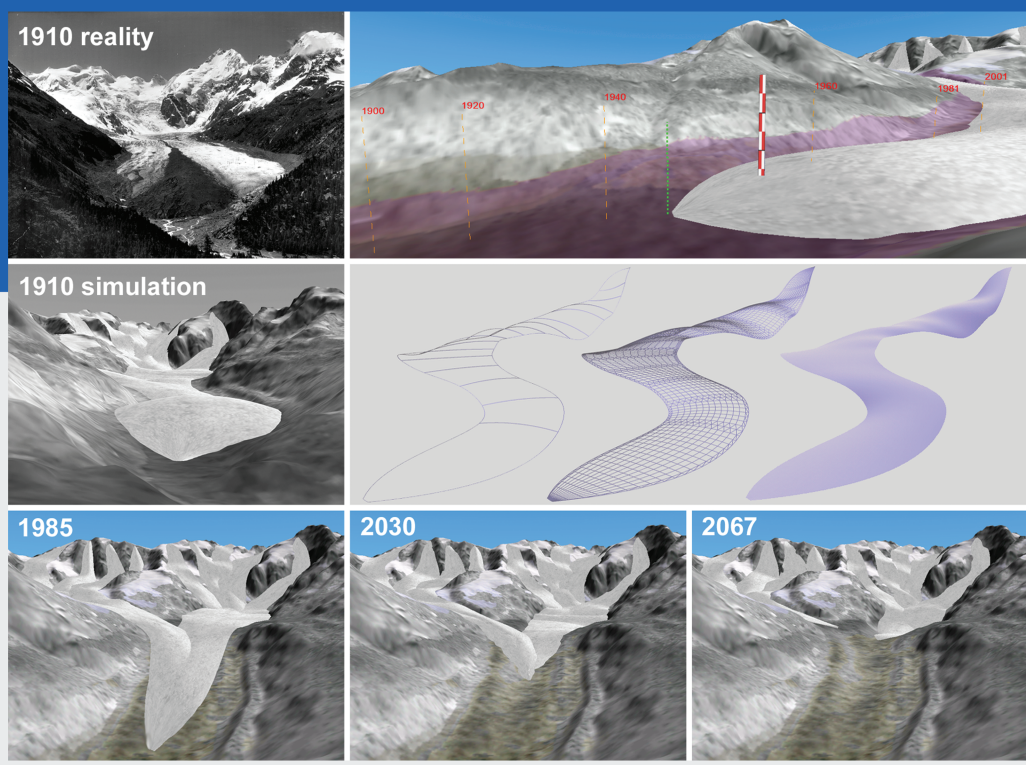
Remote Sensing Series 43

Stefan Biegger

A Visual System for the Interactive Study and Experimental Simulation of Climate-induced 3D Mountain Glacier Fluctuations



Remote Sensing Laboratories
Department of Geography
University of Zurich, 2004

Remote Sensing Series 43

Stefan Biegger

A Visual System for the Interactive Study and Experimental Simulation of Climate-induced 3D Mountain Glacier Fluctuations

Remote Sensing Laboratories
Department of Geography
University of Zurich, 2004

Author:
Stefan Biegger
Remote Sensing Laboratories RSL
Department of Geography
University of Zurich
Winterthurerstrasse 190

CH-0857 Zurich
Switzerland

http://www.geo.unizh.ch

Biegger, Stefan:
A Visual System for the Interactive Study and Experimental Simulation of Climate-induced 3D Mountain Glacier Fluctuations. Remote Sensing Series, vol. 43 (CD-ROM), Remote Sensing Laboratories, Department of Geography, University of Zurich, 2004.
ISBN 3-03703-009-7

Printed by the *Druckerei der Stiftung Zentralstelle der Studentenschaft der Universität Zürich*.

Die vorliegende Arbeit wurde von der Mathematisch-naturwissenschaftlichen Fakultät der Universität Zürich im Wintersemester 2004/05 aufgrund der Gutachten von Prof. Dr. K. I. Itten (Universität Zürich), Prof. Dr. D. Nüesch (Universität Zürich) und Prof. Dr. J. Döllner (Universität Potsdam, Deutschland) als Dissertation angenommen.

Summary

The environment is subject to change due to natural and anthropogenic causes. In order to be able to anticipate the future development of the environment and to take suitable steps in time, the understanding of the relationships and changes is of great importance to society. But the more sophisticated these processes are, the more difficult is their analysis for scientists. Furthermore, non-experts need also be aware of the background to environmental changes. In this context, an experiental way of learning is preferrable to a one-way communication that leads to a passive role for the target group. The active role in experiential learning enhances the ability to grasp knowledge and helps establish personal responsibility and environmental competence. The climate-induced changes of mountain glacier fluctuations are considered as an example of an environmental process. There is hardly another process that illustrates the impact of global change on the environment in a clearer and more obvious way.

The use of visualisation techniques is a well-suited approach to gain insights to complex processes, as they exploit the dominance of the human visual channel. Due to the huge efforts recently made in computer graphics, techniques for interactive and three-dimensional visualisation have been established.

In this study, a visual system is developed that provides functionality for the interactive study and experimental simulation of climate-induced fluctuations of mountain glaciers. The methodological basis of the system is the so-called experimental simulation loop that is complemented using visual analysis methods. An experiment corresponds to a single pass of the simulation loop. Assuming an initial hypothesis, a new state of the glacier can be simulated that is converted to a visual representation for rendering purposes. After the visual analysis of the simulation results and a comparison with reference information, the hypothesis may be verified or modified. When conducting several experiments, the results can be compared with previous values leading to new empirical knowledge.

The system is implemented as a prototype software that extends an existing system specialized in terrain visualisation using an efficient method for terrain rendering. The OpenSceneGraph library is used for the modelling of the scene graph and the implementation of more sophisticated rendering methods. It uses OpenGL for low-level rendering purposes. Since the system is implemented as a closed system, no conceptual considera-

tions are made concerning a more generic visual simulation system. Therefore, the simulation and rendering methods are integrated relying on the same data and method base without any additional interfaces. A graphical user interface is provided to steer the simulation experiments, to change physical or geometric parameters, or to navigate in time. Selected results of the conducted experiments are stored in an experiment repository allowing later analysis with conventional tools outside the system.

Appropriate possibilities for the numerical and visual evaluation of the simulation results are required for the analysis and validation of experiments. In the developed system, several approaches for the navigation in space and time and for the control of experiments are implemented. For a task-oriented navigation in space, three different navigation metaphors are used for the exploration of a glacier, two for manoeuvring purposes, and two for search tasks. The implemented methods for the visual analysis either reduce the information complexity, e.g. dimensionality, or blend in additional information for validating an experiment by applying modern techniques of computer graphics. Interactive approaches for the numerical extraction of information allow in-situ analysis in the georeferenced three-dimensional space. Changing the settings for the surface rendering, e.g. by modifying shading or level of abstraction, enables further possibilities for the visual investigation.

Five existing glacier models with a range of complexities and applicabilities were evaluated. Only one of these models is able to describe climate-induced fluctuations of mountain glaciers without previous knowledge of the shape of the glacier bed. This one-dimensional glacier model serves as the basis for a new method for simulating three-dimensional glacier fluctuations.

For simulation and visual modelling purposes, a glacier consists of several individual branches. The number of branches and their courses are defined according to the morphology that is derived from a digital terrain model. Each branch is considered independently for the simulation as well as for the visual modelling. No interactions are taken into account.

The implemented simulation method allows computation of the three-dimensional glacier bed as well as a new steady-state of the glacier due to a change to the climatic conditions. The method assumes steady-state conditions and temperate ice, does not consider any dynamic effects and is applicable only for simulating long-term fluctuations in a range between 20 and 100 years. In contrast to these limitations, the method is well-suited for modelling glaciers all over the world because only a few input parameters are required and the glacier bed needs not be known in advance. The calculation of the glacier thickness is based on the slope of the glacier surface in the flow direction and is derived from a digital terrain model. To estimate the width, an empirical relation between thickness and width is applied. A step-wise adjustment to a given change in the mass balance is assumed for the calculation of a new steady-state that is achieved after a glacier-specific time delay.

For the visual modelling, a glacier is regarded as the sum of several surfaces each of them representing an individual branch. A cubic Non-Uniform Rational B-Spline (NURBS) curve is applied to parameterise the course of a branch in flow-direction. By using a cubic NURBS surface for specifying geometrically a single branch surface an efficient and continuous description of the smooth shape and the parabolic cross-section is

possible. A computer-generated two-dimensional texture is mapped onto a branch surface to achieve a geotypical appearance. Gouraud shading improves the perception of the surface shape.

The visual simulation of mountain glacier fluctuations consists of a physically-based static deformation of the branch surfaces and a procedural animation technique applied for the dynamic morphing. The dynamic transition from one steady-state to another steady-state is generated by a three-dimensional surface-based method. The correspondence of the two surfaces is described using the control net of the parametric form. The central control points are animated along a parametric terrain curve. For simulating a time delay between the change of thickness and length, an exponential morphing is introduced at the tongue. An animated texture simulates the flow at the glacier surface.

In a case study, the developed methods are applied for the simulation and visual analysis of the Morteratschgletscher in Switzerland. The geometric modelling of a glacier as the sum of NURBS surfaces has proven itself for the visual modelling as well as for the rendering. The animation of glacier fluctuations is made possible by the combined use of a physically-based deformation and a descriptive morphing in real-time. Due to the absence of extensive measurements, it was only possible to validate the simulated glacier bed with one point measurement resulting in deviations in the range of 10%. The simulated bed shows an over-deepening of approximately 100 metres. The horizontal area of the simulated glacier in the year 1985 is 24% smaller compared to reality, mainly due to the underestimated accumulation area. Based on an extrapolation, a realistic glacier area would be achieved when using approximately 19 branches.

Regarding the simulated thickness and width of the maximum stand in 1850/60 and the extent in 1910, a good accordance with the known dimensions was observed.

In order to illustrate the principle of the experimental simulation loop and to reconstruct the measured retreat of the Morteratschgletscher since 1850, four different experiments were conducted. In the first three experiments, the influence of the simulation parameters mass balance gradient, altitude and sensitivity of the equilibrium line were identified. In the fourth experiment, it was possible to optimise the values of the mass balance gradient (0.68 mwe / 100 m) and the sensitivity of the equilibrium line (160 m / °C) in a step-wise manner. The temporal behaviour was reconstructed by adjusting the mass balance gradient. The spatial behaviour was adjusted by optimzing the local sensitivity of the equilibrium line and the trend of the air temperature. Further investigations are required to verify the presented optimisation approach.

The optimised simulation parameters were used to simulate possible fluctuations of the Morteratschgletscher from 1985 to 2100. Three different IPCC (Intergovernmental Panel on Climate Change) climate scenarios were applied to describe the climate change. The simulated retreat varies between 2 km and 4 km; the volume decreases by 31% - 77%. Extended measurements and further experiments are needed to validate the developed simulation method. A comparison with other simulation models would help to detect systematic errors.

It has been shown that the implemented visual system enables the analysis and supports experiental learning using simulated fluctuations of a three-dimensional mountain glacier. For this, a wide range of methods is required for navigation in space and time, control of an experiment, extraction of information and visual analysis.

Zusammenfassung

Unsere Umwelt ist einem ständigen Wandel unterlegen, sei es aufgrund natürlicher oder anthoropogener Einflüsse. Um zukünftige Entwicklungen unseres natürlichen Lebensraumes frühzeitig zu antizipieren und entsprechende Massnahmen rechtzeitig treffen zu können, ist das Verständnis der natürlichen Zusammenhänge und Veränderungen für die Gesellschaft von grosser Bedeutung. Mit zunehmender Komplexität der berücksichtigten Prozesse wird es jedoch für Wissenschaftler schwierig, die simulierten Sachverhalte adäquat zu analysieren. Zudem ist es notwendig, dass nicht nur Wissenschaftler sondern auch Personen ohne Expertenwissen die Hintergründe der natürlichen Veränderungen erkennen. Eine aktive Auseinandersetzung durch empirisches Lernen ist dabei einer passiven Kommunikation vorzuziehen, da sie zu einer gesteigerten Aufnahmebereitschaft und somit zu erhöhter Selbstverantwortung und Kompetenz bezüglich umweltrelevanter Aspekte beitragen kann. Als Beispiel eines Umweltprozesses werden die klimabedingten Veränderungen eines Gebirgsgletschers betrachtet, da kaum ein anderer natürlicher Prozess die Auswirkungen der globalen Klimaerwärmung deutlicher zeigt.

Zur effizienten Untersuchung komplexer Sachverhalte ist der Einsatz von Visualisierungstechniken sehr gut geeignet, da sich diese der enormen Fähigkeit des menschlichen Gehirns zur Verarbeitung visueller Information bedienen. Zudem hat sich aufgrund der rasanten technologischen Entwicklung im Bereich der Computergraphik die interaktive und dreidimensionale Visualisierung etabliert.

Das Ziel dieser Arbeit bildet die Entwicklung eines visuellen Systems, das Methoden zur interaktiven Analyse und experimentellen Simulation von klimabedingten Veränderungen eines Gebirgsgletschers bereitstellt. Als methodisches Konzept liegt dem System ein experimenteller Simulationskreislauf zu Grunde, der durch visuelle Analysemöglichkeiten ergänzt wird. Ein Experiment entspricht dabei dem einmaligen Durchlauf des Kreislaufs. Aufgrund einer Anfangshypothese wird mit der entwickelten Simulationsmethode ein neuer Zustand des Gletschers berechnet, der anschliessend in eine visuelle Repräsentation überführt wird. Aufgrund der visuellen Analyse der Simulationsergebnisse und einem Vergleich mit Referenzinformation kann die Hypothese verifiziert und allenfalls modifiziert werden. Nach mehreren Experimenten können die daraus gewonnen Erfahrungen zueinander in Beziehung gebracht werden, was zu neuen Rückschlüssen führen kann.

Die als Prototyp entwickelte Software basiert auf einem bereits bestehenden Geländevisualisierungsystem, das eine effiziente Methode zur Geländedarstellung bereitstellt. Zur Beschreibung des Szenengraphen und zur Implementierung anspruchsvoller computergraphischer Methoden wird die OpenSceneGraph-Bibliothek verwendet, welche auf OpenGL als Graphikbibliothek aufsetzt. Das System wurde zur Vereinfachung als geschlossenes System implementiert, weshalb keine konzeptionellen Überlegungen bezüglich eines generischen visuellen Simulationssystems angestellt wurden. Die Simulations- und Renderingroutinen konnten somit direkt im selben Kontext implementiert werden. Die Steuerung des Simulationsablaufs, die Veränderung von physikalischen und geometrischen Grössen sowie die Navigation in der Zeit wird über eine graphische Schnittstelle ermöglicht. Ausgewählte Simulationsresultate werden in einem Experimentarchiv abgelegt, um eine nachträgliche Analyse ausserhalb des Systems zu gewährleisten.

Die Analyse und Validierung eines Experiments erfordert entsprechende Möglichkeiten zur numerischen und visuellen Auswertung der Resultate. Deshalb wurden verschiedenste Ansätze zur Navigation im Raum und in der Zeit wie auch zur Steuerung des Simulationskreislaufs in das System integriert. Für die aufgabenorientierte Navigation im dreidimensionalen Raum wurden zwei Metaphern für die gezielte Positionierung, drei Metaphern für die Exploration und zwei Metaphern zur objektbezogenen Untersuchung der Gletscheroberfläche umgesetzt. Zur visuellen Analyse wurden Methoden entwickelt, die entweder die Informationskomplexität reduzieren oder zusätzliche Information für Validierungszwecke einblenden und moderne Techniken der Computergraphik einsetzen. Zusätzlich erlauben die implementierten Funktionen zur interaktiven Informationsextrahierung eine numerische Auswertung der Simulationsresultate im georeferenzierten dreidimensionalen Bezugsraum. Die modifizierte Darstellung eines Gletschers, zum Beispiel durch Veränderung der Beleuchtung oder des Abstraktionsgrades, stellt ein weiteres Hilfsmittel dar, um Gletscherveränderungen visuell zu untersuchen.

Es wurden fünf bestehende Gletschermodelle evaluiert, die hinsichtlich Komplexität und Einsatzbereich unterschiedlich sind. Nur eines der vorgestellten Gletschermodelle kann klimatisch bedingte Fluktuationen eines Gebirgsgletschers ohne Vorkenntnisse bezüglich der Form des Gletscherbetts beschreiben. Dieses eindimensionale Gletschermodell dient als Grundlage für eine neue Methode zur Simulation von dreidimensionalen Gletscherveränderungen.

Für die Simulation und geometrische Modellierung wird ein Gletscher als Summe mehrerer einzelner Gletscherarme betrachtet. Die Anzahl der Gletscherarme und der Verlauf in Fliessrichtung wird aufgrund eines digitalen Höhenmodells bestimmt. Sowohl für die Simulation als auch für die visuelle Darstellung wird jeder Gletscherarm als eigenständige Einheit betrachtet, da keine Interaktionen berücksichtigt werden.

Die implementierte Simulationsmethode ermöglicht einerseits die Berechnung des dreidimensionalen Gletscherbetts und andererseits die Bestimmung eines neuen Gleichgewichtszustandes des Gletschers aufgrund veränderter Klimabedingungen. Die Methode geht von einem Gleichgewichtszustand des Gletschers aus, berücksichtigt keine dynamischen Prozesse, setzt temperiertes Eis voraus und ist nur für grössere Zeiträume im Bereich von 20 bis 100 Jahre zulässig. Im Gegensatz dazu kann die Methode universell für Gletscher auf der ganzen Welt einfach eingesetzt werden, da nur wenige Inputparameter und kein Gletscherbett erforderlich sind. Die Eismächtigkeit wird mit Hilfe der aus einem

digitalen Höhenmodell extrahierten Neigung der Gletscheroberfläche in Fliessrichtung abgeschätzt. Zur Bestimmung der Gletscherbreite wird eine empirisch ermittelte Dicken-Breite-Beziehung eingesetzt. Die Kalkulation eines neuen Gleichgewichtzustandes beruht auf einer stufenförmigen Anpassung an eine veränderte Massenbilanz, welche nach einer gletscherspezifischen zeitlichen Verzögerung erreicht ist.

Für die visuelle Umsetzung wird ein Gletscher als Summe mehrerer Flächen modelliert, die jede für sich einen Gletscherarm repräsentiert. Der Verlauf eines Gletscherarmes in Fliessrichtung wird mit Hilfe kubischer *Non-Uniform Rational B-Spline* (NURBS) Kurven parameterisiert. Zur flächenhaften Beschreibung eines Gletscherarmes werden kubische NURBS-Flächen verwendet. Diese erlauben eine effiziente und kontinuierliche Beschreibung des typischen parabolischen Querschnitts und der gekrümmten sowie glatten Oberfläche. Ein geotypisches Aussehen eines Gletscherarmes wird durch eine computergenerierte zweidimensionale Textur erreicht, während durch die Gouraud-Schattierung die Form eines Gletscherarmes besser wahrgenommen werden kann.

Die visuelle Umsetzung von Gletscherfluktuationen basiert auf einer physikalischen Beschreibung der statischen steady-state Deformation und einer prozeduralen Animation, die für das dynamische *Morphing* eingesetzt wird. Die dynamische Entwicklung eines Gletschers von einem Gleichgewichtszustand zu einem anderen wird durch eine dreidimensionale und flächenbasierte Methode beschrieben. Die notwendige Korrespondenz zwischen zwei Gletscherarmflächen ist durch die Kontrollpunkte der parametrischen Beschreibung gewährleistet. Jeder einzelne zentrale Kontrollpunkt wird entlang einer parametrischen Geländekurve animiert. Für die Beschreibung der zeitlichen Verzögerung des Rückzugs gegenüber der Dickenveränderung im Zungenbereich wird ein exponentielles *Morphing* eingesetzt. Das Fliessen an der Gletscheroberfläche wird durch eine animierte Textur simuliert.

Die entwickelten Methoden wurden am Fallbeispiel Morteratschgletscher in der Schweiz überprüft. Die geometrische Modellierung eines Gletschers als Summe mehrerer NURBS-Flächen hat sich für die visuelle Beschreibung und das *Rendering* eines statischen Zustandes wie auch von animierten Veränderungen bewährt. Aufgrund fehlender flächenhafter Referenzinformation konnte das berechnete Gletscherbett nur an einem Punkt validiert werden, wobei eine Abweichung der Eismächtigkeit im Bereich von 10% festgestellt werden konnte. Die Simulationsresultate deuten zudem auf eine Bettübertiefung von 100 m hin. Die mit 11 Gletscherarmen simulierte Fläche für die initiale Ausdehnung 1985 ist um maximal 24% kleiner als in Wirklichkeit, vor allem aufgrund des zu kleinen Akkumulationsgebietes. Aufgrund einer angestellten Extrapolation kann davon ausgegangen werden, dass die reale Fläche mit ungefähr 19 Gletscherarmen erreicht wird.

Ein Vergleich zwischen simulierten Zuständen und historischen Referenzinformationen bezüglich der maximalen Ausdehnung 1850/60 und der Ausdehnung 1910 ergab eine gute Korrelation der Eismächtigkeit wie auch der Gletscherbreite.

Vier verschiedene Experimente wurden durchgeführt, um einerseits das Prinzip des experimentellen Simulationskreislaufs zu illustrieren und andererseits den beobachteten Rückzug des Morteratschgletschers seit 1850 zu rekonstruieren. In den ersten drei Experimenten konnte der Einfluss der drei Simulationsparameter Massenbilanzgradient, Höhe und lokale Sensitivität der Gleichgewichtslinie aufgezeigt werden. Im vierten Experiment wurde in einem stufenförmigen Prozess zuerst das zeitliche Rückzugsverhalten über die

Anpassung des Massenbilanzgradienten rekonstruiert. Danach konnte durch die Optimierung der lokalen Sensitivität der Gleichgewichtslinie und des Verlaufs der Lufttemperatur das beobachtete räumliche Rückzugsverhalten simuliert werden. Weitere Untersuchungen sind notwendig, um den vorgestellten Rekonstruktionsansatz zu verifizieren.

Mit den in Rückzugsexperimenten optimierten Simulationswerten (Massenbilanzgradient 0.68 mwe / 100 m, Gleichgewichtslinienanstieg 160 m / °C) wurden mögliche zukünftige Entwicklungen des Morteratschgletschers simuliert. Dabei wurden drei verschiedene Klimaszenarien des IPCC (*Intergovernmental Panel on Climate Change*) als Beschreibung der zukünftigen Klimaentwicklung angewandt. Die Simulationen ergeben einen Rückzug der Gletscherzunge um 2 - 4 km und eine Volumenverminderung von 31 - 77% im Zeitraum von 1985 bis 2100. Um die Qualität der Simulationen zu bewerten, sind erweiterte Messreihen sowie weiterführende Experimente in anderen Testgebieten notwendig. Zudem kann ein Vergleich mit anderen Simulationsmodellen helfen, systematische Fehler zu erkennen.

Es konnte gezeigt werden, dass die interaktive Untersuchung und das handlungsbasierte Begreifen von simulierten Veränderungen eines dreidimensionalen Gletschers mit Hilfe des implementierten visuellen Systems möglich ist. Dazu ist es erforderlich, dass die Benutzenden auf eine breite Palette verschiedenster Methoden zur Navigation in Raum und Zeit, zur Simulationssteuerung, zur Informationsextrahierung sowie zur visuellen Analyse zurückgreifen können.

Acknowledgments

This dissertation describes work done at the Remote Sensing Laboratories (RSL) in the years 2000 to 2004 under the supervision of Prof. Dr. Daniel Nüesch and Prof. Dr. Klaus I. Itten.

First of all, Prof. Dr. Daniel Nüesch deserves thanks not only for the confidence he put in me, but also for his encouragement and contagious enthusiasm. I thank him particularly for always having an open door and taking time for occasionally long discussions.

Prof. Dr. Jürgen Döllner from the University of Potsdam (Germany), Prof. Dr. Klaus I. Itten and Prof. Dr. Daniel Nüesch are all thanked for acting as thesis reviewers.

Without the everlasting effort of Dr. Urs Frei in providing me with free space, this work would never have achieved the quality it now has. I thank him for his confidence and interest in the work. His accurate proof-reading of the manuscript is gratefully acknowledged.

I am indebted to Dr. Martin Hoelzle for his glaciological support and for acting as my main contact to the world of glaciology during this work. His never-ending enthusiasm kept me working continuously and was especially important in times of reduced motivation.

Dr. Philipp Hirtz and Dr. Hilko Hoffmann deserve thanks for providing me with useful information about computer graphics in the initial phases of this work.

Prof. Dr. Wilfried Haeberli, Prof. Dr. Max Maisch and Dr. Frank Paul are all thanked for their contributions and interest in this work.

The meteorological data used was kindly processed by Ludwig Zgraggen and provided by MeteoSwiss, the Swiss Federal Office of Meteorology and Climatology. The Swiss Federal Office of Topography is thanked for allowing use of their *DHM25* terrain model.

Bronwyn Price and Dr. David Small deserve thanks for improving the linguistic and grammatical style of this work.

Georg Andersson, Sandra Eckert, Tanja Grap, Sarah Hangartner, Thomas Korner, Philippe Meuret, Peter Pirchl, Patrick Porchet, Michael Rossi, Roman Schurter and Yi Wu shared not only the room with me, but also many cheerful moments, computer-related and other, more important problems. I thank them all for the agreeable moments and discussions during and in between work.

RSL colleagues and the Department of Geography are all thanked for the pleasant working atmosphere.

Finally, I would like to express my deepest gratitude to my wife Corinna. She not only shared my happiness in good moments, but also particularly supported me during the most difficult times of this work.

Thalwil, Summer 2004

Stefan Biegger

Abbreviations

2D	Two-Dimensional
3D	Three-Dimensional
a	year
AAR	Accumulation Area Ratio
API	Application Program Interface
BRDF	Bi-directional Reflectance Distribution Function
BRep	Boundary Representation
BTF	Bi-directional Texture Function
CAD	Computer Aided Design
CLOD	Continuous Levels Of Detail
CPU	Central Processing Unit
CSG	Constructive Solid Geometry
DAG	Directed Acyclic Graph
DBMS	Database Management System
DTM	Digital Terrain Model
DTM-G	Digital Terrain Model without Glacier volume
E	Easting
ELA	Equilibrium Line Altitude
ESRI	ESRI Inc., Environmental Systems Research Institute
FFD	Free-Form Deformation

FOV	Field-Of-View
fps	frames per second
GDAL	Geospatial Data Abstraction Library
GIS	Geographic Information System
GLU	OpenGL Utility Library
GUI	Graphical User Interface
IPCC	Intergovernmental Panel on Climate Change
IVR	Immersive Virtual Reality
LOA	Level Of Abstraction
LOD	Level Of Detail
LSA	Least Square Approximation
m	metres
m a.s.l.	metres above sea level
MAAT	Mean Annual Air Temperature
MByte	Megabyte
mwe	metres water equivalent
N	Northing
NURBS	Non-Uniform Rational B-Spline
OSG	OpenSceneGraph
PC	Personal Computer
POI	Point-Of-Interest
RAM	Random Access Memory
RMS	Root-Mean-Square
ROI	Region-Of-Interest
RSL	Remote Sensing Laboratories
s	seconds
SDK	Software Development Kit
SGI	Silicon Graphics Inc.
SPOT	‘Système Pour l’Observation de la Terre’

SSI	Surface/Surface Intersection
STI	Surface/Terrain Intersection
TIN	Triangulated Irregular Network
UML	Unified Modeling Language
ViSC	SCientific Visualization
VRML	Virtual Reality Modeling Language
VRS	Virtual Reality System
VTP	the Virtual Terrain Project

List of Symbols

α	Slope of the glacier surface	deg
α_x	Slope of the glacier surface at x	deg
α_{limit}	Minimum size of α_x	deg
γ_κ	Length of $\mathbf{F}_\kappa$	m
$\dot{\varepsilon}$	Strain rate	a^{-1}
κ	Control variable ($0 \leq \kappa < s$)	-
λ	Distance-time function for morphing	-
Λ_{far}	Far viewing distance with minimum tessellation accuracy	m
Λ_{near}	Near viewing distance with maximum tessellation accuracy	m
Λ_{view}	Viewing distance	m
ρ	Ice density	$kg\ m^{-3}$
σ	Stress	Pa
τ	Basal shear stress	Pa
τ_0	Yield stress in case of perfect plasticity	Pa
T	Time of the old steady-state	a
T’	Time of the new steady-state	a
υ	Dot product of projection vector and xy-plane in y direction	-
ϕ	Error tolerance of υ and ϖ	-
Ψ	Accuracy of surface tessellation	-
ϖ	Dot product of projection vector and xy-plane in x direction	-

a_{const}	Constant length of the zone where shear stress is transmitted	m
a_{var}	Variable length of zone where shear stress is transmitted	m
aps	Animation rate	s^{-1}
A	Coefficient of ice-flow law	a^{-1} $Pa^{-n_{ice}}$
AAR	Size of the Accumulation Area Ratio (AAR)	-
A_{abl}	Ablation area	m^2
A_{acc}	Accumulation area	m^2
A_{ortho}	Area of the orthogonally projected glacier	m^2
b	Annual mass balance	mwe
$\overline{b}$	Mean specific mass balance	mwe
$\overline{b}_1$	Mean specific mass balance for one response time	mwe
bpf	Number of branches to animate per frame	-
b_t	Ablation at the terminus	mwe
$\delta b/\delta h$	Mass balance gradient in the ablation zone	mwe / 100 m
Δb	Change of mass balance	mwe
$\mathbf{B}_{x,\{0,1,2\}}$	Left, central and right point of the cross-section at position x	m
c_A	Transparency value	-
c	Mean velocity of kinematic waves	m a^{-1}
C^k	Parametric continuity of degree k	-
$\mathbf{C}(u)$	NURBS curve	-
$\mathbf{C_f}$	Parametric flow-line	-
$\mathbf{C_s}$	Parametric surface-line	-
$\mathbf{C_t}$	Parametric terrain-line	-
d	Branch thickness perpendicular to the glacier surface	m
$\widehat{d}$	Branch thickness in direction towards the centre of the Earth	m
d_f	Mean thickness of branch along $\mathbf{C_f}$	m

d_{max}	Maximum branch thickness	m
$\widehat{d}_{max}$	Perpendicular component of d_{max}	m
$\widehat{d}_{mean}$	Mean value of $\widehat{d}_x$	m
d_{mouth}	Branch thickness at the glacier mouth	m
d_x	d at position x on $\mathbf{C_f}$	m
$\widehat{d}_x$	$\widehat{d}$ at position x on $\mathbf{C_f}$	m
e	Factor used for computation of a_{var}	-
E	Easting	m
ELA	Equilibrium line altitude	m a.s.l.
$\delta ELA/\delta T$	Increase of ELA per 1 °C warming	m K^{-1}
f	Shape factor after Paterson (1994)	-
fov	Value of the Field-Of-View (FOV)	deg
fps	Number of frames per second	s^{-1}
f_t	Time stretch factor	-
$\mathbf{F}_\kappa$	Projection vector from branch surface to terrain used for morphing at κ	m
g	Acceleration due to gravity	m s^{-2}
$\mathbf{G}_\kappa$	Projection vector from terrain to branch surface used for morphing at κ	m
G^k	Geometric continuity of degree k	-
$\mathbf{G_b}$	Point grid consisting of the branch points $\mathbf{B}_{x,\{0,1,2\}}$	-
h_{max}	Maximum height of branch	m a.s.l.
h_{min}	Minimum height of branch	m a.s.l.
h_t	Height of the terminus	m a.s.l.
$h_{x,\{0,1,2\}}$	Altitude of $\mathbf{B}_{x,\{0,1,2\}}$ above sea level	m a.s.l.
$\mathbf{H}(E,N)$	DTM point at easting E and northing N	m
i	Index of control point in u direction	-

j	Index of control point in v direction	-
k_x	Ice cap factor at x	-
l_s	Distance between two cross-sections on the flow-line	m
δl_t	Selectable change of terminus during morphing	m
Δl_t	Change of L between T and T'	m
l_x	Length of flow-line from $\mathbf{P_s}$ to position x	m
L	Length of flow-line	m
L_{abl}	Length of ablation area	m
ΔL_m	Measured values of ΔL_t	m
ΔL_s	Simulated values of ΔL_t	m
ΔL_t	Cumulative change of L	m
$m + 1$	Number of control points of a NURBS surface in dimension v	-
m_d	Morphing factor for distance-time function	-
m_f	Morphing factor describing front behaviour during morphing	-
m_{near}	Ratio between viewing distance Λ_{near} and branch extension Λ_{branch}	-
m_{far}	Ratio between far viewing distance Λ_{far} and branch extension Λ_{branch}	-
M	Multiplicity of knot used for a parametric form	-
$n + 1$	Number of control points of a NURBS surface in dimension u	-
n_b	Number of branches	-
n_i	Number of control points of $\mathbf{S_b}$ in dimension u	-
n_j	Number of control points of $\mathbf{S_b}$ in dimension v	-
n_p	Number of flow-points	-
n_{ice}	Exponent of ice-flow law	-
n_t	Number of texture repetition in longitudinal direction of branch	-
nt_A	Number of response times	-
n_{tris}	Number of triangles used for terrain rendering	-

n_x	Number of cross-sections along flow-line	-
N	Northing	m
o	Index of a specific branch	-
p	Degree of NURBS form in dimension u	-
P	Annual precipitation	mm
$\mathbf{P}$	Control point of NURBS form	-
$\mathbf{P}_i$	ith control point of NURBS curve $\mathbf{C}(u)$	m
$\mathbf{P}_{i,j}$	ith control point in u dimension and jth in v dimension of $\mathbf{S}(u,v)$	m
$\mathbf{P_m}$	Flow-points of flow-line	m
$\mathbf{P_s}$	Source point of flow-line	m
$\mathbf{P_t}$	Terminus point of flow-line	m
q	Degree of NURBS surface in dimension v	-
Q	Discharge at a cross-section of the glacier	$m^2\ a^{-1}$
$\mathbf{Q}_i$	ith intermediate position of control point $\mathbf{P}$ during morphing	-
r_x	Relief factor at x	-
$R_{i,p}$	Rational basis function of pth-degree applied for ith control point	-
s	Number of morphing steps	-
$\mathbf{S}(u, v)$	NURBS surface	-
$\mathbf{S_b}$	Branch surface	-
$\mathbf{S_g}$	Glacier surface consisting of several branch surfaces $\mathbf{S_b}$	-
t	Current time in the time scale of the simulation system	a
t_{acc}	Acceleration time for morphing	-
t_A	Response time of a glacier branch	a
t_e	Exploration time-out	s

t_R	Reaction time of a glacier branch	a
T_{air}	Air temperature	° C
ΔT_{air}	Change of air temperature between T and T’	° C
T_r	Time scale of rendering system	-
T_s	Time scale of simulation system	-
u, v	Dimensions of the parametric form	-
u, v	Parametric values of dimension u and v	-
u_i	ith element of knot vector U	-
u_m	Mean flow velocity	m a^{-1}
u_s	Mean surface velocity	m a^{-1}
u_t	Texture coordinate in u direction	-
U	Non-uniform knot vector	-
v_t	Texture coordinate in v direction	-
vel_h	Horizontal component of ice velocity vector	m a^{-1}
vel_t	Retreat velocity at the terminus in the time scale T_r	m T_r^{-1}
vel_v	Vertical component of ice velocity vector	m a^{-1}
$\mathbf{V_b}$	Branch volume as a surface	-
$\mathbf{V_g}$	Glacier volume as a grid	-
V_{ortho}	Volume of the orthogonally projected glacier	m^3
$w_{i,j}$	Weight of control point $\mathbf{P}_{i,j}$	-
w_x	Width of branch at position x	m
x,y,z	Dimensions of a three-dimensional coordinate system	-
x	Index for position on flow-line (x = 0 at $\mathbf{P_s}$)	-

Contents

Chapter 1

Introduction

1.1 Motivation

This work is not the first motivated by a fascination for the ice streams flowing in high mountain areas from icy peaks down to deep valleys. There exist already papers and pictures of the early beginnings of systematic investigations of mountain regions documenting ever increasing interest. For example, a steel engraving is presented in Figure 1.1 showing the extension of a medium-sized glacier before 1876 with a huge glacier tongue

Figure 1.1: Steel engraving of the Morteratschgletscher (Switzerland) made by Disen in the year 1875.

split visually into two ice streams by dark rock material lying on the glacier surface and a remarkably large hole at its terminus. The background is dominated by snow and ice cov-

ered peaks illustrating the high and cold origin of this glacier. Together with this impressive image of nature, two persons are drawn in the foreground: a well dressed man with a hiking pole jammed under his arms and a distinguished lady sitting on a little rock with a parasol protecting her head. The combined view of this glacier and these two people points out the fashion then for perceiving the mighty glaciers reaching from the unspoiled mountain tops to the beginning of people's living space. People were astonished at the dimensions of these ice masses and showed respect and even fear of the unknown natural forces. Perhaps they were wondering if this glacier changes its shape over time. Probably they were not aware of glacier fluctuations since the shape of the glacier changes slowly compared to a human lifespan.

The regarded two persons did not yet imagine that this glacier would have lost about 30% of its volume and would have retreated by approximately 2 kilometres (km) or 25% by the turn of millennium (Maisch et al., 1999). Additionally, they were not aware of the fact that the dramatic loss of ice masses since the end of the 'Little Ice Age'[1] is observed not only on specific glaciers but in nearly all glaciated regions of the world. Moreover, they would not have expected that the primary reason for this ice loss was a global warming by more than 1 °C as stated in the latest report of the Intergovernmental Panel on Climate Change (IPCC, 2001[2]) and that there exists a human influence on this warming trend.

1.1.1 Appropriate strategies for climate change communication

Based on an increased theoretical and empirical knowledge of processes and feedback effects within the climate system and continuously growing computer power, it has been possible in the last two decades to predict possible climate changes using complex general circulation models (GCM's). Climate change has emerged as one of the most intensively researched and discussed environmental issues ever. Nevertheless, carbon emissions continue to increase both globally and regionally, and society continues to be vulnerabile to climate variability and change. This raises questions about the effectiveness of current communication efforts. To investigate these questions a look at how the output of climate models is communicated can be helpful. Typically, newspapers or magazines report on current climate issues in a manner such as: "The most probable global warming is expected to be in the order of 3 °C with a probable error of 1.5 °C". This type of language, while precise and meaningful to the research community, presents a major communication obstacle to other audiences. The information given is too abstract and valid only on a global and not on a regional scale. Normally, people are touched much more by regional than

1. According to climate historians, the beginning of the 'Little Ice Age' dates back to the Middle Ages. After a prolonged period of relatively warm temperatures, this colder phase set in towards the end of the fourteenth century and came to an end around the middle of the nineteenth century.

2. The Intergovernmental Panel on Climate Change (IPCC) was established in 1988 by the World Meteorological Organization (WMO) and the United Nations Environment Programme (UNEP). The role of the IPCC is to assess on a comprehensive, objective, open and transparent basis the scientific, technical and socio-economic information relevant to understanding the scientific basis of risk of human-induced climate change, its potential impacts and options for adaptation and mitigation.

global impacts. Even if a downscaling of atmospheric models from global to regional scale were feasible (still a challenge at the moment, e.g. Bray, 2000; Gyalistras, 2000), the output would still be invisible and almost inconceivable. Furthermore, probabilistic concepts can be difficult to comprehend. Thus, simply because information is 'going out', does not mean that it is 'going in'. Communication involves not just propagating information, it

"(...) involves imparting knowledge with the intent of raising awareness and promoting understanding. (...) communication is thought to be effective only when these changes in awareness and understanding result in attitudinal adjustments and/or improve the basis upon which decisions are made." (Andrey & Mortsch, 2000).

