Overcast' option is used to light the scene. Figure 4-22 shows the environment when the 'Daylight Sunny' option is used.

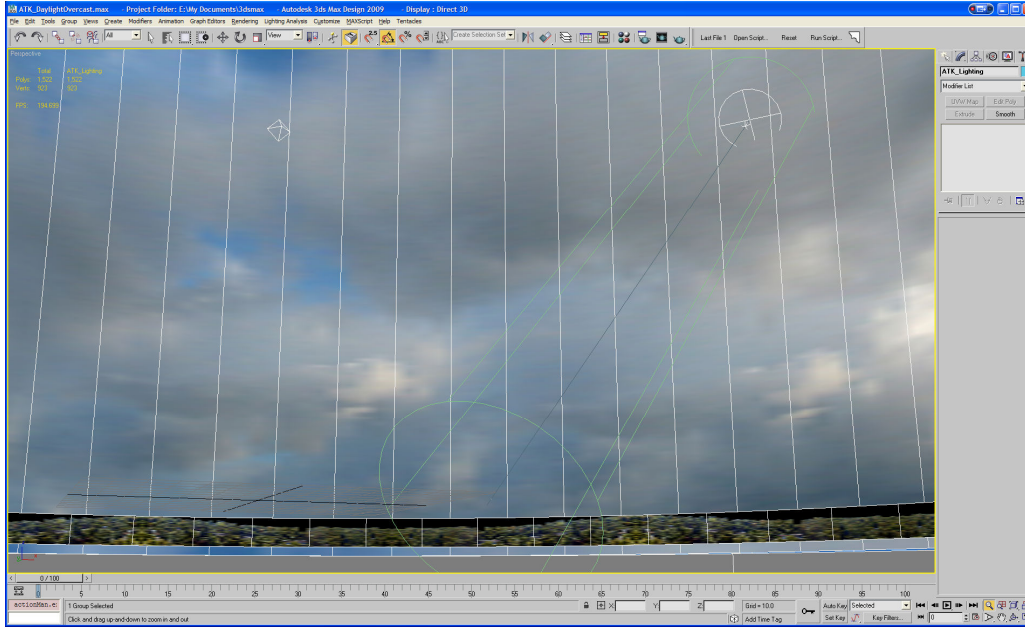


Figure 4-21 Daylight overcast lighting setup within 3D Studio Max

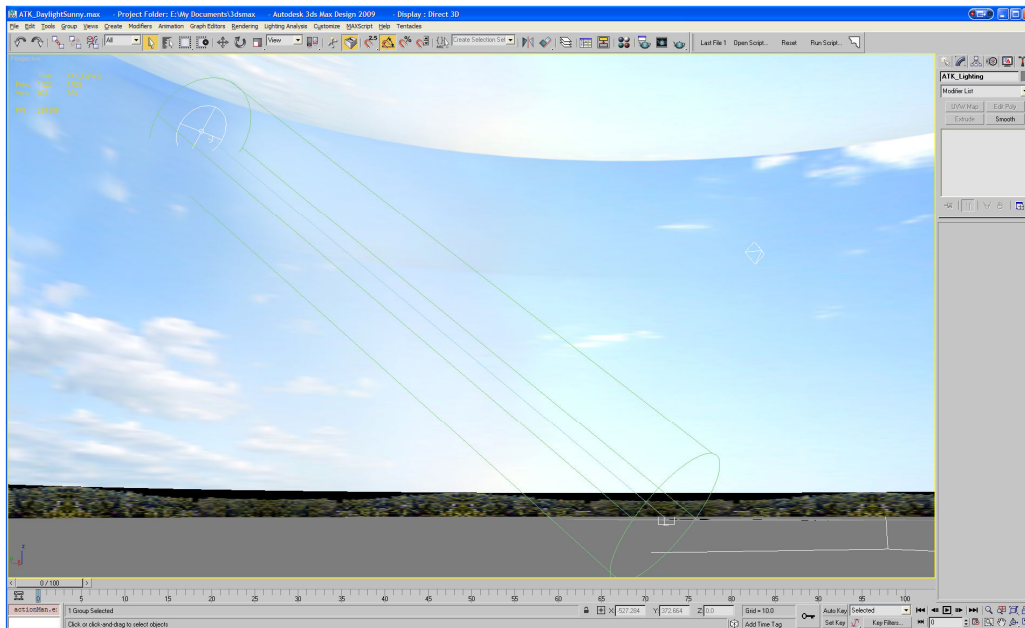


Figure 4-22 Daylight sunny lighting setup within 3D Studio Max

## 4.2. Testing

MAXScript ATK is written using MAXScript running in 3D Studio Max 2009. Some of the MAXScript functions used as part of the MAXScript ATK are unavailable in older versions of 3D Studio Max therefore the toolkit will only work in 3D Studio Max version 2009, 2010, 2011 and 2012. The script has been tested in these versions and performed identically in each. The MAXScript ATK has also been tested on 3 different machines with different hardware specifications. The Specifications used for testing are detailed in Table 4.1. The testing showed that the MAXScript ATK performed identically on each of the specifications.

	Processor	Ram	Operating System
PC 1	Intel Core 2 Quad Q6600, 2.4GHz	4GB	Windows XP
PC 2	Intel Xeon Processor E5620, 2.4 GHz	4GB	Windows 7
Laptop 1	AMD Dual-Core E2-3000M, 1.8 GHz	2GB	Windows Vista

Table 4.1 Table to show hardware specifications used for testing

A number of reference files are required for the MAXScript ATK to function. These include the reference files for geometry, ie. Door handle geometry 'ATK\_DoorHandles.max' or gutter profile geometry 'ATK\_GutterProfiles.max'. The other files required are a material library file 'ATK\_Materials.mat' which stores the preset materials used as part of the program, and the rendering setting reference files ie. ATK\_DaylightSunny.rps. This file for example stores the rendering and global illumination settings for the 'Daylight Sunny' option. The script calls these files from a set path which is always fixed. To ensure that MAXScript ATK can always see the reference files, these should be stored on a virtual drive which is mounted in windows, ie. The 'L' drive has been mounted and the reference files are stored at this location 'L:\9.0 UEA\ATK\_Files'. The user must ensure that this drive is mounted for MAXScript ATK to operate, this can be completed easily by running the following code within the command

prompt 'subst L: E:\UEA\_Working'. This code creates a virtual drive which shows the files located within 'E:\UEA\_Working'.

#### **4.2.1. Constraints**

The roof tiling function relies on the importing of an item of geometry which includes 100 no. x 60 no. roof tiles. This geometry is used to align to the roof and a trim function is performed on the roof tile geometry to remove any tiles which are not required on this particular roof surface. If one of the portions of a roof required more than 100 roof tile along the width of the roof, the MAXScript ATK in it's current form would not be able to perform a correct tiling operation as there would be spaces either side of the roof which were untiled. However the program is designed for standard modern UK estate houses which are unlikely to have a roof of this size. This flaw could easily be fixed by increasing the size of the tile geometry in the reference file however this would slow the import and tile trimming process down.

## 5. Results & Evaluation

It was considered the most efficient way to confirm that the MAXScript has met the brief would be to evaluate the software on its target audience (AutoCAD design engineers with no extensive 3D Studio Max experience). The primary goal of the program was to make the modelling and rendering functions easy to use, and also speed up the tasks. With this in mind, it was thought that the program could be evaluated using a sample of the target audience performing the tasks without assistance and being timed under exam conditions. The first part of the evaluation involved the users completing tasks using traditional methods, i.e. The methods described in 3 Analysis & Requirements which involve modelling using the tools provided with 3D Studio Max. The subjects were given the instructions included in 8 Appendix B - Evaluation Questions using Traditional Methods

Appendix C – Evaluation Questions using MAXScript ATK and watched by the specialist whilst completing the tasks. Each of the subjects was timed from when they were given the instructions, through to satisfactory completion of each of the tasks. No help was provided to any of the subjects during completion of the tasks. The second part of the evaluation involved the users completing the modelling tasks using the MAXScript Architectural Toolkit. Once again, a new set of instructions was provided to the subjects, the instructions provided are shown in 1 . Conditions for this testing were identical to the initial tests. It was thought that it would be useful to show the times required for an experienced 3D Studio Max user (shown as ‘Specialist’ in Table 5.1) as part of the tests. The experienced 3D Studio Max user is in this case the Author of the thesis. As mentioned previously, the author has strong working knowledge of 3D Studio Max software used for architectural visualisations.

The subjects used for the testing are all employed as Design Engineers for a Multinational Engineering Consultancy based in Norwich. Their role requires them to project manage small engineering schemes and also produce the basic 2D AutoCAD drawings for their projects. None of the users had any experience using 3D Studio Max before these evaluations were carried out. One of the subjects was female and 4 of the subjects were male, this was a fair

representation of the male/female divide in this engineering office therefore considered a fair representation of the engineering workforce as a whole. These subjects have been selected because they have no experience of 3D Studio Max, work in an engineering office, and thus are the target audience for the MAXScript ATK.

Method	Modelling Criteria	Specialist	Subject A	Subject B	Subject C	Subject D	Subject E	Average
Traditional	Tiling	02:38	04:54	06:31	05:23	05:43	06:58	05:21
Traditional	Gutters	03:37	04:57	06:27	05:01	04:28	05:51	05:04
Traditional	Windows	02:31	05:31	05:21	06:54	05:34	05:14	05:11
Traditional	Doors	01:16	04:50	05:43	05:21	06:32	06:03	04:58
Traditional	Lighting	03:05	06:07	06:40	05:56	06:34	06:24	05:48
<b>Total Traditional</b>		<b>13:07</b>	<b>26:19</b>	<b>30:42</b>	<b>28:35</b>	<b>28:51</b>	<b>30:30</b>	<b>26:21</b>
Method	Modelling Criteria	Specialist	Subject A	Subject B	Subject C	Subject D	Subject E	Average
ATK	Tiling	00:20	00:31	00:40	00:36	00:41	00:32	00:33
ATK	Gutters	00:35	00:42	00:27	00:41	00:37	00:24	00:34
ATK	Windows	00:31	00:37	00:42	00:37	00:30	00:34	00:35
ATK	Doors	00:12	00:16	00:13	00:14	00:16	00:16	00:14
ATK	Lighting	00:05	00:09	00:10	00:08	00:12	00:09	00:09
<b>Total ATK</b>		<b>01:43</b>	<b>02:15</b>	<b>02:12</b>	<b>02:16</b>	<b>02:16</b>	<b>01:55</b>	<b>02:06</b>

Table 5.1 Table to show evaluation results for Traditional modelling methods and MAXScript ATK

The data in Table 5.1 shows that the times required to perform each of the modelling tasks has been greatly reduced when using MAXScript ATK. On average the times are reduced to 8% of those required using traditional modelling methods.



## 5.1. Case Study – Subject C

As with all subjects, Subject C was able to complete the modelling tasks considerably faster using the MAXScript ATK. The geometry created using traditional methods of modelling was however less accurate than when using the MAXScript ATK. This was expected as part of this evaluation as the modelling process was only instructed with basic commands. If full instructions were given with exact dimensions and offsets, it was thought that the tasks would take far too long to complete. See Figure 5-1 and Figure 5-2 to see the difference between geometry created by subject C for the gutters and the geometry created by the specialist provided with the instructions.