Among other possible reasons for ineffective climate change communication, the apathy fostered by the perception that climate change is a slow and gradual change is valid for many. The public is more interested and more prepared to act when confronted with facts and immediate threats rather than probabilities and future risks. The uncertainty or lack of scientific proof is also often a barrier to action (Andrey & Mortsch, 2000). In their generic communication guidelines for climate change communication, Andrey & Mortsch (2000) propose to learn from other scientific fields. Among other measures, they suggest incorporating principles and standard tools of education like the relative effectiveness of active versus passive learning, the need for repetition and reinforcement, and a variety of visual aids (Andrey & Mortsch, 2000).

In Storch et al. (2001) Nico Stehr considers linking nature and the social world and calls for incorporation of insights from the social sciences into the scientific and political debate on environmental change (pages 11-12):

"The successful reduction of the contingencies that derive from climate allow for an increase in the contingencies that come with the socio-cultural development of knowledge. (...) it is now no longer nature which poses threats, but it is uncertain knowledge which troubles us."

Oepen (2000) states that traditional thinking was based on the false assumption that increased knowledge about the environment and its problems leads directly to a greater awareness of the environment and to environment-friendly behaviour. But as studies have shown, greater knowledge about environmental problems can lead to a state of general anxiety or a complete refusal to think about environmental problems. In-depth knowledge about and personal feelings towards environmental issues must be accompanied by skills with respect to appropriate forms of action which promote self-confidence and participation in environmental issue-solving.

Concluding the observations, there is a clear need for appropriate communication and learning strategies. Oepen (2000) envisages future trends in environmental communication that would progress from cognitive to experiential, more active learning and from isolated, factual and compartmentalized knowledge to the ability to deal with complexity, uncertainty and risk. A more active role of each individual is preferred to establish personal responsibility and environmental competence.

1.1.2 Understanding geoprocesses in a virtual 3D space

When changing the point of view from non-experts to scientists, other not yet discussed problems arise. In order to describe the future glacier evolution as a physical process accuratly, climate scientists and glaciologists generate numerical models[1] describing the most relevant interactions between climate, glaciers and its underground. These models can be applied to simulate[2] the climate-induced glacier reaction. An increase in complexity of the simulation results leads to difficulties in interpretation. Furthermore, models in geoscience typically have three-dimensional (3D) spatial components that may change over time. They are difficult to analyse with two-dimensional (2D) tools since the traditionally used paper representation forces users to fuse multiple representations into one understanding of the geoprocess.

The use of geoscientific simulation models often leads to vast data sets, usually characterised by a high dimensionality. In order to gain better insights into high-dimensional processes and data and to quickly detect trends, correlations and anomalies, the human visual sense is highly suitable. The geoprocesses can be recognised more easily by using a graphical representation of numerical data than by analysing the raw data itself. Thus, the visualization as a graphical representation of numerical data is a key element for understanding geoprocesses.

"*To visualize*" means "*to form a mental vision, image, or picture of something not visible or present to sight, or of an abstraction; to make visible to the mind or imagination*" (Simpson and Weiner, 1989).

In many branches of science, and especially in the Earth sciences, it is not possible to deduce all outcomes from known laws. Thus, geoscientists need an environment that encourages the testing and evaluation of new hypotheses concerning the structure of complex geoscientific phenomena. Hay et al. (2000) report on a collaborative development project called 'The Virtual Exploratorium' that promotes simulation and high-end visualization to support education.

In the last two decades, new technologies and methods have been developed in the field of computer graphics to overcome many limitations of 2D graphical data analysis. A natural way of exploring high-dimensional data is given by so-called virtual reality systems. These systems provide an intuitive way of navigating and interacting with virtual objects (e.g. Van Dam et al., 2000). A virtual object is an object that can appear to be real but is in fact not real. Generally, navigation is performed by means of standard input devices such as a mouse, although others like 3D input devices increasingly gain in popularity (e.g. Van Dam et al., 2000). Interactivity is a main feature in current visualization systems, as it enables a conversation between the user and application. Input and output are interleaved, allowing user's input to depend on previous output.

1. The term 'model' is used as a synonym for a system of hypothetical principles that represents the characters of a phenomenon (WordNet, 2003).
2. The term 'simulation' stands for the attempt to predict aspects of the behaviour of some system by using an approximate (mathematical) model of it (WordNet, 2003).

1.2 Problem statement

The previous chapter highlighted several aspects and problems of climate change communication and the analysis of geoscientific models in a traditional analysis framework. The communication of climate change issues still lacks an appropriate way to promote understanding and raise awareness. Since climate warming has an undisputed effect on glacier dimensions, the visual representation of glacier fluctuations helps illustrate climate effects in an obvious and easy to understand manner. People are touched much more by this than by a global phenomenon because the retreat of a glacier is a regional process.

To simulate the future glacier evolvement, the relationship between glacier and climate has to be known and expressed as a physical model. Different approaches have been developed to compute glacier changes numerically but there is no current generally working method. There are still many unsolved aspects in order to calculate the evolution of the 3D glacier geometry (see Chapter 2 for further discussion).

As already mentioned in section 1.1.2, it is beneficial to simulate and study geoprocesses, e.g. glacier fluctuations, in an interactive system supported by visual means. Although modern computer graphic techniques enable the interactive study of geoprocesses in a virtual 3D world, some important deficits of current virtual reality systems for the exploration of geoprocesses exists as argued by Uhlenküken et al. (2000). They conclude that only a joint use of methods from diverse scientific branches can overcome all required items to provide an environment for visual process exploration. Häberling (1998) states that current glacier visualizations are limited to 2D maps without any possibilities for interaction. He formulates future requirements for the use of glacier visualizations and calls for an interactive system integrating multimedia techniques and providing perspective views and interactive analysis capabilities. But this glacier information system is restricted to the extraction of static information without any simulation functionality.

Taking up the point of active learning (see chapter 1.1.1) the system's capabilities can be extended to include the possibility of interactive simulations. Thereby the user would be free to restart interactively the simulation process and perform several experiments[1] using different parameter settings. Interactive experimentation leads to a more active role for the user and is an intuitive way to gain understanding of a previously unknown phenomenon. Scientists and non-experts alike can benefit from the extended possibilities for interaction.

1. Here, an experiment is understood as an operation carried out under controlled conditions in order to discover an unknown effect or law, or test or establish a hypothesis, or to illustrate a known law (WordNet, 2003).

1.3 Objectives

Based on the above-mentioned considerations, the goal of this study is formulated as follows:

Development of a visual system supporting the synthesis of new scientific knowledge about 3D mountain glacier fluctuations through interactive study and experimental simulation and, based on that, explanation of climate-induced glacier changes in the public domain.

1.3.1 Visually supported experimental simulation

The concept of experimental simulation in a visual environment is the methodological basis for this system. It is characterised by the *visually supported experimental simulation* which can be thought of as a logical extension of the well known method of formulating a hypothesis and performing an experiment to validate or reject it (see Figure 1.2). The

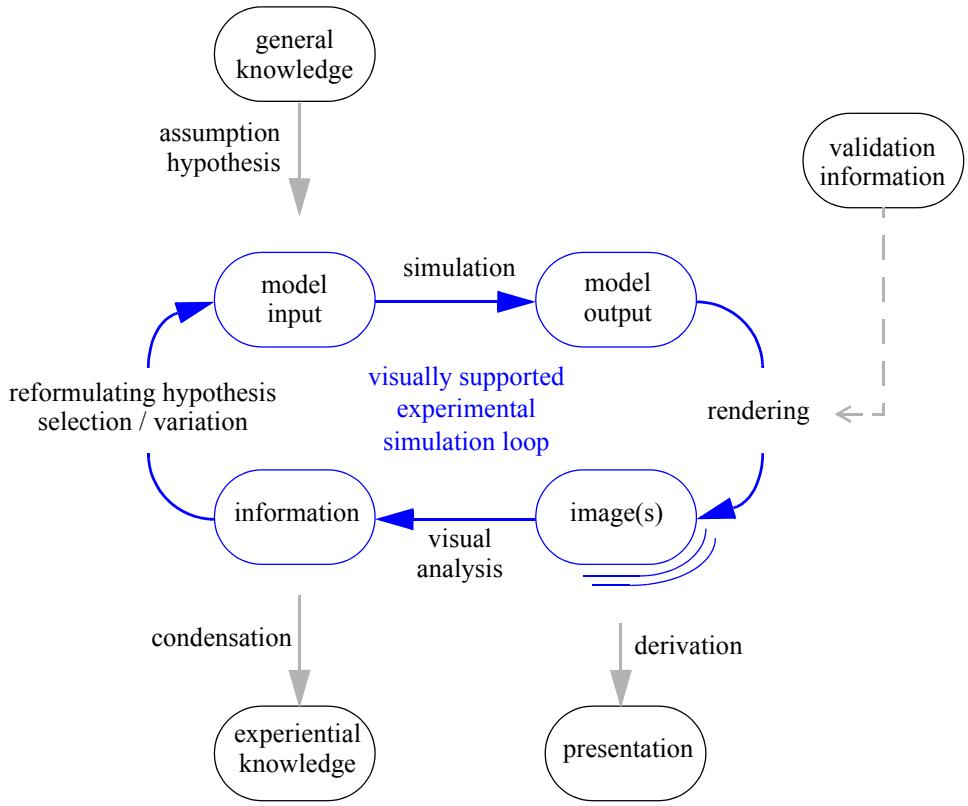


Figure 1.2: The visually supported process of experimental simulation.

main component of the process consists of an iterative and interactive loop that includes experimental simulation, visual analysis of the rendered simulation results, and formulation of new start conditions.

In a first step, general knowledge is used to assume an initial parameter set representing a distinct combination of parameter values for the model. With the aid of this set and the predefined model, the simulation produces a corresponding output. The output, or if additional transformation steps are required, its visual representation, is then converted to a 2D image for display. In the context of computer graphics, the process of converting a description of a 3D object to a 2D image is called rendering. Real-time rendering is concerned with the development of methods aiming for generating images rapidly on the computer.

The rendered image or image sequence can be used as a process derivative for presentation purposes. For example, these images are suitable to communicate scientific knowledge to non-experts and may serve as a basis for discussion. They also may act as input for further computer-aided analysis purposes.

By analysing interactively the rendered image(s) the user acquires information about the simulated phenomenon. Reflecting upon this information, either the hypothesis can be reformulated or a new simulation can be started, applying adjusted parameter settings. The experimental simulation loop has begun. After a certain number of loop passes the user is able to draw some conclusions from the visual observations and can transform the information to new experiential knowledge.

In case of optimization, the loop is interpreted slightly differently. Instead of asking what output is produced by the model, the question is what input is needed to achieve a good correlation between the output and some reference data. The reference data describes a verified state of the geoprocess and can be derived either from field measurements or other simulation models. The parameter set is then iteratively adapted until the rendered simulation output closely corresponds to the validation data.

1.3.2 Methodological and technical goals

In order to put the concept to practice, methodological and more technically oriented problems have to be solved. The methodological goal of this study consists of developing an approach for *simulating* (**1.a**) and *rendering* (**1.b**) climate-induced mountain glacier fluctuations whereas the *implementation* (**2.a**) of the developed approach into the prototype system represents the technical goal. In addition to this technical components approaches for navigation and analysis also have to be developed and integrated in the system (**2.b**). Finally, a *case study* (**3.**) shows capabilities and limitations of the developed methods and of the system.

1.a *Simulation of climate-induced mountain glacier fluctuations*: An approach has to be developed to simulate 3D glacier fluctuations. In order to retain applicability to glaciers all over the world, the model should depend on as few field parameters as possible. To simplify the complex interactions and processes, this study limits itself to mountain glaciers (see Chapter 2.1.1 and Chapter 2.1.2 for more details). If not specially declared the

term *glacier* is treated as a synonym for the term *mountain glacier* within this study. The main tasks to be performed for the simulation of glacier fluctuations are:

- development of a method for 3D geometric modelling of a glacier and its bed based on as little field data as possible,
- definition of an appropriate numerical model for simulating the 3D glacier reaction to climate change

1.b *Real-time rendering of mountain glacier fluctuations*: The generation of 2D images from 3D glacier changes includes the

- definition of a visual representation of the glacier,
- real-time animation of glacier fluctuations,
- rendering of additional glacier characteristics,
- and appropriate visualization of the glacier environment, i.e. terrain.

2.a *Implementation of the visual system*: The technical part comprises the development of a visual prototype system, in the following called VadreX[1], for the interactive study and experimental simulation of mountain glacier fluctuations. Interactive experimenting means that the user is free to test and analyse the model in an iterative way by applying different simulation parameter settings while the software is running. VadreX should

- provide functionality for interactive simulation and rendering of climate-induced glacier fluctuations,
- provide a graphical user interface (GUI) to interactively modify the scenario, simulation properties and glacier rendering properties for experimental simulation, and
- be built on top of low-end hardware to minimize costs.

2.b *Navigation and analysis*: Appropriate methods for navigating in a virtual landscape and methods for the visual analysis are crucial for a geovisualization system and thus have to be provided by the system.

The system is intended for three user-groups: scientists, students and non-experts. Scientists gain more insights into the complex glacier-climate system and use the system to communicate major aspects to the public. As an environment for educational purposes the system supports the interactive and experimental learning of glacier behaviour in 3D. Since the visualization should provide an appropriate level of realism, it is possible for non-experts to perceive future trends in a more intuitive way and to get an impression of the range of the simulation uncertainties.

3. *Case study*: The developed methods and prototype system should be tested with a well-suited glacier by performing the following tasks:

- verification of the simulation output (e.g. glacier bed and glacier fluctuations) with available reference information,

1. The name VadreX is based mainly on the Romansh language expression *Vadret* being the synonym for glacier. The letter X is derived from the two main actions eXperimenting and eXploring offered by the software system.

- application of different climate change scenarios to draw conclusions about the range of expected glacier changes,
- assessment of visual quality and performance of the methods chosen for geometric modelling and rendering, and
- testing of system functionality (e.g. visual analysis, navigation).

1.4 Thesis outline

Chapter 2 addresses the most relevant aspects of glacier research and environmental simulation. As an introduction the basic physical and visual properties of mountain glaciers are addressed in 2.1. General considerations about environmental simulation (2.2) are followed by discussion of the climate-glacier relation (2.3). Numerical approaches to compute the flow of glaciers and climate-induced glacier fluctuations are treated in 2.4.

The first section of Chapter 3 presents the main approaches in the field of computer graphics. In 3.2 an overview of animation concepts and techniques with special emphasis on fluid-like objects follows. After presenting specific approaches in geovisualization (3.4), rendering techniques for the visualization of terrain and dynamic geoprocesses are discussed in 3.5. Chapter 3.6 details several interaction possibilities in the virtual geospace. A brief overview of computer graphic systems for real-time rendering follows in 3.7.

Chapter 4 deals with the development of a method to simulate and render glacier fluctuations in real-time. Existing simulation and rendering methods are evaluated in Chapter 4.1. A new approach for generating the 3D shape of a glacier bed is presented in 4.2. In Chapter 4.3, a method for the deformation and morphing of a glacier surface follows. After pointing out possibilities to describe the visual appearance of a glacier surface, some considerations are made to increase the efficiency of the glacier rendering.

In Chapter 5 a visual system is presented that provides functionality to conduct experiments for glacier fluctuations and to study them in an interactive environment. In the first section, a suitable visualization system for further implementations is evaluated. Preliminary design aspects are then considered in 5.2. The architecture of the system is described in Chapter 5.3. Concepts for the navigation and conduction of an experiment are presented in 5.4. Strategies for the analysis of glacier fluctuations are discussed in Chapter 5.5.

A case study is carried out in Chapter 6 to evaluate the developed methods by applying them to a specific glacier. The quality of the glacier bed simulation is assessed by comparing the bed with observations (6.2). Reconstructions of historic glacier fluctuations (Chapter 6.3) are used for the calibration of the model whereas a possible range of expected changes is simulated and visualized in 6.4. Aspects of the glacier rendering and analysis are pointed out in Chapter 6.5.

The experiences made in this study are summarized in Chapter 7. They are used to formulate future requirements for interactive glacier rendering and develop perspectives. The data used for the case study is described in the Appendix.

Chapter 2

Simulating glacier fluctuations

Glacier fluctuations are the result of complex interactions between the atmosphere, the glacier surface, and the dynamic flow behaviour of the glacier. In this chapter, a brief overview of concepts developed to simulate these fluctuations is provided.

2.1 Basic concepts

In this introductory chapter, the most relevant basic aspects of glacier research are presented. Fundamental properties of mountain glaciers are followed by their major visual characteristics. For a detailed discussion, the reader is referred to the standard work Paterson (1994).

2.1.1 Fundamental properties

A glacier consists of snow and ice masses that flow from an accumulation area A_{acc} , where more mass is gained than lost, to an ablation area A_{abl} , where more mass is lost than gained (see Figure 2.1). The most important process adding mass to a glacier is snowfall. Melt and evaporation in summer are processes leading to mass loss on temperate glaciers. Most glaciers outside the polar regions or at least a significant fraction are temperate, i.e. their ice temperature is near the freezing level.

The mass balance b is the algebraic sum of all accumulation and ablation processes at the end of one balance year (1st October - 30th September). The annual mass balance b is measured in metres water equivalent (mwe) and varies at the terminus between a few meters water equivalent (mwe) for small mountain glaciers and temperate large ice fields, respectively. The mean specific balance $\overline{b}$ represents the mass balance averaged over the glacier surface and its absolute value lies between 0 and 4 mwe a^{-1} (IAHS (ICSI)/UNEP/ UNESCO/WMO, 2001). The boundary between the accumulation and the ablation area, where the mass balance b is zero is called equilibrium line, its altitude is the equilibrium

line altitude (*ELA*). The accumulation area ratio (AAR) describes the ratio between A_{acc} and A_{abl}. In nature, the zone where the mass balance is zero does not follow a simple pat-

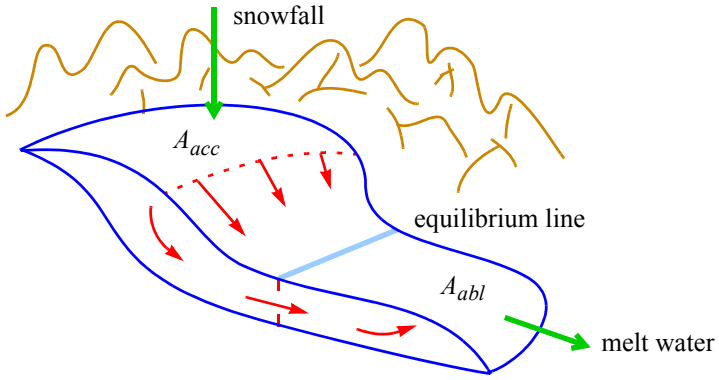


Figure 2.1: A schematic view of a mountain glacier. The equilibrium line (light blue) separates the glacier into an accumulation area A_{acc} and an ablation area A_{abl}. The red arrows indicate the flow direction and velocity in the horizontal and vertical dimensions.

tern like a line running smoothly from one side of the glacier to the other side along a constant altitude. If the accumulated gain on the glacier surface is equal the mass loss, the glacier is in steady-state (Paterson, 1994).

Glacier ice flows from the accumulation area to the ablation area because of the acceleration due to gravity. The ice is able to deform plastically (Paterson, 1994). Friction at the base and at the sides works to keep the ice in place.

Glaciers tend to exist in cold and wet places. Cold regions are encountered in high altitudes and latitudes whereas wet conditions are primarily found close to oceans. The climate requirements are reflected in the worldwide occurrence of glaciers with many active glaciers in Norway, Alaska, Patagonia, and New Zealand. These maritime glaciers close to a humidity source show a high mass turnover with high accumulation and high ablation values. Thus, the mass balance gradient $\delta b/\delta h$ is high for maritime glaciers and low for continental glaciers with a low mass turnover due to the limited supply of humidity. The mass balance gradient is a very significant value describing the glacier sensitivity and lies in the range of 0.3 - 1.3 metres per 100 metres change in altitude.

2.1.2 Visual characteristics of a mountain glacier

Glaciers can be categorized according to their size into ice caps, ice fields, mountain glaciers and glacierets. Ice caps cover a plateau-like terrain with smaller or larger outlet glaciers flowing to lower elevations. A mountain glacier is characterised by confined flow in the ablation zone and a characteristic tongue-like bottom section. An ice field is composed of several larger and connected mountain glaciers. Glacierets are very small gla-

ciers with small horizontal and vertical dimensions and without a tongue-like bottom section.

Moraines are deposits of rock material resulting from erosion of the bedrock or fallen rocks from side-walls. In Figure 2.2 a) moraines on and beside the glacier are clearly visible due to the different scattering properties or characteristic 3D shape, respectively. Distinctive moraines in the glacier forefield of many glaciers created at the end of the Little Ice Age document that these glaciers must have been close to steady-state. Over large areas, the glacier surface is a smooth image of the underlying terrain because of the material

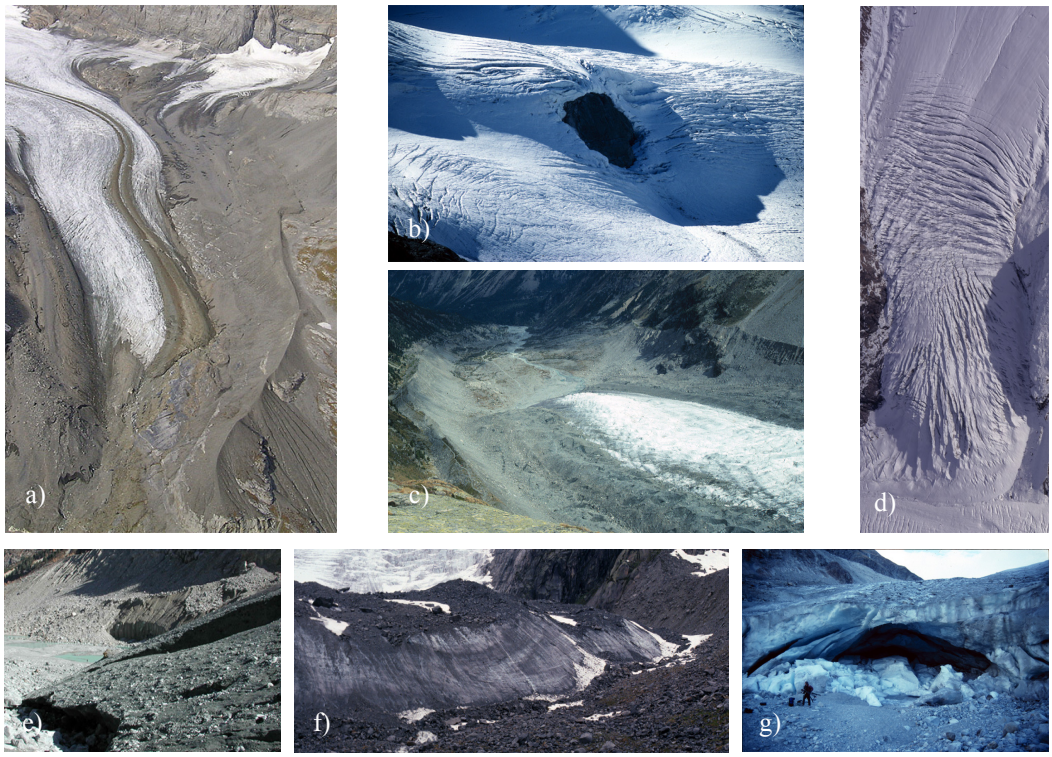


Figure 2.2: Visual characteristics of a mountain glacier: a) moraines on and beside of a glacier; b) glacier as a smooth image of the terrain; c) parabolic cross-section of the glacier bed; (d) Longitudinal and transversal crevasses; e) flat front indicating glacier retreat; f) an advancing glacier with a steep front; g) parabolic terminus. Copyrights: Max Maisch (b,c,e,f), Christine Rothenbühler (a) and Swiss Air Force (d).

and flow properties of ice (see image b). Figure 2.2 c) depicts the typical parabolic cross-section of the glacier bed that results from the erosion of the glacier at its bed. In places with extensive or compressive flow, transversal and longitudinal crevasses are characteristic small scale features caused by changes in shear stress. The example in image d) shows a glacier flowing in a steep channel compressed at the bottom by another glacier. A feature caused by the dynamic glacier response is the different shape of the tongue. When

the long-term averaged mass balance is positive, the tongue is characterised by a steep front as illustrated in image f) whereas a flat front in image e) indicates a retreating glacier.

The scattering effects at the glacier surface are dominated by the diffuse component. There is hardly any specular component aside from small melt water areas. Bare ice can appear slightly blue due to Rayleigh scattering as recognisable in image g). Additionally, in this image a distinctive arched terminus indicates a negative mass balance over a long period with a remarkable retreat.

2.2 Simulation in environmental research

In environmental sciences, models are an essential tool to simulate dynamic processes. But to investigate environmental phenomena, an approach different than that of pure sciences such as physics and chemistry is required due to the open character of environmental systems (Storch et al., 2001). An open system can be regarded as a system with many processes interacting in different ways and with an infinite number of external influences. Dynamics vary in character at different scales and can not be described with models based on similar laws. External forces are too diverse for a complete specification. The temporal and spatial scale of processes and external influences varies strongly. This implies that it is impossible to conduct laboratory experiments on the functioning of the systems as a whole (Storch et al., 2001). Moreover, an internal noise is present in the environmental system superimposing the deterministic behaviour. The presence of many chaotic subsystems leads to a pattern of variability that cannot be distinguished from random variations (Storch et al., 2001). Because of these specific features two different approaches are used in environmental modelling:

- A cognitive model reduces the process of interest to a minimum complexity considering just features of first order, i.e. the bare essentials. As such, cognitive models constitute knowledge. The formulation of cognitive models is a key method in fundamental science.
- A quasi-realistic model describes the reality in as much detail as possible. Often, a quasi-realistic model is composed of several smaller sub-models describing processes that together form the considered system. But even when complex, such a model is limited to its specific admissible domain (see Figure 2.3 right). This kind of model is used for purposes such as testing hypotheses or simulating possible future developments given certain changes in the system's ambient conditions.

Models should reflect reality, but they are smaller, simpler and closed in contrast to the real world. In Figure 2.3 left a real system is modelled as a system with fewer and simplified processes and a limited number of external forces. Some interactions between processes may be parameterised and not defined explicitly by modelling each dynamic subprocess.

A model is focused on a specific purpose and can generally not be used for several different tasks. Each model has its own admissible domain of applicability in terms of the spatial scale, the time scale and the parameter space - as shown in Figure 2.3 right (Storch

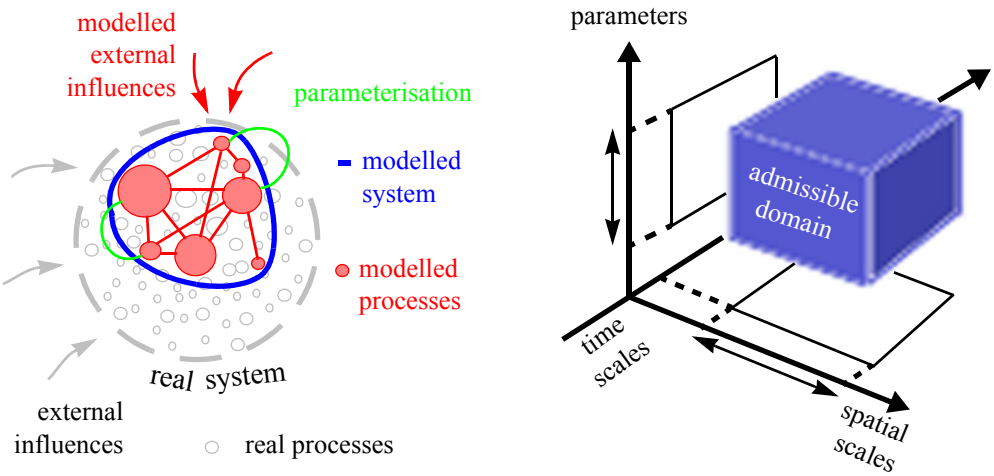


Figure 2.3: Left: Describing a real system by a simplified, smaller and closed system. Right: Admissible domain of applicability of a model (after Storch et al., 2001).

et al., 2001). In most cases, the admissible domain is set up implicitly by the scientist. Thus, the user may be unaware of this domain. The model may give incorrect descriptions and predictions for situations outside the realm of their intended use.

2.3 Climate-induced glacier fluctuations

In the first part of this chapter, possibilities and limitations of current environmental models for simulating future climate developments on global and regional scales are presented. The relation between climate and glacier and the involved processes are pointed out more schematically later in the chapter.

2.3.1 The projections of the future climate

The climate system consists of several subsystems, i.e. atmosphere, cryosphere, ocean, land surface and biosphere. These subsystems are linked by many interactions, with positive and negative feedback effects. Quantitative statements about future climate trends are based on future scenarios of forcing agents, i.e. emission scenarios, and numerical climate models.

Comprehensive climate models are based on physical laws represented by mathematical equations that are solved using a 3D grid over the globe. For this, several submodels

describing the different components of the climate system, e.g. atmosphere, ocean, cryosphere, land surface and biosphere, and interactions between them have to be applied. Using a different global climate model forced by the same emission scenario would result in a different predicted climate trend. In this study, results generated with the climate models mentioned in IPCC (2001) were used due to their wide acceptance.

IPCC (2001) established 44 emission scenarios each quantifying one of the four storylines defined by the IPCC. Scenarios based on the same storyline constitute a scenario family. For a detailed description of the emission scenarios and the four storylines A1, A2, B1 and B2 see IPCC (2001).

Local climate change is influenced greatly by local features such as mountains, which are not well represented in global models because of their coarse resolution (200-300 km). Models of higher resolution cannot practically be used for global simulation over long time periods. To overcome this limitation, regional climate models, with a higher resolution (typically 50 km) are constructed for limited areas and run for shorter periods (e.g. 20 years). These regional climate models take their input at their boundaries from global models. But currently, the downscaling of global trends to regional and local scales for mountainous regions leads to controversial results more uncertain than on global scale (e.g. Gyalistras, 2000). The selection of regionalization methods often has the same influence on the resulting scenarios as the choice of emission scenarios or the global model (Gyalistras, 2000).

To nonetheless estimate the regional climate trend a suite of projected future climate changes can be made using global models. This set of possible trends can act as a lower and upper boundary for estimating a possible range of regional climate trends. Gyalistras (2000) states that the combined view of all scenarios reflects best the current knowledge about future climate trends in the Alps. But he concludes also that it is nearly impossible to give a probability and time of the occurrence of a specific scenario that is objectively verifiable. As a rule of thumb, Oerlemans et al. (1998) assumes a uniform increase in precipitation of 10% per degree warming. Although regional changes could be larger, the number can be considered as an upper limit for the global scale (Oerlemans et al., 1998). Another approach for specifying future local climate conditions is used by Schneeberger et al. (2001): they add the simulated local climate changes to the observed present-day local climate. This ‘down-scaling’ procedure is based on the assumption that local climate changes closely follow the mean climate changes of the larger surrounding region.

2.3.2 Glacier response as a filtered climate signal

Radiation balance, air mass exchange and ocean currents are parts of a complex interplay between continental ice sheets, such as the Antarctic and Greenland, and global climate. In contrast, smaller mountain glaciers generally react passively to changes in atmospheric conditions.

Glaciers of differing geometries, located in different climatic regimes, react in different ways to a climate signal. There is no doubt that glaciers are very sensitive to climate change. The main reason for the large sensitivity lies in the nature of the melting process. Both the downward sensible heat flux and the longwave radiation balance increase as air

temperature rises while the melting point remains fixed (Oerlemans et al., 1998). As depicted in the flow scheme in Figure 2.4, mountain glaciers are related to climate change[1] through a complex process chain (Meier, 1984). Climate change expressed as a change in one or more components of the energy balance, e.g. increased air temperature T_{air} or precipitation, immediately causes a change to the mass balance due to increased or decreased mass flow from A_{acc} to A_{abl}. But it is important to note that the climate signal is filtered

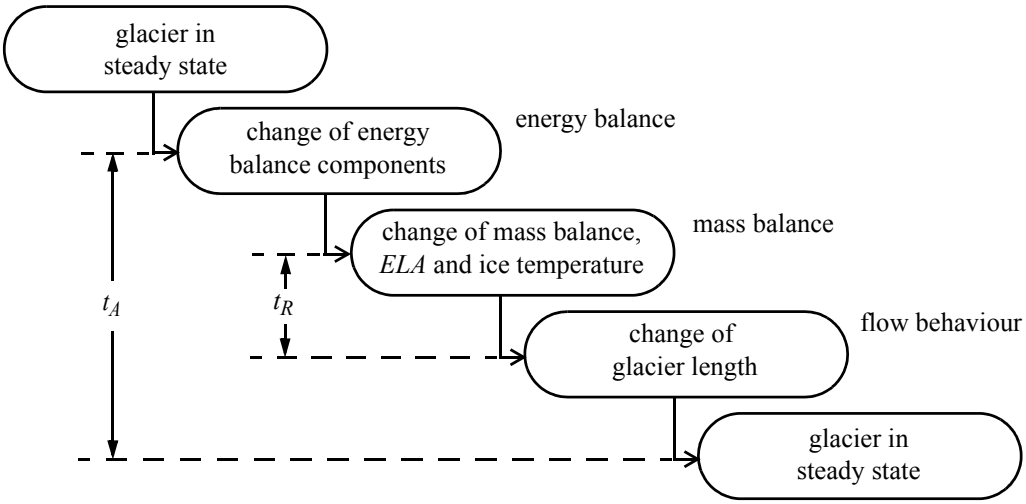


Figure 2.4: A schematic view of the glacier-climate process chain.

by the atmospheric boundary layer before it reaches the glacier surface (Oerlemans, 2001). Moreover, the *ELA* and the ice temperature T_{ice} are adjusted directly according to the changed mass balance. A quantitative description of the influence of air temperature changes on the mass balance can be found in Oerlemans et al. (1998) and Kuhn (1989). Jóhannesson et al. (1989) analyses the glacier response to a given mass balance change theoretically. Kuhn (1990) considers different feedback effects between the most important factors, i.e. radiation, air temperature, humidity and accumulation, and calculates for a warming of 1 °C an increase of *ELA* by 170 metres. Thus, if the mass balance gradient is known, the mass balance change Δb can be estimated.

Due to the mass balance change Δb, the glacier is forced to achieve a new state of equilibrium by changing its flow behaviour and geometry, i.e. responding with an advance or a retreat. The dynamic reaction of the glacier terminus does not establish itself immediately but with a delay called reaction time t_R. The time the glacier takes to adjust to Δb is called the response time t_A = T ' - T where T and T ' is the year of the old and new steady-state, respectively. Generally, large glaciers respond more slowly than small glaciers. Realistic values of t_A lie between 10 and 100 years for mountain glaciers in the Alps (Haeberli & Hoelzle, 1995). Small glaciers, i.e. glacierets, do not respond by retreat or ad-

1. According to the IPCC conventions the term climate change is used as referring to any change in climate over time, whether due to natural variability or as a result of human activity.

vance, but only in change of thickness. Their changes could be described by an ice-melting or ice-thickening simulation. If the warming ΔT_{air} is so high that $A_{acc} = 0$, the glacier reacts by down-wasting rather than changing length.

When the mass balance changes, the upper part of a glacier does not change significantly whereas the lower part does. This is because the change in ice flux, resulting from the change in mass balance, accumulates down-glacier.

The sensitivity of a glacier to climate warming is not only dependent on the mass balance gradient, but also on the distribution of glacier area according to the elevation, i.e. hypsography (Kuhn, 1989). The mass balance gradient represents the regional climatic sensitivity, whereas the hypsography reflects the local situation determining the individual behaviour of a glacier. Worldwide measurements of glacier mass balance document that the hypsography is of less importance when regarding long-term fluctuations (IAHS (ICSI)/UNEP/UNESCO/WMO, 2001).

The retreat of glaciers is a delayed and filtered signal of changes in the atmosphere. Nevertheless, the long-term signal of glacier changes since the middle of the 19th century - the end of the Little Ice Age - is among the clearest signals of ongoing warming trends existing in nature. In the latest report of the Intergovernmental Panel on Climate Change (IPCC, 2001), the observed glacier retreat is assumed to be the temperature indicator with the highest reliability, together with the directly measured temperature of the air and of the sea surface. Today, glaciers are recognised as high-confidence indicators of air-temperature trends and constitute a valuable element of a strategy for early detection of possible climate changes (e.g. Beniston et al., 1997). The recent shrinking of glaciers temporally coincides with human-induced radiative forcing which could be responsible for a major part of the additional energy flux causing the observed melt rate (e.g. Haeberli et al., 1999b).

2.4 Glacier modelling

The numerical description of changes to the 3D glacier geometry is an active research field in glaciology. As shown in Figure 2.5, two main processes can be distinguished in order to model the dynamic glacier response to climate change (Oerlemans, 2001). The mass balance model is intended to simulate all the physical processes that determine the

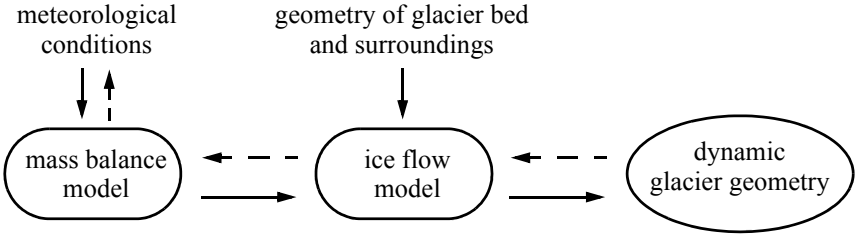


Figure 2.5: Relation between meteorological conditions and glacier geometry expressed as a combination of a mass-balance and ice-flow model (after Oerlemans, 2001).

exchange of energy between a glacier surface and the atmosphere. With regard to the variability of glacier mass balance, changes to air temperature and precipitation are the most important factors. The simulated mass gain or loss can then be used as a driving force for an ice flow model to compute the dynamic flow behaviour and geometry. The level of complexity of the mass balance and ice flow model should be similar (Oerlemans, 2001). The solid lines indicate the forward computation estimating the glacier response to a climate change, whereas the dashed lines describe the backward or inverse case of estimating past atmospheric conditions based on a known change of glacier geometry.

2.4.1 Dynamic change of the glacier surface

The change of the geometry of the glacier surface $\delta d/\delta t$ depends on the local mass balance b at the surface and the local ice flow as indicated by the kinematic boundary condition (e.g. Paterson, 1994)

$$\frac{\delta d}{\delta t} = b - vel_v - (vel_h \cdot \alpha) \qquad \textbf{(2-1)}$$

with the thickness d of the glacier, the vertical ice velocity vel_v at the surface, the horizontal component vel_h of the velocity vector and the slope α of the surface in flow direction.

The flow of glacier ice can be described mathematically as an isotropic, incompressible and viscous fluid. Ice flow is composed of basal sliding at the bed and internal ice deformation (Paterson, 1994). The basal sliding consists of direct sliding between glacier ice and bedrock, and deformation of sediments.