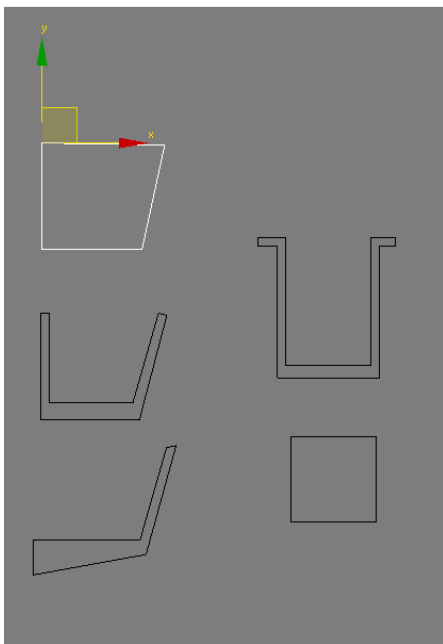


Figure 5-1 Showing Subject C gutter shapes

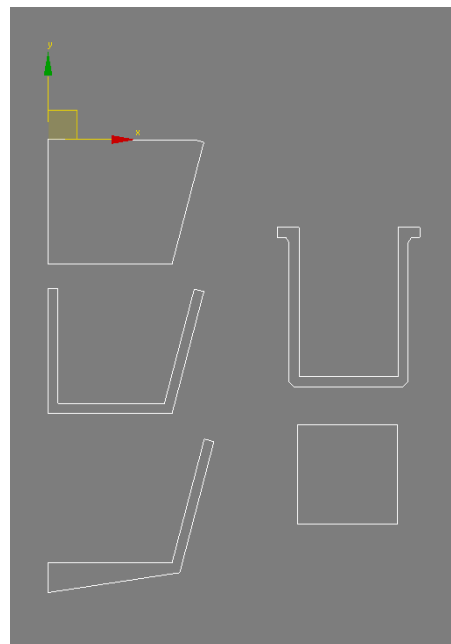


Figure 5-2 Showing gutter shapes provided with instructions

Note in Figure 5-3 the final geometry created by Subject C using traditional methods has some imperfections. The end capping of the gutter is not properly aligned to the gutter, this is also the case with the gutter brackets. The gutters modelled by Subject C using the MAXScript ATK were as expected and identical to the specialists as well as all other subjects. This proved that the tool will enforce design constraints on the model to ensure that objects are created with

clean geometry and vertices will meet if this is required. This is ensured as long as the reference files help by the MAXScript ATK are drawn correctly.

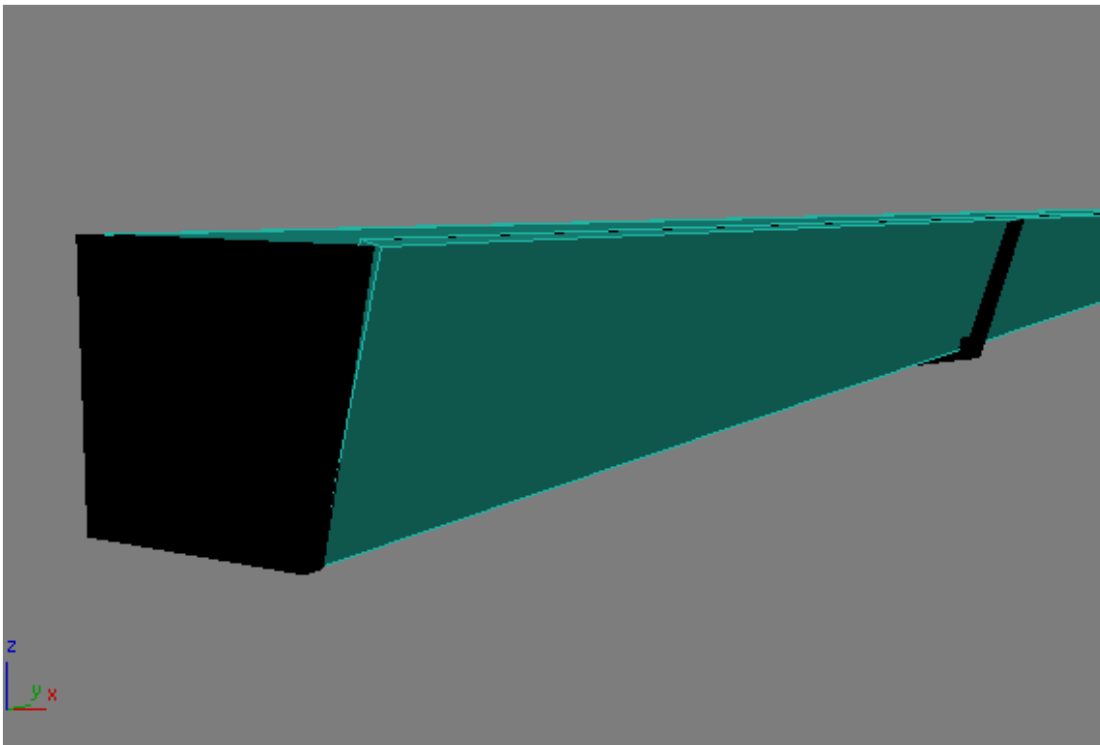


Figure 5-3 Showing the inaccuracies of the gutter model created by Subject C

The window creation did not leave any room for imperfections when Subject C was creating the geometry using traditional methods. The window plane geometry was provided and the tasks involved entering inset and extrude values which was completed accurately.

As the door creation method is similar to the window creation method, it would have been difficult to perform incorrectly or with imperfections. Subject C completed this task and the final geometry was identical to that created using MAXScript ATK.

The lighting setup involved inputting values into correct fields within an extensive set of options. There was a considerable time taken by Subject C to find the location of each of the options, although they were all found successfully and the final result was identical to that when using the MAXScript ATK.

The method used to create the roof tile geometry by Subject C traditional methods also did not leave any room for imperfections. The final result was identical to that created using MAXScript ATK.

	Traditional	MAXScript ATK	MAXScript ATK as % of Traditional methods
Tiling	05:21	00:33	10%
Gutters	05:04	00:34	11%
Windows	05:11	00:35	11%
Doors	04:58	00:14	5%
Lighting	05:48	00:09	3%

Table 5.2 Table to show the percentage of the time required for modelling using MAXScript ATK compared to traditional methods

Also, as expected, some of the tasks which took considerable time using traditional methods (notably lighting) had a huge reduction so that it took only 3% of the time required using traditional methods. What became noticeable from Table 5.1 is that the difference between the times from the Specialist and the subjects were much closer using MAXScript ATK. For example the average time required for creating gutters using MAXScript ATK is 61% of that required when using traditional methods. This is due to the fact that the specialist knows the software well and the buttons required to complete each task were found easily. When using MAXScript ATK, the difference between the subjects and the specialist is only 1 second. This equates to the time required by the specialist being only 97% of that required by the subject.

As mentioned in the software study in 3 Analysis & Requirements, the time taken by an experienced 3d modeller to complete the tasks in the traditional fashion was similar across the software packages tested. Therefore, if a tool were to be constructed which were to run on other platforms, it is likely that similar speed improvements would be seen.

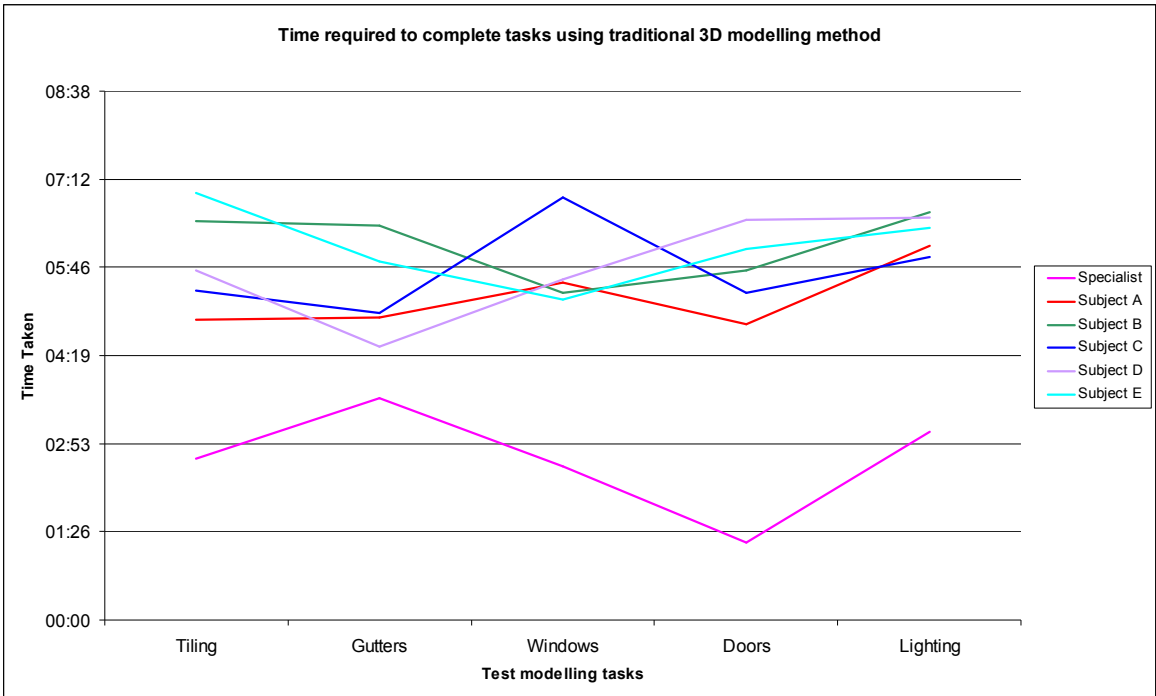


Figure 5-4 Graph to show time required to complete tasks using traditional 3D modelling method

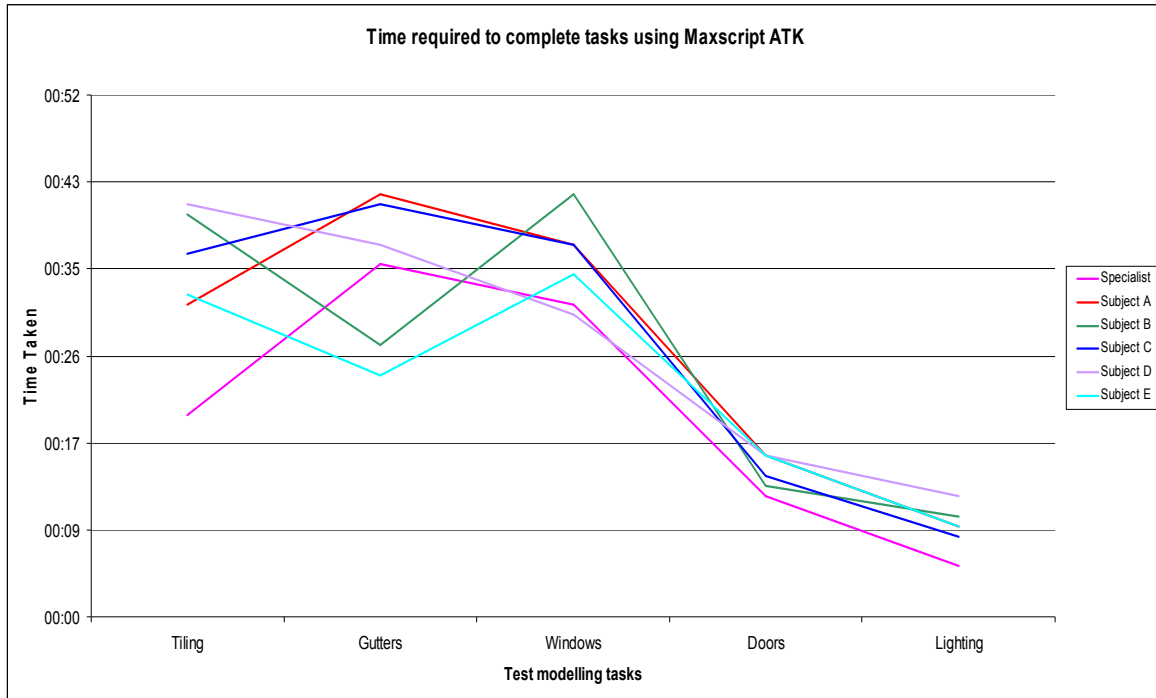


Figure 5-5 Graph to show time required to complete tasks using MAXScript ATK

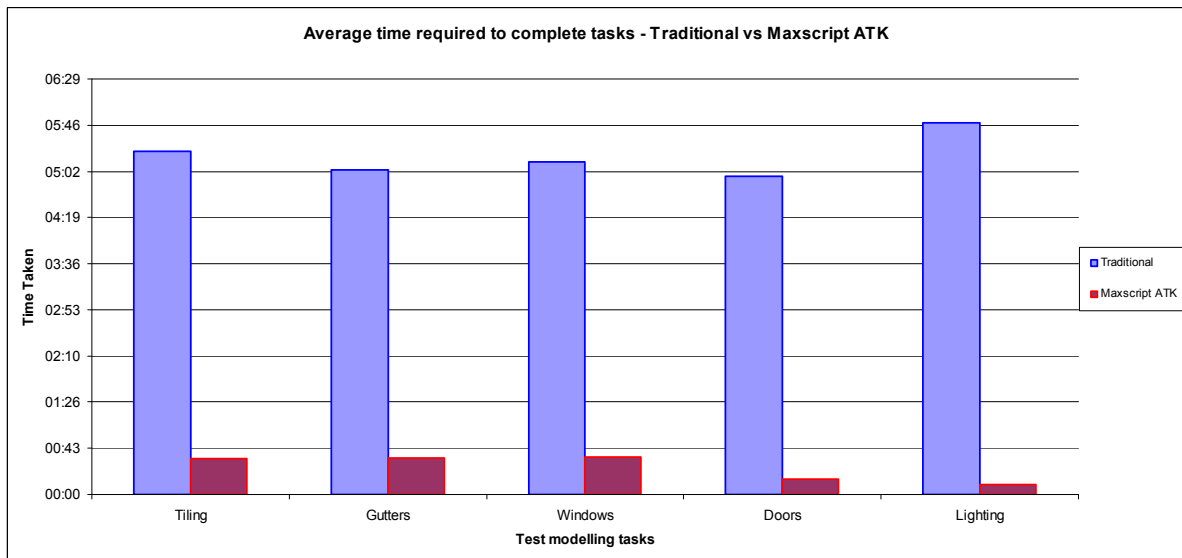


Figure 5-6 Graph to show the average time required to complete tasks – Traditional vs. MAXScript ATK

Following the practical evaluation, a number of questions were asked of the subjects to further determine the success of the tool.

Qs	Question	Function	1	2	3	4	5	6	7	8	9	10	Average	
1	On a scale of 1 to 10, how difficult do you think it was to complete the 3D Modelling using traditional methods? 10 being very easy, 1 being very difficult.	N/A			1	2		1	1				4.8	
2	On a scale of 1 to 10, how difficult do you think it was to complete the 3D Modelling using MAXScript ATK? 10 being very easy, 1 being very difficult.	N/A						2	2	1			6.8	
3	After completing the instructions to perform each modelling task the traditional way, how confident would you be that you could perform each of the tasks unassisted? 10 being very confident, 1 being not confident at all.	Tiling	3	2									1.4	
		Gutters		2	1	2								3
		Windows		1	1	1	2							3.8
		Doors			2	1	2							4
		Lighting	3	2										1.4
4	After completing the instructions to perform each modelling task using MAXScript ATK, how confident would you be that you could perform each of the tasks unassisted? 10 being very confident, 1 being not confident at all.	Tiling					1		2	2			7	
		Gutters						2	1	2				7
		Windows						1		1	3			8.2
		Doors							1	1	3			8.4
		Lighting										3	2	9.4

Table 5.3 Survey results table

Table 5.3 shows the results from the survey along with the average response for each question. The average user confidence for each question is plotted in a graph in Figure 5-7. The data shown in Table 5.1 illustrates a considerable improvement in the confidence of the user when using MAXScript ATK. The table highlights which of the tasks are seen as being the easiest to

complete. Modelling the doors and windows, with average confidence of 4 and 3.8 respectively are shown as being the easiest tasks to complete traditionally. As expected, Lighting the scene and tiling a roof are shown as being the most difficult tasks, both with average user confidence of 1.4.

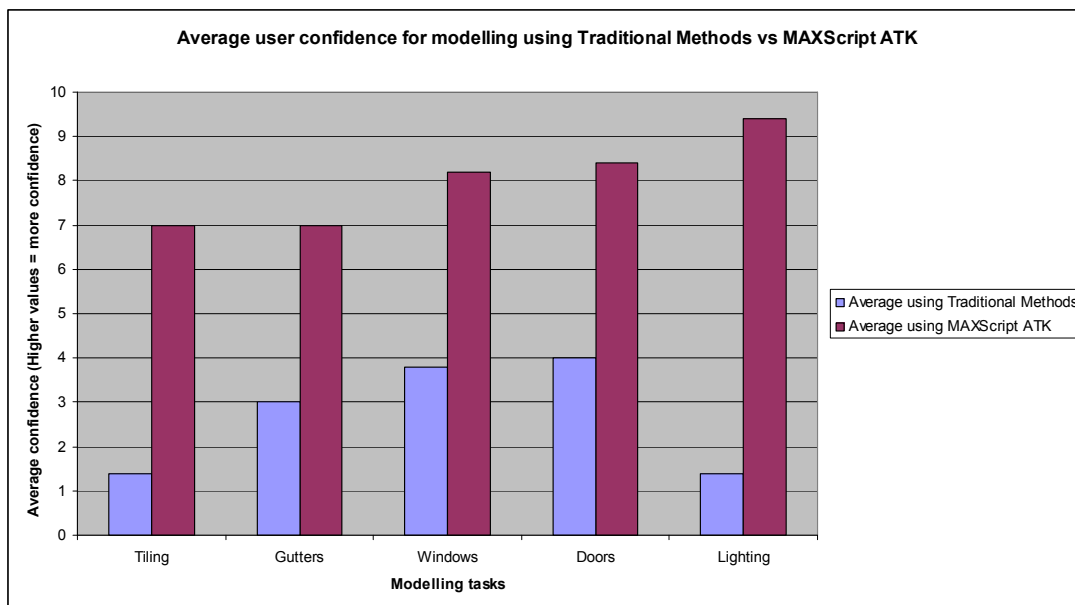


Figure 5-7 Average user confidence

Table 5.3 and Figure 5-7 illustrate that using MAXScript ATK increases the confidence of the user for all modelling tasks. The difference between the user confidences when using traditional methods and when using the MAXScript ATK is the greatest for lighting the scene, average confidence of 1.4 using traditional methods, compared to 9.4 when using MAXScript ATK. This large gap is down to the fact that there are no options required for the lighting setup using MAXScript ATK, only the rollout with buttons for 'Daylight Sunny' and 'Daylight Overcast'. The tasks completed traditionally when setting up lighting in VRay and 3D Studio Max is an extensive set of alterations to numerical options which can be meaningless to anyone unfamiliar to VRay.

Method	Modelling Criteria	Specialist	Subject A Attempt 1	Subject A Attempt 2
Traditional	Tiling	02:38	04:54	04:42
Traditional	Gutters	03:37	04:57	04:31
Traditional	Windows	02:31	05:31	05:20
Traditional	Doors	01:16	04:50	04:42
Traditional	Lighting	03:05	06:07	06:03
<b>Total Traditional</b>		<b>13:07</b>	<b>26:19</b>	<b>25:18</b>

Table 5.4 Table to show the time difference for completing the manual modelling tasks a second time.

It is thought by the author that because the subjects have completed the manual process of modelling before the automated process, they may have gained some knowledge of the workings of the software which gives them an advantage when completing the automated evaluation. One of the subjects completed an additional evaluation and the time required was comparable to the time required at the first attempt. The results from this experiment are detailed in Table 5.4.

shows the time taken to complete the required tasks in MAXScript ATK compared to traditional methods using the software packages detailed in 3 Analysis & Requirements. The data from is displayed in a graph in and highlights the sizable difference between the time required when using MAXScript ATK when compared to the other software packages evaluated. Note the tiling function required only 33 seconds to complete using MAXScript ATK compared to 47 seconds required for the Batzal Roof Designer.



	Google SketchUp	AutoCAD	3D Studio Max	Batzal Roof Designer	MAXScript ATK
Tiling	05:04	07:16	05:21	00:47	00:33
Gutters	04:40	06:12	05:04	N/A	00:34
Windows	03:06	06:39	05:11	N/A	00:35
Doors	04:28	05:28	04:58	N/A	00:14
Lighting	05:58	05:16	05:48	N/A	00:09
<b>Total</b>	<b>23:16</b>	<b>30:11</b>	<b>26:22</b>	<b>00:47</b>	<b>02:05</b>

Table 5.5 Table to show the time taken to perform modelling tasks using tested software

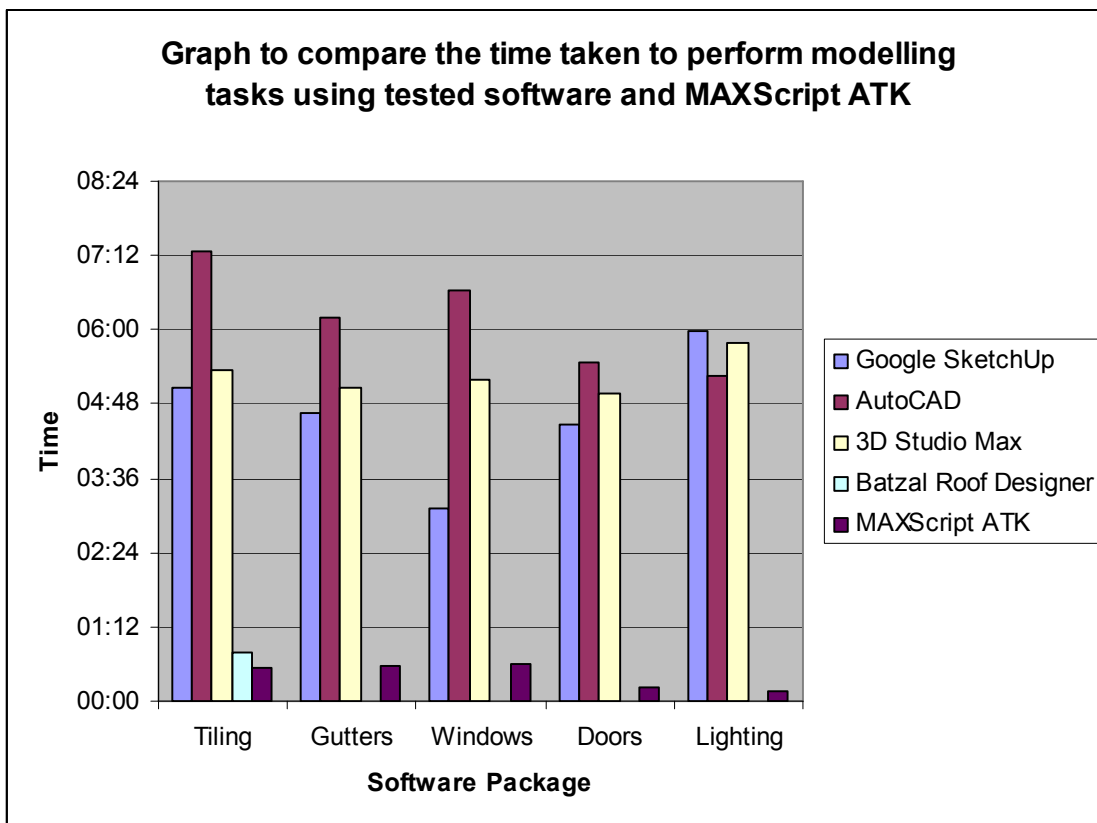


Figure 5-8 Graph to compare the time taken to perform modelling tasks using evaluated software and MAXScript ATK

## **6. Conclusions**

The survey results from 5 Results & Evaluation prove that the project has made it possible for office engineers with no previous visualisation experience to produce high quality visuals without having to outsource work to visualisation specialist freelancers or engineering companies who have visualisation capabilities, fulfilling the aims of this Masters by Research project.

Following the work carried out, it is thought that additional benefit could have been gained by the initial suggestion of questionnaires and contacting software developers as suggested in the methodology section. If the work were to be continued, it is thought that this would be a solid starting point for further research. The research which was conducted however has been useful and this has been proven by the fact that the testing results for the program were positive.

The methods used to complete the brief involved the production of the Script using MAXScript. It is thought that this was a suitable solution to the problem although as with many high level scripting languages, some of the processing could have been completed faster if the solution was programmed using the 3D Studio Max Software Development Kit (SDK) which is a mixture of C and C++ code. The advantage of the solution which has been created is that the program could be easily manipulated by any semi-advanced user who is able to alter variables within MAXScript. If the program were created using the MAXScript SDK, this code would be considerably harder for the end user to manipulate.

Building design using computer aided methods has become invaluable over recent years and many clients request BIM expertise at the bidding stage. During a recession the construction industry is generally the first to be affected (Martin, 2009). With this in mind, the work completed as part of this thesis should fit well into the workflow of visualisation specialists. Clients are demanding more work for less money and bidding on projects has become so competitive in the current market, engineering firms are bidding at a loss to keep their workforce employed and try to regain repeat business with a client. When modelling processes

can be streamlined as demonstrated in this thesis, this will in turn equate to cost savings for a project. When for example the time to create a model for visualisation is reduced by around 92%, the cost saving for an hourly charged modeller would be relative to this.

## **6.1. Improvements / Future Work**

Following the completion of the MAXScript ATK, the script was evaluated by 3 visualisation specialists, the response from this brief testing contributed to the 'future work' section.