Many laboratory experiments have been conducted to study the deformation of ice (polycrystaline ice in the case of glaciers). The experiments can be performed either driven by a constant stress σ or a constant strain rate $\dot{\varepsilon}$. If a constant stress is applied, ice starts to deform after an initial elastic response, with a decreasing (primary creep), then with a constant (secondary creep or steady-state creep), and finally with an increasing strain rate. In nature, glacier ice is assumed to deform with a constant strain rate given constant stress (steady state creep). Glen's flow law (Paterson, 1994) describes the steady state creep by defining $\dot{\varepsilon}$ as a function of σ

$$\dot{\varepsilon} = A\sigma^{n_{ice}} \qquad \textbf{(2-2)}$$

in combination with the flow law parameters A and n_{ice}. The size of the coefficient A depends on ice temperature, moisture and debris content, ice fabric and composition and is estimated by field measurements. The exponent n_{ice} describes the non-linearity of the relation between stress and strain rate and is typically set to 3 (e.g. Paterson, 1994). Glen's flow law has traditionally been used for modelling purposes in glaciology, although there are reasons to believe that it may be based on an oversimplification of the actual rheological behaviour of glacier ice (Gudmundsson, 1999). The concept of perfect plasticity is widely used for glaciological studies. It can be seen as an asymptotic case of Glen's law where n_{ice} is infinite. Perfectly plastic material does not start to deform until the stress reaches a critical value called yield stress τ_0 (Paterson, 1994).

For ice deformation, the most important factor is the basal shear stress τ (Paterson, 1994) at the bedrock

$$\tau = f \cdot d \cdot \rho \cdot g \cdot \sin(\alpha) \quad \textbf{(2-3)}$$

with the shape factor *f*, the glacier thickness *d*, the ice density $\rho = 917$ kg m^{-3}, the acceleration due to gravity $g = 9.81$ m s^{-2} and the slope α of the glacier surface. The shape factor *f* (Paterson, 1994) considers friction at the glacier sides which depends on the shape of the cross-section. Nye (1965) carried out some experiments describing the flow of glacier ice in a channel of parabolic, rectangular or elliptical cross-section.

A simple method to estimate the mean thickness of a mountain glacier was formulated by Paterson (1994). Regarding Equation 2-3, the product of thickness *d* and slope α of the glacier surface is constant when perfect plasticity with the yield stress τ_0 is assumed. This explains the often observed fact that the thickness is smaller where the slope is steep and vice versa.

So far, the strain rate was considered to be the dependent factor. Departing from that assumption, Haeberli & Schweizer (1988) regarded the shear stress τ as variable and the strain rate as defined by the mass turnover, i.e. the climate and topography. This explains why maritime glaciers show an increased basal shear stress. Similarly, within one individual glacier, shear stresses are higher in steep slopes.

Besides deformation, basal sliding is the second component contributing to ice flow. It is extensively but also controversially discussed due to the complexity of the involved processes (e.g. regelation, cavitation and bed deformation) and the limited accessibility. There is an obvious relation between sliding speed and basal water pressure, often leading to the application of a sliding law combining regelation and water pressure (Oerlemans, 2001).

The resulting pattern of ice deformation may be high dimensional. Due to the incompressibility of ice, an increased deformation in the flow-direction (extending flow) leads to an ice compression in the transverse and vertical directions with different strain rates. This three-dimensionality of the flow field makes the use of analytical methods difficult unless some strongly simplifying assumptions about the flow field can be made. An often applied approach is the use of a flow-line to reduce the dimensionality of the problem. For most mountain glaciers it is not difficult to identify a major flow-line located in the middle of the glacier following the valley in which the glacier is flowing. Possible tributaries can be considered by adding additional flow-lines that merge with the major flow-line.

2.4.2 Numerical modelling of glacier flow and fluctuations

Glacier flow is a complex process commonly described by continuum mechanics. Detailed knowledge of the 3D pattern of stress and strain rates is essential to calculate the dynamic and small-scale changes of mountain glaciers. Physically-based 3D flow modelling using numerical techniques such as finite element methods is a current research area in glaciology and leads to complex flow systems.

As an example, a dynamic 3D flow model of the Unteraargletscher developed by Gudmundsson (1999) is presented. Numerical flow models generally require detailed

knowledge about the glacier and its underground, presenting a severe constraint on their usability. The bedrock geometries of the Unteraargletscher and most parts of the Finsteraar- and Lauteraargletscher were derived from radio echo soundings (Funk et al., 1994). In Figure 2.6 left the bedrock and the surface generated from topographic maps are shown

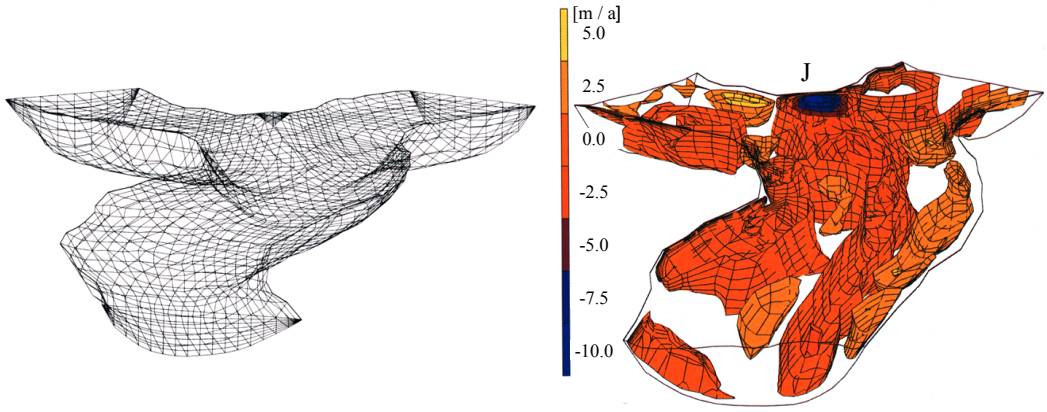


Figure 2.6: The Finite-Element mesh (left) used to simulate isosurfaces of vertical ice velocities (right). Both perspective plots are taken from Gudmundsson (1999).

as a Finite-Element mesh. This mesh is used by Gudmundsson (1999) to study the flow dynamics of the confluence area of the Unteraargletscher. The developed numerical model calculates the internal ice deformation respecting conservation of mass and angular and linear momentum. Glen's flow law is used as a constitutive relation. The upper boundary is defined as a free surface with no forces exerted up on it. Since the focus is put on the winter flow pattern, and basal sliding is assumed to be insignificant during the winter velocity regime of this specific glacier, a no-slip condition is imposed along the lower boundary. In this way, the model is kept as simple as possible, having only two adjustable parameters A and n_{ice} of Glen's flow law. Across all three cross-sections of the model (see Figure 2.6 left), either stress or velocity distributions were defined as boundary. Measured winter velocities were used for the model verification. The optimal size of the parameter A was found by minimizing the root-mean-square (RMS) error between the measured marker velocities and calculated velocities. This RMS error was calculated for different values of n_{ice} leading to an optimal size of $n_{ice} = 3$ where the best overall fit was achieved. For $n_{ice} = 3$, the optimal value for A is found to be much smaller than often recommended in literature, e.g. Paterson (1994). But despite some systematic errors (possibly due to neglected basal sliding or inadequate flow law) the overall qualitative agreement of the numerical calculations with field observations was good. As an example, the calculated spatial distribution of the annual vertical ice velocities is shown in Figure 2.6 right with increased subsidence at the junction point J.

For glacier-climate studies, describing the response of a glacier to changes in meteorological quantities the physics can be reduced to a problem of mass continuity (Oerlemans, 2001). Three examples of different complexities are presented in the following for modelling the climate-induced glacier response: two flow-line based 1D models (Jóhan-

nesson, 1997 and Haeberli et al., 1999a) and a comparative study incorporating 12 different models (Oerlemans et al., 1998).

Flow-line based and dynamic 1D model: Jóhannesson (1997) investigated the response of two temperate Icelandic glaciers to climate warming with a degree-day glacier mass balance model coupled to a dynamic glacier model. The used mass balance model computes the glacier mass balance as a function of altitude based on observed temperature and precipitation data measured at a meteorological station. Temperature and precipitation on the glacier surface are linearly extrapolated with two different precipitation gradients for ice-free and glaciated areas. Variation of temperature within the year was modelled as a sinusoidal function with superimposed statistical fluctuations and the measured MAAT as amplitude. Melting of snow and ice is computed from the number of positive degree-days. The mass balance model was calibrated by alternating runs of a runoff model and the mass balance model. Parameter adjustments were required to ensure that simulated mass balance components agreed with mass balance observations and that the simulated runoff fed by meltwater from the glaciers correlated with measured river runoff. The used dynamic glacier model describes the glacier as a 1D flow system based on the equation of continuity of ice, assuming a unique ice density and a flux-geometry relationship incorporating Glen's flow law. The width of the flow channel was estimated from recent maps and independently from the glacier thickness in order to ensure the same hypsography as observed for the investigated glaciers. Each outlet glacier is considered independent of other outlet glaciers from the same ice cap. Two longitudinal glacier profiles are shown in

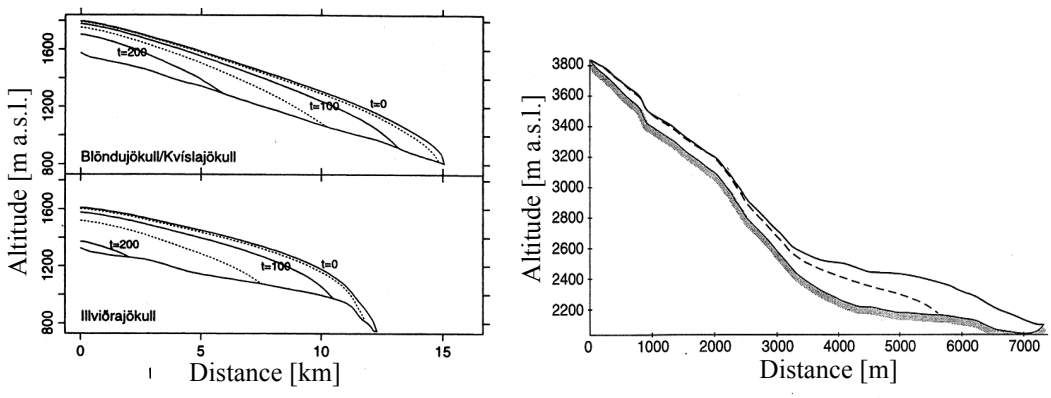


Figure 2.7: Left: Longitudinal profiles of two Icelandic glaciers as a function of time (from Jóhannesson, 1997). Right: A calculated bedrock and two surfaces of the Morteratschgletscher. The upper solid line refers to the surface in 1973. The dashed line represents the simulated surface assuming a warming of 1 °C (from Haeberli et al., 1999a).

Figure 2.7 left as a function of time for t = {0, 100, 200} (solid lines) and t = {50, 150} (dotted lines) years after the climate starts to warm. Air temperature is assumed to increase by 0.25 °C (mid-summer) or 0.35 °C (mid-winter) per decade and precipitation by 5% per °C of warming which has recently been defined for the Nordic countries.

Steady-state conditions combined with flow-line approach: Haeberli et al. (1999a) developed a 1D flow-line model based on the idea from Paterson (1994), i.e. combining Equation 2-3 with the concept of perfect plasticity to estimate the glacier thickness, and performed simulations of the retreat of several glaciers in the Alps forced by two different climate warming scenarios. A calculated bedrock and surface of the Morteratschgletscher (Switzerland) assuming a warming of 1 °C is visualized in Figure 2.7 right. The thickness along the flow-line is computed as a combination of a so-called ice cap factor and relief factor (Haeberli et al., 1999a) derived from the glacier surface. Due to the complexity of the climate-glacier relation, they parameterised the glacier response and approximated it as a step-wise static adjustment from one steady-state to another whereby no dynamic effects are considered (Haeberli & Hoelzle, 1995). This steady-state approach is restricted to time scales larger than the response time of the investigated glacier: short-term fluctuations are not considered. Since this method supports computation of a rough image of the glacier bed (in contrast to other methods), it requires no prior information about the glacier bed. The mass balance change is approximated using the method of Kuhn (1990) as described in Chapter 2.3.2. The correspondence between simulated reconstructions of historic glacier states and measured reference data is satisfying (Hoelzle et al., 2003). Differences may be caused by simplifications introduced by the parameterisation, limited data and/or changing climate-mass balance circumstances (Haeberli et al., 1999a).

Comparative study: Oerlemans et al. (1998) compared dynamic ice flow models for 12 temperate glaciers and ice caps of different sizes by forcing them with various climate change scenarios. One of the long-term goals of their project is to find ways to deal with all glaciers in a simplified scheme. The majority of the used models assumes that ice velocity is determined by the local basal shear stress τ. As prognostic equation the vertically integrated continuity equation, describing the relation between changes in ice thickness d and flux divergence and specific balance, is applied. For most glaciers, an approach with one or more flow-lines parameterising the 2D geometry is used. The flow models were calibrated with historic length records by seeking a mass balance history in such a way that observed and simulated length variations match. They found no straightforward relationship between glacier size and fractional change of ice volume. Furthermore, differences in hypsography between glaciers makes it difficult to generalize results.

It is unclear how the behaviour of glaciers can be generalized to make inferences about all glaciers on earth (Oerlemans et al., 1998). Attempts to deal with all glaciers in a single equation may be illusionary (Gudmundsson, 1999). Oerlemans et al. (1998) remark that it may be promising to apply simpler models, such as Haeberli & Hoelzle (1995), to more easily study large data sets of glaciers.

Schneeberger et al. (2001) combined a glacier mass balance and a 3D ice flow model of high complexity with an Atmosphere-Ocean General Circulation Model (AO-GCM) to forecast the evolution of the mass balance and the ice volume of a small well-studied glacier in northern Sweden. Despite promising simulation results, they conclude that the need for a good database for each glacier modelled with their method limits applicability to few glaciers worldwide. But these glaciers can in turn be used to assess the accuracy of simplified models that are applicable to a larger set of glaciers (Schneeberger et al., 2001).

To validate and calibrate specific models it is indispensable to continue and extend observations, e.g. variation in length or mass balance, in a systematic way (e.g. Oerle-

mans et al., 1998, Haeberli et al., 2000). The role of the World Glacier Monitoring Service (WGMS) in collecting and compiling glaciological data all over the world is therefore crucial. Operational elements of a worldwide glacier strategy for monitoring glacier fluctuations are published in Haeberli (1995).

Chapter 3

Methods in computer graphics and geovisualization

In the first part of this chapter, approaches for the visual modelling and rendering of animated objects are presented. The second part comprises dedicated aspects of visualizing objects in the geospatial domain.

3.1 Modelling and rendering of 3D objects

In general, computer graphics comprises all methods, algorithms[1] and data structures used for the mathematical or geometrical description of a real object or process and its transformation into a visualization by means of computers.

The conversion of a high-level object-based description into a graphical 2D image for display is called rendering. The rate at which images are displayed is measured in frames per second (fps) or Hertz (Hz). In the context of computer graphics, an application is real-time if the display rate is higher than 15 fps[2]. Below 6 fps the user does not have a sense of interactivity (Möller & Haines, 2002). In this context, hardware-accelerated graphic systems gain more importance.

An application-oriented reason for the huge effort and the resulting fast evolution in the past, is the aim to simulate objects and processes most realistically. Different techniques like geometric modelling, culling or lighting developed in the 1970's and 1980's are used in so-called photo-realism to render images that are visually identical to the real object.

From a technical point of view, the motivation for developments in computer graphics comes from the improvement of hardware and advances in software techniques accessing

1. Hereby an algorithm is a computing procedure that produces a result based on a given input and after a defined number of computing steps (WordNet, 2003).
2. In the context of operating systems, a similar but different definition of 'real-time' is used.

image-generating hardware. Most important was the development of the raster display where the whole image is stored in a hardware buffer called a frame buffer. Aside of the de-facto standard for image-generating developed in the 1970's a new perspective is provided by programmable graphics hardware and the use of shading languages (Rost, 2004).

In the following, the rendering process is explained schematically as implemented in most of the low-level hardware.

3.1.1 From 3D objects to 2D images

The path from a geometric object to a two-dimensional image or projection is the core of each graphic system. Because it is a sequentially organized procedure, it is called a pipeline or more exactly a rendering pipeline. The main function of the rendering pipeline is to render a two-dimensional image, given a virtual camera, 3D objects and light sources. The geometric model of the object can be 2D or 3D, but it is assumed in the further course of this study to have three dimensions in space. Figure 3.1 shows the different conceptual stages within the rendering pipeline (Möller & Haines, 2002).

The first stage of the pipeline is the *application* stage that is fully implemented in software and where the developer has direct control over performance. At this stage, geometry, appearance, and dynamic behaviour of the objects are modelled by application software. The application stage is also the place where input from other devices such as a keyboard or mouse is processed.

The *geometry* stage comes next and includes the majority of all per-polygon and per-vertex[1] operations (Möller & Haines, 2002). There is a current trend to move the implementation of this stage from software to hardware. Objects defined in a specific object space are transformed into the common world space where normally lighting is also per-

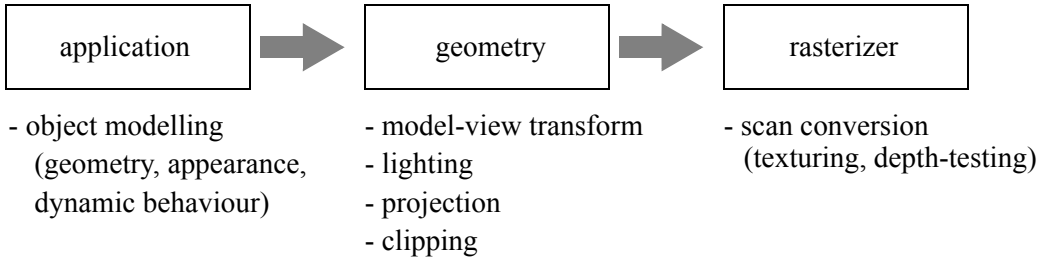


Figure 3.1: Conceptual stages of the rendering pipeline: application, geometry and rasterizer stages.

formed. After setting camera position and viewing angle, the view transform places the camera at the origin. The space thus delineated is commonly called the eye space. For the following projection the so called viewing frustum must first be defined. In defining an orthographic projection, the viewing frustum is a simple rectangular box. A truncated pyramid with a rectangular base is applied for a perspective view. Only the primitives com-

1. A vertex is a point in three-dimensional space.

pletely within the viewing frustum need to be passed to the next stage. Primitives partially inside are clipped against the projected viewing frustum (*clipping*).

The last stage is called *rasterizer* stage and aims to assign actual colours to the picture elements (pixels) in order to render the actual image. Modern graphic systems implement this stage exclusively in hardware. The main procedure is the scan conversion that converts two-dimensional vertices in screen-space - with a depth-value, a colour and possibly a texture coordinate associated with each vertex - into pixels. The rasterizer stage handles per-pixel operations and stores the pixel information in the colour buffer, a rectangular array of colours consisting of the three components red, green and blue. The z-coordinate or depth-value is stored separately in the Z-buffer (also called depth-buffer) used to perform visibility tests. Together with the Z-Buffer and the colour buffer, there exist additional buffers such as the stencil buffer and the alpha channel. The stencil buffer can be used to control rendering into the colour buffer and Z-buffer. The alpha channel is associated with the colour buffer and stores a related opacity value.

When the primitives have passed the rasterizer stage they are displayed on screen. In order to avoid seeing the update of the frame buffer on screen, double buffering is used. The primitives are rendered into a back buffer and after completion the content of this back buffer is swapped with the front buffer that was previously displayed on the screen. The digital information is then converted by the video controller on the graphic card to an analog signal for the output device.

It is obvious that the complexity of the object influences the time necessary to traverse the pipeline. The more computation needed for lighting or transformation of vertices, the lower is the rendering speed. The slowest of the pipeline stages, i.e. the bottleneck, determines the overall rendering speed. To increase the frame rates the bottleneck in the rendering pipeline must be located, and optimization methods have to be applied at this stage (Möller & Haines, 2002).

3.1.2 Geometric modelling of 3D objects

The representation of an object as a mathematical or geometrical model for efficient rendering has been a major research topic since the early beginning of computer graphics. Most approaches are approximations that nevertheless result in acceptable images. The appropriate way to mould an object depends on its characteristics, on the desired computer graphic technique, and the application.

In the context of geometric or object modelling, the term modelling refers to the process of describing the shape of an object. An object is an entity that has a 3D geometry. The shape of an object includes a set of points in object space that make up the object's surface or volume. The genus of an object refers to how many holes there are through it. The term topology refers to the vertex-edge-face connectivity. By twisting an object, the topology remains the same but the geometry is changed. The separation between specification and representation is important when dealing with modelling approaches. In geometric design, the specification depicts the creative step that specifies a geometric object. Representation is the productive step that produces a concrete representation of the object

for later storage or rendering. In the following, the six most important techniques for object modelling in computer graphics are discussed (Watt, 2002).

Polygonal representation: This is the most commonly used approach although it is more of a technical-oriented approach than a user-oriented description for intuitive manipulation. Objects are approximated by a set of several planar polygons. The achieved accuracy is arbitrary and increases with the number of used polygons. There are three reasons why this approach is still omnipresent in computer graphics: 1. Many objects show a data structure which is suitable for the generation of a polygonal description, 2. there is no limitation in shape complexity and 3. its rendering is well implemented in hardware and software with many different algorithms. Because control of polygonal meshes (e.g. for deformation) can be difficult, this approach is poor for animating complex geometries. An example of a polygonal mesh representing a cow is shown in image a) of Figure 3.2.

Parametric patches: An object can be exactly defined or approximated by one or more parametric patches. Such a patch is similar to a quadrilateral used in a polygonal description except that the patch can be curved instead of planar. Each parametric patch is described by a mathematical function and control points indicating its position and shape, enabling the calculation of any point on the surface. The parametric approach is well suit-

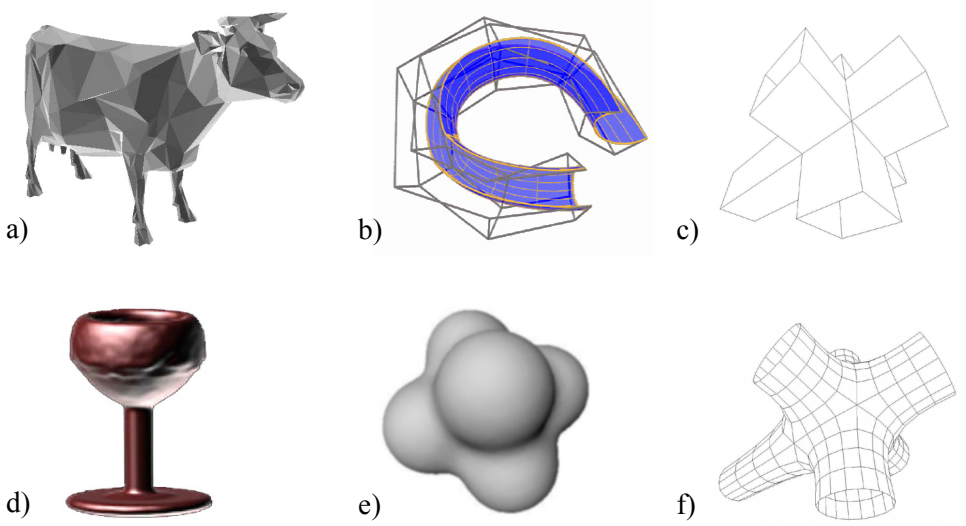


Figure 3.2: Examples of object representations: a) a polygonal representation of a cow (Möller & Haines, 2002); b) a NURBS surface with its control polygon (TU Wien, 2002); c) a BRep of a CSG model (Möller & Haines, 2002); d) a volumetric model of a melting goblet (Wei et al., 2003); e) a six point-skeleton implicit surface (Cani, 1999) and f) a subdivision surface generated from the CSG model shown in image b) (Möller & Haines, 2002).

ed for geometric operations like shape modification and efficient description of curved objects. Limitations on its use are the more difficult rendering and discontinuity problems that occur when a patch is changed within a net of several patches (Farin, 2002). The most important representatives of parametric patches are the Bézier-surface (Farin, 2002) and

the Non-Uniform Rational B-Spline (NURBS) surface (Piegl & Tiller, 1997). An example of a NURBS surface is shown in image b) of Figure 3.2.

Constructive solid geometry (CSG): The main idea of the CSG approach is to construct an object by combining a collection of simpler primitives called solids, e.g. sphere, cone, cylinder. A solid or solid model is defined as a surface and its interior. The CSG representation is a volumetric description and aside from advantages in constructing complex parts in engineering applications a major drawback is the time-consuming rendering. Thus, a CSG model is often converted to its boundary representation (BRep) for rendering purposes. See image c) in Figure 3.2 for a BRep of a CSG model. A novel approach for faster CSG rendering is provided by image-based rendering algorithms. They rely on modern graphics hardware and multi-pass rendering that includes reading and writing large amounts of pixel data from and to the frame buffer (Kirsch & Döllner, 2004).

Volumetric methods: These are methods that subdivide the object space three-dimensionally into volumetric elements (voxels) of equal size and assign to each element information about the object being represented. The use of voxels is memory-intensive - this approach is therefore generally not preferred for real-time rendering. But there exist justifications for appropriate use of volumetric methods. One reason may be that the available raw data may already be of volumetric nature, as is the case in medicine or fluid dynamics. A volumetric model of a melting goblet (Wei et al., 2003) is shown in image d) of Figure 3.2. The kind of application also plays a role. E.g., ray tracing is performed more efficient for a volumetric representation.

Implicit representation: In general, an implicit surface is defined by the collection of points $\mathbf{Q}$ that satisfy the implicit function $f(\mathbf{Q}) = 0$ (Bloomenthal, 1997). The surface is referred to as implicit because it is only implicitly defined, not explicitly represented. More complex objects can be defined by using the weighted sum of several implicit primitives, leading to a blend of these implicit primitives (see image e of Figure 3.2). To render an implicit surface, an explicit representation of the surface is generated by sampling the implicit function at the vertices of a 3D mesh (polygonization). Advantages of the implicit approach are the ability to enclose the volume and to represent blends of volumes. The point classification, i.e. if a point lies inside or outside the surface, is easier to perform with implicit surfaces than with parametric patches or polygonal meshes. In contrast to this, only a few analytical shapes can be defined by one implicit function leading to difficulties in specifying objects of high complexity.

Subdivision surfaces: A polygonal mesh can act as an initial mesh or as a so-called control mesh for a subdivision scheme. In a first phase, new vertices are created based on the control mesh and connected to new smaller triangles. The second phase is the smoothing phase, where the position of some or all new vertices are recomputed to get a smoother triangle mesh. In this way, the resulting subdivision surface is a continuous and curved surface (see image f of Figure 3.2). The subdivision surfaces provide infinite levels of detail, as the number of new triangles or polygons is unlimited. Furthermore, there are different subdivision schemes interpolating or approximating the control mesh (Warren & Weimer, 2001).

Summarizing the above considerations, there exists no general solution for modelling an object. The appropriate choice depends on the object(s), the intended modifications such as animation, the desired rendering speed, and the required accuracy of the object

representation. In order to specify a suitable method for the geometric modelling of glacier fluctuations the presented modelling methods are compaired in more details in Chapter 4.1.2.

3.1.3 Lighting and shading

An object is visible when it reflects some fraction of incoming light or when it is a light source itself transmitting some light. Each object has its own material properties describing the reflection behaviour of its surface. The material is specified by parameters for the ambient, diffuse, specular and shininess components.

The computer graphics term *lighting* describes all interactions between materials and one or more light sources. A light source can be a directional light positioned infinitely far away, e.g. the Sun, or a positional light as a spot or point light. For the computation of the illumination at one vertex, the Phong model can be applied as a local lighting model. It simulates scattering effects as a linear combination of specular, diffuse and ambient components.

Phong's illumination model has little physical validity; it just looks reasonable under most circumstances. More realistic multiple scattering approaches simulate indirect light sources and are included in global lighting models as used in ray tracing or radiosity algorithms (Watt, 2002). High computing costs diminish or even exclude their use in real-time applications. In reality, light intensity depends not just on the direction of reflection but also on the lighting angle. This circumstance is considered in the bi-directional reflectance distribution function (BRDF) specific to each material and wavelength of incoming light. The BRDF used in Phong's model is extremely simplified, reducing the computation of light intensities to vectors describing light position, surface shapes and viewing angle.

Shading is the process of performing lighting computations and determining pixel colours from them. There are three different techniques: flat shading, Gouraud shading and Phong shading. These correspond to computation of the light per-polygon, per-vertex or per-pixel (Foley et al., 1995). The Phong shading algorithm is much better suited to simulation of specular effects, whereas Gouraud shading computes the diffuse reflection much faster and with equal quality. The flat shading results in clear discontinuities in the case of curved surfaces, as the light intensity is calculated just for one polygon normal and the same intensity is applied over the whole polygon. Shading can dramatically improve the retrieval of 3D shape and the perceptual grouping of objects (Kleffner & Ramachandran, 1992).

3.1.4 Texture mapping

Texturing or *texture mapping* is the process of storing information in a texture space and using it for the simulation of a texture during rendering. Generally, the texture space is two-dimensional, but can also be one- or 3D. The goal of texture mapping is to map a generally two-dimensional image onto an object specified in 3D object space (or rarely in world space). The surface properties are modified by replacing the original diffuse colour

with the texture information from the image. The texture mapping can be explained in two steps: the mapping of the texture image onto the 3D object (parameterisation) and the projection of the texture-assigned object to the screen space. Rather than this forward mapping, the backward mapping is often preferred in practice where the corresponding texture pixel (texel) is assigned directly to the screen pixel (Watt, 2002). If backward mapping results in texture coordinates outside the texture dimensions, i.e. u_t or $v_t \notin [0, 1)$, special functions, e.g. repeat, mirror, clamp and border, can be defined (see Möller & Haines, 2002).

To specify the way a texture modifies the original surface colour so-called texture blending operators can be applied. The 'replace'-operation simply replaces the surface colour (including the shading) by the texture colour. The colour is multiplied with the texture when the blending operator 'modulate' is chosen.

There are three different approaches to generate texture images. Procedural texturing includes all methods where mathematical functions are used to synthesize a texture. The noise-function serves as a typical procedural approach (Ebert et al., 2003). The artistic creation of textures is still a widespread technique for generating texture images. The third possibility is to measure the spectral properties of the object with a sensor and to store the measured scattering values as an image.

Since rendering involves the projection from 3D to 2D as well as the sampling of the projected object, aliasing may occur. Aliasing is prevented when the sampling frequency is at least twice the maximum frequency of the signal frequency (Nyquist-theorem). Several strategies are discussed in Ebert et al. (2003).

One way to produce changes in appearance is the use of animated textures, whereby the nominally static relation between texture and object coordinates is released and the texture coordinates are changed from frame to frame.

3.1.5 Programmable graphics hardware

The recent trend in graphics hardware has been to replace fixed functionality of the rendering pipeline with programmability in areas that have grown exceedingly complex, e.g. vertex processing and fragment processing. This provides new perspectives for the rendering of complex (shading) effects in real-time applications by extending the functionality of the standard rendering pipeline. Thereby, it is possible to write so-called vertex shader and fragment shader programs for the individual and complex usage of the graphics processing hardware (Rost, 2004).

3.2 Animation of 3D objects

The introduction of animated objects into the world of computer generated visualizations brings them to life. Visualizations are no longer restricted to static scenes: objects can change their shape, position and appearance. The factor time becomes an essential part. This chapter covers the main animation concepts and techniques in the field of computer graphics. Furthermore, methods for the animation of fluid-like objects are presented.

3.2.1 Animation control

Animating is moving something (or making something appear to move) that cannot move itself (Parent, 2002). A wide range of animation techniques were already developed that apparently have very little in common making it difficult to categorize them. Concerning the main approaches, a classification proposed by Cerezo et al. (1999) is used. They grouped the different animation techniques based on the control models used in the motion modelling and behaviour modelling categories. Motion modelling includes *descriptive* or guided models (a), *generative* models (b) and *low-level task-oriented* models (c). The behaviour modelling consists of *higher-level task-oriented* models (d).

a) *Descriptive* models: They reproduce a motion effect without knowledge of the causes. Key-framing is one of the most traditional and widespread animation techniques, interpolating so-called key-frames linearly, constrained to a point, based on parametric curves or quaternions (Parent, 2002). Another descriptive approach is the procedural model, using a procedure to control or animate some attribute of the object. Procedural models can be script-based systems using animation languages, models using kinematics or geometric deformation models. Examples of deformation models are the non-linear deformation approach (Barr, 1984), the more general free-form deformation technique (Sederberg, 1986) and an animation technique based on NURBS (Preston & Hewit, 1994). The animation of deformable models using implicit surfaces is presented in Cani-Gascuel & Desbrun (1997). Motion capture of an object involves sensing, digitizing and recording the object in motion.

b) *Generative* models: The search for realism in other computer graphics fields and the increasing computer power in the late 1980's made it possible to describe animated phenomena in terms of physics. In dynamic systems, objects become masses with forces and torques acting on them. An advantage of such dynamic systems over classical kinematic approaches is the greater realism and the automatic reaction to the environment. Since the physically-based systems use complex mathematical models to control and simulate the movement of bodies, the computational requirements are very high and difficult to achieve in real-time. Furthermore, the motion is difficult to control. The current challenge in physically-based animation is the struggle to obtain user-controlled realistic motion with small computation times (Cerezo et al., 1999).

c) and d) *Task-oriented* models: Since the 1990's task-oriented animation systems have been developed providing animated objects with environment perception, decision, action and communication ability. The motor or *low-level* control (c) simulates the movement based on dynamic or kinematic models. In motor planning (or *high level*) systems (d), perception is connected to action.

3.2.2 Basic animation techniques

In the following basic animation techniques such as interpolation, deformation, morphing, the use of implicit surfaces, and particle systems are introduced. A more exhaustive overview of animation approaches is given by Parent (2002). The selection of a method depends on the desired accuracy, complexity, continuity and spatial domain.

Interpolation is a fundamental technique in animation tasks. Since the control of an interpolation procedure can either be script-based, key-framed or analytical the animation can be exactly defined in contrast to other more sophisticated algorithms such as task-oriented modelling. The motion along an interpolating curve can be controlled by specifying a distance-time function (Parent, 2002). An example of a smooth distance-time function with first order continuity is shown in Figure 3.3. The first part of the in total three time segments is described by the so-called ease-in function simulating in this example a constant acceleration. The middle segment has constant speed whereas the last one is defined

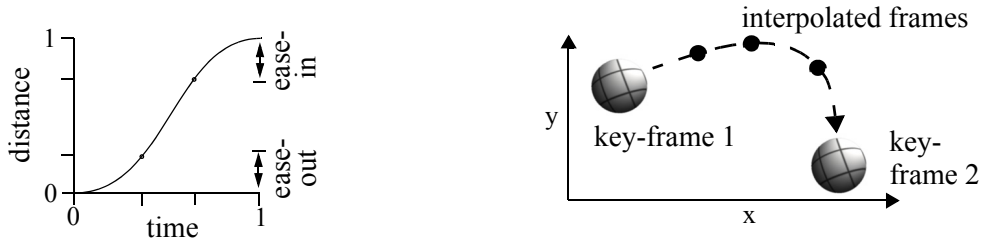


Figure 3.3: Left: An example of a distance-time function with ease-in/ease-out. Right: A non-linear key-frame interpolation.

by an ease-out function with constant deceleration. A prominent approach using interpolation is key-framing, where the transition between two key frames is achieved through several interpolation steps. On the right side of Figure 3.3, a textured sphere is non-linearly interpolated between two key-frames along a predefined motion path. If the animated frames are displayed at a rate slower than the period of visual retention, the animation becomes jerky because the frame changes are perceptible.

Deformation of objects is a technique producing realistic animations of flexible bodies. Different techniques to deform an object with much less computations and more control about the transformations has already been developed. Barr (1984) presents a global deformation method transforming the space in which the object is defined using a 3 x 3 matrix. One of the most popular and more general methods is the so-called free-form deformation (FFD) introduced by Sederberg (1986). During the 1980's the computer graphics community began exploring physically-based methods for animating and modelling deformable objects (Terzopolus & Fleischer, 1988), e.g. mass-spring models, continuum models and low degree of freedom models. A survey of studies on deformable models is given by Gibson and Mirtich (1997).

The 3D morphing includes all methods making a transition from a 3D source object to a target object. A survey of volume- and surface-based metamorphosis (morphing) approaches is given by Lazarus & Verroust (1998).

Implicit surfaces provide unique possibilities to animate their shape. Organic-looking shapes and their deformation can be realistically reproduced and since changes in genus are easily handled with implicit functions, they are well suited for the modelling and animation of fluids and elastic material.

Animated particle systems are used mainly in the field of physically-based animation. A particle system consists of many individual elements that together represent one larger object. The individual particles typically behave according to simple physical models and may change their attributes (e.g. position, velocity, shape parameters, colour, lifetime) during their lifespan.

Carlson & Hodgins (1997) describe a method for reducing the computational cost of simulating groups of creatures by using less accurate simulations for individuals when they are less important to the observer or to the action in the virtual world. The dynamic simulations (higher level of detail) are approximated either by dynamics with fewer degrees of freedom or kinematic animation (lower level of detail).

3.3 Advanced rendering techniques

3.3.1 Multi-pass and multi-texturing

Hardware-accelerated processes are much faster than software-based techniques but they lack the flexibility for combining more complex tasks. The *multi-pass* approach makes it possible to perform different tasks within the rendering pipeline, e.g. lighting, each in a separate pass. After computing the different passes they are combined by 'add' or 'blend' operations. Adding two passes means adding (or subtracting) the texture colour to (or from) the surface colour. Blending is used to blend textures having transparent colours. Obviously, rendering performance decreases as the number of passes increases.

Some graphics systems support the combination of several textures in the same pass. This approach, called *multi-texturing,* takes advantage of the fast blending capabilities of graphics hardware and avoids the rendering overhead of a multi-pass system. An example is shown on the left side of Figure 3.4, where in a first stage the second texture is added to

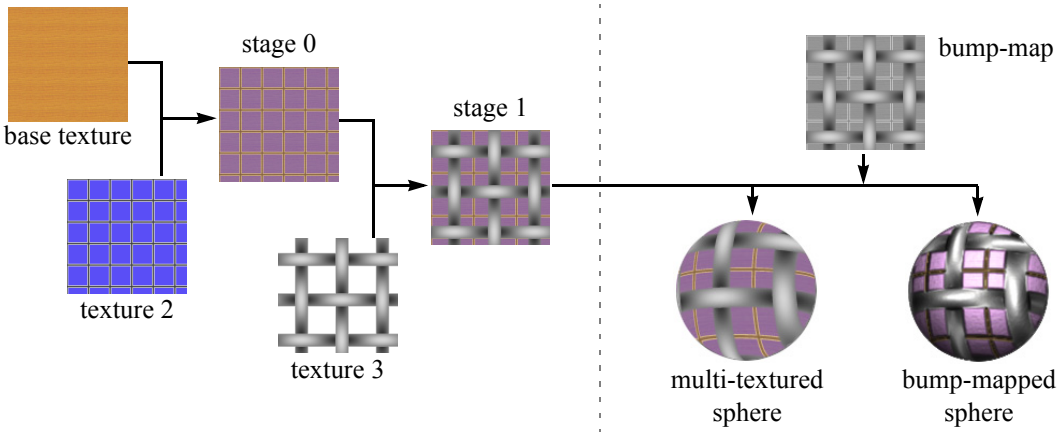


Figure 3.4: Left: multi-texturing of three textures in two stages (stage 0: modulating base texture with texture 2; stage 1: blending of texture 3 onto result from stage 0). Right: bump-mapping of a multi-textured sphere.

the base texture, and in a second stage the third texture is blended onto the result from the first stage of the same pass. *Alpha mapping* uses the alpha value to make some part of the third texture transparent or invisible.

Another texturing method is *bump-mapping*, where the normals-per-pixel of a surface used for the lighting are modified based on information from a texture (bump map). Bump-mapping gives the illusion of detailed geometry although the surface remains smooth in the geometric sense (see Figure 3.4 right).