To make the program easier to use, the main task would be to implement a function which would show a preview of the gutter profile, door handle and roof tile geometry. This would make it possible for the user to make an informed judgement on which item to select and thereby avoid the requirement for prior knowledge of the program. A preview of the materials could also be included in the MAXScript to ensure ease of use.

The most important part of the program was to provide lighting and rendering presets, this function has the potential to be expanded. The program could include many more lighting setups including different HDRI images. These could include some background, for example trees or mountains, instead of simply providing a sky dome.

The gutters function could be streamlined so that the gutters, brackets, and end caps are completed with one press of a button. If a solution could be found to automatically align all downpipe brackets to the outer face of the building, this functionality could be added to further streamline gutter modelling.

Another part of the roof tile modelling process is to provide 'Flashing'. Flashing is the material added to a tiled roof which creates a water tight seal around windows and joins. Flashing was not included in this project however future work could involve adding this functionality.

To assist the user when using the MAXScript ATK, future work could involve altering the functions so that they operate in real time. For example, instead of inputting a value and

clicking the 'Model Windows' button, the program could create the geometry automatically and when any of the values are altered, the model would update in the viewport.

A requirement which is often requested by a client is that of producing a phased construction animation. This would include animating various parts of the building process so the video highlights the construction process. A tool could be included as part of the MAXScript which would assist when creating an animation of this kind. This could involve automatically adding different parts of the model to a relevant layer and animating the visibility of all geometry on that layer. The software could easily create key frames for this animation which would save the animator time.

To take the MAXScript further some helper images could also be provided which assist the user when entering values into the spinners. For example, a novice user may not initially understand what the values relate to when entering values for the windows, ie. If the frame inset was labelled up on a helper diagram, this could be beneficial. The theory here could be applied to the doors, garage doors and gutter creation elements.

One of the elements of the Batzal Roof Designer program enabled the user to use an automated function to model the actual roof structure. This element would need to be modelled before hand when using the MAXScript ATK. There could be some benefit of including an option as part of the MAXScript ATK although most of the time this wouldn't be applicable due to the complicated nature of some of the faces on the roof.

To further improve the project, the testing stage could have been conducted on a wider range of subjects to provide a more balanced set of results.

## 7. Appendix A



Figure 7-1 Completed render 01



Figure 7-2 Completed render 02



Figure 7-3 Completed render 03



Figure 7-4 Completed render 04





Figure 7-5 Completed render 05



Figure 7-6 Completed render 06

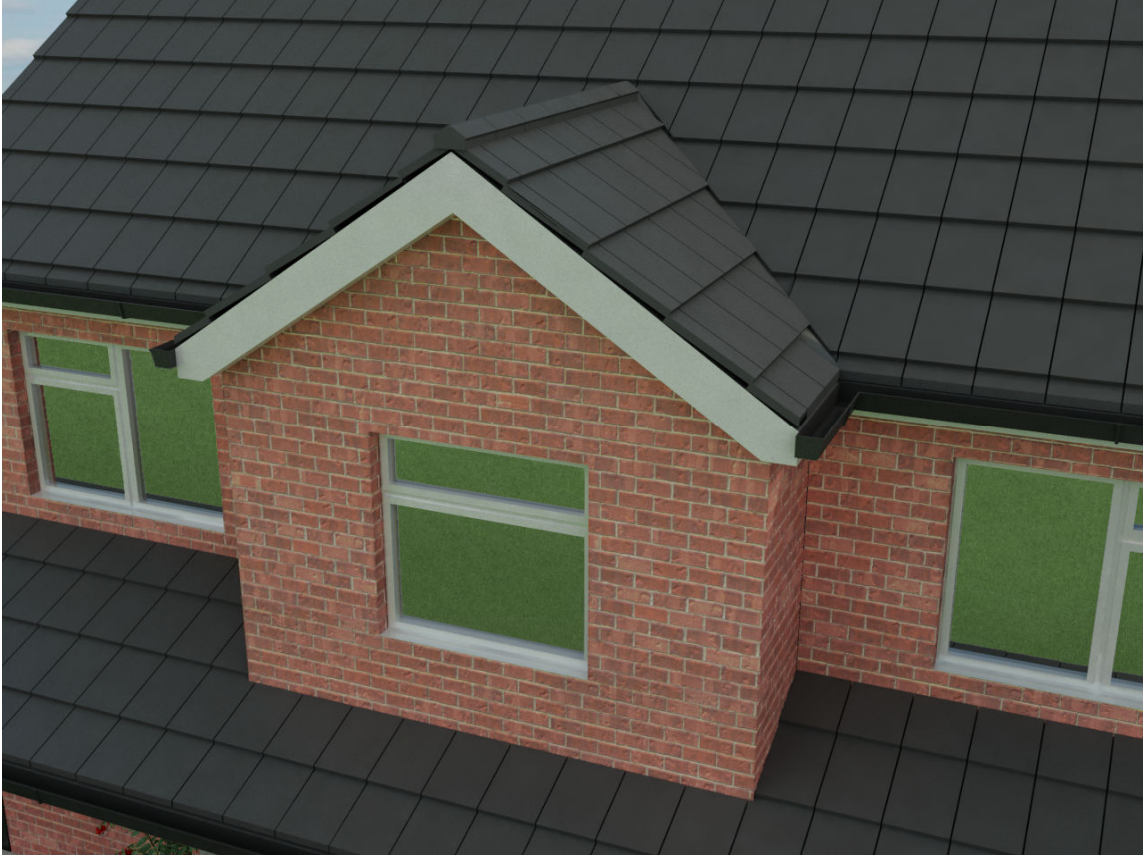


Figure 7-7 Completed render 07



Figure 7-8 Completed render 08



Figure 7-9 Completed render using HDRI lighting option

**8. Appendix B - Evaluation Questions  
using Traditional Methods**

## Roof Tiling

1. Draw a box with 4 length, 6.5 width as shown in image 1
2. Press the 'e' key to activate rotate
3. type 5 in the y entry box as shown on image 2
4. Click 'tools' > 'Array'
5. Enter 6 in x, and 60 into the 'count' box and click 'ok' as shown in image 3
6. Click 'tools' > 'Array'
7. Enter 4 in y, and 100 into the 'count' box and click 'ok' as shown in image 3
8. Hold down 'ctrl + a' to select all
9. Click 'tools' > 'Array'
10. Select one of the boxes
11. Right click > convert to > editable mesh
12. Click 'attach list' as shown in image 5
13. Press 'ctrl + a' and click attach as shown in image 6
14. Select your roof tile geometry
15. Use the 'align' tool as shown in image 7
16. Check the 'x axis' button as shown in image 8
17. Slect 'Plane01' using the 'h' key
18. Right click > convert to editably poly
19. Click the 'border' button as shown in Image 9
20. Press 'ctrl + a' to select all
21. Press the black rectangle button next to extrude, see image 10
22. Set extrusion height to 20 and extrusion base width to 0 and press ok, as shown in image 11
23. Press 'ctrl + a' to select all
24. Press 'cap' as shown in image 10
25. Select your roof tile geometry
26. Convert it to a Boolean, see image 12
27. Set Boolean settings setting to cut and remove outside as per image 13
28. Click 'pick operand b' and select 'plane01'

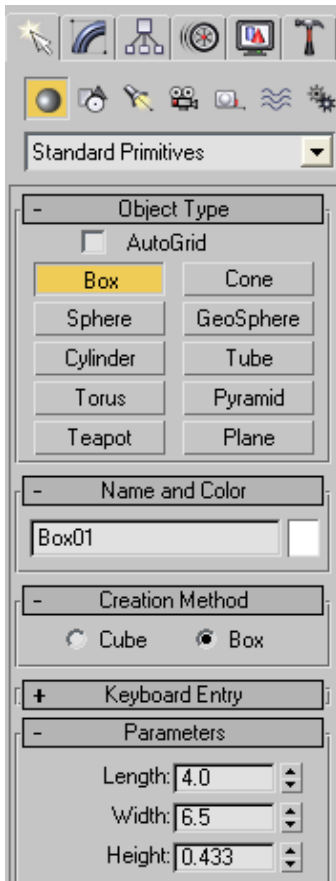


Image 1

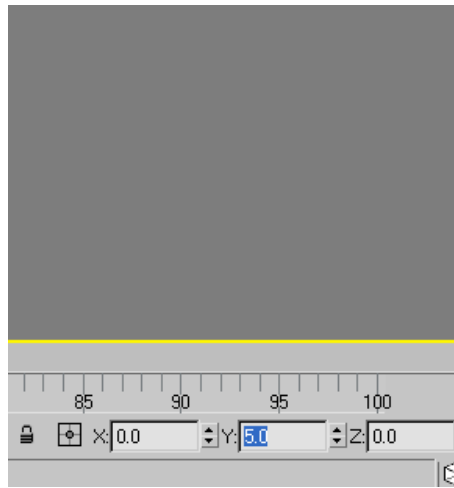


Image 2

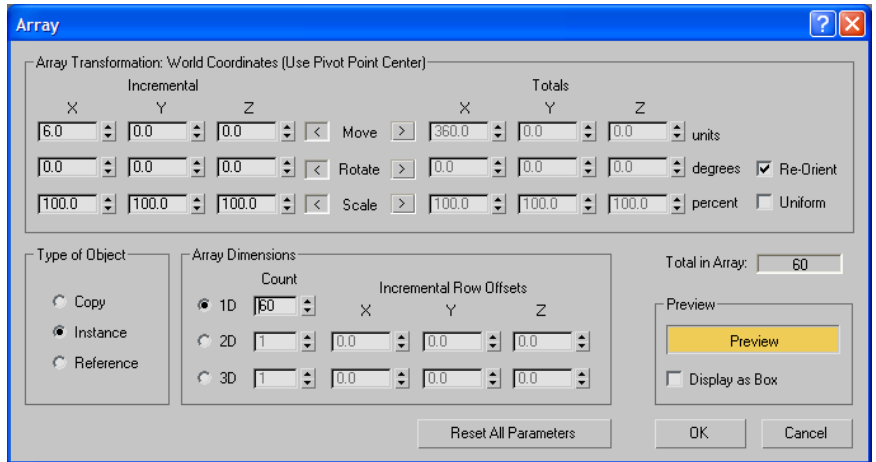


Image 3

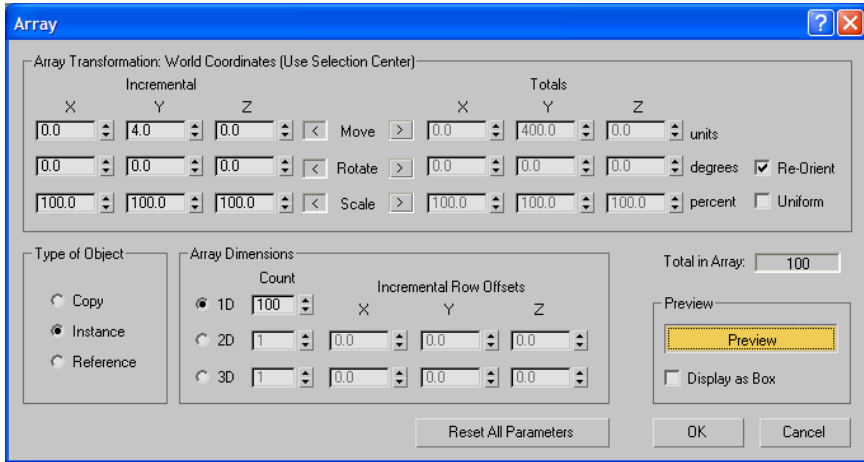


Image 4

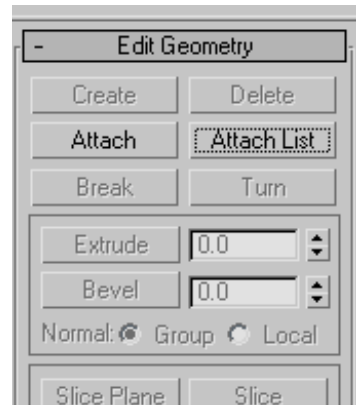


Image 5

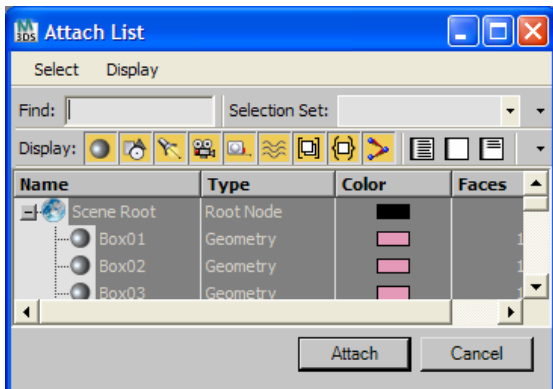


Image 6



Image 7



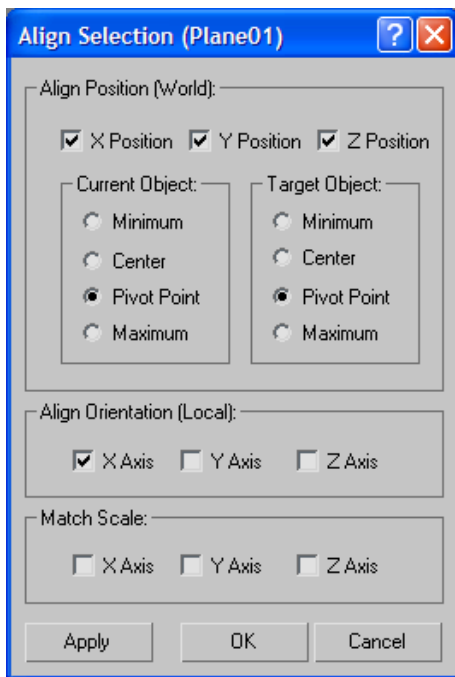


Image 8

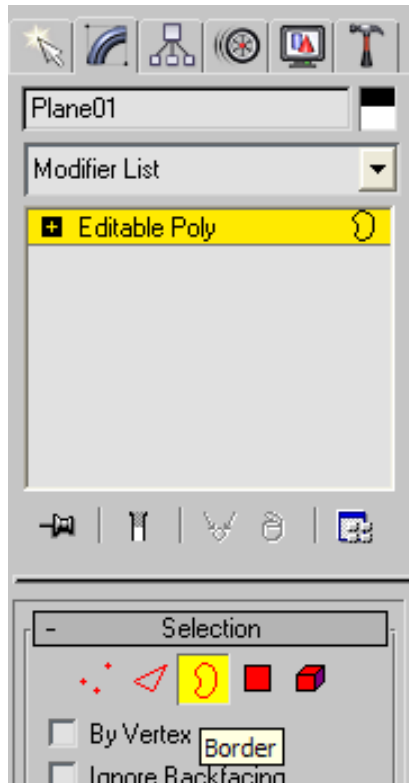


Image 9

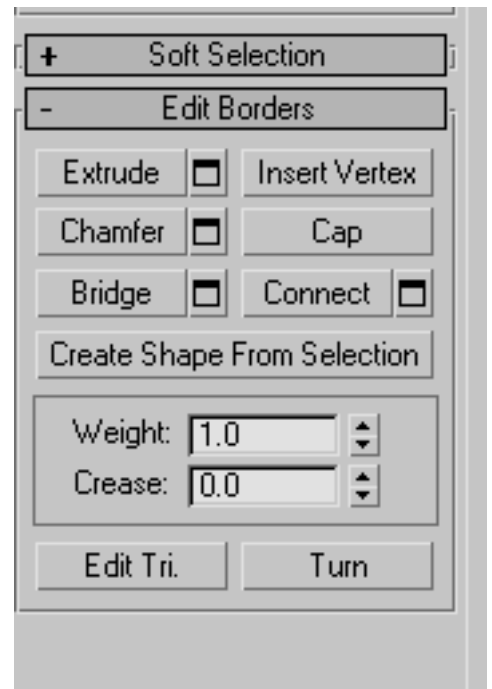


Image 10

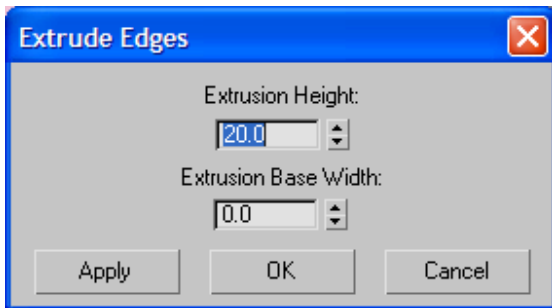


Image 11

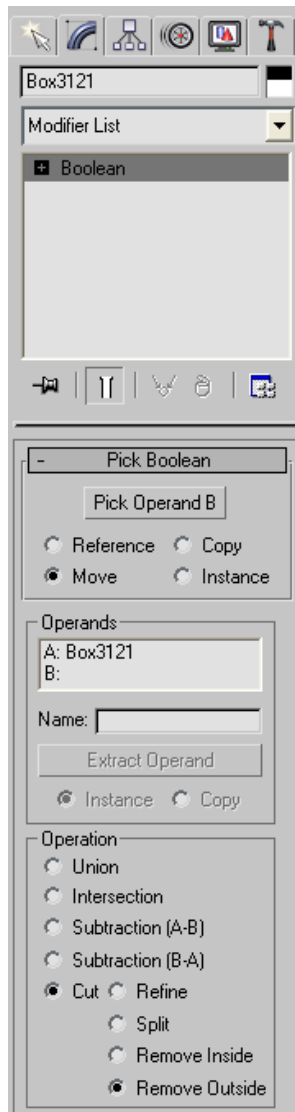


Image 13

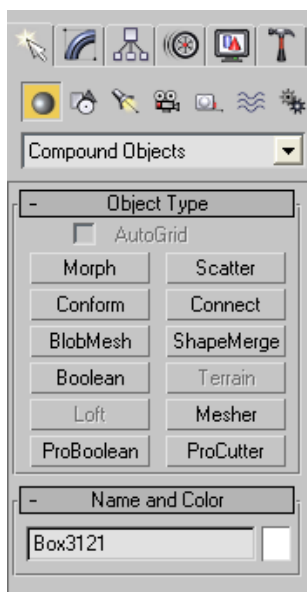


Image 12

## Gutters

29. Draw the shapes shown in image 1 using the line drawing tool shown in image 2. Hold down the shift key when drawing to ensure the lines are orthographic mode, ie constrain drawing to 90 degree angles. Press the S key to snap to the grid
30. Click the affect pivot only button shown on the image 3
31. Select each object and move the x/y arrow to the top right of each object
32. Click the affect pivot only button shown on the image 3
33. Press 'H' then Select 'Line01' and click 'OK'
34. Apply the Loft function, as shown in image 4
35. Click the 'Get Shape' button as shown in image 4 also
36. Select the middle left shape you have drawn
37. Select the top left, bottom left and top right objects you have drawn, hold down ctrl to select multiple objects.
38. Apply the extrude modifier as shown in image 5, select it from the 'modifier list', enter an 'amount' of 0.1
39. Select the top left object you have drawn
40. Use the spacing tool by pressing 'shift + I'
41. Press 'pick path' as shown on image 6 and select 'Line01'
42. Change the count to '2' and click 'apply'
43. Select the bottom left object you have drawn
44. Use the spacing tool by pressing 'shift + I'
45. Press 'pick path' as shown on image 6 and select 'Line01'
46. Check the spacing checkbox and enter '20' into the value box and click apply
47. Select the top right object you have drawn
48. Use the spacing tool by pressing 'shift + I'
49. Press 'pick path' as shown on image 6 and select 'Line02'
50. Check the spacing checkbox and enter '20' into the value box and click apply
51. Press 'H' then Select 'Line02' and click 'OK'
52. Apply the Loft function, as shown in image 4
53. Click the 'Get Shape' button as shown in image 4 also
54. Select the bottom right shape you have drawn