The interactive generation and rendering of textures that are applied to an object is called *dynamic texturing* (Döllner & Hinrichs, 2000b). It is a useful method for improving the interactive functionality of a virtual reality system.

3.3.2 Rendering optimizations

As long as man dreams of producing ever more realistic visualizations, computer power will be the limiting factor in computer graphics. Thus, optimization strategies for improving the rendering efficiency remain important. One main approach to free processing power is reduction of the number of rendered primitives. Other solutions are based on software adjustments and considerations about general rendering properties.

Taking into account the geometrical properties of the rendering process, several approaches exist to lower the number of primitives without any significant visual loss. The level of detail (LOD) approach reduces the resolution of an object far away. It generates several levels of detail of the object, then selects the appropriate level based on specified criteria, and switches from one level to the next as necessary (Möller & Haines, 2002). Culling is the process of removing everything from a scene that will not contribute to the final image, including things that are behind the observer. Special types of culling are hierarchical view-frustum culling, backface culling and occlusion culling (Möller & Haines, 2002). Image-based rendering techniques provide useful methods to substitute 3D geometric models with their two-dimensional texture-mapped image. Billboarding is a powerful approach that makes polygons facing towards the viewer. Simplification methods lower the geometrical resolution by collapsing edges of less importance as they do not contribute much to the shape of the object. An efficient description of several triangles or polygons can be achieved by sharing common vertices. Strips, fans or polygon meshes are examples of such a compact description (Möller & Haines, 2002).

Furthermore, software-related issues are presented by Möller & Haines (2002) in order to improve the rendering speed. A display list stores a series of graphics API calls in an optimized command sequence, typically for objects that have an unchanging shape and material. The use of display lists (or retained mode) leads to an improved access pattern in the case of static objects. Animated objects (possibly in combination with changed texturing or lighting circumstances) are better rendered in immediate mode (e.g. OpenGL ARB, 1999) without any display lists. Increased performance can also result by grouping objects with similar properties to minimize the number of state switches. Callback-procedures, e.g. reading the frame buffer, are expensive, as they read intermediate results from the pipeline, and should therefore be avoided.

3.3.3 Scene graph modelling

To efficiently perform the culling, it is beneficial to organize the objects into a hierarchy or tree propagating any shared information towards its root. There are different ways of partitioning the scene into a hierarchical structure, but the most often used approach is the directed acyclic graph (DAG), holding hierarchical organized bounding volumes. A bounding volume is a volume that encloses completely a set of objects and is used to increase the performance of intersection tests. The rendering of a scene organised in a DAG is done generally by traversing the scene graph three times. After the application traversal including moving objects, new texture applications etc., the culling traversal removes all nodes outside the viewing frustum. During the last so-called drawing traversal, visible objects are sent to the geometry stage of the rendering pipeline (see chapter 3.1.1).

The view-frustum culling of the scene becomes faster using a DAG as objects completely outside the viewing frustum can be removed at an early stage, avoiding additional transformations. In the case of rendering in retained mode, additional improvements can be achieved by culling and drawing in parallel and by state sorting. A state contains information about the used texture map, lighting values, transparency, blending properties and others, and can be assigned to a node.

The evolution of scene graph approaches was driven by the increased culling performance over low-level rendering. As materials and lighting became more and more important, state sorting was the next big issue of scene graphs. Normally, state nodes were included in the former spatial organized structure, leading to an inefficient node hierarchy concerning the aim to aggregate common states. Future scene graph strategies will be directed towards multiple views of the scene, resulting in more than one scene graph. The different tasks could be separated into different scene graphs representing spatial, state, semantic and application views (Bar-Zeev, 2003).

3.4 Geovisualization

Until now, general methods to model objects by specifying their geometrical and visual characteristics, to animate changes in shape and appearance and to render the objects efficiently were presented. The difference between geovisualization and basic visualization approaches lies primarily in the kind of data and processes of interest. Geovisualization is characterised by rendering georeferenced objects (geoobjects), i.e. objects linked to a geospatial domain. The main task of geovisualization is to provide means for presenting geographic data in a virtual environment.

3.4.1 Geoobjects and geographic data

A geoobject is an entity that has a 3D georeferenced geometry and semantic information indicating the meaning of the object. It is used as a synonym for the term geographic data which is a formal description of *what* is present, and *where* it is.

One can distinguish the terms *information* corresponding to real world phenomena and *data,* a formalized encoding of the information (Hirtz, 2003). Several data models are used to store the complex information about the real world in digital representations. Such models represent the geometric and thematic properties of the real world. They allow the user to analyse their content regarding geometry and theme. The thematic model defines the semantic object properties. Geographic data can be characterised by the properties of acquisition, spatial localization, scale / resolution, dimensionality, and quality (Hirtz, 2003).

Geographic data is typically managed by a geographic information system (GIS), storing the geometric and the semantic information of geographic data in a database management system (DBMS). A GIS offers primarily functionality for the collection, management, analysis and visualization of geographic data. Geographic data may be inadequate with respect to accuracy, completeness and structure. The need to validate, edit and update the data has led to the inclusion of powerful graphical editors in GIS. Due to the widespread limitation of current GISs to two dimensions (usually easting and northing) one of the major research topics in GIS sciences is the incorporation of the third spatial dimension and time as a fourth dimension into GIS.

Traditionally, in GIS sciences the term geographic data is the commonly used expression. Nevertheless, the expression *geoobject* is preferred here over geographic data to guarantee consistency with the definition of the term object in Chapter 3.1.2 and to make clear that one is dealing with objects rather than data.

Each geoobject contains either a well-known or an undefined error according to its visual and/or geometrical properties issued from inaccurate measurements or simulations. The visualization of uncertainty is beneficial to data interpretation. Pang et al. (1997) presents several possibilities for converting numerical errors into graphical representations.

Often geoobjects are visualized in combination with their spatial reference object, generally, the terrain. The terrain / landform represents the outermost layer of the Earth as an infinitly thin layer without additional man-made objects such as buildings or trees. As rule of thumb, half of the terrain cannot be seen when exploring the geospace since the backface of hills is traditionally occluded by the frontside. This implies that geovisualization has to be interactive, allowing the observer to change his / her viewing position and direction. The overall system performance could be further improved by applying an appropriate technique for terrain rendering as presented later in Chapter 3.5.1.

3.4.2 Visualization of the landscape and of scientific data

In geovisualization, the objects to visualize can be non-tangible (e.g. air temperature) or tangible (e.g. a building). Contrary to this broad sense of geovisualization, landscape visualization is restricted to the modelling and rendering of landscape models consisting of the main elements terrain, vegetation, water, man-made structures, animals (including people) and atmosphere (Ervin, 2001). A list of well established and current methods for rendering landscape elements can be found in Discoe (2004).

Another field of activity related to geovisualization is scientific visualization (ViSC). ViSC uses computer graphics principles and methods for visualizing data resulting from

scientific measurements and computations. The emphasis of ViSC is on using graphics to gain insights into multi-dimensional data sets produced by scientific simulations or measurements. Therefore, methods of ViSC are dedicated to visualizing objects with a high degree of abstraction and dimensionality. Due to the typical high dimensionality in ViSC, immersive techniques are applied especially in this area to give the user the experience of being surrounded by a virtual environment (Van Dam et al., 2000). Concepts and methods of visualizing information in general are presented in Spence (2001).

Until now research in geovisualization has mainly been focused on developing viewing methods for tangible geoobjects, whereas methods for displaying non-tangible items have been almost neglected (Kraak, 2002).

3.4.3 Adaptive levels of abstraction

The visualization of geoobjects may vary from abstract to realistic. In this context the term abstract means "...not representing or imitating external reality or the objects of nature.", whereas the term realistic refers to "... aware or expressing awareness of things as they really are." (WordNet, 2003).

According to the level of abstraction (LOA), geovisualization can be subcategorized into cartographic, schematic and realistic visualization. The cartographic visualization of geoobjects depicts a very abstract kind of visualization as symbols, glyphs or other means of traditional cartography are used. Schematic visualization is assumed to be closer to reality, as the main characteristics of a geoobject were likely to be noticeable.

Typically, abstract information for the cartographic visualization is managed by a conventional 2D GIS, where different thematic and geometric properties of geoobjects are stored in a layered structure. It is easy to increase the level of abstraction by using the semantic description of a geoobject and assigning an adequate simplification strategy. In the inverse case, i.e. adding more information to decrease the abstraction of the visualization, additional information about the geoobject or an assumption-based approach has to be available. To improve the realistic impression of landscape visualizations, elements such as trees, buildings, roads and atmospheric effects can be added to the terrain.

It is generally known that a better visualization of a problem leads to an improved understanding of the underlying principles. But which visualization technique is most useful for communication and teaching depends on the problem, the intended statements, and the audience. Bishop (1994) discusses the role of visual realism in communicating and understanding spatial changes and concludes that

- the non-scientific audience wants abstraction minimized and information context maximized, in an easy-to-understand fashion.
- if visualization is to extend beyond the communication role and thus, leads to scientific insights not otherwise available, a well tuned combination of symbolic with concrete and realistic information must be used.

Thus, sending more information to the audience does not automatically means that more is understood by non-experts. Too much information can confuse the observer within the virtual geospace. An adapted level of abstraction helps both, scientists and non-experts, to

focus attention to the important facts. Taking this into account, the run for realism in computer graphics forced by the game industry is a trend that should be questioned critically and adapted carefully to the field of geovisualization.

Geoscientific models simplify reality by abstracting relationships and real processes. The abstraction level applied for the visualization should be adapted to the level used in the physical model to preclude an overestimation of the capabilities of the underlying physical model. The correctness and the limitations of the simulation have to be declared clearly. Otherwise, a misinterpretation of the simulation could result. Although Bishop (1994) proposes a very low abstraction level for the non-scientific audience, a critical remark has to be added. Increasing the realism of a visualization causes the non-experts to trust the images less critically. They take the shown images for granted because they look realistically. This is a blindfold and sometimes fatal statement.

3.5 Modelling and rendering of geoobjects

In geovisualization, it is the terrain that usually acts as the geometric reference for the various geoobjects. It is therefore essential to visualize the terrain most efficiently with high visual accuracy and low rendering costs to provide a reliable visual reference and to save processor power for more theme-oriented issues.

3.5.1 Terrain modelling and rendering

The terrain information is usually stored digitally in a regularly sampled and rectangular grid, commonly referred to as a *digital terrain model* (DTM). As with maps, DTMs are available in a specific cartographic coordinate system serving as the geometric reference. A DTM simplifies the real terrain by approximating the true terrain at discrete steps and by neglecting the Earth curvature. Information about the shape of the terrain is produced either by survey techniques, remote sensing techniques or a procedural description.

A common way of viewing a terrain is to render a set of triangles that approximates the terrain surface. Nearly all terrain rendering approaches in computer graphics represent the terrain as a piecewise linear surface. Due to the massive data provided by a DTM, the application of a terrain decimation method is generally inevitable. There are two main approaches to simplify the terrain: the simplification of *triangulated irregular networks* (TIN) and the *continuous level of details* (CLOD) methods.

TIN methods create a strip of triangles containing the fewest possible triangles that satisfy an error criterium. Significant improvements over a straightforward transformation to a triangle mesh can be expected in areas of high linearity. As an extension, different LODs can be applied by using a set of TINs with different error thresholds as discrete LOD models. This simple approach can be combined with a tile-based subdivision of the terrain to vary the LOD depending on the observer distance (e.g. Hirtz, 2003). An advantage of TIN approaches lies in the fact that they are not restricted to regular grids and can be applied for the approximation of arbitrarily distributed points. A view-dependent triangulation, called progressive mesh, is presented by Hoppe (1996). To minimize the notice-

able popping caused by switching between discrete LOD models, he extend his method with the *geomorph* approach, interpolating the different geometries (Hoppe, 1998)

To reduce visual discontinuities caused by LOD switching, Lindstrom et al. (1996) propose the continuous LOD (CLOD) technique. The CLOD approach uses a hierarchical quadtree structure and a dynamic triangulation dependent on the observer distance, resulting in a smooth change of terrain resolutions. Duchaineau et al. (1997) present a CLOD algorithm called ROAM (*Real-Time Optimally Adapting Meshes*) for optimizing triangulations given view-dependent error metrics resulting in well adapted triangle meshes at high frame rates. In order to eliminate vertex popping, a geomorphing technique is applied by Roettger et al. (1998) that interpolates the vertex positions. All presented CLOD techniques require a regular terrain grid with identical sizes in both dimensions since they are all based on a recursive splitting algorithm producing regular triangles. Generally, CLOD approaches are distinguished by high performance combined with acceptable visual quality.

Baumann et al. (1999) introduced a generic data structure for the combined use of TINs and multi-resolution approaches, e.g. the CLOD algorithm of Lindstrom et al. (1996). The authors argue that many applications visualize simulation data that should not be converted to a uniform representation as required by a CLOD algorithm.

3.5.2 Rendering of thematic data in geovisualization

Thematic information plays an important role in geovisualization, as it depicts the real information to communicate or explore. Two different types of thematic information can be distinguished concerning their representation in geovisualization. Geotextures refer to surface-like information to be draped on the terrain using traditional texture mapping procedures (see Chapter 3.1.4). The second group of thematic data is formed by graphical symbols not coupled with the terrain.

These graphical symbols are described in an abstract manner using image-based rendering methods, i.e. techniques portraying objects as images. The simplest image-based rendering primitive is the *sprite*, an image that can be positioned arbitrarily in the geospace. This image is shaped by a two-dimensional, often semi-transparent, texture applied to a rectangular polygon. Billboards are another type of image-based primitives that are oriented towards the observer (Möller & Haines, 2002). Basic 3D shapes providing any type of 3D geometry can be used for visualizing the third dimension of a standard geoobject, e.g. height of a building (Döllner & Hinrichs, 2000b).

Geotextures are superimposed on the terrain and can be derived from the terrain, i.e. topographic textures. They can also describe additional thematic information, i.e. thematic textures. Topographic textures are procedurally generated based on information from a DTM and may include geomorphological properties such as slope (see an example in Figure 3.5 left), exposition or hypsometry. The topographic shading map represents the best

known topographic texturing technique and can provide great morphological detail combined with high rendering performance (Döllner & Hinrichs, 2000b).

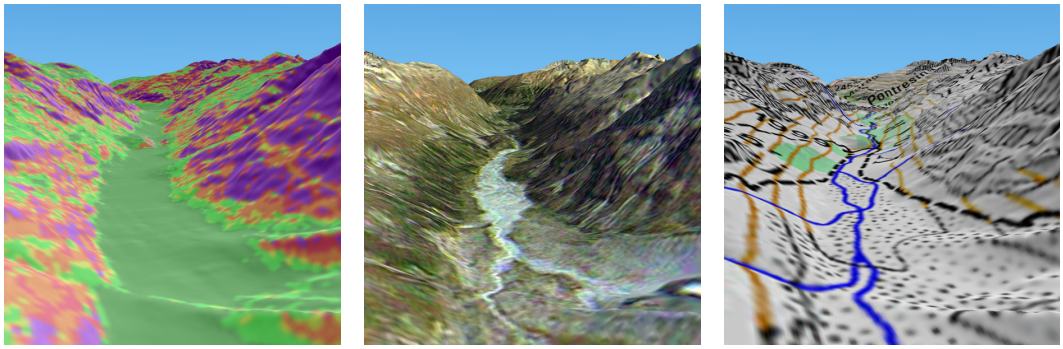


Figure 3.5: Different geotextures draped onto a DTM. Left: a procedurally derived slope texture where the colour indicates a specific range of slope. Centre: a satellite image as a geospecific texture. Right: the cartographic pixel map PK200 (Swiss Federal Office of Topography, 2000) as geotexture. Data sources: DTM DHM25 © Federal Office of Topography (BA024148), Enhanced Satellite Map of Switzerland (10 m pixel size) and pixel map PK200 © Federal Office of Topography (BA024148).

Depending on the chosen abstraction level, thematic information can be visualized either as a geospecific, geotypic or cartographic texture (Suter, 1997). Geospecific textures contain physically measured visual properties of real objects. In geovisualization, satellite images or airphotos are typically used for realistic visualizations of landscapes (see Figure 3.5 centre). A geotypical texture is a generic, procedurally or artistically generated image map representing a specific surface. To increase realism of geotypical textures, knowledge about BRDF can help to simulate real scattering effects. In this context, Dana et al. (1999) discuss a new texture representation called the BTF (bidirectional texture function), which captures the variation in texture with illumination and viewing direction. Cartographic textures consist of traditional information derived from cartographic maps as shown in Figure 3.5 right.

The approaches of multi-resolution modelling, multi-texturing, dynamic texturing and multi-pass rendering are identified by Döllner & Hinrichs (2000b) as belonging to the basic functionality of geovisualization systems. The ‘thematic lens’ is an example of combining multi-texturing and dynamic texturing with high-level blending operations in real-time (Döllner & Hinrichs, 2000b). It is a visual working tool for the combined analysis of multiple theme layers.

3.5.3 Animating fluid-like and geomorphological processes

There is a wide range of natural phenomena like plants, clouds or fire that have already been animated over the last years. Since these phenomena do not have a lot in common

with glacier flow and long-term fluctuations of glaciers, the following discussion is restricted to methods related to the animation of fluid-like and geomorphological processes.

In Fournier et al. (1998), a method is proposed to efficiently simulate the flow of liquid droplets by separating the shape and the motion of a droplet. The shape of a droplet is mainly based on mass-springs, whereas the motion is characterised by a particle system. Foster & Metaxas (1996) present a comprehensive method for animating liquid phenomena by solving the Navier-Stokes equations in two or three dimensions. Layton & van de Panne (2000) introduce a physically-based model based on the two-dimensional shallow-water equation to animate realistically water waves. The approach is restricted to thin layers of fluids so phenomena such as 3D flows cannot be modelled.

Nixon & Lobb (2002) propose a fluid-based soft object model to animate soft objects represented as closed elastic membranes filled with viscous compressible fluid (see Figure 3.6 left). The object moves under the influence of external forces and deforms in contact with other objects. Depending on the applied parameter settings, the simulated object's behaviour can vary from elastic to viscous fluid. Due to the polygon representation, the object's surface is rendered quickly, whereas implicit surfaces would have to be approximated by polygons or particles before they could be rendered at interactive rates. Until now the approach has been restricted to compressible fluids because the algorithm is stable only with a moderately high viscosity.

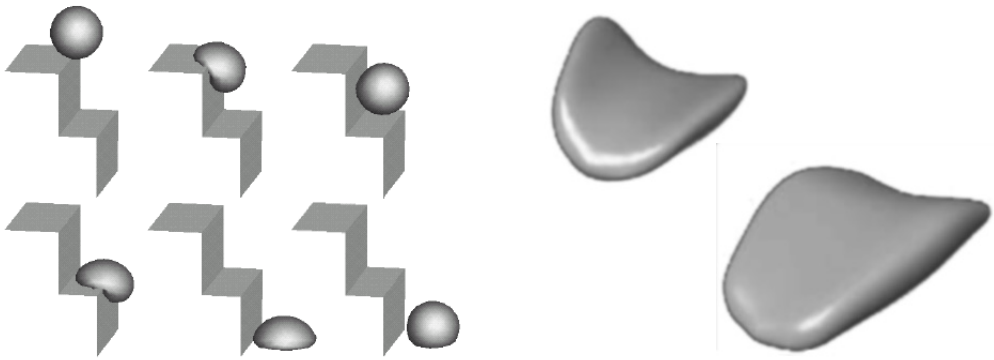


Figure 3.6: Left: A soft object falls down a pair of steps (Nixon & Lobb, 2002). Right: A parametric tongue model (King & Parent, 2001).

King & Parent (2001) present a 3D parametric tongue model for animated speech. There are obvious similarities in shape between a tongue and a glacier (see Figure 3.6 right). They use a geometric model composed of a B-Spline surface to produce a smooth surface that can be quickly deformed by displacing the control points. The real-time animation of the tongue is based either on a procedural, finite element, or keyframing approach. Volume preservation and collision detection between the tongue and the rest of the oral cavity, e.g. teeth, are neglected due to their computational requirements.

Stora et al. (1999) address the problem of animating lava flow (see Figure 3.7 left). Their approach relies on smoothed particles governed by a state equation for animating the flow. The animation relies on a macroscopic model for the lava flow including only a few intuitive parameters (e.g. mass, density, stiffness etc.). The appearance of lava is de-

scribed by a moving procedural texture controlled by the flow and combining colour and displacement information. The displacement is computed as the sum of a large scale shape and of Perlin's noise component (Perlin, 1985). On a SGI[1] O2 workstation, the rendering of one animated clinker takes less than one second per frame.

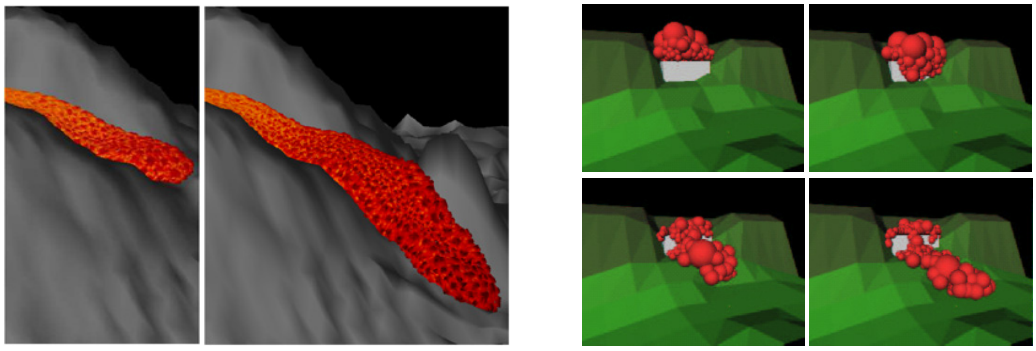


Figure 3.7: Left: a sequence of two images showing an animated lava flow (from Stora et al., 1999). Right: four images depicting a landslide (from Gascuel et al., 1998). A high LOA is applied.

Gascuel et al. (1998) discuss a simulator that handles both rock-falls and mud-flows using a space-time adaptive algorithm for animating a particle system (see Figure 3.7 right). They coat the adaptive particle system with an active implicit surface to filter the changes of granularity of the internal particle system and to achieve a smooth rendering.

An appearance-based simulation model of ground surfaces that can be deformed by the impact of rigid body models can be found in Summer et al. (1999). A simple method for simulating the topography of eroded mountains based on velocity fields of water flow is described by Chiba et al. (1998).

Wei et al. (2003) present a linear 3D cellular automata approach for animating the melting process of solid volumetric models such as lava flow in real-time. Their method relies on volume data sets. Fujishiro & Aoki (2001) implement mathematical morphology to model the ice thawing effect based on volume models. Since some of the ice may sublimate, mass is not conserved in their method.

3.6 Interactivity in virtual geospace

Two forms of interactivity in a virtual geospace can be distinguished. The interactive change of the viewing position and direction is called *navigation*. Selecting, manipulating and extracting information from the geoobjects is the second form of interactivity.

1. SGI stands for Silicon Graphics, Inc. (http://www.sgi.com).

3.6.1 Navigation

Moving through space is an integral part of our existence. Our choice of movement has been conceptualized as a process where an individual perceives and forms preferences for different action alternatives. A virtual 3D world is only as useful as the user's ability to get around and interact with the information within it. Van Dam et al. (2000) classify navigation within three categories: *exploration*, *search* and *manoeuvring* tasks. Exploration implies movement in space without any specified target. A search task exists if the observer has a concrete location in mind and would like to go to this target. Manoeuvring tasks are short precise movements to position the observer better for performing other tasks. Van Dam et al. (2000) divide travel interaction into five different categories: physical movement, manual viewpoint manipulation, steering, target-based travel and route planning. 1) Physical movement means physically moving the user's body to travel through the environment. 2) Manual viewpoint manipulation depicts a technique where the user grab points in space and pulls themselves along as if holding a virtual rope. 3) Controlling the motion by moving the viewing direction is called steering and is traditionally used for the 'fly-through' metaphor. 4) Target-based travel consists of all navigation methods where the user specifies the destination and the application handles the movement of the camera. 5) Route planning exists if the motion follows a predefined path.

Aside from these general navigation considerations, geospecific issues need to be considered for precise and non-confusing navigation. It should not be possible to go 'through' the terrain, i.e. move under the surface. Additionally, a horizontal alignment of the camera has to be ensured. The up-vector of the camera has to be directed upwards, otherwise the observer could become lost in virtual space. Döllner et al. (1999) propose further constraints on navigation in terrain. They suggest three different navigation strategies: the look-around, sphere and cone metaphors. All are designed for a precise navigation constrained to a small-area. The look-around metaphor fixes the viewer at a certain position and lets him freely choose the viewing direction. If the sphere metaphor is used, the viewer can examine a point-of-interest (POI) from a constant distance and from any viewing angle. The cone metaphor can be applied when the viewer should be able to look to the POI from any viewing angle and any distance. The viewer then moves on a cone whereby the cone point corresponds to the POI.

Verbree et al. (1999) propose a multiple view approach for better orientation and navigation in virtual geospace. The user can easily switch between the different views depending on the desired task to perform. The plan-view refers to a conventional digital two-dimensional map, where the user can create and manipulate the geoobjects as symbols. The model-view is a perspective view of a simple symbolic abstraction of the data. There is no immersivity, as the user just looks at the data. The world-view provides the most realistic view of the data or geoobjects. The observer is immersed within the object and can explore the data inside the object. The world-view is well suited for analyzing 3D and dynamic phenomena, as both the position and object size may change, and the observer has to adapt position and viewing direction. This kind of interaction requires real-time rendering of the geoobjects.

3.6.2 Interaction with geoobjects

Selecting a geoobject and extracting relevant meta-information is fundamental to analysis in a geovisualization system. For the verification of simulation models, Belleman et al. (1999) state that a major problem in the quantitative comparison of simulation results with actual phenomena is that in many cases there is no single discriminative feature by which they can be differentiated. They propose to obtain measurements on multiple properties in local areas of the data sets that together form a discriminative measure. They present software that can be used to instrument interactive environments with virtual probes to obtain quantitative information from geometric representations.

In Kreuseler (2000), a visualization system is presented for the visual investigation of marine ecosystems by analyzing large quantities of spatially and temporally dependent data, e.g. simulated fishery data. Several visualization techniques are applied to the 2D and 3D display of the spatial dependency frame, e.g. direct and indirect display of the spatial dependence, together with data reduction approaches (focusing, selecting).

Changing the size of an object, or positioning a new geoobject onto the terrain are manipulation tasks. The introduction of manipulation tasks makes the geovisualization system a working tool suited to the extraction and the generation of information and data.

3.7 Computer graphic systems

Computer graphic systems are defined as software or hardware that can be used for computer-based modelling and rendering of objects. For this, they implement a selection of the above-mentioned methods and algorithms at one or more system levels and provide an interface for accessing the implemented procedures and data structures. The computer graphic systems are often divided into low level systems (or basic rendering systems), high level systems encapsulating the functionality of low level systems and application systems using both or just the low level systems.

3.7.1 Rendering systems

A rendering system is a computer system for generating images based on computer graphic objects. It can be implemented in software, or as a combination of software and hardware for increased performance. The rendering system OpenGL (Shreiner, 1999) is the de-facto standard for 2D and 3D real-time rendering and defines a rendering pipeline as described in Chapter 3.1.1. It is a software interface to graphics hardware and is supported by most current operating systems either in a native hardware-accelerated implementation or as a fully software-based implementation. It provides an application programming interface (API) in the programming language C consisting of about 250 commands and several data types that can be integrated as a library into a real-time application. As a standard part of every OpenGL implementation, the OpenGL Utility Library (GLU) provides many additional modelling features such as parametric surfaces. OpenGL is a rendering system usable for immediate- and retained-mode rendering. The rendering system

Direct3D is very similar to OpenGL but restricted to the Microsoft Windows operating system.

Other rendering systems are dedicated to generation of high-realistic images using ray-tracing or radiosity techniques. However, they are unsuitable for real-time applications due to their high rendering costs and limited possibilities for integration with applications (Döllner, 2000).

3.7.2 High level computer graphic systems

OpenGL does not provide possibilities for defining an efficient scene representation in a hierarchical structure useful for culling, state sorting and geometric transformations. High level computer graphic systems are dedicated to development of computer graphics applications based on low level rendering systems such as OpenGL. They simplify the scene graph modelling and handling of object groups in a 3D world concerning their geometry and visual appearance. Most systems use an object-oriented approach.

OpenInventor is an object-oriented library coded in the programming language C++ based on OpenGL. It uses a DAG for scene representation and provides property nodes for attributes, geometry nodes for shapes, as well as additional grouping nodes and camera nodes. Although OpenInventor efficiently uses OpenGL commands in immediate-mode, it lacks flexibility and support for multi-pass rendering.

OpenSceneGraph (OSG) is a C++ library based on OpenGL still in development (Burns & Osfield, 2004). It is open source and can be redistributed and/or modified under the terms of the OpenSceneGraph Public License (OSGPL). There exist distributions for Windows, Mac OSX, Solaris, Irix, Linux and FreeBSD. A DAG is applied for the scene representation in a scene graph with group nodes arranging geodes as leaf nodes which have drawables holding the geometry. A second DAG is used as a StateGraph for efficient state sorting and inheritance. OSG supports LOD techniques, multi-texturing, multi-pass rendering, and different culling methods.

Other high level computer graphic systems are the IRIS Performer Toolkit using OpenGL and the freely available Java3D from Sun Microsystems. Java3D is written in the object-oriented programming language Java and is also based on OpenGL. Both rely on a DAG for scene modelling and offer state sorting and culling techniques as well. OpenInventor serves as the basis for the Virtual Reality Modelling Language VRML, often used for interactive rendering of small scenes on the web.

The Virtual Reality System (VRS) is a freely available high level computer graphics library written in C++ (Döllner & Hinrichs, 1997). It is a generic system integrating several low level rendering systems using an adapter subsystem. Real-time rendering systems like OpenGL or Direct3D are integrated as libraries whereas external rendering systems, e.g. Radiance for highly-realistic images, are integrated by exchanging files in the appropriate format. The system consists of two main parts: the rendering layer and the graphics layer. In the rendering layer, the commands of low level systems, e.g. OpenGL, are encapsulated without any significant performance loss. In the graphics layer, the geometry and the behaviours of objects are organized separately within two different DAGs, the geometry and the behaviour graph.

Chapter 4

Visual simulation of mountain glaciers

In the previous two chapters the most relevant aspects of simulating glacier fluctuations, computer graphics and geovisualization were summarised. In this chapter, several ideas from the fields are taken up and brought together to develop a combined method for simulating and rendering climate-induced mountain glacier fluctuations.

4.1 Evaluation of existing methods

To develop an approach for the simulation and rendering of glacier fluctuations, the methods pointed out in Chapter 2.4.2 and 3.1.2 will be evaluated, keeping the limitations and requirements of the simulation and computer graphics components in mind.

4.1.1 An appropriate glacier model

The most important properties and requirements of the different glacier models are compared in Table 4.1 together with some criteria concerning the rendering .

Dynamic 3D flow models make it possible to compute a detailed image of the 3D pattern of glacier flow and strain rates at a high temporal resolution. But since these models require detailed knowledge of glacier properties and bed geometry, their applicability is restricted to a very small set of glaciers. They need to be adjusted for each glacier individually, although the main physical principles of ice flow are included. That the time for the computation of one state is significantly higher than for simpler models remains a further drawback. Because the 1D glacier models (Oerlemans et al., 1998 and Jóhannesson, 1997) simplify the 3D geometry, they are able to simulate new states faster, but they also need information about the geometry of the bedrock. The third category of glacier models, the steady-state approach using a 1D flow-line, is the only one not requiring bedrock information, allowing simple application to glaciers all over the world. Only very few glacier parameters need to be known for a simple but robust description of the glacier bed

and long-term behaviour. For a long-term view of glacier fluctuations, the parameterised adjustment from one steady-state to another without any dynamics is sufficient, resulting in low computing costs. Thus, the method described in Haeberli & Hoelzle (1995) and

Table 4.1: Comparison of selected glacier models.

Feature		Dynamic 3D mesh	Dynamic 1D flow-line	Steady-state flow-line
Referenced examples		Gudmundsson (1999), Schneeberger et al. (2001)	Oerlemans et al. (1998), Jóhannesson (1997)	Haeberli & Hoelzle (1995), Haeberli et al. (1999a)
Properties	Dimensionality	3D	1D	1D
	Dynamics	yes	yes	no
	Time period	short - long	long	long
	Applicable for glaciers all over the world	no	no	partly
Require-ments	Glacier knowledge	high	medium	low
	Glacier surface	yes	yes	yes
	Geometry of bedrock	yes	yes	no
Rendering criteria	Estimated computing time	high	medium	low
	Geometry of simulation output	volume grid	discretised 3D flow-line	discretised 3D flow-line
	Appearance	no	no	no

Haeberli et al. (1999a) seems to be the most appropriate basis for a combined simulation and rendering of climate-induced glacier fluctuations.

The reduced dimensionality (i.e. 1D) of the simulation output implies that the glacier model should be extended to enable specification of a glacier surface based on a 3D flow-line.

4.1.2 A surface-based method for the geometric modelling

As mentioned in Chapter 3.1.2 volumetric and surface-based approaches exist for geometric modelling of an object. To describe the surface of the glacier, the volumetric and the CSG methods can be neglected for the evaluation since no completely 3D glacier model will be used in this study.

The four different surface modelling methods are compared with respect to the methodological and application-oriented aspects listed in Table 4.2. Considering the properties

of the different methods, one notices that each method has specific drawbacks. The major drawbacks of polygonal meshes are the continuity problem and the difficulty to specify a smooth parabolic shape, that contours well to the cross-section of a glacier surface. Using a parametric description, problems arise when trying to compute intersections or dealing with objects with complex topologies (control mesh must be a quadrilateral). Implicit surfaces are not well suited for the interpolation or approximation, which is required for the generation of a surface from a simulated flow-line. Another limitation of implicit surfaces is that they do not provide local support, i.e. a local modification affects the shape of the whole surface. This makes the modelling of local features, e.g. flat tongue during retreat, more complicated. Drawbacks of subdivision surfaces are the difficult specification and inefficient intersection tests.

Table 4.2: Comparison of surface modelling techniques (modified after Funkhouser, 1999)

Feature		Polygonal mesh	Parametric surface	Implicit surface	Subdivision surface
Methodological aspects	Guaranteed continuity	no	yes	yes	yes
	Interpolation and approximation	yes	yes	no	yes
	Intuitive specification	no	yes	no	no
	Local support	yes	yes	no	yes
	Modelling of parabolic cross-section	no	yes	yes	yes
	Affine invariant	yes	yes	yes	yes
	Arbitrary topology	yes	no	no	yes
	Efficient intersection	no	no	yes	no
Application-oriented aspects	Supported by low-level rendering system	yes	yes	no	no
	Efficient rendering	yes	yes	no	yes
	Toolkit available	yes	yes	no	no

Some application-oriented aspects of importance during implementation are provided in the lower part of Table 4.2. Only the polygonal mesh and the parametric surface are explicitly supported by low-level rendering systems, e.g. OpenGL. Additionally, software toolkits for modelling and modifying an object are only available for polygonal meshes and parametric surfaces. Currently, implicit surfaces are very hard to use due to the very poor support for their implementation and the high rendering costs.

Concluding, the presented facts show that the most suitable method for modelling a glacier surface from a previously simulated flow-line (and later implementation in VadreX) seems to be the parametric approach. It offers a simpler rendering method than implicit surfaces, and geometric operations as position or curvature control are easier to perform compared to polygonal meshes. In the next chapter, additional theory on parametric surfaces is outlined.

4.1.3 Details on NURBS surfaces

Non-Uniform Rational B-Spline (NURBS) curves and Bézier curves are the best known types of parametric curves. Since Bézier curves (and surfaces) consist of just one polynomial segment, they are often inadequate when dealing with complex shapes (e.g. Farin, 2002, Piegl & Tiller, 1997). To overcome the shortcomings of the Bézier form, B-Spline and NURBS curves can be used which are piecewise polynomial and rational, respectively. Since NURBS curves (or surfaces) contain non-rational B-Spline as well as rational and non-rational Bézier curves (or surfaces) as special cases (Piegl & Tiller, 1997), the basic theory is introduced for NURBS only.

A NURBS curve is composed of Bézier segments joined at points called knots. A *p*th-degree NURBS curve $\mathbf{C}(u)$ is defined by

$$\mathbf{C}(u) = \sum_{i=0}^{n} R_{i,p}(u) \cdot \mathbf{P}_i \qquad \textbf{(4-1)}$$

where $u \in [0, 1]$, the $\{\mathbf{P}_i\}$ are the control points, $\{R_{i,p}\}$ are the piecewise rational basis functions of *p*th-degree, and n +1 the number of control points. The basis functions are defined on the non-uniform knot vector U with its knot elements u_i. See Figure 4.1 left for an example of a cubic NURBS curve. To ensure geometric (G) and parametric (C) conti-

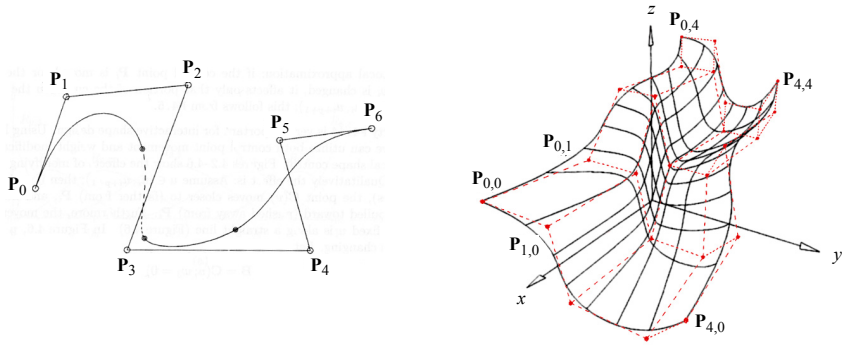


Figure 4.1: A cubic NURBS curve with 7 control points consisting of 4 segments (modified from Piegl & Tiller, 1997) and a biquadratic NURBS surface with its control net in red colour (right, modified from Piegl & Tiller, 1997).

nuity at the end of each segment, constraints are introduced. The curve $\mathbf{C}(u)$ is said to be C^k continuous, i.e. parametric continuous of degree k, at the knot u_i if $C_i^{(j)}(u_i) = C_{i+1}^{(j)}(u_i)$ for all $0 \le j \le k$. A NURBS curve is infinitely differentiable in the interior of knot intervals, and it is at least *p*-*M* times continuously differentiable at a knot of multiplicity *M*. A curve of G^1 continuity must have coplanar curve segments.

A NURBS surface $\mathbf{S}(u, v)$ of degree p in the u direction and of degree q in the v direction is a bivariate piecewise rational function. Its mathematical formulation is straightforward described as

$$\mathbf{S}(u, v) = \sum_{i=0}^{n} \sum_{j=0}^{m} R_{i,p}(u) \cdot R_{j,q}(v) \cdot \mathbf{P}_{i,j} \qquad \textbf{(4-2)}$$

where $n + 1$ ($m + 1$) are the number of control points in u (v) direction, $\{\mathbf{P}_{i,j}\}$ form a bidirectional control net arranged in a rectangular fashion and $\{R_{i,p}\}$ ($\{R_{j,q}\}$) are the rational basis functions in the u (v) direction. An example of a biquadratic NURBS surface is shown in Figure 4.1 right. Some important geometric characteristics of NURBS surfaces follow from the mathematical properties of the basis functions such as the strong convex hull property, local support and corner point interpolation (Piegl & Tiller, 1997). Since NURBS surfaces are piecewise rational, the geometric shape can be modified not only by control point movement but also by changing so-called weights specified for each control point. The local support property is important for shape modification purposes: if the control point $\mathbf{P}_{i,j}$ is moved, or the weight $w_{i,j}$ changed, it affects only the portion of the surface inside of $[u_i, u_{i+p+1}) \times [v_j, v_{j+q+1})$. The continuity of a NURBS surface is determined by the basis functions, hence the control points can be modified without altering the surface's continuity (as opposed to other parametric forms, e.g. Bézier). The NURBS surface $\mathbf{S}(u, v)$ is infinitely differentiable in the interior of a knot. At a knot $\mathbf{S}(u, v)$ is p-M times continuously differentiable in the u direction and q-M in the v direction where M is the multiplicity of the knot (Piegl & Tiller, 1997).