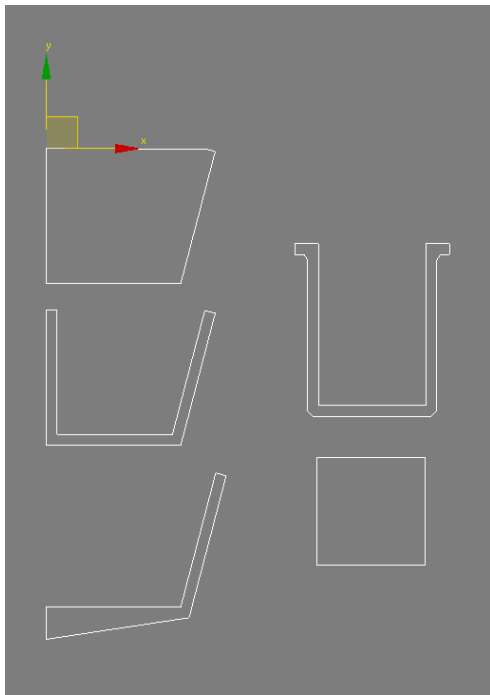


Image 1

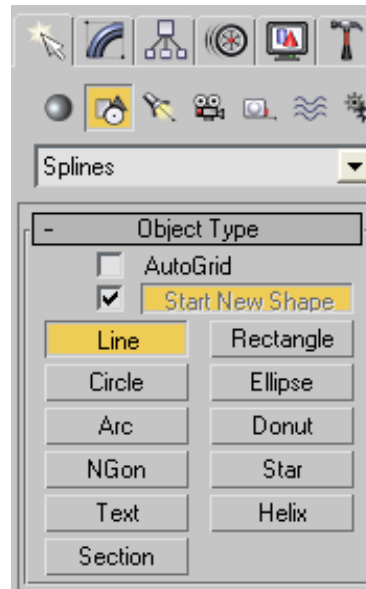


Image 2

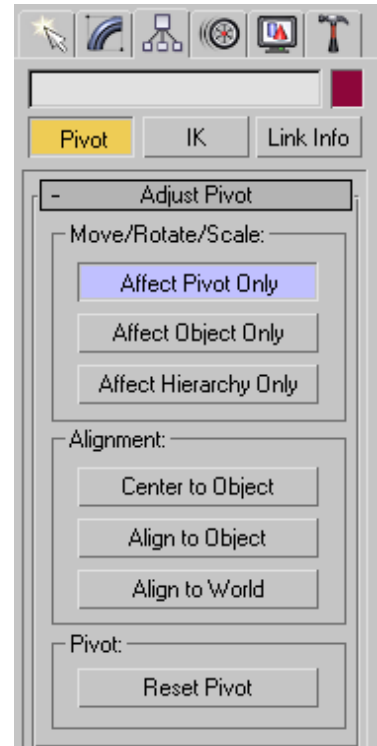


Image 3

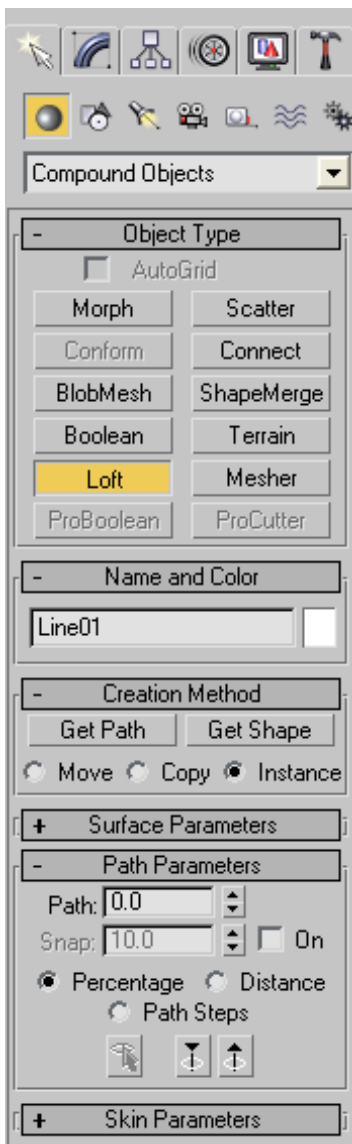


Image 4

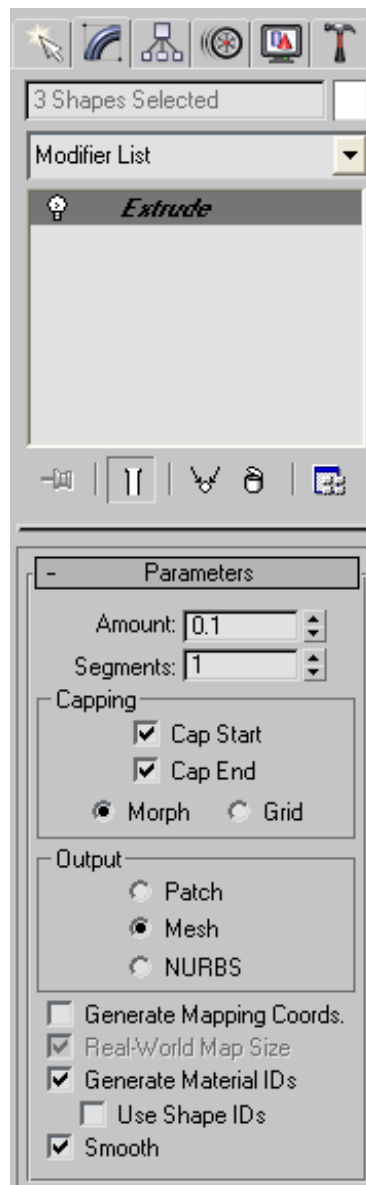


Image 5

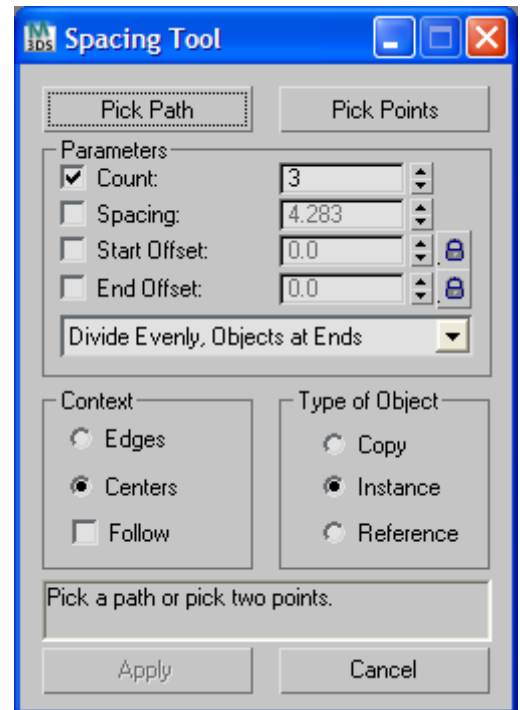


Image 6

## Windwos

55. Select 'window01' using the 'h' key
56. Right click 'convert to editable poly'
57. Select the 'polygon' mode see image 1
58. press 'ctrl + a' to select all
59. Press the 'inset' button see image 1
60. Press the black rectangle button next to the 'inset amount' to '0.5' and press 'ok' see image 2
61. Press the black rectangle button next to the 'extrude' button see image 1
62. Set the 'Extrusion height' to '0.5' and press 'ok' see image 3
63. Press the 'inset' button see image 1
64. Press the black rectangle button next to the 'inset amount' to '0.5' and press 'ok' see image 2
65. Press the black rectangle button next to the 'extrude' button see image 1
66. Set the 'Extrusion height' to '-0.5' and press 'ok' see image 3
67. Set the 'set ID' under 'Material IDs' to '2' see image 4
68. Repeat steps 1-13 for all windows in the scene.

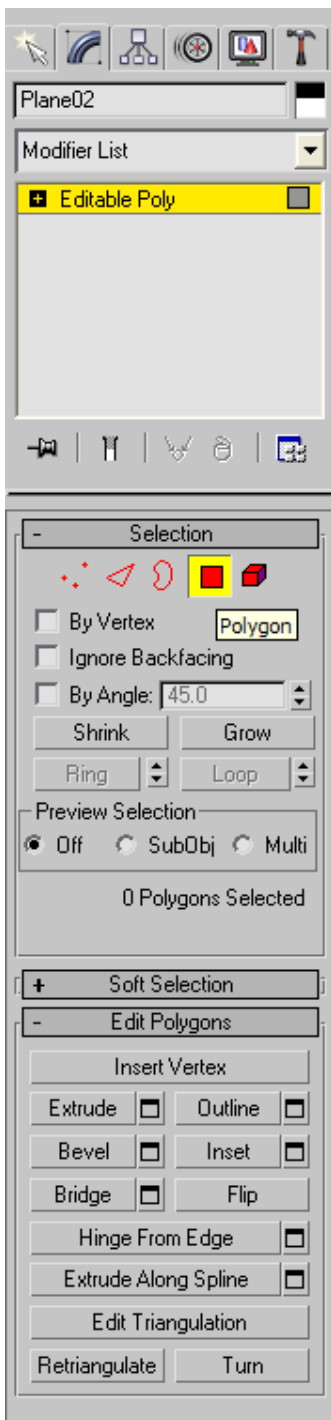


Image 1

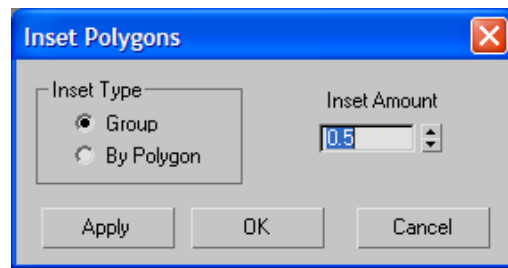


Image 2

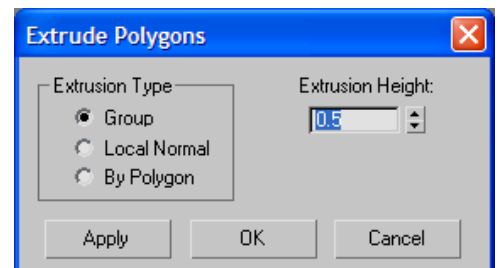


Image 3

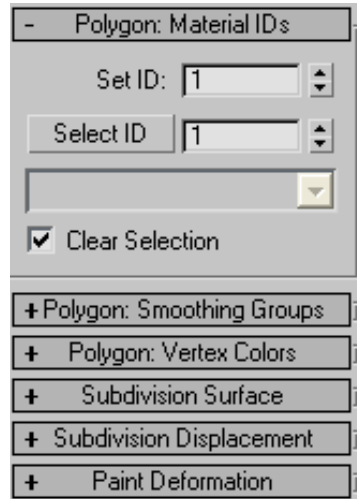


Image 4

## Doors

69. Select 'door01' using the 'h' key
70. Right click 'convert to editable poly'
71. Select the 'polygon' mode see image 1
72. press 'ctrl + a' to select all
73. Press the 'inset' button see image 1
74. Press the black rectangle button next to the 'inset amount' to '1' and press 'ok' see image 2
75. Press the black rectangle button next to the 'extrude' button see image 1
76. Set the 'Extrusion height' to '0.5' and press 'ok' see image 3
77. Press the 'inset' button see image 1
78. Press the black rectangle button next to the 'inset amount' to '1' and press 'ok' see image 2
79. Press the black rectangle button next to the 'extrude' button see image 1
80. Set the 'Extrusion height' to '-0.5' and press 'ok' see image 3
81. Set the 'set ID' under 'Material IDs' to '2' see image 4
82. Repeat steps 1-13 for any doors in the scene.

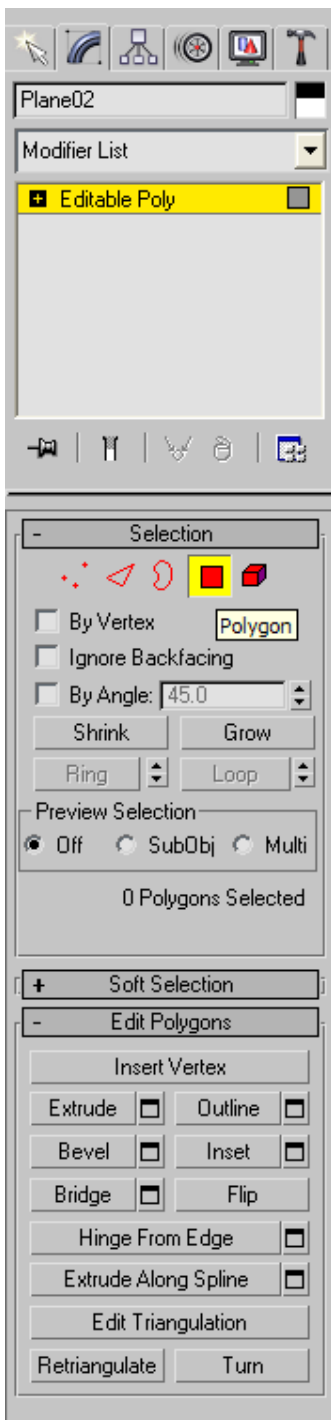


Image 1

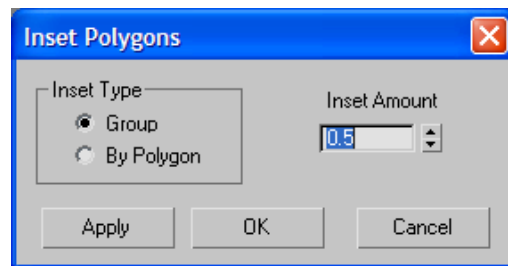


Image 2

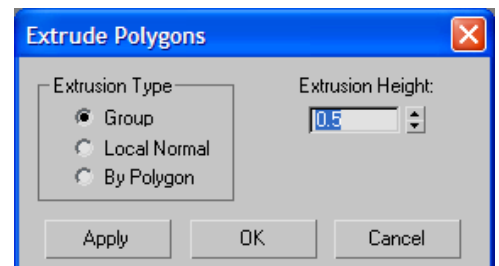


Image 3

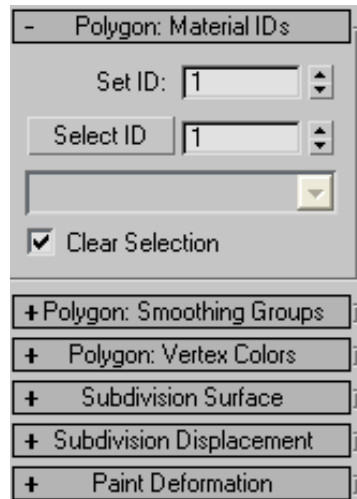


Image 4

## Lighting

83. Set 'Adaptive DMC' as Type in 'image sampler' as shown in image 2
84. Set 'area' as 'antialiasing filter' as shown in image 2
85. Set 'min rate' to 2 as shown in image 2
86. Set 'max rate' to 5 as shown in image 2
87. Uncheck 'use DMC sampler thresh'
88. Set 'clr thresh' to 0.01 as shown in image 2
89. Turn on the checkbox underneath 'gi environment (Skylight) override' see image 2
90. Set the multiplier to 2.5 underneath 'gi environment (Skylight) override' see image 2
91. Change the type of vray Colour mapping to 'exponential' see image 2
92. Set the 'dark multiplier' to 1.1 see image 2
93. Set the Bright multiplier to 0.8 see image 2
94. Turn on 'Vray Indirect Illumination' see image 1
95. Set the 'secondary bounces' 'Gi engine' to 'Light Cache' see image 1
96. Set the 'irradiance map' current preset to 'very low' see image 1
97. Set the 'HSph subdivs' to '20' see image 1
98. Set the 'vray light cache' 'subdivs' to '150' see image 1
99. Set the 'number of passes' to '4' see image 1
100. Check the 'show calc phase' checkbox see image 1
101. Insert a 'vraysun' light see image 3
102. Set the 'intensity multiplier' to '0.01' see image 3
103. Set the 'shadow subdivs' to '8' see image 3
104. Insert an 'omni light' see image 4
105. Uncheck the 'shadows' 'on' checkbox see image 4
106. Click the 'Exclude' option and select 'Include' and click 'ATK\_Sky' and press the '>>' button then 'ok' see image 5



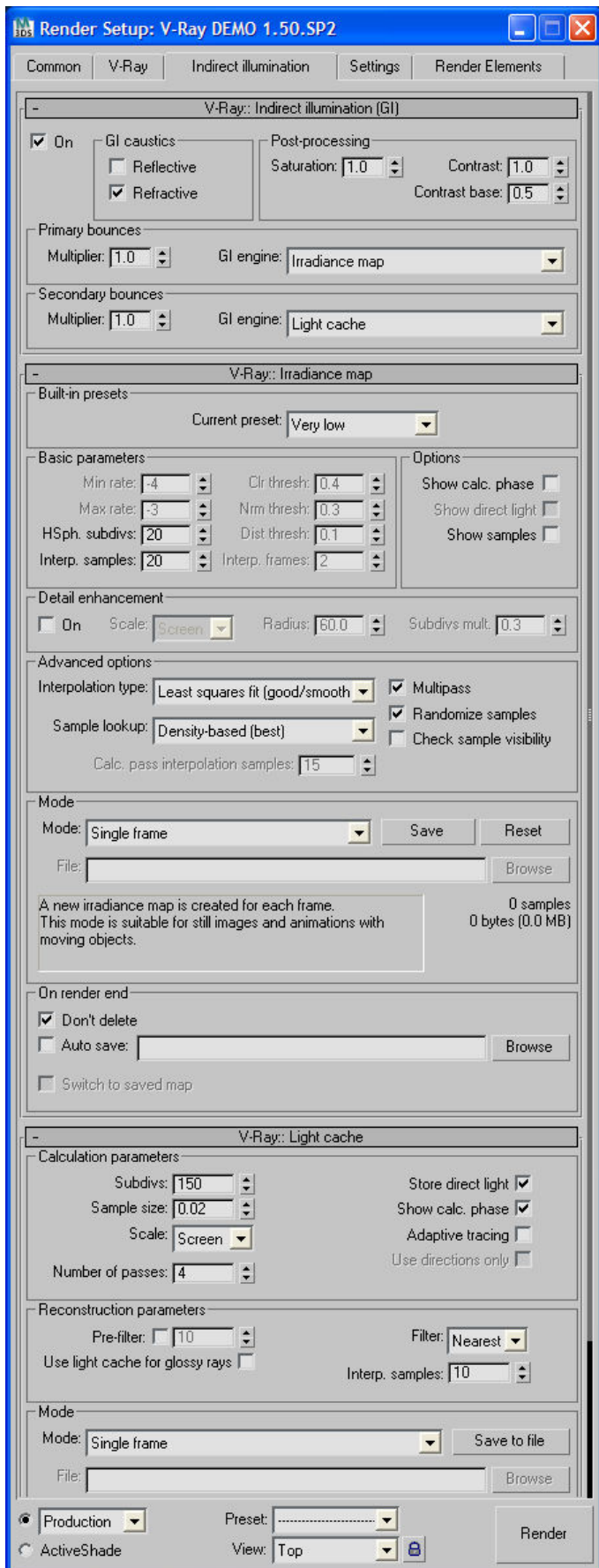


Image 1

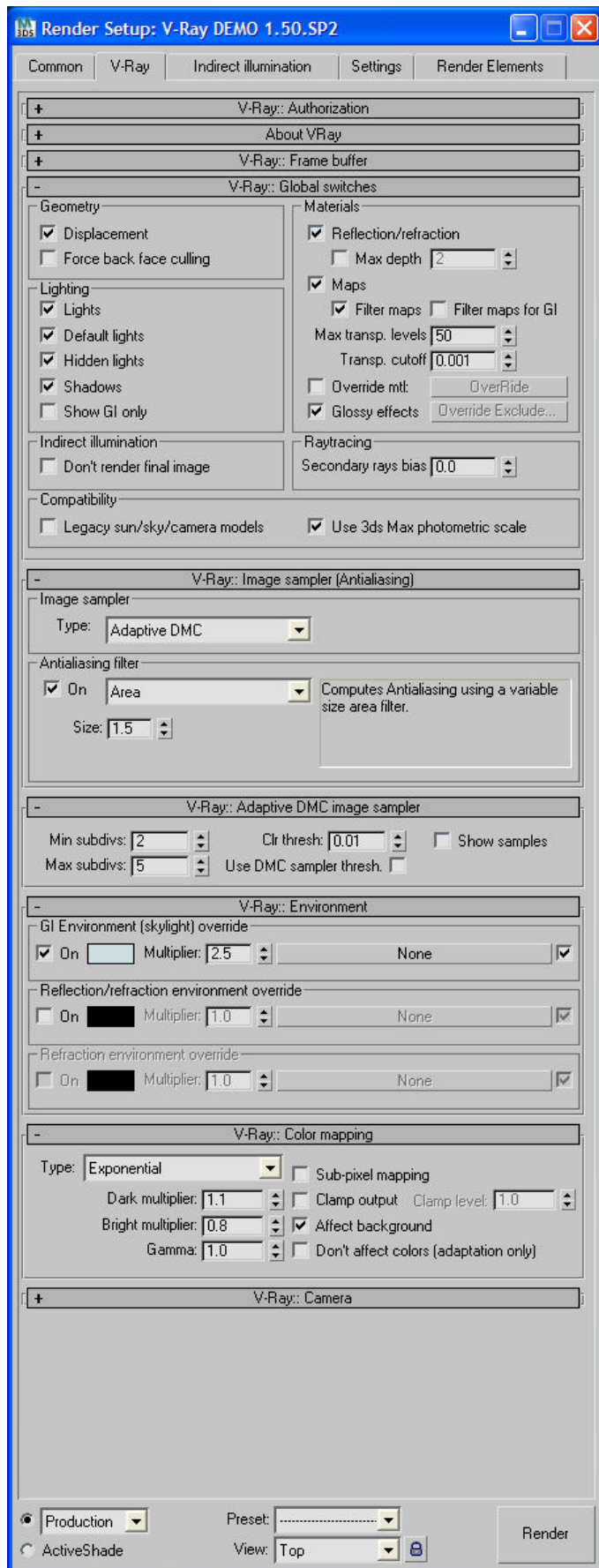


Image 2

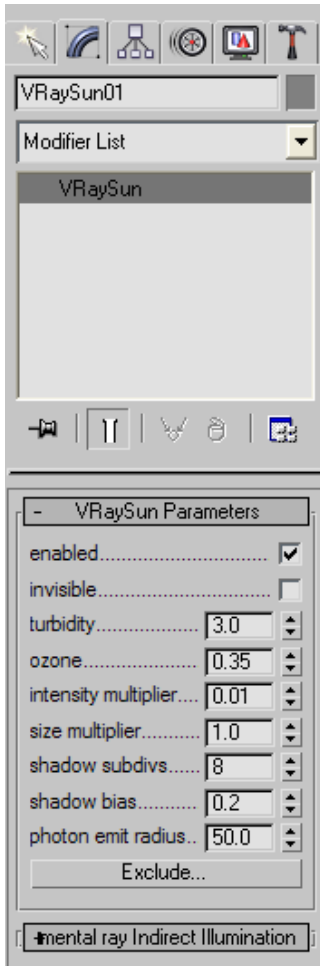


Image 3

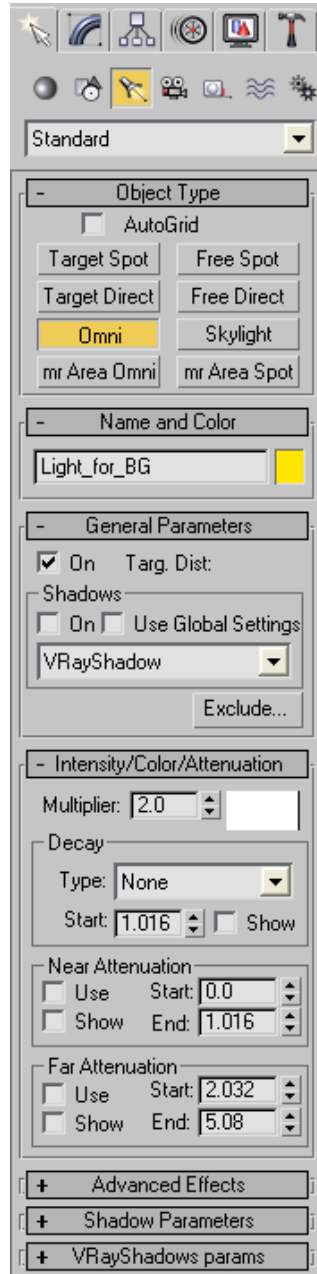


Image 4

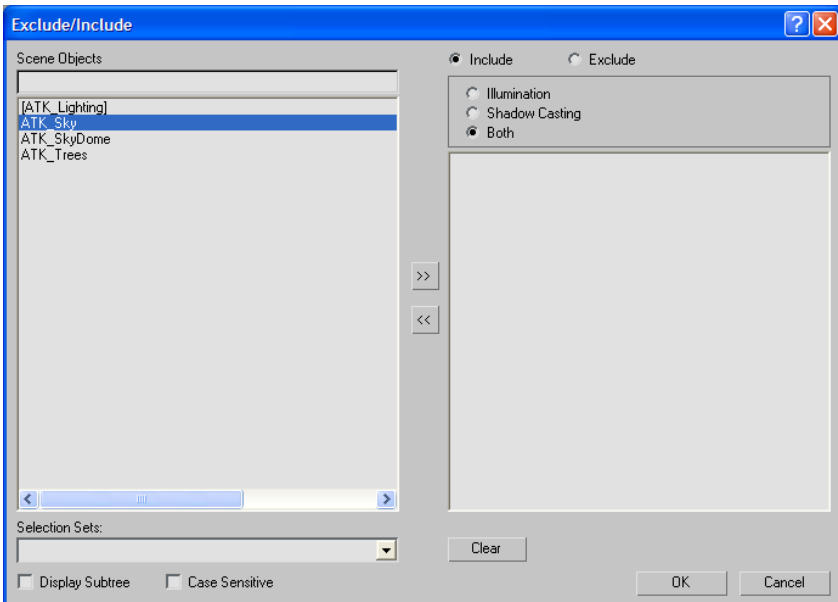


Image 5

**9. Appendix C – Evaluation Questions  
using MAXScript ATK**

## Tiling ATK

1. Select 'Tile Reference' '3' under the 'Roof Tiles' rollout of the MAXScript ATK see image 1



Image 1

## Gutters ATK

1. Set the 'Profile Ref' to '1' in the 'Guttering' rollout of the MAXScript ATK see image 1
2. Set the 'Support Spacing' to '20' see image 1
3. Press the 'Guttering' button see image 1
4. Press the 'Downpipe' button see image 1
5. Press the 'Gutter Brackets' button see image 1
6. Press the 'End Caps' button see image 1
7. Press the 'Downpipe Brackets' button see image 1

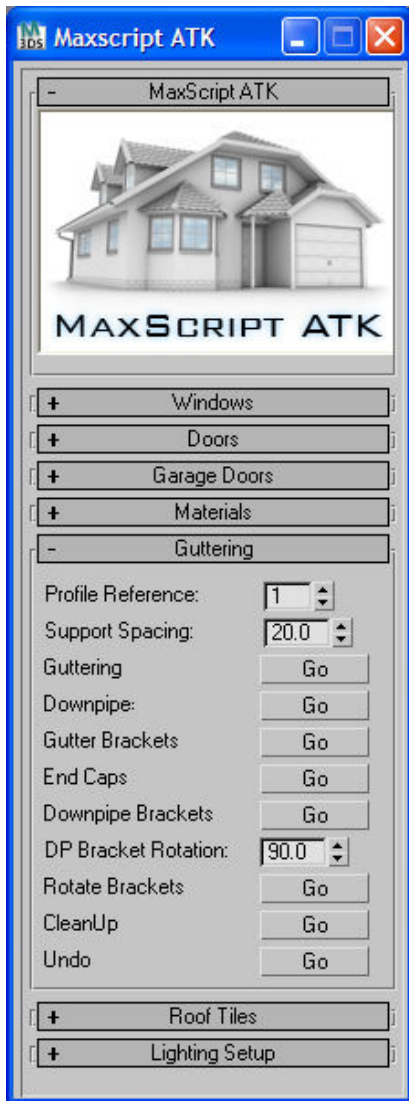


Image 1

## Windows ATK

1. Set the 'Frame Width' to '0.5' under the 'Windows' rollout of the MAXScript ATK see image 1
2. Set the 'Frame Width 1' to '0.2' see image 1
3. Set the 'Frame Return' to '0.5' see image 1
4. Set the 'Frame Offset' to '0.15' see image 1
5. Set the 'Frame Width to '0.5' see image 1
6. Press the 'Model Windows' button see image 1