The 3D point $\mathbf{P}(x,y,z) = \mathbf{S}(u, v)$ on the NURBS surface $\mathbf{S}$ can be analytically computed from the parametric values u and v. The inverse case of finding the parametric values u and v from the given 3D coordinates (x,y,z), i.e. the point inversion problem, must be solved numerically (Piegl & Tiller, 1997).

Different methods were developed to fit parametric surfaces to scattered points (e.g. Hoppe et al., 1994, Piegl & Tiller, 2000). The points can be interpolated by solving analytically a system of linear equations, or they can be approximated by satisfying given constraints, e.g. maximum deviation bound or parametric continuity (Piegl & Tiller, 1997). Derivatives and the number of control points to apply can also be specified. Global fitting algorithms set up an optimisation problem and try to solve it respecting the data set as a whole. Local algorithms construct the surface segment-wise using only local data for each step. The shape and the parameterisation of the surface is affected not only by the control points and their weights w_i but also by the knot vector U and its elements u_i. Three common methods of choosing u_i are (a) equally spaced, i.e. uniform, (b) chord length and (c) centripetal. While the uniform approach is inappropriate given unevenly distributed points, the chordal parameterisation is generally adequate. The centripetal method can give better results when the data take very sharp turns (Piegl & Tiller, 1997).

The rendering of NURBS surfaces is done usually by tessellating the NURBS surface, i.e. converting the parametric description into a polygonal mesh (Watt, 2002). The rendering of this mesh is then performed straightforwardly. The tessellation consists of two steps: transforming the NURBS surface into several Bézier-patches, and polygonising these patches either by a uniform or adaptive subdivision method (e.g. Lane et al., 1980). An adaptive method divides the patches as long as the generated polygons have not achieved a predefined flatness criteria that can be specified either in object space or

screen space. Theoretically, the adaptive approach is preferable. But since the additional computing time for that method is often greater than the benefit, and cracks can appear between patches of different subdivision levels, its application is often questionable (Watt, 2002).

4.2 Modelling the glacier bed and surface

Typically, a mountain glacier can be thought as a combination of n_b smaller parts called branches where the number of branches n_b varies for each glacier. Each branch is characterised by one flow-line representing the central line reaching from the top to the terminus of the branch. In order to geometrically model a glacier surface, several steps should be distinguished as shown in Figure 4.2. Since the glacier model explained in Haeberli &

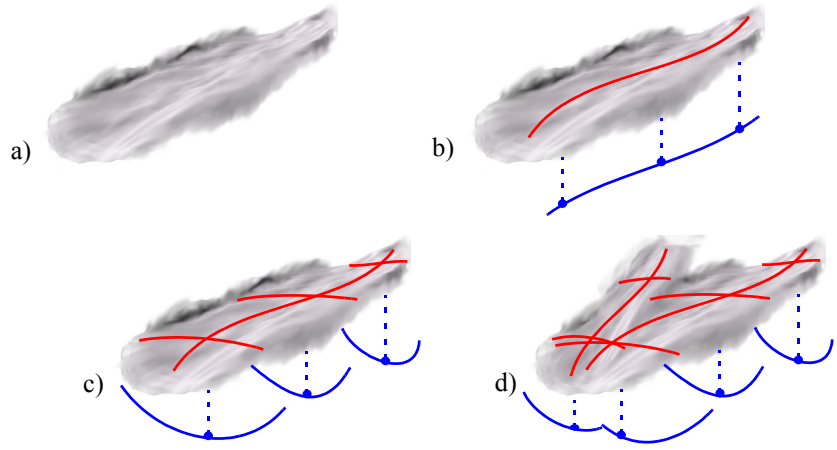


Figure 4.2: Steps for modelling of the glacier bed and surface: a) taking surface information from a DTM; b) defining a flow-line and computing a longitudinal profile; c) generating cross-sections; d) combining several branches.

Hoelzle (1995) and Haeberli et al. (1999a) is based on flow-lines, an approach has to be developed for their generation as described in the first subchapter. In Chapter 4.2.2 a modelling technique for defining a longitudinal profile along one branch as sketched in image b) of Figure 4.2 is explained. This profile is then applied as a skeleton for several cross-sections (see image c) in Figure 4.2), as pointed out in the third subchapter. Finally, a description of how to model the glacier bed and surface is given in Chapter 4.2.4.

4.2.1 The course of a flow-line

The flow-line is important not only for the physical glacier model but also for the final shape of the glacier, as it acts as the spine of a branch. It is required to represent the major line of mass transportation on the surface and should lie in the middle of the transverse direction. For the course of the flow-line, a computational approach similar to classical drainage pattern algorithms was tested. Starting at a manually set source point $\mathbf{P_s}$, the algorithm computes the course downwards on the real glacier surface derived from a DTM by applying the principle of lowest resistance and respecting the inertia of glacier flow. Due to the parabolic cross-section of a branch, the flow-lines tend to the sides of the glacier. In order to make the flow-line run in the centre of the valley, the surface information could be filtered spatially depending on the local surface roughness. But based on first hand experiences, one can draw the conclusion that such an algorithm produces flow-lines of low quality. It is not ensured that the computed flow-line lies in the centre of the cross-section - a strong requirement for the surface modelling later on. A more robust method is to manually define the flow-lines by setting n_p flow-points $\mathbf{P_m}$ along the central line on the branch surface as proposed by Haeberli et al. (1999a). To produce a continuous form, the flow-points can be fitted by a cubic NURBS curve leading to a smooth representation of the flow-line. The resulting parametric flow-line $\mathbf{C_f}$ is defined within the range $u \in [0,1]$.

In Figure 4.3, a manually defined set of 29 flow-points is approximated by the global least square approximation (LSA) method using a cubic NURBS curve, chordal parameterisation and 4, 8, 12, 16 or 20 control points.The lower the number of control points, the lower is the elastic energy of the curve, more details are blended out, and the accuracy is lowered. The corresponding RMS error of the approximation is 224 metres (4 control

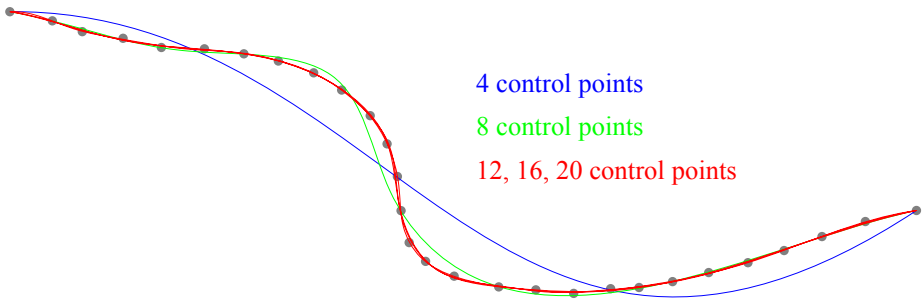


Figure 4.3: Global least square approximation (LSA) of one flow-line defined by 29 manually set flow-points (grey dots) with a cubic NURBS curve and 4, 8, 12, 16 or 20 control points.

points), 53 (8), 15 (12), 13 (16) and 10.9 (20). More than just one data set is considered in Figure 4.4 left. The RMS error of the approximation of 6 sets of flow-points with different geometry is plotted. Each flow-point set is fitted by global interpolation (black stem) and least square approximation (grey stem for 10 and white stem for 20 control points). The

data sets 1-6 in the bar plot are between 7000 and 8000 metres long and sorted by increasing number of specified flow-points (22-27). The interpolation algorithm results in small deviations, i.e. less than 5 metres[1]. The fitting accuracy of the global least square approxi-

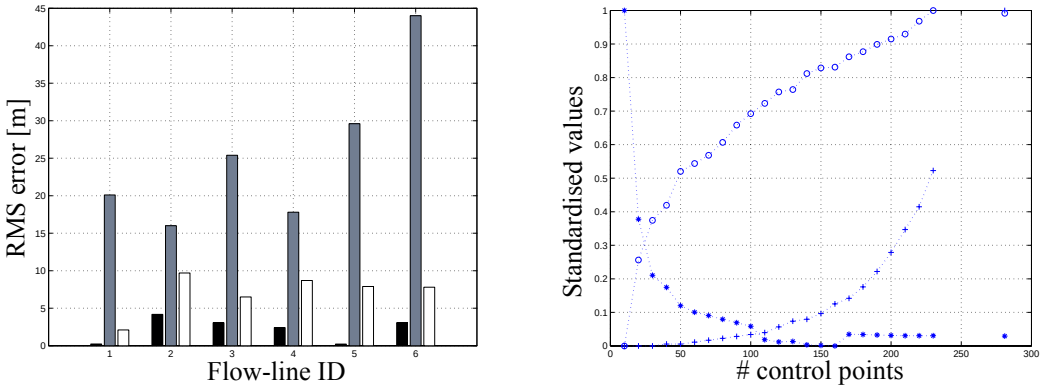


Figure 4.4: Quality of the global LSA of flow-points. Left: The RMS errors of interpolating (black) or approximating 6 flow-point sets of different geometry with 10 (dark-blue) and 20 (white) control points. Right: The impact of the number of control points applied in a global LSA of 281 flow-points. The resulting RMS error (*), length of the flow-line (o) and computing time (+) are plotted as standardised values.

mation (LSA) varies between 16 and 44 metres for 10 control points and between 2 and 9 metres for 20 control points. Local methods are not considered, as perturbation in a data item can be excluded, and the flow-points are treated equally describing one data set.

Results of an accuracy-performance test of the global LSA with a flow-line described by procedurally generated 281 flow-points are presented in Figure 4.4 right. It shows that the standardised RMS error and the total length of the flow-line converge exponentially towards the value achieved by global interpolation (single values at 281 control points). The computing time for the global LSA increases exponentially with the number of control points. Comparing the three graphs, an optimal number of control points is approximately half the number of the flow-points. Doing so, the variation diminishing property is combined with low computation time and sufficient accuracy.

Other error metrics could be applied for the approximation (see Chapter 4.1.3). But since the animation method presented later in Chapter 4.3 requires precise control of the number of control points, these approaches are not evaluated.

The choise of which fitting configuration should be applied is influenced by the specific data set and cannot be generalised. It is part of the whole specification process of a flow-line. Satisfying results can be expected with the global least square approximation, chordal parameterisation and $n_i = n_p / 2$ where n_p is the number of flow-points and n_i the number of control-points.

1. The RMS error is not zero as probably expected because of the limitations of the numerical solution of the shortest distance problem.

4.2.2 Longitudinal profile

The upper boundary of the longitudinal profile is defined by the parametric flow-line $\mathbf{C_f}$. The flow-line $\mathbf{C_f}$ runs from the source point $\mathbf{P_s}$ to the terminus point $\mathbf{P_t}$ as illustrated in Figure 4.5. The vertical dimension of the longitudinal profile is estimated using steady-

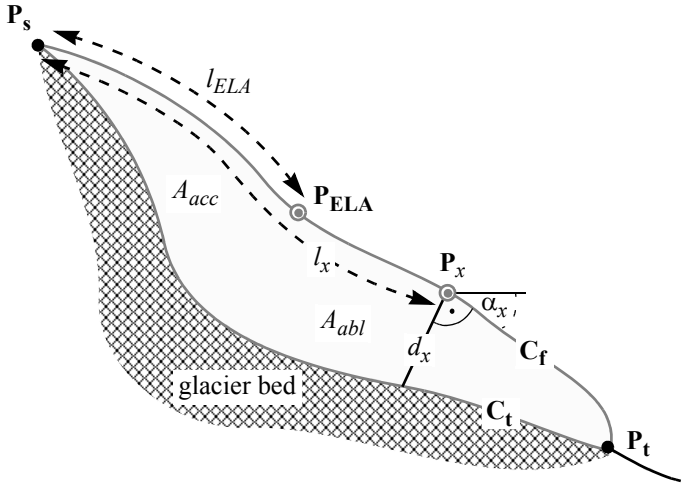


Figure 4.5: A longitudinal profile with the flow-line $\mathbf{C_f}$ and the terrain-line $\mathbf{C_t}$.

state conditions as proposed by Haeberli et al. (1999a). The glacier thickness along the flow-line $\mathbf{C_f}$ can be expressed as a combination of the ice cap factor k_x and the relief factor r_x (Haeberli et al., 1999a). The ice cap factor k_x describes a parabolic shape (in the flow direction) of a glacier when situated on a horizontal plane given perfectly-plastic ice (Paterson, 1994). The maximum thickness d_{max} of the profile would result at the equilibrium line. The ice cap factor k_x can be written as

$$\begin{aligned} k_x &= \sqrt{l_x / l_{ELA}} && \text{for } l_x \le l_{ELA} \\ k_x &= \sqrt{(L - l_x)/(L - l_{ELA})} && \text{for } l_x > l_{ELA} \end{aligned} \qquad \textbf{(4-3)}$$

where x is an index indicating the position on the flow-line relative to $\mathbf{P_s}$ (see Figure 4.5). The variable l_x represents the length from $\mathbf{P_s}$ to the surface point $\mathbf{P}_x$ at x, l_{ELA} is the distance from the point $\mathbf{P_s}$ to $\mathbf{P_{ELA}}$ and $L = \overline{\mathbf{C_f}}$ is the glacier length. It is important to notice that the glacier length L is measured along the sloped flow-line $\mathbf{C_f}$, generally not equal to the horizontal distance from $\mathbf{P_s}$ to $\mathbf{P_t}$ as derived from a topographic map.

Introducing a sloped plane, ice flows due to ice deformation, ice sliding at the bed, and bed deformation (Paterson, 1994). The two processes at the bed can be neglected by assuming long time scales (Greuell, 1992). Thus, ice flow due to ice deformation depends on the basal shear stress $\tau_x = f \cdot d_x \cdot \rho \cdot g \cdot \sin(\alpha_x)$ (see Paterson, 1994), where d_x is the profile thickness perpendicular to the branch surface and α_x the slope of the glacier surface at position x. The shape of the cross-section is described by the shape factor f which

can vary between 0.4 for narrow and 0.9 for wide valleys, respectively (Paterson, 1994). Since valley walls support part of the weight of the glacier, τ_x is less than its value for a wide channel. The size of f is typically 0.75 for mountain glaciers with confined flow (Haeberli & Hoelzle, 1995). The slope α_x is interpreted as an average over a constant distance $a_{const} = L/8$. The size of a_{const} is equal the zone where the shear stress can be transmitted. Values of α_x where $l_x < L/8$ or $l_x > L - L/8$ are set to α_x where $l_x = L/8$ or $l_x = L - L/8$, respectively (see Figure 4.17 on page 71). Regarding ice as a perfectly-plastic material, the basal shear stress τ reduces to the constant yield stress τ_0. Normally, the size of τ_0 for a specific glacier is unknown and has to be estimated. Thus, either a theoretically derived value of 50-150 kPa (Paterson, 1994) or, as preferred in this study, the empirical relation

$$\tau_0 = \left(0.005 + 1.598 \cdot \left(\frac{h_{max} - h_{min}}{1000}\right) - 0.435 \cdot \left(\frac{h_{max} - h_{min}}{1000}\right)^2\right) \cdot 10^5 \text{ [Pa]} \qquad \textbf{(4-4)}$$

can be used (Haeberli & Hoelzle, 1995). Resolving the shear stress equation for d_x and replacing d_x by r_x, a description of the relief influence r_x on the thickness at position x is given by

$$r_x = \tau_0 / (f \cdot \rho \cdot g \cdot \sin(\alpha_x)) \qquad \textbf{(4-5)}$$

This relation explains the observation that the thickness is higher where the slope is small due to increased shear stress (Haeberli et al., 1999a). The combination of the ice cap factor k_x and the relief factor r_x results in the glacier thickness d_x

$$d_x = k_x \cdot r_x \qquad \textbf{(4-6)}$$

and allows to compute a discrete profile along $\mathbf{C_f}$. A procedurally generated glacier surface (solid line in blue colour) and a simulated glacier bed (brown) are shown in the upper plot of Figure 4.6. The solid line in brown colour indicates a glacier bed generated with L = 5000 metres, l_{ELA} = 1500 metres, f = 0.75, a_{const} = L / 8 and $\Delta h = h_{max} - h_{min}$ = 2500 metres. Given a stepwise shaped glacier surface, the thickness is dominated by the relief influence, while the ice cap factor is less critical.

Until now, τ was assumed to be constant, equal to τ_0 although τ depends on the thickness d. In nature there is no a priori basal shear stress that controls the glacier flow. Considering that the glacier surface is also influenced by the energy and mass exchange with the atmosphere, it is rather the glacier geometry and mass turnover or, i.e., the combination of shear stress and flow velocity that is determined (Haeberli & Schweizer, 1988). A refined profile can be recomputed using a variable shear stress τ derived from the previously calculated version. Due to the incompressibility of ice, the shear stress is transmitted to zones within a distance $a_{var} = e \cdot d_x$ with $1 < e < 20$ ('longitudinal stress coupling'). For mountain glaciers, e = 8 seems to be a practical way (Haeberli & Schweizer, 1988). Therefore α_x is averaged over the range a_{var} for the refined profile.

The influence of e on the slope and thickness calculation depends on the local slope pattern of the surface and cannot be easily quantified. Local changes to the slope and thickness introduced by using a_{var} instead of a_{const} are depicted in Figure 4.6 where 4 dif-

ferent values of e (2, 8, 14, 20) are applied. High values of e lead to smaller thicknesses in flat areas when a_{var} becomes bigger than the flat area. For steep slopes, the thickness decreases independently of the size of e when using a_{var} instead of a_{const}. At both ends of

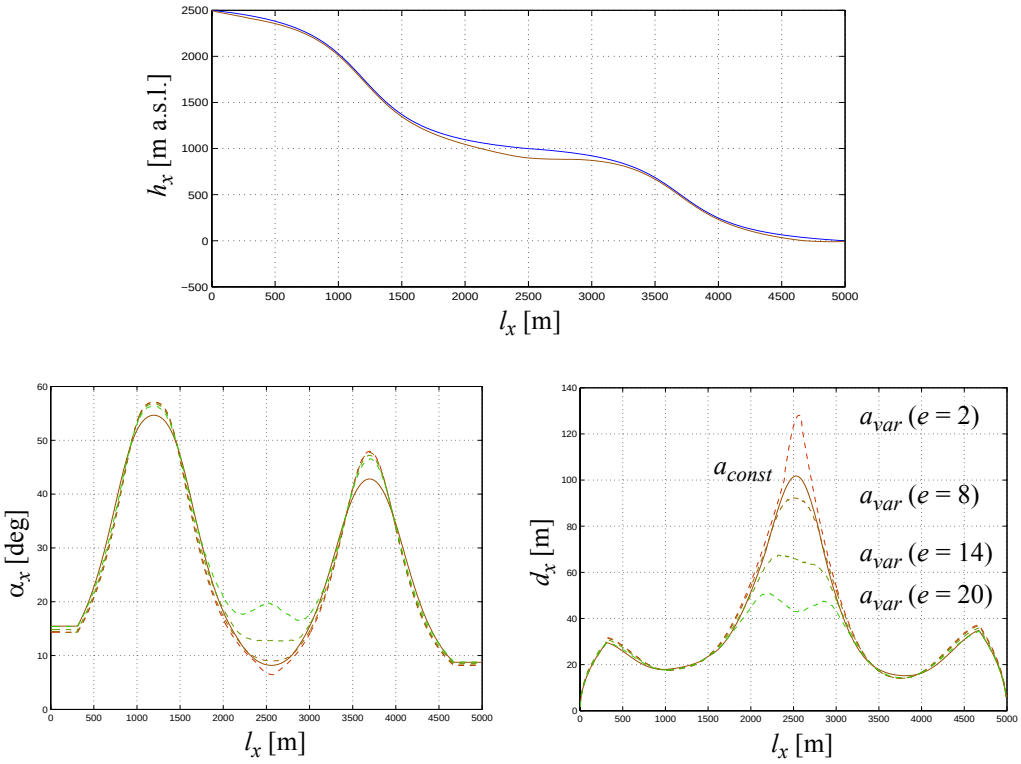


Figure 4.6: Simulation of a longitudinal profile. Top: Idealised glacier surface (blue) and a simulated glacier bed (brown) using $a_{const} = L / 8$ ($l_{ELA} = 1500$ metres, $f = 0.75$). Bottom: Sensitivity of the estimated slope (bottom left) and thickness (bottom right) in relation to a_{var} ($e = 2, 8, 14, 20$). For comparison the solid brown line indicates the use of $a_{const} = L / 8$.

the profile the shape is dominated by the ice cap factor as clearly recognizable in the plot bottom right. The term $\sin(\alpha_x)$ in the denominator of Equation 4-5 may require that the slope α_x has to be limited to α_{limit} to prevent an unrealistically high relief factor.

4.2.3 Branch surface

To geometrically model the branch surface, the longitudinal profile has to be extended with n_x cross-sections. In general, the cross-section has a parabolic shape due to its material properties and flow behaviour (Paterson, 1994), and does not vary significantly along the flowline (Haeberli & Schweizer, 1988). Since the glacier model is not based on a physically-based 3D computation, the width w_x has to be derived from general assump-

tions. A widely used method is to apply a constant relation between width and thickness, i.e. relying on a constant shape factor. In Figure 4.7, four different lines are plotted assuming typical shape factors and a parabolic cross-section. Empirical studies (Haeberli & Schweizer, 1988) show that the branch width w_x and thickness d_x are correlated by

$$w_x \approx 2500 \cdot \sqrt{0.0023 + 0.0008 \cdot d_x} - 120 \qquad \textbf{(4-7)}$$

for mountain glaciers. The empirical relation is plotted in Figure 4.7 as a red curve. In thin regions, the cross-section becomes wider than at thick locations comparing to a linear relation with a constant shape factor. In special cases, it may be required to limit w_x to pre-

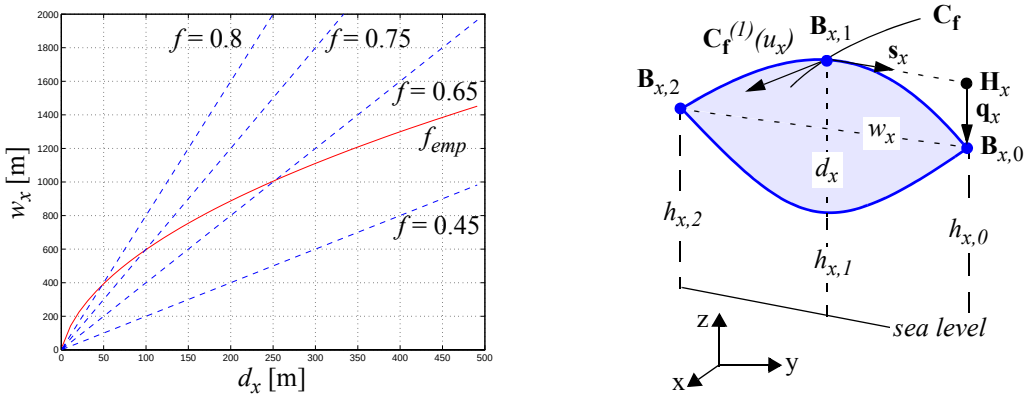


Figure 4.7: Left: The relation between thickness and width assuming different shape factors and a parabolic cross-section. Right: Geometric modelling of a cross-section of a branch at position x.

vent an unrealistic width, e.g., if the glacier flows from very steep to flat areas high water pressure at the glacier bed results in increased erosion and over-deepening of the bed.

For the geometric modelling of a cross-section, at least two side-points have to be specified, as illustrated in Figure 4.7 right. The side points $\mathbf{B}_{x,0}$ and $\mathbf{B}_{x,2}$ for each central point $\mathbf{B}_{x,1}$ on the flow-line $\mathbf{C_f}$ are provided by

$$\mathbf{B}_{x,(0,2)} = \mathbf{B}_{x,1} \pm \frac{w_x}{2} \cdot \mathbf{s}_x + \mathbf{q}_x \qquad \textbf{(4-8)}$$

with the side vector $\mathbf{s}_x$ and the projection vector $\mathbf{q}_x$ at position $x = [0..n_x)$. The parameter n_x identifies the number of cross-sections along the flow-line. The side vector $\mathbf{s}_x$ is calculated by projecting the unit vector of the first derivative $\mathbf{C_f}^{(1)}(u_x)$ to the xy-plane and rotating it by ninety degrees in the xy-plane. The projection vector $\mathbf{q}_x$ translates the side point H_x either to the z0-plane in the case of bed computation, or to the terrain.

If the glacier bed is already known, the branch surface can be fitted to the glacier bed using the terrain height $h_{x,(0,2)}$ for the two side points and $h_{x,1} + \widehat{d_x}$ as height for the central point where $\widehat{d_x}$ is the perpendicular component of d_x. Otherwise, the branch volume $\mathbf{V_b}$ is modelled by setting the z-value of $\mathbf{B}_{x,0}$ and $\mathbf{B}_{x,2}$ to 0 and that of $\mathbf{B}_{x,1}$ to $\widehat{d_x}$. Repeat-

ing the procedure for n_x cross-sections results in the point grid $\mathbf{G_b}$ consisting of $3 \cdot n_x$ points. The number of cross-sections n_x depends on the chosen sampling size l_s and L but a minimum of $n_x > n_p$ is preferred. Typically, l_s is about 200 metres what leads to $n_x = 30$ given $L = 6000$ metres.

The terminus of the branch, i.e. the branch mouth, is modelled geometrically by setting the height of the central grid point $\mathbf{B}_{x,1}$ to the predefined size of d_{mouth}.

A geometrical problem may arise during the modelling of the grid if the flow-line includes regions of very sharp turns (in comparison to the width). In such cases, the order of

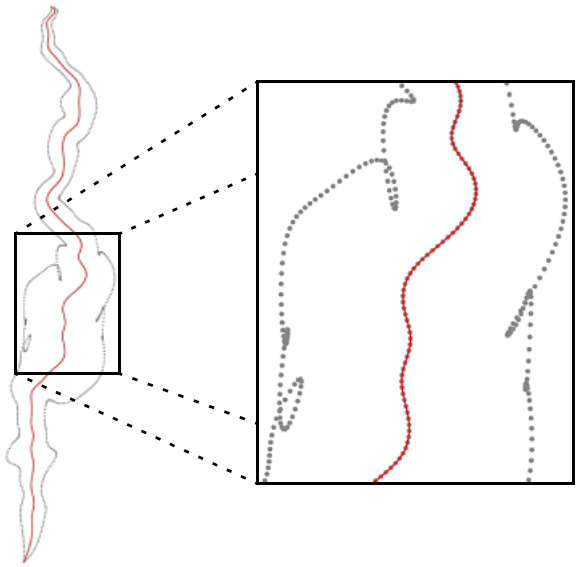


Figure 4.8: Geometrical inconsistencies in regions of too high curvature.

the side points $\mathbf{B}_{x,(0,2)}$ becomes inconsistent (see Figure 4.8). To prevent such artifacts, the number of control points can be decreased, resulting in reduced variation of the flow-line.

To get a continuous description, the grid $\mathbf{G_b}$ is fitted by the NURBS surface $\mathbf{S_b}$ with n_i control points in the longitudinal direction and n_j control points in the cross direction. The use of a NURBS surface allows continuous and efficient specification of the parabolic cross-section with $n_j = 3$ where n_i lies in the range $n_i \in [4, n_x - 2]$ for a cubic fitting depending on the number of cross-sections.

The histogram in Figure 4.9 shows the achieved RMS errors when interpolating or approximating 6 different branch grids, each consisting of $3 \cdot 50$ points. Similar to the flow-point fitting, the appropriate number of control-points appears to lie in the range of $n_x / 2$. Results from fitting two branch grids are shown in Figure 4.10. The depicted parametric branch surfaces contain $n_i = 20$ control points in the u (longitudinal) and $n_j = 3$ control points in the v (transverse) direction. For the rendering, the parametric surfaces are tessellated using an adaptive subdivision method.

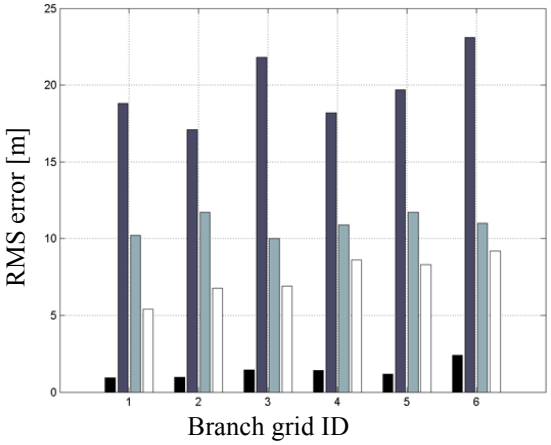


Figure 4.9: Achieved quality when fitting a branch grid based on 150 branch points: global interpolation (black) and global LSA with n_i = 10 (dark-blue), n_i = 20 (light-blue) and n_i = 30 (white).

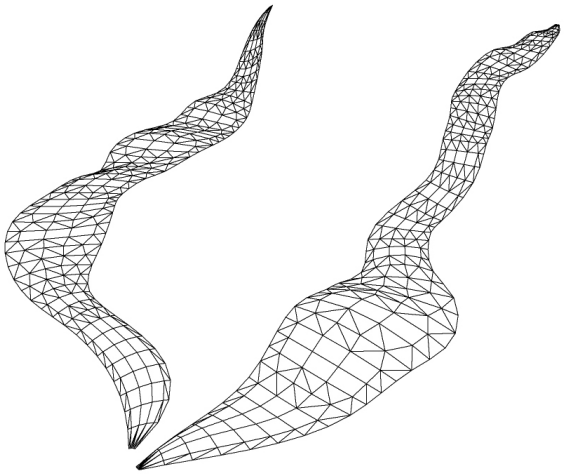


Figure 4.10: Two adaptive tessellated branch surfaces.

4.2.4 Glacier bed and surface

Although a glacier may consist of more than one branch, i.e. $n_b > 1$, each branch has to this point been treated separately. Each branch runs from the source point $\mathbf{P}_{\mathbf{s},o}$ to the terminus point $\mathbf{P}_{\mathbf{t},o}$ where n_b is the number of involved branches and $0 \le o < n_b$. The parametric flow-lines $\mathbf{C}_{\mathbf{f},o}$ and $\mathbf{C}_{\mathbf{f},o+1}$ join at the confluence points $\mathbf{P}_{\mathbf{c},o}$. Since the interactions between two or more branches are included implicitly in the shape of the glacier surface

derived from the DTM, the branch surfaces are combined geometrically without any additional physically-based interaction computations taken into account.

In order to compute the glacier bed, the volume $\mathbf{V_b}$ of each branch is calculated as described in Chapter 4.2.3, setting the height of $\mathbf{B}_{x,0}$ and $\mathbf{B}_{x,2}$ to 0 and that of $\mathbf{B}_{x,1}$ to $\widehat{d_x}$. The continuous volume $\mathbf{V_b}$ is then discretised by projecting each grid point of the DTM onto the volume. The projection problem can only be solved numerically by searching the value pair (u,v) iteratively for points fulfilling the following equation system

$$\begin{aligned} \upsilon(u,v) &= (\mathbf{S_b}(u,v) - \mathbf{H}(E,N)) \bullet \begin{pmatrix} 1 \\ 0 \\ 0 \end{pmatrix} = 0 \pm \phi \\ \varpi(u,v) &= (\mathbf{S_b}(u,v) - \mathbf{H}(E,N)) \bullet \begin{pmatrix} 0 \\ 1 \\ 0 \end{pmatrix} = 0 \pm \phi \end{aligned} \qquad \textbf{(4-9)}$$

where υ (ϖ) is the dot product of the projection vector and the xy-plane in the x (y) direction, $\mathbf{H}(E,N)$ is the 2D point given on the DTM grid with easting E, northing N and h = 0 m a.s.l., $\mathbf{S_b}(u,v)$ is the point on the branch surface at (u,v) and ϕ the predefined error tolerance (see Figure 4.11). The projected point is found when $\upsilon(u,v)$ and $\varpi(u,v)$ lies inside

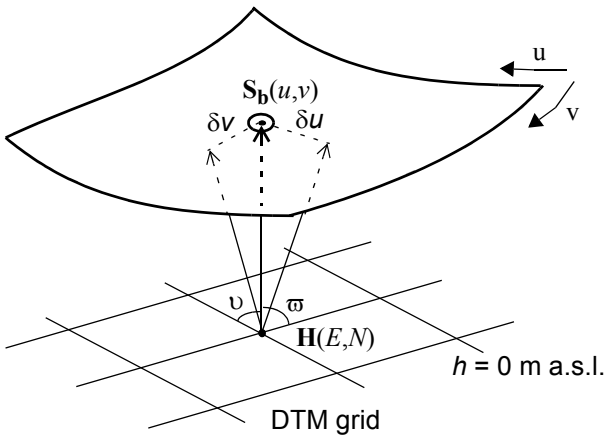


Figure 4.11: Rasterization of the branch volume by projecting the DTM point $\mathbf{H}(E,N)$ onto the branch surface $\mathbf{S_b}$.

the range $0 \pm \phi$, i.e. the projection vector is the normal vector of the xy-plane. Critical for a stable solution is the choice of ϕ ($\phi \sim 0.001$) and the shifting values δu and δv, which have to be adjusted analytically for each grid point by solving a linear equation system. The exact thickness at (E,N) is then equal to the z-value of $\mathbf{S_b}(u,v)$. Superimposing the discrete volume of each branch and taking just the maximum values in overlay regions re-

sults in the grid $\mathbf{V_g}$ describing the glacier volume. Subtracting the grid $\mathbf{V_g}$ from the DTM leads to the final DTM-G, i.e. the digital terrain model reduced by the simulated glacier volume.

4.3 Animation of mountain glacier fluctuations

In the last chapter, methods were developed for defining a steady-state glacier as the sum of n_b branch surfaces relying on physical principles and NURBS surfaces. For the dynamic evolution a smooth movement of the terminus, and an appropriate way of changing the surface thickness and width needs to be achieved.

Since dynamic systems are inappropriate as control models for the animation of long-term glacier fluctuations (see Chapter 4.1.1), and costly in computing time, a simplified system has to be developed in this study, combining physical approaches with procedural elements.

Several publications exist on modifying the shape of a NURBS surface, e.g. Piegl (1989) or Sánchez-Reyes (1997). However, NURBS surfaces have been viewed mainly as purely geometric primitives that require manual adjustment of multiple control points and associated weights in order to design shapes. Terzopolus & Qin (1994) pointed out drawbacks to the shape design (aside from the extraordinary flexibility) and introduced dynamic NURBS (D-NURBS), a dynamic generalisation of the NURBS model. D-NURBS are physics-based models that incorporate mass distributions, internal deformation energies and other physical quantities into the NURBS geometric substrate.

Considering the work of King & Parent (2001), a mountain glacier can be regarded as a set of deformable branch models. Furthermore, taking up the idea of dynamic NURBS (Terzopolus & Qin, 1994), the behaviour of these deformable branch models can be governed by physical laws describing the climate-glacier relationship. To achieve a continuous shape metamorphosis, the physically-based deformation can be improved temporally using geometry-based techniques.

4.3.1 Climate-induced branch deformation

The world's climate is continuously changing with apparently random fluctuations from year to year superimposed on long-term trends: such changes are reflected in variations of the glacier extent. Thus, the presented animation technique describes the long-term change of glacier shape due to climate change. Instabilities such as rapid advances, i.e. surges, are physically complex, not well understood exceptions and therefore not considered.

For the deformation of a glacier each branch is considered individually and its reaction is parameterised as proposed by Haeberli & Hoelzle (1995). Dynamic processes are disregarded although they exist in reality and are important to understand the glacier flow at small time scales.

Climate change is introduced numerically by changing the mass balance of the glacier. Given a step-wise temperature change ΔT_{air} assuming a certain climate scenario, the change Δb of the mass balance can be estimated by

$$\Delta b \approx \frac{\delta b}{\delta h} \cdot \frac{\delta ELA}{\delta T_{Air}} \cdot \Delta T_{Air} \qquad \textbf{(4-10)}$$

where $\delta ELA/\delta T$ describes the vertical increase of *ELA* per 1 °C warming and integrates the change of all climate parameters, i.e. humidity, radiance, accumulation and air temperature, as well as feedback effects (Kuhn, 1990). It is set to 170 m / °C within a range of ±50 m / °C (Kuhn, 1990). Changes in precipitation have in addition an influence on the mass turnover and thus on the mass balance gradient $\delta b/\delta h$. In this simplified mass balance model time is disregarded. Since the ice temperature of cold glaciers is also affected by a temperature change (together with the mass balance) the method is restricted to temperate glaciers where ice is at its melting point and the effect of climate change is restricted to a change in mass balance.

The new branch length $L' = L - \Delta l_t$ is calculated as a static function of the previous length L and its change

$$\Delta l_t \approx L \cdot \frac{\Delta b}{b_t} \qquad \textbf{(4-11)}$$

expressed as a function of the mass balance change Δb and the ablation at the terminus

$$b_t \approx (ELA - h_t) \cdot \frac{\delta b}{\delta h} \qquad \textbf{(4-12)}$$

where h_t is the height at the terminus (Paterson, 1994). Instead of changing T_{air} to disturb the steady-state conditions, the inverse case is also possible. Thereby a change in glacier length is given, e.g. from historical glacier measurements, from which the corresponding change of T_{air} can be derived. Additionally, if the real size of T_{air} is also measured, conclusions about the simulation quality, i.e. applied parameters and underlying physical principles, can be drawn.

The response time $t_A = T' - T$ that the glacier takes to adjust to Δb can be estimated by

$$t_A \approx d_{max} / b_t \qquad \textbf{(4-13)}$$

where d_{max} is the maximum branch thickness and T and T' the time of the old and new steady-state, respectively (Jóhannesson et al., 1989).

The mean specific balance $\bar{b}$ is calculated with

$$\bar{b} = \frac{\Delta b}{2 \cdot nt_A} \qquad \textbf{(4-14)}$$

depending on the change of the mass balance and the number of response times nt_A.The variable $\bar{b}_1$ refers to $\bar{b}$ where $nt_A = 1$.

Since the new glacier length *L'* is known, the new location $\mathbf{P_t'}$ of the terminus can be found by traversing the terrain-line $\mathbf{C_t}$. The constant terrain-line $\mathbf{C_t}$ is a cubic NURBS curve interpolating all manually set points $\mathbf{P_m}$ and limits the potentially maximum length for branch simulations. Please note that the length of $\mathbf{C_t}$ is at least as big as the one of $\mathbf{C_f}$, and usually longer to handle an advance over the initial glacier state (see Figure 4.5 on page 55).

The new thickness d_x' along the new flow-line $\mathbf{C_f'}$ from $\mathbf{P_s}$ to $\mathbf{P_t'}$ is calculated corresponding to section 2.1 with $d_x' = k_x' \cdot r_x'$. The new ice cap factor k_x' is adjusted to the new branch dimension whereas the new relief factor r_x' is based on the simulated surface described either by the old flow-line $\mathbf{C_f}$ or the surface-line $\mathbf{C_s}$ as in the case of advance. The difference between $\mathbf{C_s}$ and $\mathbf{C_t}$ is that $\mathbf{C_s}$ lies on the initial glacier surface derived from the DTM instead of the glacier bed, i.e. DTM-G. The further calculations for the branch surface and glacier surface are the same as presented in Chapter 4.2.3 and 4.2.4.

The resulting glacier surface $\mathbf{S_g}$' represents the new steady-state shape of the glacier adjusted to the new climate after the response time t_A. As a consequence there is an old and a new surface $\mathbf{S_g}$ and $\mathbf{S_g'}$ which can be thought of as two key states for a continuous morphing procedure.

4.3.2 Branch morphing

Due to the laminar flow behaviour of the glacier, surface-based morphing techniques are sufficient to describe the transformation from the old surface $\mathbf{S_b}$ at time T to the new branch surface $\mathbf{S_b'}$ at time T*'* (see Figure 4.12). In order to generate a robust but reliable morphing procedure with low computation costs, a geometrical approach should be applied.

The required shape metamorphosis can be divided into two subproblems: the correspondence problem and the interpolation problem. The correspondence problem refers to the mapping from one branch to the other. The interpolation problem consists of creating a sequence of intermediate branch states that visually represents the morphing from one branch to the other. For matching two parametric surfaces Surazhsky & Elber (2001) developed a method based on a discrete representation of the surfaces and a resemblance metric. Their approach deals with the general case of simple surfaces that are topologically equivalent to a disc. Zöckler et al. (2000) presented a surface-based method for establishing a geometric shape transition between two triangulated models by 1. defining manually corresponding regions, 2. parameterising these corresponding regions individually and 3. interpolating the parameterised mesh. But for the special case of morphing two branch surfaces on a terrain model, another method may be more appropriate and computationally faster.

Since the parametric form of the branch surfaces is given by $\mathbf{S_b}$ and $\mathbf{S_b}$' and they share a control net with the same topology (see Figure 4.12), the correspondence is given by the

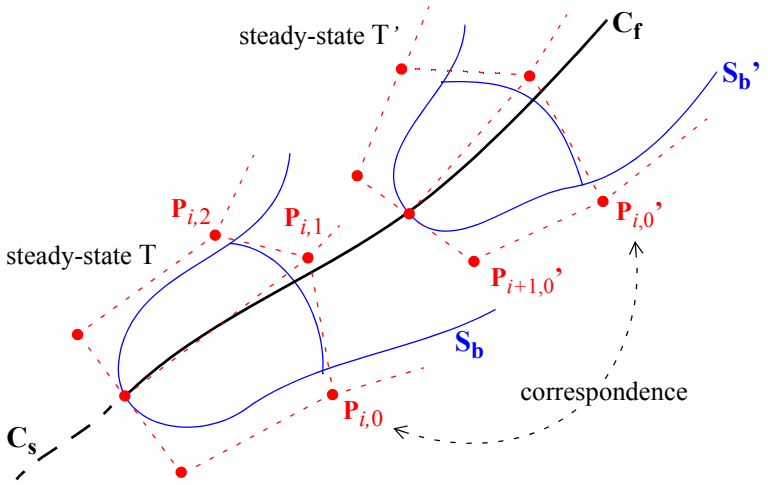


Figure 4.12: The branch surfaces $\mathbf{S_b}$ and $\mathbf{S_b}$' and their control meshes (red) of the new and old steady-state.

control points of the NURBS surfaces $\mathbf{S_b}$ and $\mathbf{S_b}$'. Thus, the control point $\mathbf{P}_{i,j}$ of $\mathbf{S_b}$ is modified in such a manner that it lies at the position of $\mathbf{P}_{i,j}$' after $t = t_A$. Both surfaces should be specified with the same number of control points, otherwise the correspondence would not be given. Consequently, a LSA with the same number of control points is applied for fitting $\mathbf{G_b}$' and $\mathbf{G_b}$.

For the morphing between two branch surfaces the control points $\mathbf{P}_{i,j}$ are changed independently whereby the central control point $\mathbf{Q}_0 = \mathbf{P}_{i,1}$ and the control points on the outside are modified in a different way.

The interpolation between $\mathbf{Q}_0 = \mathbf{P}_{i,1}$ and its corresponding point $\mathbf{Q}_{s\text{-}1} = \mathbf{P}_{i,1}$' is depicted in Figure 4.13 where s-2 is the number of intermediate positions and $0 \le \kappa < s$. Since

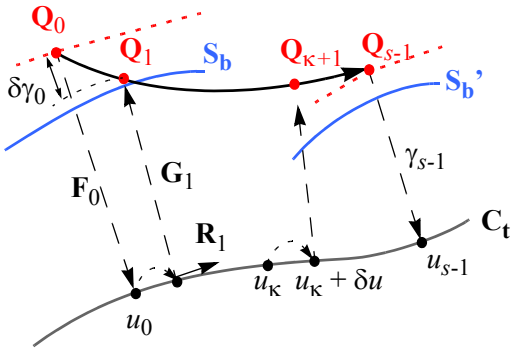


Figure 4.13: Terrain-followed morphing of a branch surface. Interpolation of a central control point $\mathbf{Q}$ between the two steady-states positions $\mathbf{Q}_0$ and $\mathbf{Q}_{s\text{-}1}$.

the transverse position of the central points is given by the terrain-line $\mathbf{C_t}$ the first intermediate position $\mathbf{Q}_1$ is defined by

$$\mathbf{Q}_1 = \mathbf{q_c}(D_c \cdot \mathbf{p_c}(\mathbf{Q}_0)) \tag{4-15}$$

where $\mathbf{p_c}$ is the projection of $\mathbf{Q}_0$ to $\mathbf{C_t}$, D_c is the increment operator increasing the parametric value u_0 by δu and $\mathbf{q_{c,1}}$ the inverse projection at $\mathbf{C_t}$ (u_1). Solving a linear system of 3 equation leads to the inverse projection vector $\mathbf{G}_\kappa$ at u_κ

$$\mathbf{G}_\kappa = \begin{pmatrix} G_x \\ G_y \\ G_z \end{pmatrix} = \begin{pmatrix} \dfrac{R_x}{R_y} \cdot G_y \\ \sqrt{\dfrac{R_x^2 + R_y^2 + R_z^2}{1 + \left(\dfrac{R_x}{R_y}\right)^2 + \left(\left(\dfrac{R_x^2}{R_y} + R_y\right) / R_z\right)^2}} \\ -G_y \cdot \left(\dfrac{R_x^2}{R_y} + R_y\right) / R_z \end{pmatrix} \tag{4-16}$$

expressed as a function of $\mathbf{R}_\kappa(u_\kappa) = \mathbf{C_t}(u_\kappa)^{(1)}$. The projection vector $\mathbf{F}_\kappa$ is estimated numerically by Newton iteration. Thus, it is not ensured that $\mathbf{G}(u_0) = \text{inv}(\mathbf{F}(u_0))$ leading to a probable small spatial shift of $\mathbf{P}_{i,1}$ at the beginning of the morphing. To prevent this popping the points $\mathbf{P}_{i,1}$ are projected to $\mathbf{C_t}$ with $\mathbf{F}_0$ and reprojected with $\mathbf{G}_0$ at the end of each branch deformation.

Because u_0 is known and $u_{\kappa+1} = u_\kappa + \delta u$ the interpolation of the further positions $\mathbf{Q}_{\kappa+1}$ is simplified to

$$\mathbf{Q}_{\kappa+1} = \mathbf{q}_{\mathbf{c},\,\kappa+1}(\mathbf{C_t}(u_{\kappa+1})) \tag{4-17}$$

The length of the projection vector $\mathbf{F}_\kappa$ is represented by γ_κ and is linearly varied by $\delta\gamma = \Delta\gamma / s$ where $\Delta\gamma = \gamma_{s-1} - \gamma_0$.

The number of interpolations $s = \Delta l_t / \delta l_t$ is determined by the selectable change of the terminus δl_t per interpolation step. The size of δl_t depends on the viewing distance and angle and lies reasonably in the range of 1 - 5 metres, resulting in a visually smooth transformation. Assuming that the curve length Δl_t from $\mathbf{C_t}$ (u_0) to $\mathbf{C_t}$ $(u_s\text{-}1)$ is linearly interpolated, i.e. $\delta l_t = \Delta l_t / s$, the parameterisation of δu by the arc length ensures a constant speed along $\mathbf{C_t}$. The arc length parameterisation of the terrain-line is performed by the forward differencing technique (Parent, 2002).

The interpolated positions of the two side control points $\mathbf{P}_{i,0}$ and $\mathbf{P}_{i,2}$ depend on the current position of $\mathbf{Q}$ and are calculated similarly to the method used for $\mathbf{B}_{x,0}$ and $\mathbf{B}_{x,2}$ in Chapter 4.2.3. However, the distance from the side control points to $\mathbf{Q}$ and their height difference to the current DTM-G position is interpolated linearly. Therefore, a terrain-following transition is also guaranteed for the side points.

4.3.3 Delayed and non-linear morphing

Some characteristics have not been taken into account so far. A glacier does not react immediately to a mass balance change. It takes reaction time $t_R \approx L_{abl}/c$ until the glacier begins to change its shape (Paterson, 1994). The parameter L_{abl} is the length of the ablation area and c represents the mean velocity of kinematic waves. These waves are regions of increased ice thickness and their velocity is 3-5 times larger than the mean surface velocity u_s (Paterson, 1994). Thus, the morphing speed along $\mathbf{C_t}$ is defined by applying the distance-time function $\lambda(t)$ which is expressed as

$$\begin{aligned} \lambda(t) &= \frac{t - t_R}{t_A - t_R} \cdot \Delta l \cdot m_d(t) \qquad \text{for } t_R < t \le t_A \\ \lambda(t) &= 0 \qquad\qquad\qquad\qquad\qquad\quad \text{for } t \le t_R \end{aligned} \tag{4-18}$$

with $\lambda(T') = \Delta l$, $\Delta l = \overline{\mathbf{C_t}(u_{s-1}) - \mathbf{C_t}(u_0)}$ and $0 \le t \le t_A$. To provide smooth movement with constant acceleration at T and T', a parabolic ease-in/ease-out function (Parent, 2002) is applied - described by the distance morphing factor $m_d(t)$. The parameter $m_d \in [0,1]$ computed with three different acceleration times, t_{acc} = 0.0, 0.1 and 0.3, is shown in Figure 4.14. The acceleration remains constant in the range [t_{acc}, 1 - t_{acc}] whereas it increases or decreases linearly within [0, t_{acc}) and (1 - t_{acc}, 1], respectively.

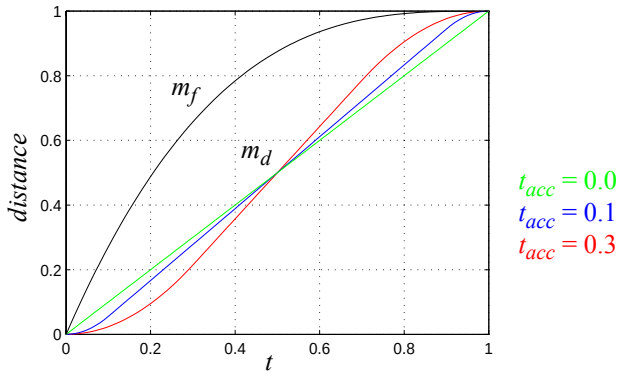


Figure 4.14: Morphing factors applied for the parabolic distance-time relation (m_d) and the exponential thickness-time relation (m_f) at the tongue. The distance-time relation is plotted for three different acceleration times, t_{acc} = 0.0, 0.1 & 0.3.

Another often observed phenomenon is the different dynamic behaviour of the glacier front (see images e) and f) of Figure 2.2). In case of retreat the front first begins to flatten before moving backward or, in the inverse case, front swelling is predominant during advance. This observation is simulated by the exponentially increasing morphing factor $m_f(t) \in [0,1]$. In Figure 4.14 an exponent of third order is applied to compute m_f. The length of $\mathbf{F_K}$ is then given by

$$\gamma_\kappa = \gamma_{\kappa-1} + m_f \cdot \Delta\gamma . \quad \textbf{(4-19)}$$

with $\Delta\gamma = |\mathbf{F}_{s-1}| - |\mathbf{F}_0|$ and $\delta\gamma_\kappa = |\mathbf{F}_\kappa| - |\mathbf{F}_{\kappa+1}|$ (see Figure 4.13). Linear behaviour is assumed in the accumulation area and the upper part of the ablation area. The morphing factor m_f is also applied for the transition of the control points $\mathbf{P}_{i,0}$ and $\mathbf{P}_{i,2}$.

The morphing method is repeated for each branch separately. After each interpolation step the branch surfaces are rendered in immediate mode.

The presented branch morphing approach for the in-betweening is a procedural animation technique based on mathematical principles taken from physical sciences. In contrast, the generation of key frames by deforming a branch uses a physically-based method limited to static processes.

4.4 Appearance

To specify the appearance of the glacier surface different techniques such as lighting, shading, texturing and transparency can be applied. The level of abstraction for the glacier appearance should be variable between low and medium degree, i.e. from abstract to geotypical representation, in order to adjust the amount of information. The use of geospecific textures would not be suitable since they represent a static view of a glacier at a specific point in time.

4.4.1 Abstract view of a branch surface

For a scientific study, e.g. analysis of future mass changes, the shape of the surface is the most relevant visual property. Therefore an abstract view of the branch surface has to be provided limiting the communicated information.

BRDF field measurements can be carried out to get information about the scattering of glaciers. Among very few other scientists Zhou & Li (2000) compared the BRDF of an ice pack floe covered by a pure and homogenous snow cover with the BRDF of a multiyear and more anisotropic ice floe, consisting of an ice layer and refrozen clusters. The presented results show that both ice floes have shown a nearly lambertian scattering behaviour with a slight forward scattering in the blue channel.

Thus, the scattering effects at the glacier surface are dominated by diffuse scattering. There are almost no specular highlights aside from small melt water areas or pure ice parts. A diffuse Gouraud shading with no specular lights is therefore sufficient to describe the scattering at the glacier surface. Surface brightness then depends only on the angle between the illumination direction and the surface normal. The normal vectors used are computed from the tessellated NURBS surface. An adaptive tessellated surface in object space is visualized in the centre of Figure 4.15 as wire-frame. A restrictive flatness criteria has been applied to obtain a smooth shading later on.

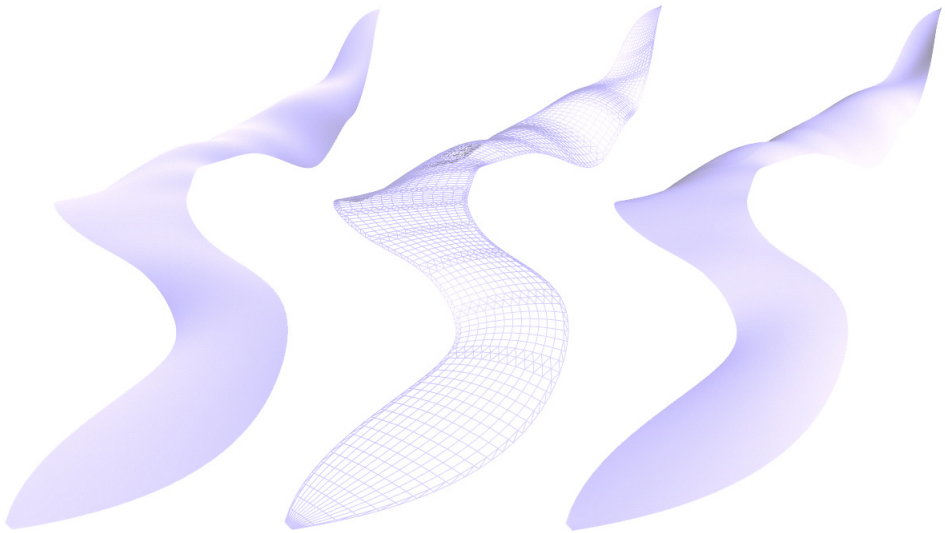


Figure 4.15: Abstract representation of a branch: adaptive tessellated surface rendered as wireframe (centre) and surface shaded from the left side (left) and the right side (right).

Additionally, shading of the surface improves determination of shape (e.g. Rodger & Browse (2000), Kleffner & Ramachandran, 1992). In Figure 4.15 left and right, two shaded branches are shown shaded from different directions to point out the influence of the shading direction on perceiving morphological details of the surface.

4.4.2 Geotypical representation of a branch surface

The visual impression of a real glacier surface is dominated by small ice structures on the surface, the scattering properties of ice, the typical moraines on the surface and crevasses in the area of extended flow as depicted in Figure 2.2. For a non-scientific audience the level of abstraction should be minimised, but well adapted to the level of abstraction of the physical model being used. A schematic representation using geotypical textures is therefore preferred over a highly realistic approach. The primary goal is to visualize scientific information and not the photo-realistic reconstruction of the glacier.

To get a geotypical description of a branch surface, a noise function is used to generate off-line a procedural 2D branch texture representing randomly distributed ice structures (see Figure 4.16 centre). For mapping the branch texture to the surface the texture is applied n_t times in longitudinal direction of the branch surface to ensure minimum distortion (see Figure 4.16 left). The size of n_t is calculated with $n_t = L / w_{max}$ where w_{max} is the maximum glacier width. To obtain a continuous appearance of the textured surface the mirror function is used as a corresponder function (Möller & Haines, 2002) and two-dimensional evaluators (OpenGL ARB, 1999) are used for the texture mapping. Thus, texture coordinates outside the range [0..1] are mirrored. Due to the multiple application of

the same texture, the texture size is minimized, e.g. 128 x 128 pixels, even for the representation of surface details. A geotypically textured and shaded branch surface is depicted in Figure 4.16 right.

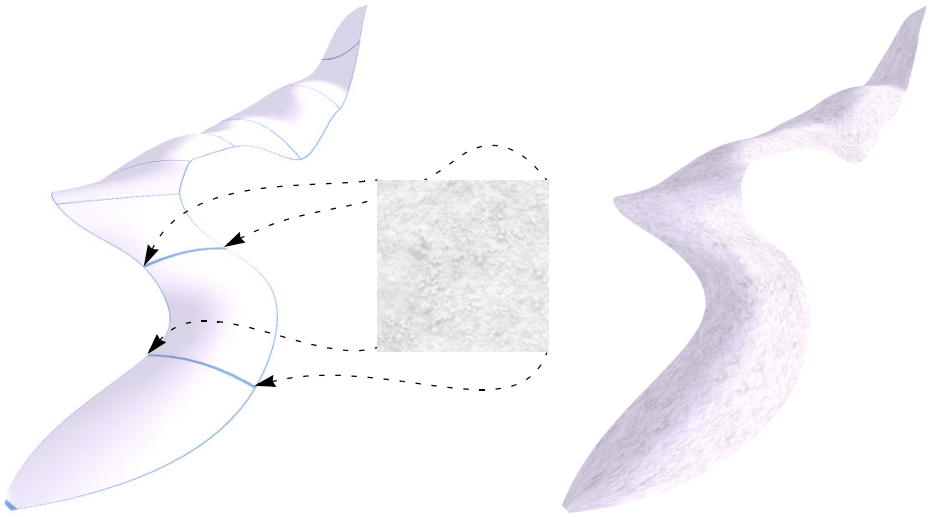


Figure 4.16: The geotypical representation includes a branch texture (centre) simulating ice structures. This branch texture is mirrored several times in longitudinal direction (left) resulting in a texture-mapped surface (right).

To simulate the slight translucency of ice, alpha blending is applied reducing slightly the alpha value of the texels to $c_A = 0.95$. Since each branch is rendered independently several transparent polygons may be added in regions where two or more branches overlap, leading to reduced visual quality. To make sure that each fragment is covered by not more than one transparent portion the stencil-buffer can be used (OpenGL ARB, 1999). The shading and texture colour are combined by the 'modulate' blending operation. A linear minification and magnification filter are applied when the pixel size of the texture does not match the size of a screen pixel. All branch surfaces share the same branch texture.

A possibility to describe the ice motion on the glacier surface is provided by using animated textures. In doing so, the texture coordinates are uniformly changed to simulate the effect of a moving surface. The amount of texture displacement depends on the mean surface velocity $u_s = 1.12 \cdot u_m$. The mean flow velocity u_m is estimated by

$$u_m \approx \frac{3 \cdot b_t \cdot L_{abl}}{8 \cdot d_f} \qquad \textbf{(4-20)}$$

where d_f is the mean thickness along $\mathbf{C_f}$ (see Figure 4.17). The factor 1.12 is the numerically computed ratio between u_s and u_m, assuming a parabolic cross-section with a shape factor of 0.75.

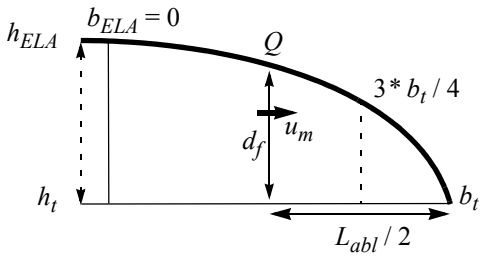


Figure 4.17: The mean flow velocity u_m.

The size of u_m is derived from the estimated mass balance b_t at the tongue and the computation of the discharge Q

$$Q = u_m \cdot d_f \approx \frac{3 \cdot b_t}{4} \cdot \frac{L_{abl}}{2} \qquad \textbf{(4-21)}$$

A highly realistic representation could be achieved by simulating crevasses in areas of high extended flow and adding moraines spanning in the lateral direction. However, the required physics would exceed by far the level of detail of the other used physical principles. The user would get a misleading impression of the applied level of resolution in space and time.

4.4.3 Appearance of the glacier environment

For an appropriate visual representation of the terrain the level of abstraction for terrain texturing has to be similar to the one used for the glacier. Otherwise, if a photo-realistic texture would be applied for terrain texturing an inconsistent image of the simulated glacier and its environment would result.

An abstract way of terrain texturing is provided through topographic textures. They can be generated by pixel-based shading of a DTM using an illumination model, e.g. hill shading. A detailed image of the terrain morphology is obtained as illustrated in the left image of Figure 4.18. Due to the pixel-precise shading the geometrical resolution of the visual representation of the terrain can be dramatically reduced, i.e. the frame rate can be increased, without any major visual loss (Döllner & Hinrichs, 2000b). However, a lower limit of the mesh resolution has to be ensured to preserve the characteristic shape of mountain silhouettes for orientation, especially in high mountain areas. A shaded terrain acts as a useful basis for adding any thematic geodata, e.g. geotextures or graphical symbols, supporting further scientific analysis.

In order to produce a more natural appearance with a medium level of abstraction, geotypical textures of various cover types, i.e. materials, can be brought in. Each material represents one type of surface in an idealised manner, as shown in the centre of Figure 4.18. The idea is now to apply the materials in areas of dedicated slope and elevation range and to blend them in transition zones. General qualitative rules, e.g., that steep slopes at high altitude are characterised by the existence of rocks but no snow, are valid

for many regions, but the detailed quantitative settings in numbers have to be adjusted for each specific terrain. The geotypical texture used for the terrain rendering in the right image of Figure 4.18 is generated by using the seven materials presented in Figure 4.18 centre. Additionally, the snow material is applied in two different ways to achieve a dif-

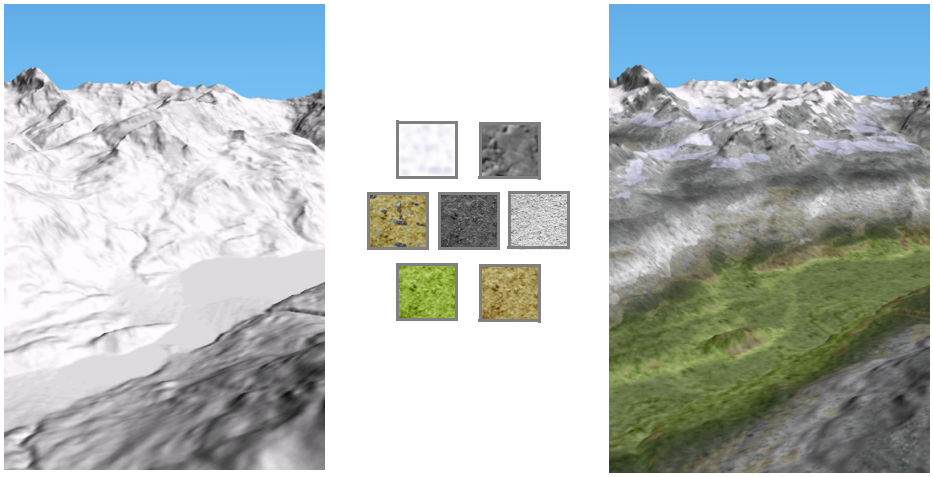


Figure 4.18: Topographic and geotypical geotextures for terrain texturing. Left: perspective view of a shaded terrain. Centre: seven geotypical materials representing snow, rock, river bed, dark gravel, light gravel, grassland and bare soil (arranged in left-right and top-down order). Right: terrain draped with a geotypical texture which is based on the seven materials. Data source: DTM DHM25 © Federal Office of Topography (BA024148)

ferent snow appearance in low altitudes. Computing such a geotypical texture with a sufficient resolution is a computational high-demanding task. Therefore, this is done off-line.

In extensive regions of homogeneous materials, e.g. snow covered areas, the presented material blending approach is lacking in visual detail. The left image of Figure 4.19 shows a high-mountain area with steep slopes represented by dark rock material and white snow-covered regions where it is nearly impossible to detect morphological structures. For an improved 3D perception the bump-mapping technique is applied to enhance the perception of morphology (see Figure 4.19 right).

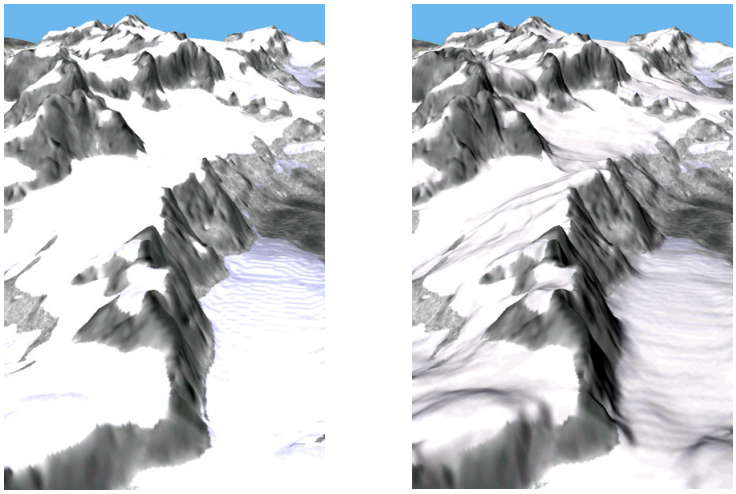


Figure 4.19: Adding morphological details in snow-covered regions using DTM information for bump-mapping. An original geotypical texture (left) and an improved version with a shading texture (right). Data source: DTM DHM25 © Federal Office of Topography (BA024148).

A dynamic change of the terrain appearance could be achieved through the proposed method by blending several precomputed static geotextures that represent the terrain appearance for a given point in time. However, since this study is focused on visualizing glacier fluctuations, the elevation and slope range of each material is assumed to be constant over time.

4.5 Reducing rendering costs

To increase the performance of the branch rendering, i.e. the frame rate, a general approach is to minimize the number of vertices to transform either by using a lower geometrical resolution of the object or by culling invisible parts.

4.5.1 Continuous change of tessellation accuracy

Since NURBS surfaces provide implicitly the possibility of representation in different resolutions, the tessellation accuracy is continuously changed as a function of the distance Λ_{view} to the viewer (see Figure 4.20). This ensures a smooth transition without visual discontinuity between two accuracy levels. The tessellation accuracy Ψ and the viewing distance Λ_{view} are related by

$$\begin{aligned} \Psi &= \Psi_{min} && \text{for } \Lambda_{view} \geq \Lambda_{far} \\ \Psi &= \Psi_{min} + \frac{\Lambda_{far} - \Lambda_{view}}{\Lambda_{far} - \Lambda_{near}} \cdot (\Psi_{max} - \Psi_{min}) && \text{for } \Lambda_{near} < \Lambda_{view} < \Lambda_{far} \\ \Psi &= \Psi_{max} && \text{for } \Lambda_{view} \leq \Lambda_{near} \end{aligned} \quad \textbf{(4-22)}$$

where the distances Λ_{near} and Λ_{far} depend linearly on the branch extension $\Lambda_{branch} = \overline{(\mathbf{P_s} - \mathbf{P_t})}/2$, i.e. $\Lambda_{near} = m_{near} \cdot \Lambda_{branch}$ and $\Lambda_{far} = m_{far} \cdot \Lambda_{branch}$. The sizes of m_{near} and m_{far} are set by the user, typically $m_{near} \sim 1.5$ and $m_{far} \sim 3.0$. If the viewer is closer to the branch centre $\mathbf{P_m}$ than the near distance Λ_{near} the maximum resolution Ψ_{max} is applied. If the viewing distance Λ_{view} is bigger than the far distance Λ_{far} the NURBS surface is tessellated with a minimum accuracy Ψ_{min}. Finally, the accuracy parameter Ψ is mapped to the value(s) of the applied tessellation method, e.g. the sampling rate in both parametric dimensions u and v.

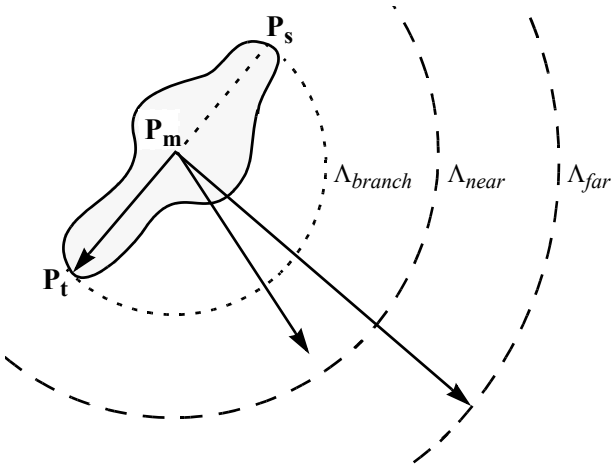


Figure 4.20: Continuous change of the accuracy of the branch tessellation.

4.5.2 Occlusion culling

For the culling of overlapping branch surfaces, i.e. occlusion culling, a surface-based approach would be an appropriate solution. This requires a surface/surface intersection (SSI) method for computing intersection curves, which are then used for trimming the branch surfaces. Barnhill et al. (1987) describe a robust SSI method that generates parametric intersection curve(s) of rectangular parametric patches by finding first one point on some intersection curve and following this intersection curve to produce more points in a sequential fashion. Their approach reduces the problem to finding an intersection of a line segment with a flat triangle. Although priorities of four different tolerances can be chosen to meet the criteria *quality* and *efficiency*, the additional culling time is expected to be in the same or higher order of magnitude as the reduced rendering costs.

A more specific culling approach was carried out that takes advantage of the known course of the flow-lines. It reduces the SSI problem to a line/line intersection problem. The idea is to set a trimming curve at each confluence point $\mathbf{P_c}$ in transverse direction for removing parts of branch surface(s) below $\mathbf{P_c}$. At each confluence point $\mathbf{P_c}$ the first-passing branch surface is not trimmed, whereas the next one is trimmed if it is not already trimmed at a former confluence point. The proposed approach works well if longer branches are thicker and wider at $\mathbf{P_c}$ than shorter ones. Although this is a plausible assumption it is not of general validity and leads to easily recognizable artefacts.

Therefore, a much simpler possibility to interactively disable the rendering of nonrelevant branches is introduced for adaptive performance control. This helps additionally to focus on specific branches during exploration.

Chapter 5

The visual system VadreX

The visual system VadreX implements the developed simulation and rendering method presented in Chapter 4. Furthermore, approaches for visual and numerical information extraction and validation, and also for navigation are integrated. As a prototype, VadreX will not serve as an operational system but as an environment for evaluating the implemented techniques and their suitability for studying and simulating mountain glacier fluctuations interactively.

5.1 Evaluation of a suitable base system

For practical reasons it is not intended to design and implement a new system from scratch. Thus, for evaluation of a visualization system serving as a basis for the own implementation, some system requirements are first identified (Chapter 5.1.1). Then, related visualization systems being already developed are described (Chapter 5.1.2). Considerations of a more practical kind are discussed and later used for comparison of existing systems (Chapter 5.1.3).

5.1.1 System requirements for VadreX

In order to identify criteria for simulating geoprocesses within a visual system, requirements desribed in Uhlenküken et al. (2000) are extended to cater for the simulation part.

Uhlenküken et al. (2000) outline some general requirements for future geoscientific visualization systems. They divide the requirements into four categories: high-dimensional data visualization, software-technical issues, user-specific issues and cartographic issues. Since VadreX is restricted to simulation and rendering of glacier fluctuations not all of the issues mentioned in Uhlenküken et al. (2000) are critical. Relevant points in the scope of high-dimensional data visualization are the support of

- multiple visualization techniques (shading, slicing, animation, flow visualization), and
- all existing dimensions (spatial, temporal and thematical dimensions).

For VadreX the consideration of all dimensions is crucial due to the 4D character of the visualized phenomenon. As consequence, the high dimensionality requires additionally fading-out of some information in order to symplifying analysis. The user-specific issues are to provide

- an appropriate level of abstraction depending on the target group, and
- a convenient navigation metaphor

to support orientation and efficient exploration in 3D space. Cartographic-oriented requirements include

- considering graphic and cartographic guidelines, and
- the support of geodetic coordinate systems

for handling data specified by different cartographic projections. Additionally, some criteria concerning the simulation part can be formulated such as

- representation of simulation validity and quality,
- tracking the course of the simulation,
- plausibility control of parameter settings (especially for non-experts), and
- efficient terrain rendering.

For experimental simulation it is essential to give the user visual or numerical information about the simulation run and its validity. Moreover, in order to reproduce a specific scenario in an objective manner the course of a simulation has to be tracked. The application of parameter settings outside the realm of the intended use of the simulation model (see Chapter 2.2) has to be made impossible, at least for non-experts. Since an efficient and suitable terrain rendering algorithm is required in order to free computing time for additional computations, and its development is beyond the scope of this study, it must be offered by the base system.

Aside from mandatory functionality of the system as listed above, software-related issues are also important. Even more, they are often of higher significance for practical implementation. For evaluation of existing visualization systems the following requirements were identified:

- open system architecture,
- source code or extensive and well-documented API available,
- low level rendering layer accessible,
- flexible and open scene graph architecture for animated scenes,
- providing meta-information,
- providing GIS functionality (e.g. overlaying or querying), and
- efficient data management.

To extend an existing geovisualization system with own software components an open system architecture is fundamental (Döllner, 2000). Furthermore, the source code or an extensive and well-documented API is required and the low-level rendering system, e.g. OpenGL, has to be accessible. Otherwise, a limited scope of rendering methods would be available within the application layer. The employment of animation methods should be well supported by the high-level computer graphic system. To improve the understanding of visualized simulation results it is valuable to provide meta-information, e.g. information about applied climate scenarios, especially for non-experts. GIS functionality such as overlaying thematic layers or querying information, have to be available during 3D exploration. Finally, two application-specific issues are derived from the need of modelling and rendering NURBS surfaces:

- software library for NURBS handling is available and linkable, and
- NURBS rendering has to be possible on low-level hardware.

5.1.2 Related visualization systems

Usually, geovisualization systems are built on low level rendering systems and use high level computer graphic libraries for scene modelling. Several geovisualization systems combine an interactive visualization system with a spatial database management system or GIS for spatial data handling. Pajarola & Widmayer (2001) propose a visualization system for large scale terrain data. To overcome the problem of exceeding available memory when rendering massive data in real-time, their system ViRGIS loosely couples two components, the visualization and the data handling component. The data handling system is based on a commercial object-oriented database management system and can serve queries for different geographic regions. The IRIS Performer Toolkit acts as the high level computer graphic system for the visualization component. Most of the visualization work is done by the scene manager which takes care of updating the actual scene dynamically according to the user's movement. The terrain is triangulated dynamically using a restricted quadtree when a given threshold of LOD is achieved.

An integrated global GIS and visualization system has been designed and implemented by Lindstrom et al. (1997). The use of hierarchical multi-resolution spatial data structures combined with multiple coordinate systems allows visualizing and manipulating huge terrain data sets covering the whole Earth at resolutions below one meter. The system is built on OpenGL and supports multiple windows with independent, stereoscopic views.

The Virtual Terrain Project (VTP) is an on-going open source project aiming to foster the easy recreation of the real world (Discoe, 2004). Aside from software for the preprocessing of geographic data, e.g. terrains, roads or vegetation, the runtime environment Enviro has been developed for the real-time rendering of terrain, plants and buildings. The interactive application Enviro is running on the Windows operating system[1] using the OSG library for scene graph modelling and wxWindows (wxWindows, 2004) for graphi-

1. Date of evaluation: autumn 2001. In the meantime a version running on Linux is also available, may be with reduced functionality.

cal design of the user interface. For terrain rendering the method of Roettger et al. (1998) is provided among others. The included Geospatial Data Abstraction Library (GDAL) supports the handling of many cartographic projection systems (Warmerdam, 2004).

Geovision is a landscape visualization system developed at the RSL (Remote Sensing Laboratories, Department of Geography, University of Zürich). Its focus lies primarily in the photo-realistic reconstruction of the real world and its interactive visualization. Geovision is built on the IRIS Performer Toolkit and runs on SGI workstations. Different methods for rendering terrain, buildings, plants and meteorological effects are implemented (Hirtz et al., 1999). For terrain rendering a tile-based approach with discrete LODs is used. Methods for realistic modelling of landscape elements have been improved within the context of several case studies. Geovision was also ported to Java (JGeovision) using the Java3D library as a high-level layer in order to be more platform independent.

LandExplorer (Döllner et al., 2003) is a geovisualization system that was not available at time of evaluation (Autumn 2001). For the sake of completeness, this interesting system is also briefly mentioned. It is built on VRS as a high-level layer and OpenGL as a low-level rendering layer. The concept of interactive 3D maps (Döllner & Hinrichs, 2000b) is implemented using an approximation tree for terrain representation and a multi-resolution texturing tree.

5.1.3 Comparison of systems

As a fundamental requirement the source code of the system or an API has to be available, which is true for three of the presented systems, currently: the two inhouse visualization systems of RSL (Geovision and JGeovision) and the run-time environment of the open-source project VTP. Consequently, these three systems will be evaluated in more detail. Nevertheless, some ideas from the other approaches may be integrated into the new system.

Own experiences with Java3D used in JGeovision have shown that speed and quality of the terrain rendering did not differ significantly when compared with a Performer/ OpenGL solution, e.g. Geovision. The major shortcoming of a Java3D based visualization system is the very limited extensibility since Java3D does not provide access to the lower rendering level (e.g. Döllner, 2000). Thus, several modelling and rendering functions of OpenGL or GLU, e.g. NURBS evaluators (OpenGL ARB, 1999), are not accessible from JGeovision. Additionally, no NURBS library written in Java was available whereas the source code of an object-oriented NURBS library (Lavoie, 2004) in C++ is freely available and considered to be well implemented and reliable.

Comparing the two systems Geovision and Enviro it is observed that the CLOD approach for terrain rendering used in Enviro should be preferred over the discrete and tile-based method in Geovision due to 1. visual quality, 2. no continuity problems at tile edges when rendering simulated objects and 3. rendering speed. Furthermore, many geodetic systems are supported by GDAL which is integrated in Enviro. This is important when dealing with glaciers all over the world. Another advantage of Enviro is the use of the powerful library wxWindows for the GUI design. Finally, Geovision requires high-end graphic hardware (e.g. SGI Onyx) whereas Enviro is running on low-end hardware, i.e.