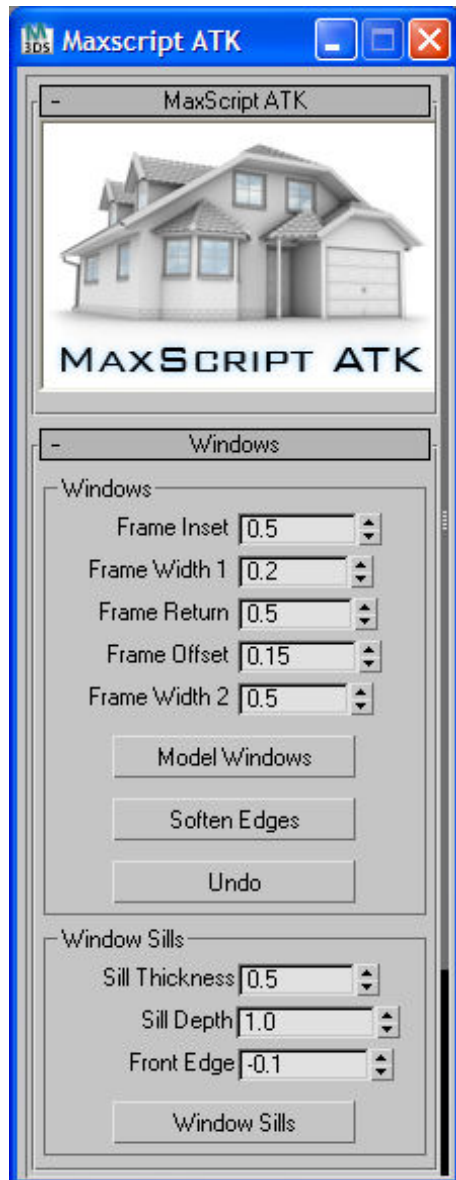


Image 1

## Doors ATK

1. Insert 0.75 into the 'Frame Inset' box see image 1
2. Insert 0.5 into the 'Width of Frame' box see image 1
3. Insert 0.1 into the 'Frame Depth' box see image 1
4. Press the 'Model Doors' Button see image 1

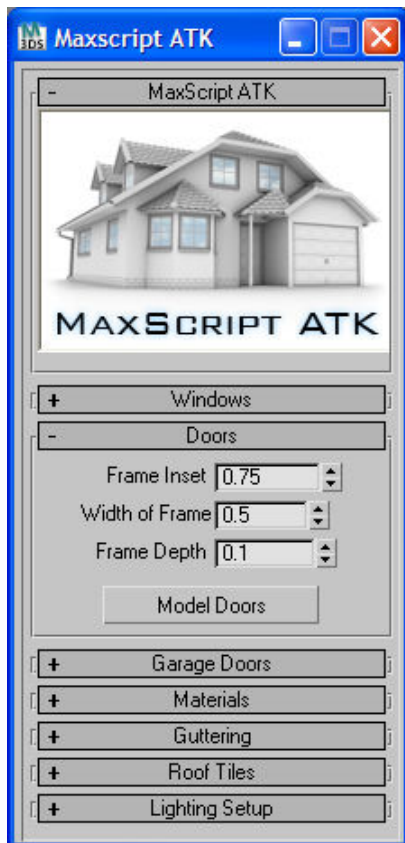


Image 1

## Lighting ATK

1. Hit the Daylight Sunny Option which is Located in the 'Lighting setup' rollout of the MAXScript ATK, see image 1



Image 1



## **10. Glossary**

DTM – Digital Surface Model

DSM – Digital Terrain Model

iSM - Instant Scene Modeller

LiDAR - Light Detection and Ranging

## BIBLIOGRAPHY

- ALEJANDRO F. FRANGI, D. R., JULIA A. SCHNABEL, AND WIRO J. NIESSEN 2002. Automatic Construction of Multiple-Object Three-Dimensional Statistical Shape Models: Application to Cardiac Modeling. *IEEE TRANSACTIONS ON MEDICAL IMAGING*, VOL. 21, 1151-1166.
- ALIAGA, D. G., ROSEN, P. A. & BEKINS, D. R. 2007. Style Grammars for Interactive Visualization of Architecture. *IEEE Trans. Vis. Comput. Graph.*, 786-797.
- ALLEN, P., TROCCOLI, A. & BLAER, P. 2005. Automating the 3D Modeling Pipeline. *International Symposium on the CREST Digital Archiving Project*.
- BARNES, D. G. & FLUKE, C. J. 2008. Incorporating interactive three-dimensional graphics in astronomy research papers. *New Astronomy*, 599-605.
- BLUMBERG, B. & GALYEAN, T. A. 1995. Multi-Level Direction of Autonomous Creatures for Real-Time Virtual Environments. *SIGGRAPH '95 Proceedings of the 22nd annual conference on Computer graphics and interactive techniques*, 47-54.
- BOULOS, M. N. K., HETHERINGTON, L. & WHEELER, S. 2007. Second Life: an overview of the potential of 3-D virtual worlds in medical and health education. *Health Information & Libraries Journal*, 24, 233-245.
- BUXTON, W., FITZMAURICE, G., BALAKRISHNAN, R. & KURTENBACH, G. 2000. Large Displays in Automotive Design. *IEEE Comput. Graph. Appl.*, 20, 68-75.
- CALDAS, L. & NORFORD, L. 2001. Architectural Constraints in a Generative Design System: interpreting energy consumption levels. *Seventh International IBPSA Conference*, 1397-1404.
- CHASSAING, F., CHOQUET, B. & PELÉ, D. 1991. A stereoscopic television system (3D-TV) and compatible transmission on a MAC channel (3D-MAC). *Signal Processing: Image Communication*, 4, 33-43.
- DEBEVEC, P. 1999. Modeling and Rendering Architecture from Photographs. *In SIGGRAPH 99*, 11-20.
- DEUSSEN, O., HANRAHAN, P., LINTERMANN, B., RADOM, M., R., CH, PHARR, M. & PRUSINKIEWICZ, P. 1998. Realistic modeling and rendering of plant ecosystems. *Proceedings of the 25th annual conference on Computer graphics and interactive techniques*. ACM.
- DICK, A., TORR, P. & R., C. 2000. Automatic 3d modelling of architecture. *British Machine Vision Conference*, 372-381.
- DOERR, K.-U., RADEMACHER, H., HUESGEN, S. & KUBBAT, W. 2007. Evaluation of a Low-Cost 3D Sound System for Immersive Virtual Reality Training Systems. *IEEE Transactions on Visualization and Computer Graphics*, 13, 204-212.
- EBERT, D. S., MUSGRAVE, F. K., PEACHEY, D., PERLIN, K. & WORLEY, S. 2002. *Texturing and Modeling, Third Edition: A Procedural Approach*, San Francisco, Calif., Morgan Kaufmann.
- ENCYCLOPEDIA-BRITANNICA 2011.

- FLACK, P. A., WILLMOTT, J., BROWNE, S. P., ARNOLD, D. B. & DAY, A. M. 2001. Scene assembly for large scale urban reconstructions. *Proceedings of the 2001 conference on Virtual reality, archeology, and cultural heritage*. Glyfada, Greece: ACM.
- FUNKHOUSER, T., TELLER, S., SÉQUIN, C. & KHORRAMABADI, D. 1996. The UC Berkeley System for Interactive Visualization of Large Architectural Models. *Presence*, 5, 13-44.
- GLENCROSS, M., WARD, G. J., MELENDEZ, F., JAY, C., LIU, J. & HUBBOLD, R. 2008. A perceptually validated model for surface depth hallucination. *ACM Trans. Graph.*, 27, 1-8.
- HARLEY, J. B. 1968. The Evaluation of Early Maps: Towards a Methodology. *Imago Mundi*, 22, 62-74.
- HOWARD, R. & BJÖRK, B.-C. 2008. Building information modelling - Experts' views on standardisation and industry deployment. *Advanced Engineering Informatics*, 22, 271-280.
- HU, Z., MELLOR, J., WU, J. & DELISI, C. 2004. VisANT: an online visualization and analysis tool for biological interaction data. *BMC Bioinformatics*, 5, 17.
- IBRAHIM, M. & KRAWCZYK, R. 2004. Two Approaches to BIM: A Comparative Study. *Education in Computer Aided Architectural Design in Europe*, eCAADe 2004 Conference, Copenhagen, Denmark.
- JACOB, R. J. K. 1998. User Interfaces. *Encyclopedia of Computer Science*, Fourth Edition.
- KERWIN, J., SCOTT, M., SHARPE, J., PUELLES, L., ROBSON, S., MARTINEZ-DE-LA-TORRE, M., FERRAN, J., FENG, G., BALDOCK, R., STRACHAN, T., DAVIDSON, D. & LINDSAY, S. 2004. 3 dimensional modelling of early human brain development using optical projection tomography. *BMC Neuroscience*, 5, 27.
- LAMA, S., JOHNSON, C., MAFFEI, K. & BOUSQUET, M. 2007. *3ds Max MAXScript Essentials*, Focal Press.
- LAPEER, R. J., ROWLAND, R. S. & CHEN, M. S. 2004. PC-based Volume Rendering for Medical Visualisation and Augmented Reality based Surgical Navigation. *Proceedings of the Information Visualisation, Eighth International Conference*. IEEE Computer Society.
- LEE, S. C., JUNG, S. K. & NEVATIA, R. 2002. Automatic Pose Estimation of Complex 3D Building Models. *Proceedings of the Sixth IEEE Workshop on Applications of Computer Vision*. IEEE Computer Society.
- LIPP, M. A. W., PETER AND WIMMER, MICHAEL 2008. Interactive visual editing of grammars for procedural architecture. *ACM Trans. Graph.*, 27, 102:1--102:10.
- MACMILLAN-DICTIONARY 2011.
- MARTIN, P. 2009. The Recession and Migration: Alternative Scenarios. *Working Papers*, Paper 13, 14.
- MATUSIK, W. & PFISTER, H. 2004. 3D TV a scalable system for real-time acquisition, transmission, and autostereoscopic display of dynamic scenes. *ACM Transactions on Graphics*, 23, 814-824.
- MCKENZIE, F. D., PETTY, M. D., KRUSZEWSKI, P. A., GASKINS, R. C., NGUYEN, Q.-A. H., SEEVINCK, J. & WEISEL, E. W. 2008. Integrating crowd-behavior modeling into military simulation using game technology. *Simulation & Gaming*, 39, 10-38.

- MELLENDEZ, F., GLENCROSS, M., WARD, G. J. & HUBBOLD, R. J. 2011. Relightable Buildings from Images. *EuroGraphics 2011*, Areas Paper, 33–40.
- NUNES CORREIA, F., CASTRO REGO, F., DA GRAÇA SARAIVA, M. & RAMOS, I. 1998. Coupling GIS with Hydrologic and Hydraulic Flood Modelling. *Water Resources Management*, 12, 229-249.
- PAAR, P., APPLETON, K., CLASEN, M., GENSEL, M., JUDE, S. & LOVETT, A. 2008. Interactive Visual Simulation of Coastal Landscape Change. *Digital Earth Summit on Geoinformatics 2008: Tools for Global Change Research*. Wichmann, Heidelberg., 153-159.
- PANAS, T., EPPERLY, T., QUINLAN, D., SAEBJORNSEN, A. & VUDUC, R. 2007. Communicating Software Architecture using a Unified Single-View Visualization. *Engineering of Complex Computer Systems*, 12th IEEE International Conference on Engineering Complex Computer Systems, 217-228.
- PARISH, Y. I. H. & MÜLLER, P. 2001. Procedural Modeling of Cities. *Proceedings of ACM SIGGRAPH 2001*, 301-308.
- ROBERTS, A. 2001. ECOTECT: Environmental Prediction in Architectural Education. *19th ECAADE - Education for Computer Aided Architectural Design in Europe, Helsinki, Finland*.
- SALISBURY, K., CONTI, F. & BARBAGLI, F. 2004 Haptic rendering: introductory concepts. *Computer Graphics and Applications, IEEE* 24, 24 - 32
- SE, S. & JASIOBEDZKI, P. 2005. Instant Scene Modeler for Crime Scene Reconstruction. *Computer Vision and Pattern Recognition Workshop*, 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPRW'05), 123.
- SHEPPARD, S. R. J. 2001. Guidance for crystal ball gazers: developing a code of ethics for landscape visualization. . *Landscape and Urban Planning*, 54(1-4), 183-199.
- SHIODE, N. 2000. 3D urban models: Recent developments in the digital modelling of urban environments in three-dimensions. *GeoJournal*, 52, 263-269.
- STASKO, J. T. 1992. Three-Dimensional Computation Visualization. *In Proceedings of the 1993 IEEE Symposium on Visual Languages*, 100-107.
- WHYTE, J., BOUCHLAGHEM, N., THORPE, A. & MCCAFFER, R. 2000. From CAD to virtual reality: modelling approaches, data exchange and interactive 3D building design tools. *Automation in Construction*, 10, 43-55.
- YAN, W. 2006. Integrating web 2D and 3D technologies for architectural visualization: applications of SVG and X3D/VRML in environmental behavior simulation. *Proceedings of the eleventh international conference on 3D web technology*. Columbia, Maryland: ACM.