PCs with Windows operating system and an ordinary consumer graphic card, resulting in low application costs.

Thus, the run-time environment Enviro (Discoe, 2004) is considered to be the most suitable visualization system for the implementation of the simulation and rendering methods presented in Chapter 4.

5.2 Preliminary design considerations

Before setting up the new system, some preliminary considerations are necessary to identify relations, structures and dependencies within the system. In a first step, major functional requirements of the system are captured by formulating the most relevant use cases (Chapter 5.2.1). Then, various kinds of static relationships within the system are derived from a data-oriented view of the system (Chapter 5.2.2).

5.2.1 Use cases

Use cases are a valuable tool to identify the functional requirements of a system (Fowler, 2004). Use cases work by describing the typical interactions between the user and the system itself, and may be a set of scenarios tied together by a common user goal. In Chapter 5, the terms *use case* and *scenario* are used according to the specification of UML 2[1] and do not have anything in common with the terms *case study* or *climate scenario* used in the other chapters.

Four different use cases are discussed to describe the major requirements of VadreX: *initialization* use case (a), *simulation* use case (b), *information* use case (c) and *validation* use case (d):

a) Considering the initialization use case two slightly different scenarios can be distinguished: computation and study of glacier bed (scenario 1.1) and study of initial glacier surface (scenario 1.2).

b) The second use case refers to the main action of experimental simulation and comprises three scenarios (2.1, 2.2 & 2.3) describing the simulation of glacier fluctuations and their rendering. Scenario 2.1 describes the involved interactions between user and system when changing T_{air}:

- user studies interactively static glacier and DTM-G
- user assumes change of T_{air} by selecting a specific climate scenario to apply for a future or past steady-state
- user may modify settings for simulation or geometric modelling
- system checks for plausibility of entered values
- system computes Δb and Δl_t

1. It is assumed that the reader is aware of the major concepts of the Unified Modelling Language (UML). Otherwise, Fowler (2004) is recommended as a brief guide.

- system computes new tongue position(s), longitudinal profile(s) and cross-sections using current settings for simulation (SimPar)
- system approximates new cross-sections by parametric branch surface(s) using current settings for geometric modelling (ModPar)
- if morphing is enabled (morphing extension):
    - system verifies that number of control points of new and old steady-state are equal
    - system updates current simulation time
    - system defines active branches which have to be animated at this time
    - system moves surfaces of active branches according to current simulation time and morphing behaviour that is defined by the current modelling settings (ModPar)
    - system animates branch texture according to flow velocity at glacier surface
    - user studies interactively animated glacier and DTM-G
- system is rendering the new branch surface(s) according to current rendering settings
- user studies interactively static glacier and DTM-G

The other two scenarios of use case 2 are scenario 2.2: modifying L according to a known retreat or advance, and scenario 2.3: user applies new settings and system recomputes the current state.

c) Use case 3 (‘information extraction’) consists of scenarios describing mechanisms for extracting numerical or graphical information about the simulation process.

d) Use case 4 (‘visual validation’) includes two scenarios where the user carries out a simulation and validates visually the result. The scenario 4.1 refers to adding reference data that is prepared outside the system. Adding dynamically generated validation data is described by scenario 4.2 and includes the following steps:

- user studies interactively static glacier and DTM-G
- user defines time/state of reference
- system generates a representation of the reference state
- system is rendering additionally the added reference representation
- user studies interactively static or animated glacier, DTM-G and reference representation

Use cases provide an external view of the system from a user’s perspective as they describe the way in which a user interacts with the system. In contrast, no correlations between the use cases and the internal structure of the system can be expected. Thus, a look at more static relationships and structures will follow in the next subchapter.

5.2.2 Static view of the main components

To identify static relationships and types of objects within the new system a data-oriented view as shown in Figure 5.1 is helpful. The essential part, i.e. the *glacier module*, is depicted as a gray box. It comprises the three modules glacier simulation, modelling and rendering. In order to experiment with different settings an interactively modifiable repository of simulation parameters, i.e. SimPar, specifies and controls the simulation. The same mechanism is available for the modelling and rendering part. For each simulation

run the currently applied parameter settings SimPar, ModPar and RenPar are tracked and stored in an experiment repository.

Only a DTM, flow-points and glacier parameters are needed as input to the simulation procedure. The DTM and the flow-points can be extracted from a GIS database whereas the glacier parameters, i.e. $\delta b/\delta h$, *ELA* and $\mathbf{P_s}$ of the initial glacier state, are derived from a glacier database, e.g. that of the WGMS and saved in the configuration file.

To interactively start a simulation run, the user selects a specific climate scenario, or measured climate data in case of a reconstruction task, both provided by a climate database (DB). A plausibility layer is introduced to ensure that climate scenario variables are within a reasonable range. Functionality for the modelling and rendering of the validation data is covered by the *validation module* relying partially on the run-time environment Enviro which will be discussed later.

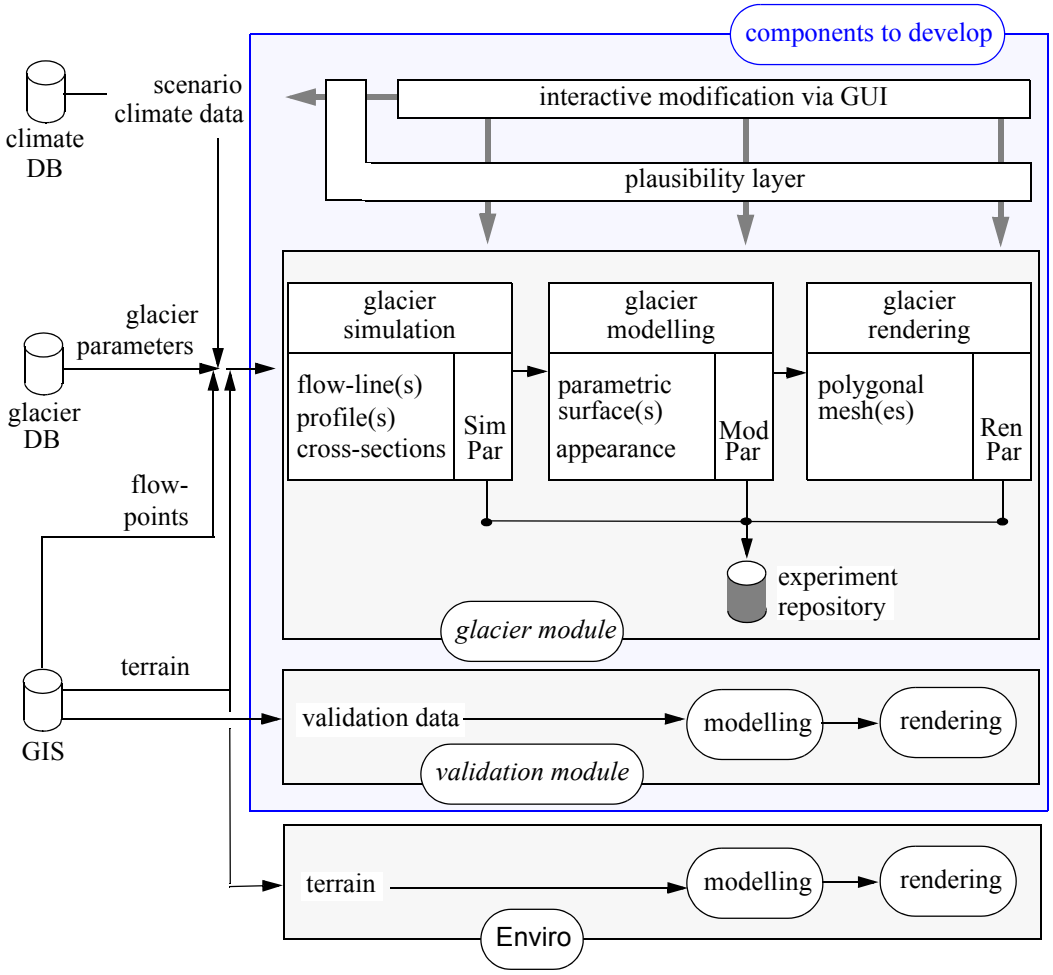


Figure 5.1: Data-oriented view of the main components.

5.3 System architecture

The goal of VadreX is to provide functionality for interactive rendering and simulation of glacier fluctuations. This restrictive formulation of its usage is the rationale behind many of the design considerations.

5.3.1 Integration approach on single platform

The capabilities of modern computer systems are nearing the requirements for performing both simulation and rendering tasks on the same machine. Nevertheless, some performance increase may be attained by running these tasks on distributed platforms (Belleman et al., 2000). However, as components may execute on different, possibly heterogeneous computing platforms substantial design and implementation issues result (e.g. Belleman et al., 2000). Additionally, the computational effort of the developed simulation model is considerably low compared to complex dynamic physically-based models. Based on these practical considerations VadreX is designed pragmatically as a single-platform system.

An important design issue is the way the two components simulation and visualization are linked. There are two main strategies to combine these components: coupling or integration. Coupling means that the simulation process and visualization process coexist independently and that data is exchanged either by transferring files or by calling a method of the other system (e.g. Wittmann, 2000). Using the integration approach results in an integration of one component into the other relying on common data and method base. Major disadvantages of the coupling approach are data redundancy, loss of semantic information and low efficiency due to format transformations. Hence, integration has been shown as the suitable strategy to avoid inconsistency due to data and method redundancy and to prevent information loss due to non-appropriate data exchange formats and heterogeneous data models (Bernard & Krüger, 2000). To prevent the development of a monolithic application by using the integration approach, the concepts of object-orientation[1] and interoperability[2] are widely accepted (Bernard & Krüger, 2000). The work of Becker et al. (1999) is an example of an interoperable and object-oriented prototype integrating simulation, visualization and GIS components. Döllner & Hinrichs (2000a) apply mediator classes, e.g. iterator classes, to couple geoobjects with their visual representation.

In VadreX the GIS component is linked to the visualization component by file-based coupling as already provided by Enviro. Therefore, geographic data such as flow-points or DTM's are loaded by appropriate software components.

1. An overview of object-oriented modelling is given by, e.g., Rumbaugh et al. (1991).
2. Interoperability means the ability of two or more systems or components to exchange information and to use the information that has been exchanged (IEEE, 1990).

5.3.2 Layered structure of VadreX

The architecture of the single-platform system VadreX is characterised by its layered structure, bringing the object-oriented components of each layer together as depicted in Figure 5.2. The two bottom layers provide functionality for low-level rendering and high-level scene modelling. The third layer consists of a terrain component and a glacier com-

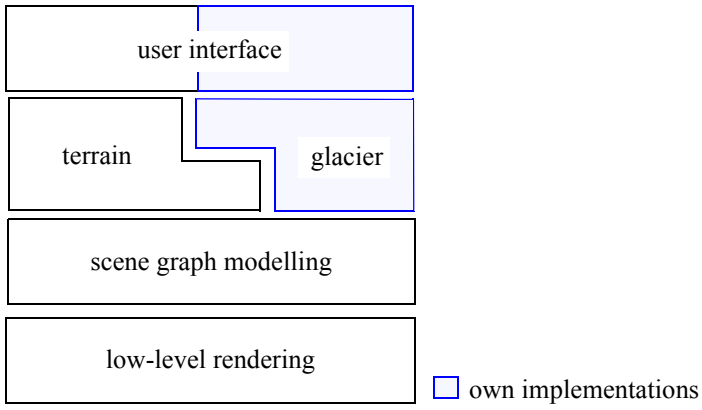


Figure 5.2: Layered structure of VadreX (black: existing components, blue: components to be developed).

ponent which are both accessible from a common user interface. The parts to be developed are highlighted in blue whereas the other parts in black are already implemented and distributed as open-source.

The package diagram of VadreX presented in Figure 5.3 shows the involved packages and their dependencies in UML notation. VadreX uses the run-time environment of VTP, i.e. Enviro, for the terrain visualization and extends the functionality of Enviro by adding the package VadrexSDK which contains libraries for glacier-specific simulation, modelling and visualization.

The object-oriented NURBS++ package (Lavoie, 2004) provides methods for the geometric modelling and the OpenSceneGraph (OSG) package (Burns & Osfield, 2004) for visual glacier modelling. The parametric branch surfaces are rendered using the evaluators of the OpenGL GLU library (OpenGL ARB, 1999) which are accessible from VadrexSDK due to the extendable and open character of OSG. The glacier simulation procedures and the validation methods are integrated in the package VadrexSDK.

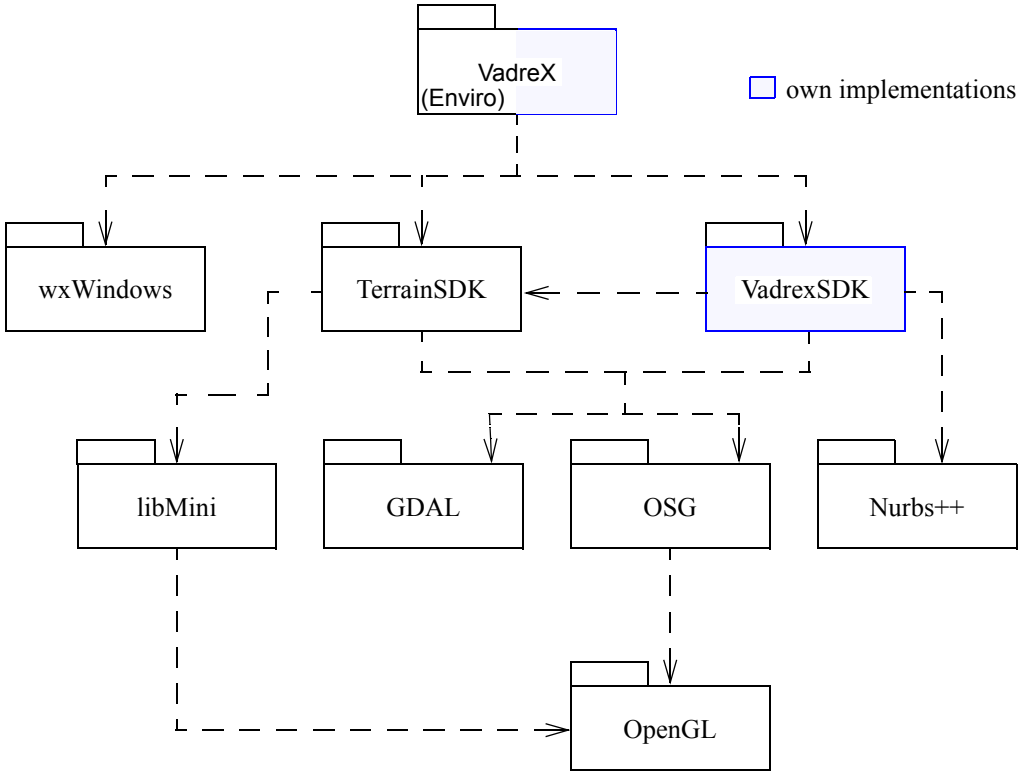


Figure 5.3: Package diagram for VadreX with tabbed folders representing the packages and dashed arrows showing dependencies (UML 2 notation).

Classes of the GDAL package (Warmerdam, 2004) are integrated for reading the manually generated and georeferenced flow-points in a file-based manner. Information about the terrain is accessed in VadrexSDK by referencing terrain classes of the package TerrainSDK, which is providing the core techniques of Enviro for terrain handling and rendering. The package libMini consists of dedicated methods for CLOD terrain rendering described in Roettger et al. (1998). More information about Enviro and TerrainSDK is provided by Discoe (2004).

The user interface for interactive simulation steering and for other user-oriented functionality is based on C++ classes from the wxWindows package (wxWindows, 2004).

5.3.3 Classes of VadrexSDK

The package VadrexSDK contains the two C++ libraries NurbsOSG and SimGlacier whereby SimGlacier depends on the NurbsOSG library.

NurbsOSG encapsulates the behaviour of the scene graph nodes of OSG and provides classes for the usage of NURBS surfaces in combination with the scene graph of OSG (see Figure 5.4). To strictly separate the drawable and its representation in the scene

graph, i.e. the geometry node or geode, the abstract class *osg::Drawable* is subclassed by the class NurbsSDrawable. This derived class implements the pure virtual operation *drawImmediately()* which renders the drawable directly ignoring OpenGL display lists. An object of type NurbsSDrawable is instantiated by an object of type NurbsSGeom and assigned to the scene node osg::Geode. NurbsSGeom extends the functionality of the class NurbsSurface defined in the package Nurbs++ by providing material properties, a GLU interface for NURBS tessellation and the virtual drawing operation doRender() called by NurbsSDrawable::drawImmediately(). The main features

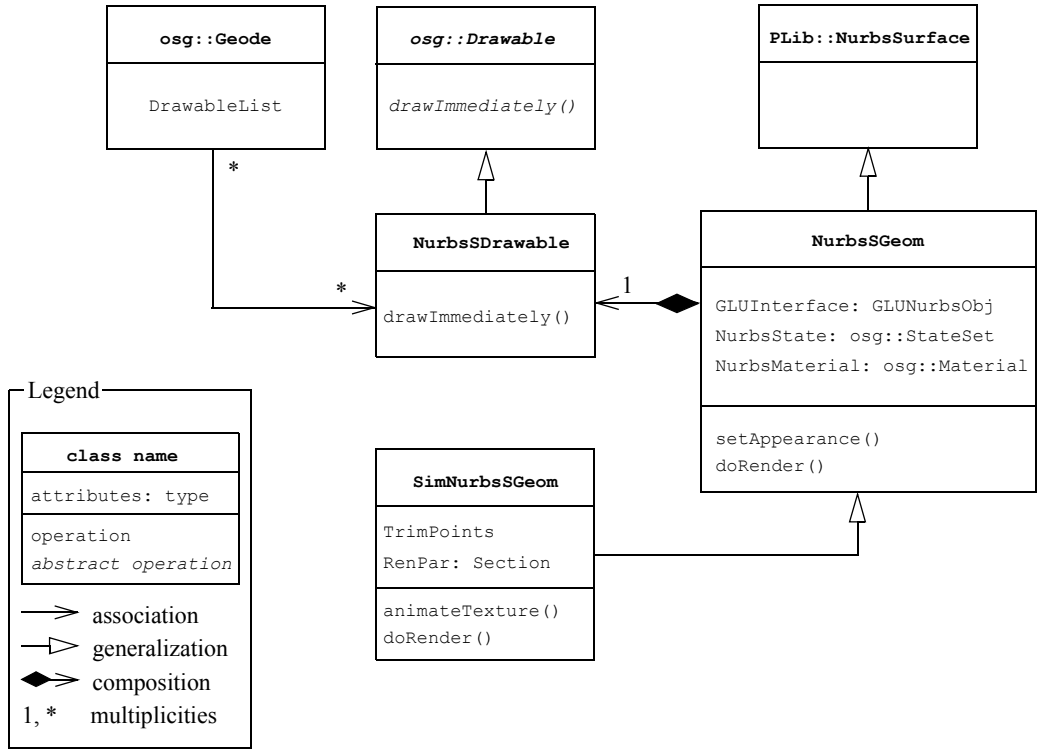


Figure 5.4: Class diagram of NurbsOSG (UML 2 notation).

of the derived class SimNurbsSGeom are the support of trimming, animated textures and a parameter repository of type Section that controls the rendering of the NURBS surface. Section[1] is a kind of a container class that stores any value pairs and supplies operations for their ASCII-based reading/writing. In SimNurbsSGeom the virtual operation doRender() is overridden and called dynamically by NurbsSDrawable::drawImmediately() using the Run-Time-Type-Information (RTTI) mechanism.

The library SimGlacier is made of several classes suitable for modelling and simulation of a dynamic glacier (see Figure 5.5). The main class Glacier encapsulates the com-

1. The class Section has been developed at RSL.

mon attributes and behaviour of several branches and has a group node for adding instances of `osg::Geode`, i.e. the nodes for the branches, as leaf nodes. An instance of `Glacier` contains one or more objects of type `DynBranch` which is a class for a dynamic branch able to adapt its shape to the current climate. Since all branches have the same appearance the branch texture is an attribute of the `Glacier` class. This provides the useful ability to define the branch texture once in memory, and have multiple objects of type `DynBranch` referring to it. behaviour properties of a glacier are defined in several different modules of type `Section` and divided in the object `GlacierPar` which is of type `SectionGroup`. These modules are generated dynamically by the `Glacier` constructor based on the values set in the configuration file.

A bi-directional association links `Glacier` with the glacier environment class `GlacierEnviro`. The class `GlacierEnviro` acts as an interface between the terrain-related software components and the glacier and implements the method for computing the glacier bed. It also includes a scenario object of type `ScenarioDB`. The purpose of `ScenarioDB` is to describe the future development of the air temperature, either by applying an IPCC scenario or a user-defined one.

The class `Branch` is derived from `SimNurbsSGeom` and describes properties and behaviour of a static branch. It provides operations for computing the longitudinal profile, branch surface and volume as well. The attribute `BranchProperties` stores relevant glaciological parameters of the branch. The operations of `Branch` are controlled by variables derived from the attribute `BranchPar`. `BranchPar` is an object of type `SectionGroup`, which holds any number of `Section` objects, one for each module. Considering a `Branch` object, the three modules 'FlowLineModul', 'ProfileModul' and 'BranchModul' are loaded. The 'ProfileModul' contains, for example, value-pairs steering the computation of the longitudinal profile, e.g. the size of the variable *e* (see Figure 4.6 on page 57).

As a specialized class `DynBranch` inherits the properties of `Branch` and implements the deformation and morphing capabilities described in Chapter 4.3.1 and 4.3.2. For this, two `Branch` instances are included referring to the old and new steady-state branch. After performing a deformation the objects are mirrored by pointer assignment, i.e. the old branch becomes the new one and vice versa. Additionally, an `ArcLengthParameterizer` object takes care of a constant morphing speed along the terrain line in case of a linear distance-time function. Of more practical importance is the attribute `TerminusPosition` which defines the terminus position of the `NewBranch` in parametric coordinates of the `OldBranch`. A `SectionGroup` object controls the deformation and morphing operations in a similar way as the object `BranchPar` in the `Branch` class.

Several subclasses of the abstract class *`vtp::Engine`* can be associated with `Glacier` in order to add and manage flexibly additional behaviour occuring in every frame. The subclass `GlacierEngine` provides methods for the asynchronous morphing of the branches and for dynamic trimming of the branch surfaces. Specific techniques for visually validating the glacier simulation are encapsulated in the `ValidationEngine` whereas run-time information of the simulation is visualized by the `MessageEngine`. The `ExperimentTracer` is also a subclass of *`vtp::Engine`* and gathers all relevant information about the current experiment and writes it to the experiment repository.

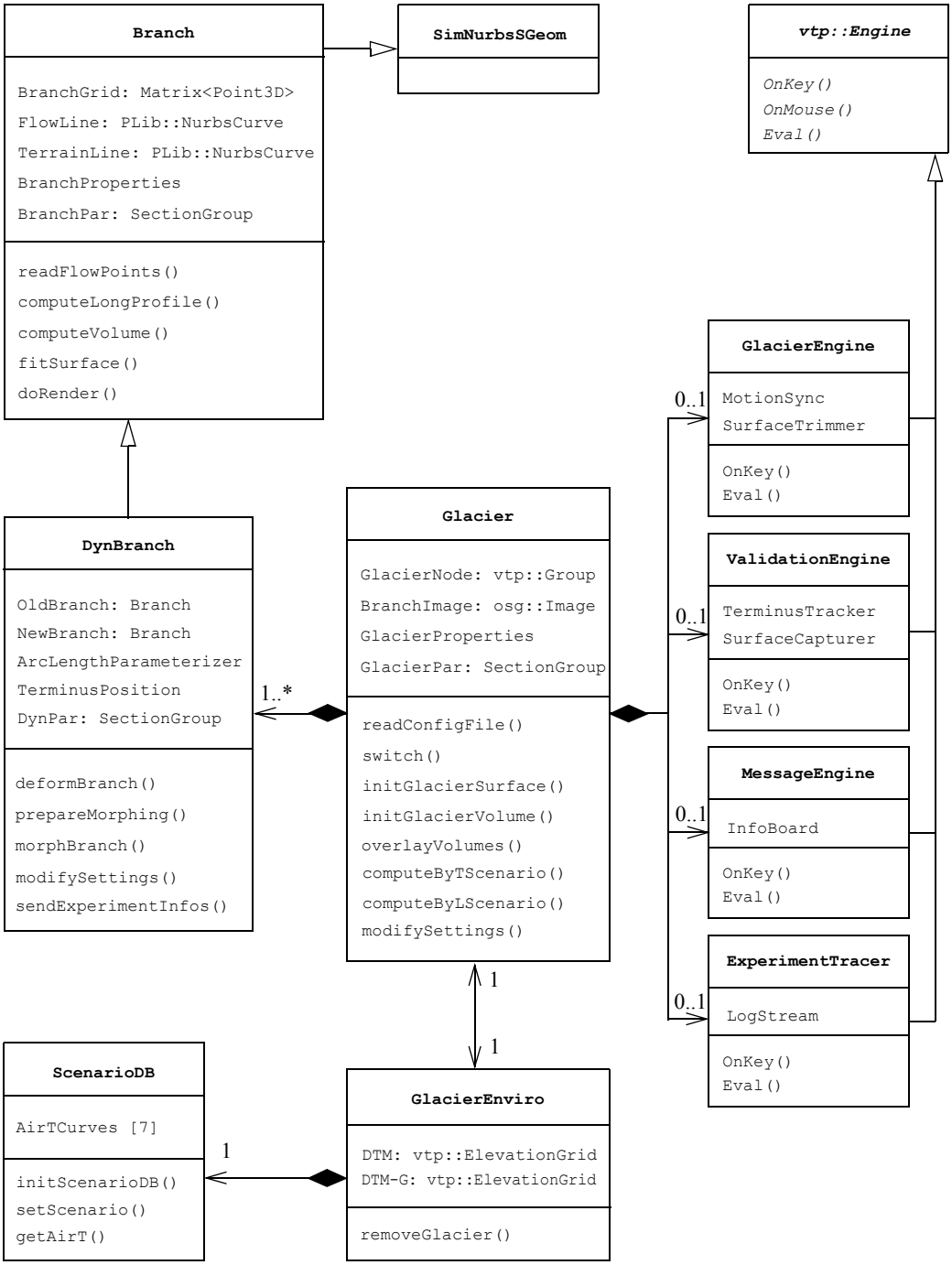


Figure 5.5: Class diagram of the library SimGlacier (UML 2 notation).

5.3.4 Bed computation from the system's perspective

To illustrate the behaviour and interplay of system objects some use cases are shown in more process-oriented kind of UML diagrams. The sequence diagram in Figure 5.6 represents the scenario 1.1 as presented in Chapter 5.2.1 on page 81, where the user is providing a few input parameters and the system is computing the glacier bed. The instance of `GlacierEngine` gets a reference of the original DTM and requests for computing the volume of the glacier by calling `Glacier::initGlacierVolume()`. Then, the branch volume is computed sequentially by reading the flow-points, computing the longitudinal

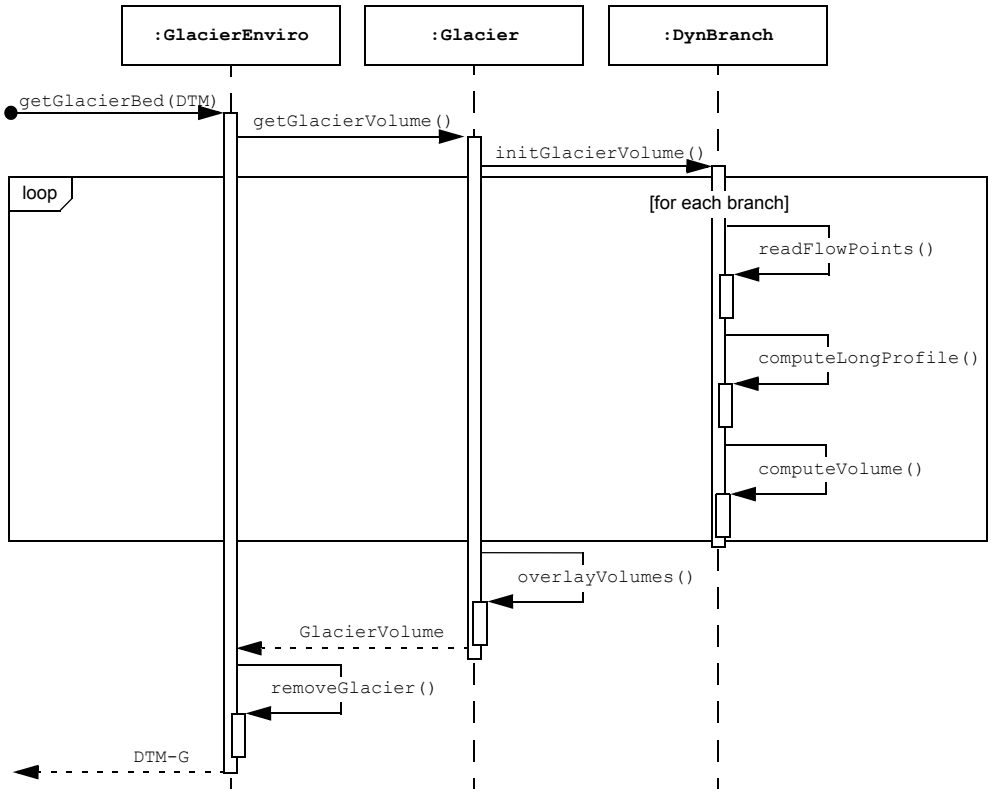


Figure 5.6: Sequence diagram of computing the glacier bed (UML 2 notation).

profile and rasterizing the surface-fitted branch grid for each `DynBranch` object included in the branch container. The `Glacier` object takes care of combining the rasterized branch volumes and the `GlacierEnviro` object removes the final glacier volume from the original DTM.

The format of the input parameters varies. The flow-points are stored as ArcInfo (ESRI Inc.) shapefiles and read by the appropriate classes of GDAL. The DTM is available in the proprietary format of VTP (Discoe, 2004) or that of RSL. Numerous terrain formats can be converted into that of VTP using the additional VTP software for geospatial data

processing (Discoe, 2004). The configuration file containing the module settings and initial glacier parameters is stored in the format supported by the class `Section`.

5.3.5 Experimental simulation from the system's perspective

The second exemplary diagram points out the complexity of processing a user request for rendering animated glacier fluctuations. The activity diagram in Figure 5.7 comprises all three scenarios of use case 2 including the morphing extension (see Chapter 5.2.1 on page 81) and illustrates the involved procedural logic.

The work flow is initiated by calling `ScenarioDialog::Apply()`, which is triggered by an external event such as pressing a dialog button. The class `ScenarioDialog` is a subclass of `wxDialog` which is provided by the wxWindows package. As a standard class for a window, `wxDialog` is used to allow the user to make a choice or answer a question (wxWindows, 2004). The following plausibility check of the input variables, e.g. whether correspondence of the control points is given when morphing is enabled, is done by the object of type `ScenarioDialog`.

After the quality of the input has been ensured the two `Branch` objects included in the `DynBranch` instances are switched, i.e. `NewBranch` becomes `OldBranch`. Then, if no simulation or modelling parameters were modified and T = T' a message informs the user about the ambiguous input. Otherwise, if T' not equal T, each branch is deformed separately by calling `DynBranch::deformBranch()` where the change in length has to be previously computed in the case of scenario 2.1. If some inconsistencies occur during branch deformation, i.e. L' > length of $\mathbf{C_t}$, the previous computations are revoked. Normally, the dialog is updated and the branches are animated according the subsidiary morphing diagram in Figure 5.7.

The morphing loop takes place as long as the condition $| t - T | < t_A$ is true whereby the branch is just animated when the difference between two morphing states is significant, i.e. the tongue has changed its position by more than δl_t. Additionally, it is not necessary to change the branch surface every frame as the asynchronous animation mechanism is controlled by the `GlacierEngine` instance. Before rendering the new shape of the branch surface the `ExperimentTracer` is made to compile and send the latest experiment informations to the experiment repository. If morphing is enabled the simulation time t is increased by $\delta t = (fps \cdot f_T)^{-1}$. Otherwise, t is set to T', forcing to leave the loop already at the first time step. However, at the end of the work flow the new steady-state surface is rendered and a new experiment can be started.

A possibility not yet discussed is given by scenario 2.3 where only simulation or modelling settings are modified. In this case, the surface has to be recomputed using the updated settings without any additional considerations such as morphing.

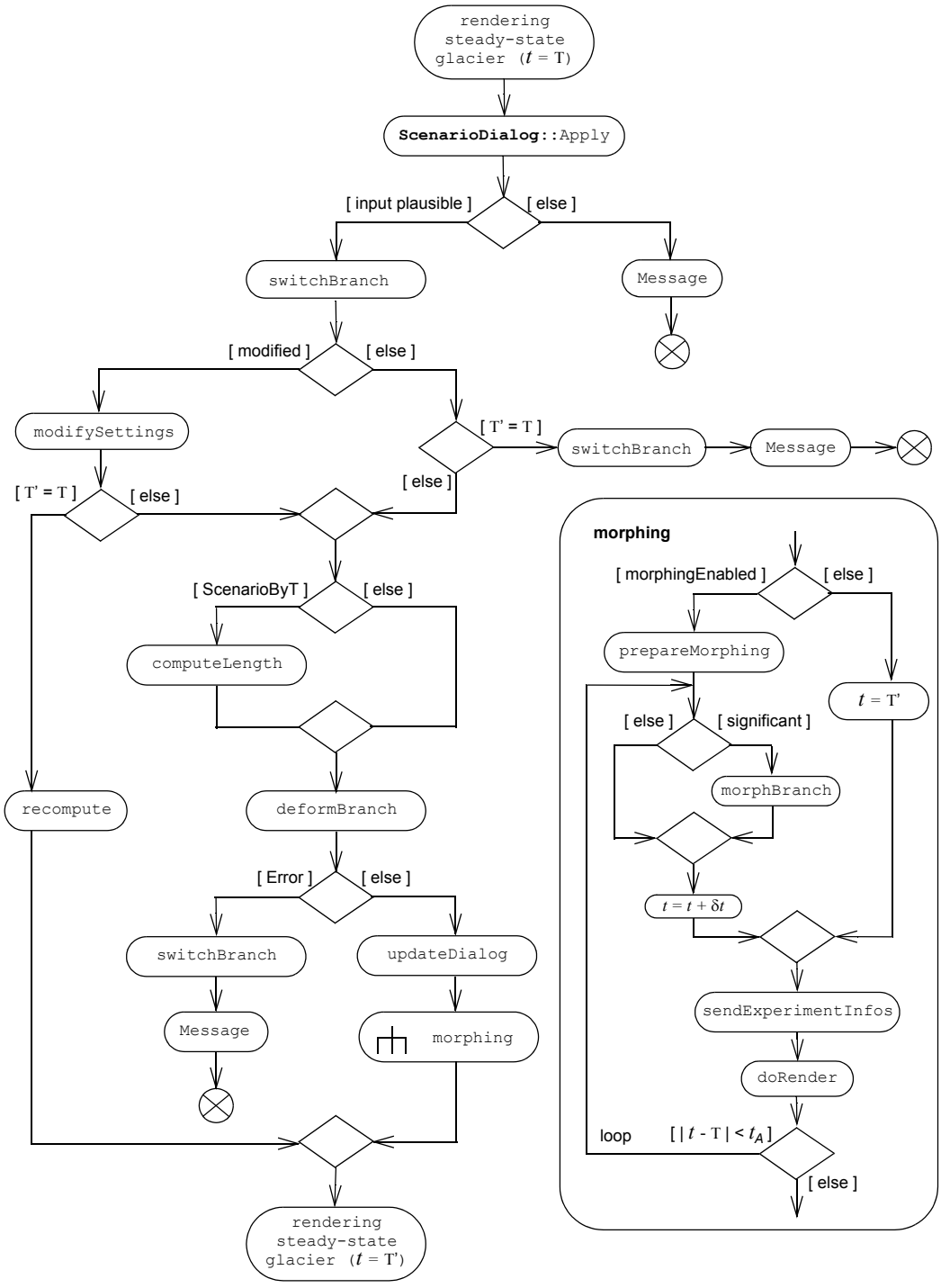


Figure 5.7: Activity diagram describing work flow when a new scenario is applied (UML 2 notation). The morphing procedure is shown as a subsidiary activity diagram.

5.3.6 Chronology of involved processes

The four most demanding components in a dynamic environment like VadreX are *simulation*, modelling the visual representation of the simulated results (*visual modelling*), *rendering* and *exploration*. The time diagram in Figure 5.8 shows the chronology of these four processes in the context of VadreX. Due to presentation issues the time axis is delineated as a non-continuous axis with interruptions. Thus, in contrast to the chronological

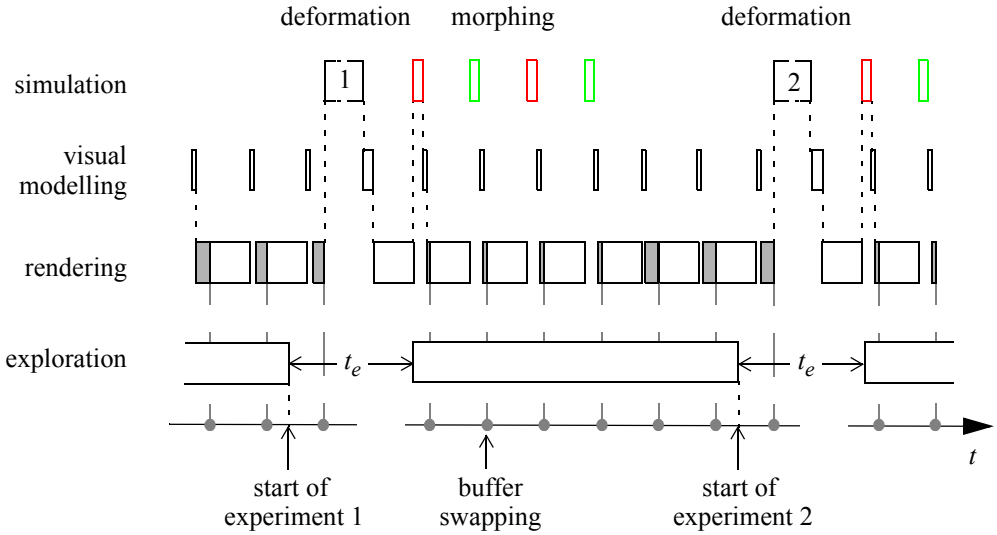


Figure 5.8: Time diagram illustrating the chronology of the processes involved in experimental glacier simulation and visual exploration (example with $n_b = 2$ and $bpf = 1$).

order, the lengths of the time frames typically do not correspond with reality. For each process rectangular boxes or so-called time frames illustrate when the process is active whereas the horizontal gaps between these boxes represent the idle time. The small points on the time axis highlight the swapping of the frame buffer which occurs only in synchronization with the frequency of the monitor. The shaded rectangles in the rendering line refer to the time the graphics pipeline is idle.

While the user is exploring the glacier, the branch surfaces are tessellated (part of visual modelling) and rendered, whereas the simulation process sits idle. When the user is starting a new experiment, e.g. applying a future climate scenario, the branches are deformed sequentially after a small delay. Exploration is disabled as long as the rendering, tessellation and fitting of the deformed branches take place. Then, if morphing is enabled, the branch surfaces are changed according to the current simulation time. In the shown example the glacier consists of two branches represented by the red and green boxes and the number of branches to animate per frame (*bpf*) is assumed to be one. Since the computing time for the morphing of one branch is smaller than the idle time of the graphics

pipeline the frame rate does not decrease and interactivity is ensured during the morphing process.

In order to be able to explore the glacier while the branches are deformed, i.e. to minimize the exploration time-out t_e, the simulation component must be running asynchronously (e.g. Belleman et al., 2000). This strategy is rejected for two reasons. First, the user could be confused if he starts a new experiment and wants to explore the new results but the previous steady-state glacier is still shown. Second, because the deformation is expected to take a small amount of time, the user should not be irritated by the small interruption. In this way, the user can clearly notice the beginning of an experiment.

5.4 Navigation and control of experiment

Although the terrain rendering application of VTP provides basic features for navigation and data probing, e.g. interactive picking of terrain position, more elaborate and appropriate techniques for navigation and information extraction should be developed and implemented in VadreX. Furthermore, strategies for conducting an experiment have to be included. In the third subchapter a method is presented for an asynchronous morphing of the branch surfaces.

5.4.1 Navigation strategies in VadreX

Virtual environments encompass typically more space than can be viewed from a single point, especially in rugged terrain. Thus, users have to be able to navigate intuitively and efficiently within the environment to get different views of any point of interest (POI) and to be placed optimally for further analysis tasks. The dynamic change of objects, e.g. glacier fluctuations, requires a navigation metaphor providing different modes of moving not restricted to observing static objects.

In VadreX all three basic categories of moving in space are covered by the implemented navigation metaphors, namely exploration, search tasks and manoeuvring tasks. In the following, the navigation approaches are presented ordered by their navigation task. Moreover, all of them respect the geospecific issues mentioned in Chapter 3.6.1 on page 44.

For exploration, i.e. navigation without any explicit target, three different metaphors are implemented. In case of the *fly metaphor* the user is able to navigate freely in 3D space in order to gain survey knowledge. The moving direction is given by the 2D cursor position on the screen whereas the speed can be modified continuously for moving forward or backward. In mountainous regions it may be useful to have different viewing and moving directions, e.g. to increase the visible part of terrain when flying horizontally. Additionally, the latency of the navigation is adaptable for more precise positioning. Aside from the fly metaphor and the *look-around metaphor* (see Chapter 3.6.1 on page 44) the *plane metaphor* is the third kind of exploring 3D environments that is included in VadreX. Here, the user is moving in a plane transversal to the viewing direction which may be beneficial, e.g., for exploring the cross-section of a branch.

Search tasks include travel interactions where the location the user would like to go to is not yet within the current field of view. To travel to a destination Enviro provides a list of previously defined views of the environment. The user can then select an entry based on the location name for immediately jumping to this location. Considering a glacier it is often desirable to look at the terminus since the position of the terminus is of high importance for evaluating the glacier condition. In VadreX a *teleportation metaphor* is implemented which makes it possible to be placed in front of the terminus of a selected branch and look towards the terminus.

Navigation metaphors for manoeuvring have to be well adapted to the kind of object to be inspected. To study the fluctuations of the whole glacier surface the sphere and cone metaphor (Döllner et al., 1999) are less adequate since they are dedicated to look at a well-defined and static point on the terrain. For observation of glacier fluctuations, the user is more interested in inspecting the whole surface(s) from a specific location in the environment. Thus, the point of interest (POI) is rather a region of interest (ROI). A *mirror metaphor* is defined that is based on the teleportation metaphor. It allows the user to travel to an interactively defined destination on terrain whereby the viewing direction is set towards the former location. In the case of large viewing distances the ability to zoom in and out is added to virtually expand the ROI. The focus is set to a distinct ROI with cursor selection.

Although the implemented navigation metaphors show different degrees of freedom, all of them are controllable by a 2D input device, e.g. a mouse. The fly metaphor is the only one allowing modification of each attribute of the viewpoint motion, i.e. position, orientation and speed. In case of the look-around metaphor and the mirror metaphor just one parameter is variable, namely the orientation or the position. The plane metaphor allows modification of position and speed. Additionally, navigation can be disabled resulting in a fixed position and orientation. This *locked-view metaphor* is well adapted for object interactions, i.e. selection, manipulation or analysis tasks.

A plan-view (or top-down-view) of the environment improves the navigation in wide areas since it provides a better scene overview. The original run-time system Enviro supports this feature by implementing an additional camera that allows switching between plan- or world-view. A dynamically-linked plan-view rendered in an additional window could also be helpful but is not yet included in VadreX as it is of less importance when exploring smaller areas, e.g. mountain glaciers.

While navigating in the virtual world, information about the current navigation state is displayed in a status bar. Information such as the position in units of the cartographic reference system, height above ground, viewing direction (azimuth and elevation) and angle between moving and viewing direction is provided.

5.4.2 Conducting an experiment

To conduct an experiment multiple times and to vary the simulation and modelling variables interactively, visual dialogs are essential and act as the interface to the underlying system functionality. A range of dialogs implemented in VadreX are shown in Figure 5.9. The dialogs are presented in left-right and top-down order with decreasing relevance for

non-experts. Although they are presented side by side they are actually arranged in a layered structure with the core dialog, i.e. the Scenario dialog, on top. The Scenario dialog offers the possibility to start a new experiment that runs from the current state (T) to the new steady-state (T'). Additionally, the user is able to select either ΔT_{air} or Δl_t as the driving force. The size of ΔT_{air} is dependant on the chosen IPCC climate scenario or a user-defined temperature curve. To repeat an experiment with modified settings the Scenario

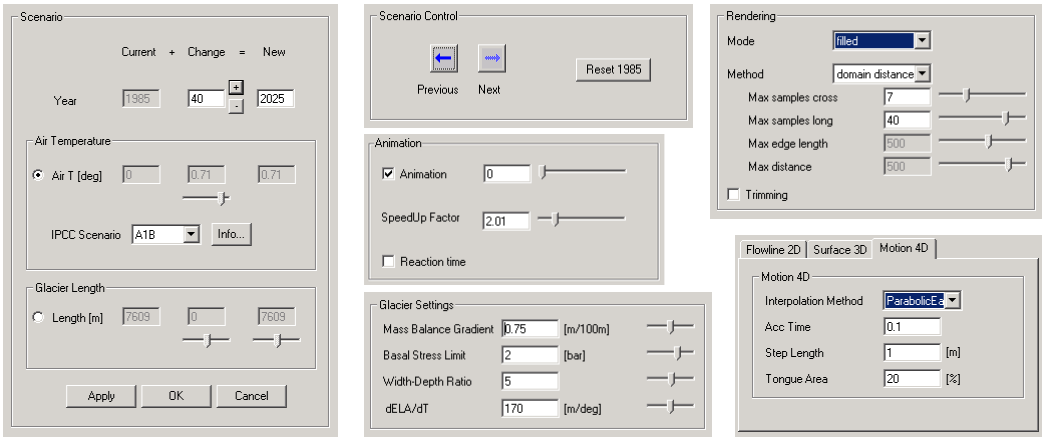


Figure 5.9: The graphical user interface for varying experiment settings represented by a range of dialogs (in left-right and top-down order): Scenario, Scenario Control, Animation, and dialogs for modifying glacier parameters (Glacier Settings), rendering settings (Rendering) and modelling settings (e.g. Motion 4D).

Control dialog allows switching between two simulation results or to recompute the initial state by pressing the appropriate button. The Animation dialog lets the user determine the time stretch factor f_T and whether the branch morphing and the reaction time t_R should be considered. The user-defined navigation in time between two steady-state branches is provided by a time slider. The remaining three dialogs can be used to vary physical properties of the glacier (Glacier Settings), settings of the surface rendering (Rendering) and variables for the geometrical modelling of the longitudinal profile (Flowline 2D), of the surface (Surface 3D) and of the morphing (Motion 4D).

5.4.3 Distributed branch rendering

The number of steps s for the complete morphing from the old to the new steady-state is defined by the total retreat Δl_t at the terminus and the maximum spatial shift δl_t per interpolation step. Thus, it would be unnessecary to change the surface at each frame as the terminus would move just slightly, i.e. the change cannot be noticed visually. Therefore, the animation costs for several branches are distributed over a few frames to save computing time. Given the size of δl_t the animation rate aps is expressed as

$$aps = \frac{vel_t}{\delta l_t} \qquad [s^{-1}] \tag{5-1}$$

where vel_t is the known retreat velocity at the terminus in the time scale T_r of the rendering system. The time scale T_r is linked to the time scale T_s of the simulation system by

$$T_r = f_T \cdot T_s \tag{5-2}$$

where f_T is a modifiable time stretch factor. Thus, vel_t is given by

$$vel_t = \frac{\Delta l_t}{t_A} \cdot f_T \qquad [s^{-1}] \tag{5-3}$$

The number of frames per animation step is defined by

$$fpa = \frac{\overline{fps}}{aps} \tag{5-4}$$

using the moving average of fps. The number of branches bpf to animate per frame is then calculated by

$$bpf = \frac{n_b}{fpa} \tag{5-5}$$

Given an example with T_r = [s] and T_s = [a], Δl_t = 240 metres and t_A = 20 a, the tongue velocity vel_t is equal 12 m/s. Further, assuming n_b = 10, $\overline{fps}$ = 30 and δl_t = 2 metres two branches would have to be animated for each frame using this asynchronous morphing approach.

5.5 Analysis of glacier fluctuations

Within the simulation loop (see Figure 1.2 on page 6) the visual analysis depicts the process of deriving information from the rendered image(s). In order to enable an efficient analysis the content of the image(s) has to be reduced to the relevant aspects. The decision of what is relevant is taken by the user, but the visual system should provide the potential approaches to (de)select information, which is described in Chapter 5.5.1 and Chapter 5.5.2. In contrast to reducing the content of the image(s), additional information can be displayed when validating the simulation results as described in Chapter 5.5.3.

5.5.1 Task-dependent variation of information complexity

Three different strategies are considered to vary the complexity of visual information. The first approach refers to optimization of the thematic content according to the actual needs for performing a dedicated task. This is done by choosing an appropriate level of abstrac-

tion (LOA) of the visualized geoobjects. In Figure 5.10 the lower part of a glacier branch and the terrain are visualized applying different LOA's. The selection of the most suitable LOA depends on the analysis task to perform. A representation as shown in image a) is

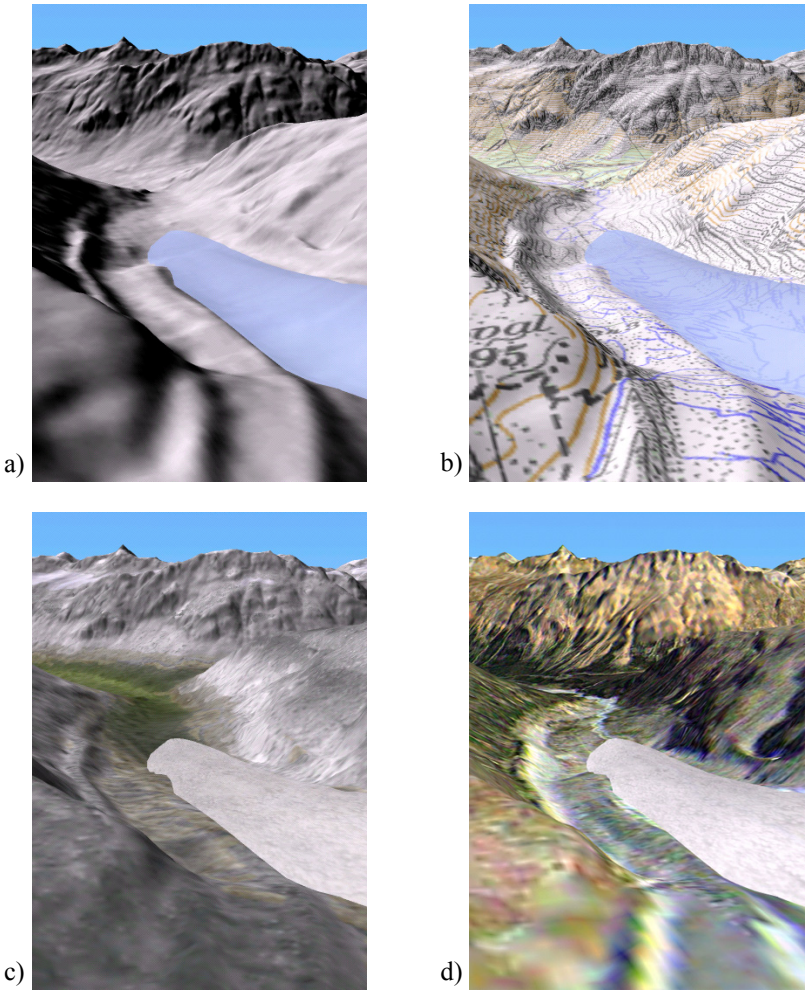


Figure 5.10: Glacier tongue and environment represented in different LOA's: a) topographic terrain texture and shaded branch surface (high LOA), b) shaded terrain with pixel map PK50 (Swiss Federal Office of Topography, 2000) and transparent branch surface (high LOA and high information content), c) geotypical representation of terrain and branch (medium LOA), d) realistic view using the Enhanced Satellite Map of Switzerland (10 m pixel size) as geospecific terrain texture (low LOA). Data source: DTM DHM25 © Federal Office of Topography (BA024148), pixel map PK50 © Federal Office of Topography (BA024148).

preferred for establishing spatial relationships between the terrain morphology and the glacier surface. Cartographic textures show a high LOA with variable information con-

tent. Their use supports efficiently the visual correlation between abstract geoinformation, e.g. signatures for footpaths or rivers, and the simulated glacier leading to an improved orientation (see image b). Furthermore, the cartographic geotexture may serve as a reference for validation, e.g. if the historic extension of a glacier is also included. In image c) the geotypical texturing method as presented in Chapter 4.4.2 and Chapter 4.4.3 is applied for both, glacier and terrain to achieve a natural representation. A geospecific texture draped on the terrain as shown in image d) adds details of high realism and may be used for presentations in the public domain. Each representation of an object corresponds to a specific and changeable LOA. VadreX supports the interactive selection of three different levels of abstraction.

Another approach to modify the information content is to change the complexity of the scene. For this, the branches can be individually faded out or in depending on the intended task. In order to see terrain information that is normally hidden by the glacier surface, e.g. cartographic signatures as shown in image b) of Figure 5.10, the transparency of each branch can be modified.

The last method is based on the idea of reducing the dimensionality of the representation of the object to simplify the analysis of a specific feature. The implemented methods to derive and visualize the most important properties such as numbers or 2D profiles, are discussed in the following chapter.

5.5.2 Interactive retrieval of glacier information

In order to form a discriminative measure of glacier simulations, measurements of multiple properties of the glacier should be performed. Consequently, the system has to provide methods and tools for the interactive retrieval of glacier information. To ensure a precise object interaction most of the following tasks are only possible when the locked-view navigation metaphor is applied.

The selection of the branch of interest is done either by choosing the branch from a drop-down list or by the mouse. Instead of performing a costly ray/surface intersection test a fast method is implemented relying on a ray/terrain intersection method (Discoe, 2004) and the numerical algorithm previously used for the discretization of the branch surface (Chapter 4.2.4 on page 60) to determine if there is a branch surface above the selected terrain location.

The most important variables of a running simulation are rendered simultaneously as a message board. Examples of critical variables are simulation time, air temperature, length, area and volume of the selected branch, and also their relative changes with regard to the initial simulation time. For high-quality rendering the message board is realized as an `osg::Text` object. The message board is drawn with blending to get an anti-aliased rendering of the text. More detailed information about simulated physical properties of steady-state branches and climate scenarios are provided in a separate window.

A feature of general interest is the interactive measurement of the distance and height difference in 3D between two terrain locations, used in VadreX for example for measuring glacier retreat.

When exploring or inspecting a branch surface it is hard to get an image of the thickness variation in longitudinal or transversal direction of a branch. For improved analysis a well-adapted strategy is reduction of the geometrical dimensionality of the 3D phenomenon to 2D. Since the thickness values along the flow-line are already in memory no additional computing tasks are necessary to provide the longitudinal 2D profile. The longitudinal profile shown in the left image of Figure 5.11 is embedded in a separate window that

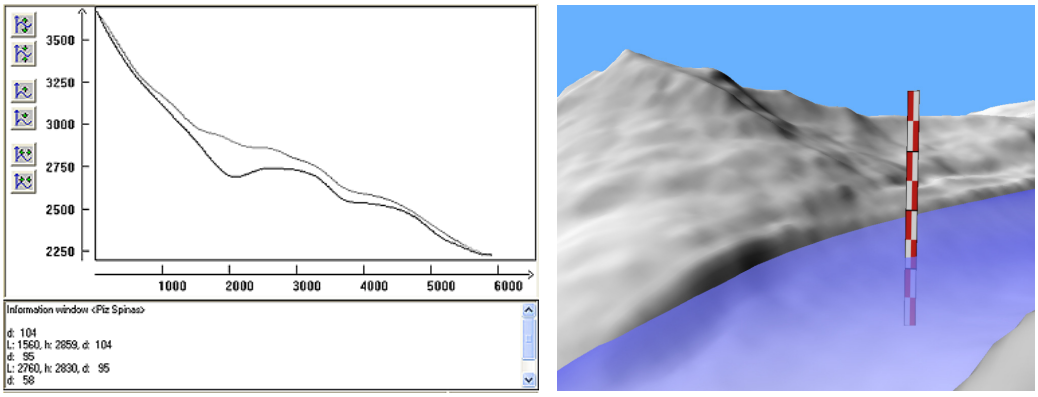


Figure 5.11: Interactive retrieval of thickness information: 2D longitudinal profile with information area in a separate window (left) and in-situ measurement using the measuring bar (right).

contains in addition an information area displaying key properties of the selected location. Thus, the spatial dependence between the frame of reference, i.e. the 3D environment, and the displayed object, i.e. longitudinal profile, is displayed indirectly and linked by highlighting the corresponding branch surface. The 2D plot of a cross-section is generated similarly but the user has to first define a reference location in longitudinal direction of the branch surface.

Glaciologists have a general interest in knowing the hypsography to assess the local situation. The simulated hypsography may act as a further variable helping to evaluate discrepancies between simulation and reality. Therefore, the simulated hypsography of a branch can be displayed as a 2D plot in the same manner as used for the longitudinal profile and the cross-section.

To give the user a working tool to measure the main geometrical properties in the 3D environment, i.e. branch thickness and width, VadreX provides a measuring bar that is represented visually by a 2D glyph, i.e. textured billboard (see Figure 5.11 right). Applying the locked-view metaphor for navigation, the measuring bar is interactively placed by cursor movements. As a first estimate, the depth at a specific location can be derived visually via the pattern of the applied bar texture. The height of the bar in the example of Figure 5.11 right is 400 metres and the branch thickness is thus estimated to be about 120 metres. Additionally, the width and thickness of the branch at the current terrain location can be computed on demand using a Newton iteration method. In contrast to the longitudinal profile visualized on the left side where the thickness is measured perpendicular to

the branch surface, the thickness estimated with the measuring bar is computed in vertical direction and refers to the variable $\widehat{d_x}$.

5.5.3 Visual validation of glacier simulations

Every model in environmental sciences, such as the model developed in this study for computing glacier fluctuations, describes just a small part of the whole ecosystem. In addition, this part is modelled as a closed system. Therefore, each environmental model is wrong to a certain degree when compared to reality. To rate the error of the simulation, validation is a fundamental process. Together with the methods for information probing, visual methods for validation purposes are helpful and exploit the dominance of the human visual channel. In order to validate simulation results three main different approaches can be distinguished:

- modifying rendering settings for an appropriate branch appearance,
- adding precompiled reference data, and
- adding dynamically generated reference data.

Applying well-adapted settings for the rendering of a branch surface is a straight-forward and efficient approach to examine the shape carefully and to allow further investigation using any of the methods for information picking. For example, to compare the dynamic change of the shape of several branches, the surfaces can be rendered using reduced transparency or as wireframe. Another effective validation method is to change interactively the light position that is used for shading of the branch surface (see Figure 4.16 on page 70) since shading has a big influence on the perception of shape. This allows to easily detect inconsistencies of the surface(s).

Precompiled reference data that is stored persistently in a DBMS may be visualized in conjuntion with a simulated 3D reconstruction of a glacier state. If the reference information exists as a 2D thematic map, multi-texturing provides an efficient way of rendering the additional information without increasing geometrical complexity. Since modern graphics hardware supports at least 2 or more texture units for fast texture blending in screen space and OSG is able to handle multiple textures, the 2D thematic maps are combined with the terrain texture in real-time[1]. The size of the textures is limited by the graphics hardware, i.e. the maximum texture dimension and the memory, because no paging algorithm is considered. The user of VadreX is free to select interactively any number of texture layers within the limits of the graphics hardware.

For some glaciers 1D reference data is available, documenting the retreat of the glacier terminus relative to a reference point. A labelled representation of one measured sample is generated by using a static glyph combined with a text billboard. Furthermore, an animated glyph shows the observed ‘real’ position of the tongue corresponding to the current simulation time which helps to dynamically perceive discrepancies in fluctuations.

To evaluate the accuracy of simulations of future fluctuations there exist no real measurements that can be used as reference data. But to assess the influence of varying pa-

1. Although it is slower, multi-texturing could also be realized using multiple rendering passes.

rameter settings the ‘reference’ data has to be generated dynamically based on results from previous experiments. One simple possibility is to switch between two simulation results (`OldBranch<->NewBranch`) which is provided by the `Scenario Control` dialog.

Dynamic glacier capturing allows dynamic generation of a 2D map of the current glacier and takes advantage of basic techniques of modern rendering systems. The method uses multi-pass rendering where in a first pass only the glacier is sent to the frame buffer applying an orthographic projection matrix and setting the viewing frustum to the terrain extensions (see Figure 5.12). Then, a transparency value of 1 is assigned to non-glaciated

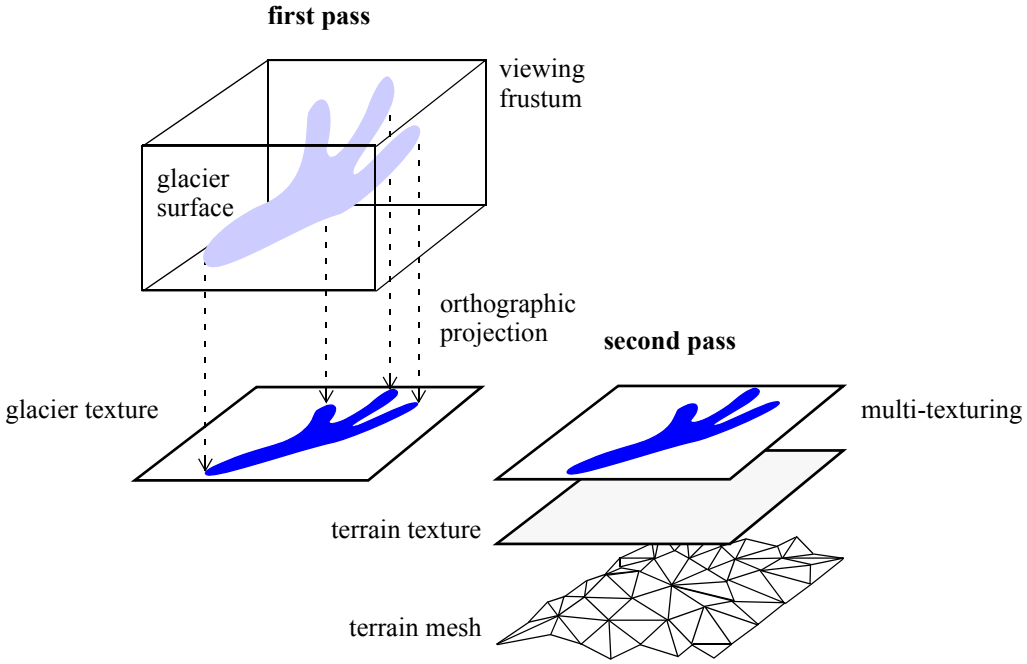


Figure 5.12: Dynamic capturing of the glacier.

regions and the branch surfaces are coloured blue. In a second pass the rendered glacier image is combined with the terrain texture (and other thematic textures) using multi-texturing. The image-based representation of the surface helps to save rendering time and allows therefore visualization of the extension of a second glacier in real-time. Since dynamic capturing is done on graphics hardware, the size of the resulting texture is limited by the size of the frame buffer.

Moreover, by calling the buffer read back routine the dynamically generated 2D pixelmap of the glacier can be exported as a georeferenced image for later use in a conventional 2D GIS. Another positive side-effect of the new method for dynamic glacier capturing is that it is now possible to compute efficiently the projected glacier area A_{ortho} containing several branches. Previously, the calculation of the area was restricted to the ‘real’ area of one branch surface using functionality of the `Nurbs++` package. However, to compare the

simulated size of the glacier area with values from literature, the area A_{ortho} of the projected surface has to be known. Furthermore, the glacier volume V_{ortho} can be estimated for comparison purposes by multiplying the glacier area A_{ortho} with the mean vertical thickness.

Chapter 6

Case study

A case study is carried out in order to evaluate the implemented methods for simulation, rendering and analysis of glacier fluctuations. As test site the Morteratschgletscher in the Eastern part of Switzerland is chosen because it consists for the most part of temperate ice, as required by the simulation model. Since this glacier is already well-studied, reference information exists for validation purposes. Sufficient glaciological knowledge (Hoelzle, 1994, Maisch, 1992) about this glacier and digital terrain information (Swiss Federal Office of Topography, 2001) is available.

Considering the restrictions of the simulation model ($L > \sim 2$ km, temperate ice), there are approximately 400 glaciers in the Alps and thousands of glaciers all over the world that could be simulated (Hoelzle, 2004).

6.1 Investigated glacier

The Morteratschgletscher lies in the Upper Engadin which is a high-valley at a mean altitude of approximately 1750 m a.s.l. with some smaller side-valleys and mountains at heights of 4000 m a.s.l.. The climate of the Upper Engadin is characterised as an inner-alpine continental climate and is mainly influenced by the weather conditions of the southern part of the Alps.

The Morteratschgletscher is located in one of the many side-valleys of the Upper Engadin and spans currently an altitude range of nearly 2000 metres. There exists two main branches contributing to the glacier in the Morteratsch valley: the Morteratschgletscher and the Persgletscher that flows into the lower part of the Morteratschgletscher. Within this study both glaciers are generally considered as the same glacier and called in the following just Morteratschgletscher. In Figure 6.1 the extent of the Morteratschgletscher in the year 1973 is shown. The applied method for generating the shown glacier area is described in Paul et al. (2002). At this time the height of the glacier terminus was about 2050 m a.s.l. and its highest point reached 4000 m a.s.l.. The glacier has been manually

divided into n_b = 11 different branches for further modelling purposes whereby each branch is defined by an individual set of flow-points. In Figure 6.1 these flow-points are represented by red dots. The labelling of a branch is based on the map name of a characteristic location that is close to the source point.

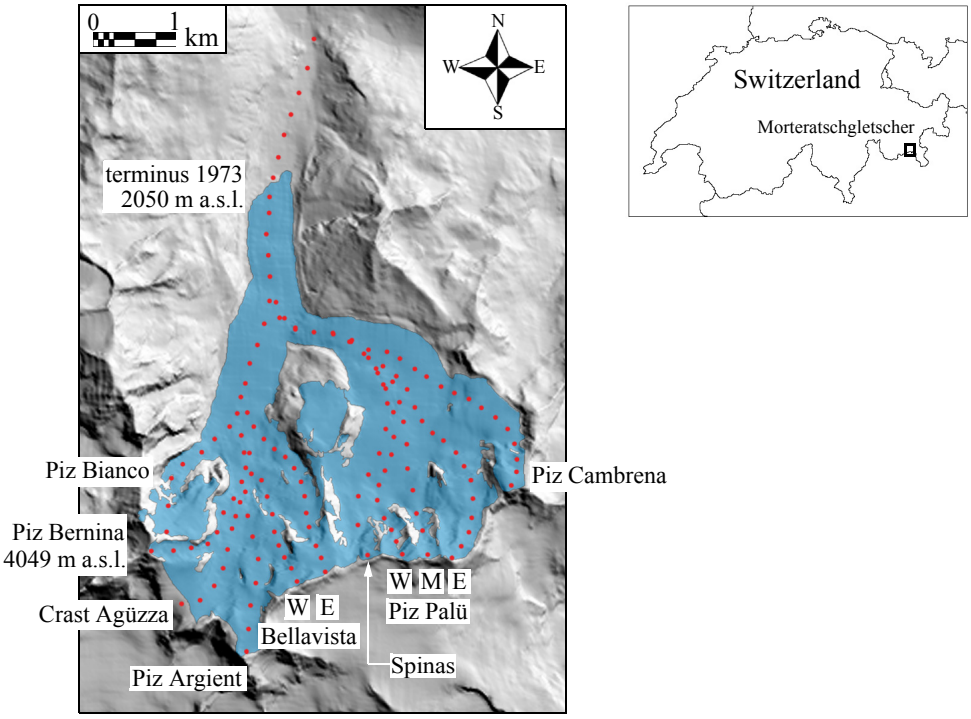


Figure 6.1: The Morteratschgletscher in the Upper Engadin (Switzerland) and its extent in the year 1973 (blue). The manually-defined flow-points of 11 branches are represented by red dots. Data source: DTM DHM25 © Federal Office of Topography (BA024148).

6.2 Initial glacier state 1985

To compute and visualize fluctuations of the Morteratschgletscher the initial state has to first be simulated. Its surface is defined by the morphology derived from the digital terrain model DHM25 (Swiss Federal Office of Topography, 2001) and thus, the initial date corresponds to the time of the photogrammetric measurements. In case of the Morteratschgletscher these measurements were carried out in the year 1985. Furthermore, a few parameters have to be assumed as initial input to the simulation (see Table 6.1). The location of the terminus $\mathbf{P_t}$ is determined either from the DTM or from topographic maps and has to be projected onto the flow-line $\mathbf{C_f}$. The height of the equilibrium line (*ELA*) is an important factor but nevertheless difficult to estimate since it is variable over time and space. As a first estimate the ELA is set to the mean height of the glacier for practical rea-

sons. Standard values are taken for the shape factor f (0.75), for the mass balance gradient $\delta b/\delta h$ (0.75 mwe / 100 m), for the coefficient A (0.16 a^{-1} $bar^{-n_{ice}}$) and exponent n_{ice} (3) of Glen's flow law and for e (8). The sampling size l_s is 200 metres which leads to n_i = 15 using the rule of thumb that $n_i \approx n_x / 2$. The size of α_{limit} is set to 0 deg, i.e. there is no limitation of the surface slope.

Table 6.1: Used parameters for the computation of the glacier bed (t = 1985)

P_t	$\delta b/\delta h$ [m / 100 m]	ELA [m a.s.l.]	f	l_s [m]	e	α_{limit} [deg]	n_i	A a^{-1} $bar^{-n_{ice}}$	n_{ice}
E 791'925 N 145'170	0.75	3000	0.75	200	8	0.0	15	0.16	3

In the following section some simulated properties of all branches of the Morteratschgletscher are discussed. The maximum thickness $\widehat{d}_{max}$ varies between 151 and 264[1] metres and the mean thickness $\widehat{d}_{mean}$[2] between 50 and 56 metres. Although they seem to be in a reasonable order it is difficult to validate them. Maisch (1992) estimates the mean thickness of the Morteratschgletscher in 1973 to be around 70 metres using an empirically derived relation. Unpublished field measurements with radio echo soundings carried out in the late eighties of the last century resulted in a maximum thickness in the order of 290 metres (Haeberli, 2003). A remarkable difference is observed between the response times t_A of the different branches (23 - 40 a) whereas the reaction time t_R (21 a) is more or less constant[3]. The computed surface velocities (56 - 71 m/a) and deformation velocities (18 - 21 m/a) are in a reasonable range and a velocity ratio around 0.7 is typical for glaciers of this size. The simulated mass balance b_t at the terminus is -6.7 metres and compared to field measurements it lies at the upper limit. Possible reasons could be the relative high mass balance gradient (0.75) or ELA. The basal shear stress τ_0 varies slightly around 1.45 bar (+/- 0.02 bar).

6.2.1 Simulated extension 1985 of the Morteratschgletscher

The validation of the simulated surface is done by comparing the orthographic projection of the simulated glacier volume with the extent of the real glacier derived from remote sensing data (Paul et al., 2002) as depicted in Figure 6.2. The real extent is based on the glacier stand 1973 whereas the simulation is performed for the year 1985. Thus, the slight difference in length at the terminus is due to this time gap. Otherwise, a good agreement is achieved. Main differences can be noticed in the accumulation zone and at the tongue where the simulated area is too small. Also, some isolated glacierets are not taken into considerations during simulation. At the location $\mathbf{P_o}$ the simulated area (or width) is too large. It seems that the used relationship between thickness and width has to be modified so that additional processes such as increased erosion play a role. An interesting fact is

1. The RMS error of the parametric surface fitting is 15 metres.
2. The calculation of the mean thickness is based on equally spaced sampling points lying on the branch surface.
3. The relevant velocity of the kinematic waves is estimated by $c \approx 4 \cdot u_s$ (Paterson, 1994).

that the difference in width at the tongue correlates with the passive part of the glacier tongue where no dynamics occur (indicated in reality by rock material on the surface).

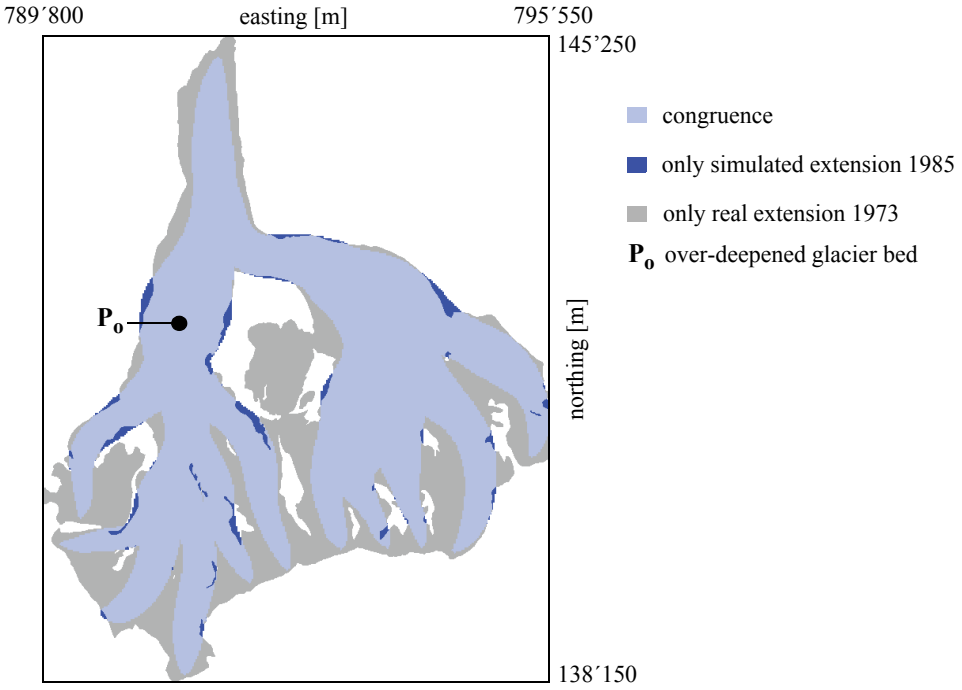


Figure 6.2: Comparison between the simulated and real horizontal extensions of the Morteratschgletscher (date of real extension: 1973, date of simulation: 1985)

Maisch (1992) used planimetric methods to estimate the area and the volume of the Morteratschgletscher 1973 and an empirical relation to calculate the mean thickness. He obtains an area of 16.401 km^2, a mean thickness of 71.3 metres and a volume of 1.17 km^3. A comparison between these measurements and the simulated values (A_{ortho}: 12.4 km^2, $\widehat{d}_{mean}$: 53 m, V_{ortho}: 0.66 km^3) results in a remarkable difference, especially for the volume. The main reason for the differences in area and volume is assumed to be the underestimation of the accumulation area when using just 11 branches.

6.2.2 Hypsography

In contrast to the mass balance gradient the hypsography of a glacier reflects the local situation that determines the individual behaviour of a glacier. In Figure 6.3 the area-elevation distribution of four selected branches is illustrated whereby each stem refers to a height range of 40 metres. The plots show that a significant amount of the area is located in the bottom half of the glacier, especially in case of the branches 'Piz Bianco' and 'Piz Argient'. Due to the uniform slope the area of the branch 'Piz Cambrena' is evenly distributed around the mean height.

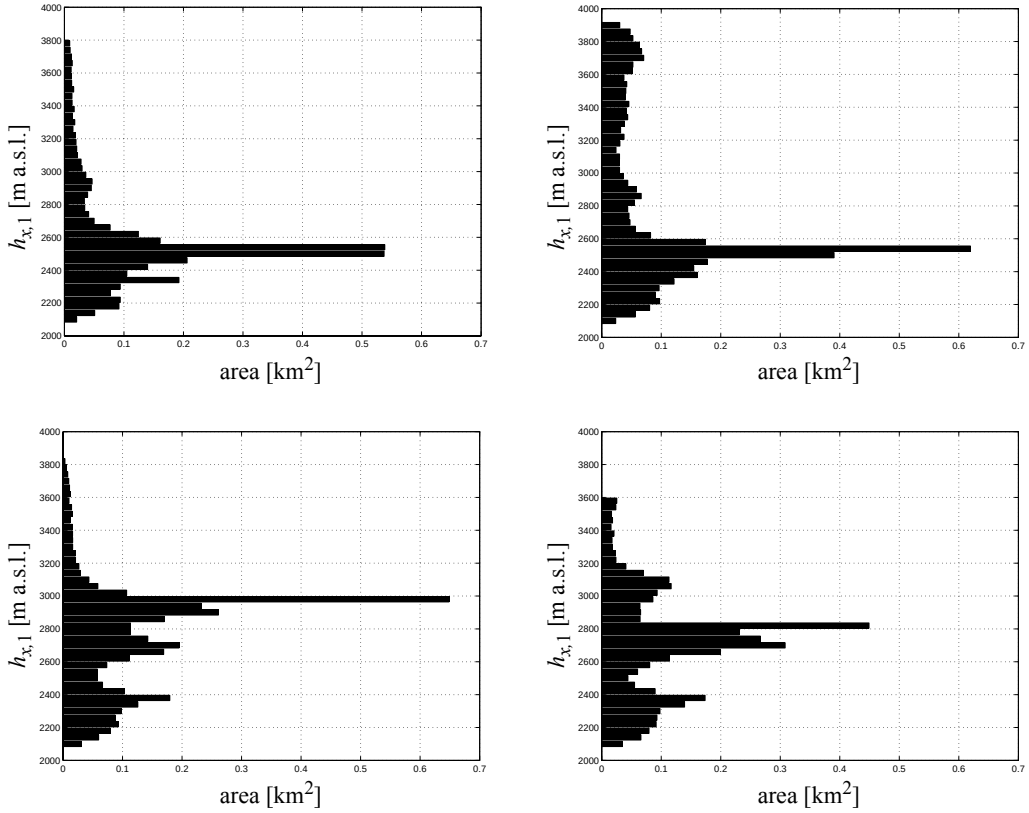


Figure 6.3: Area-elevation distribution of the simulated branches Piz Bianco (top left), Piz Argient (top right), Piz Palü W (bottom left) and Piz Cambrena (bottom right). Height resolution: 40 metres.

For comparison purposes the *ELA* can now be estimated based on the hypsography. According to the two widely-applied assumptions that the *ELA* divides the glacier area in such a manner that the *AAR* is either 0.5 or 0.66, the corresponding heights $h_{1:1}$ (AAR = 0.5) and $h_{2:1}$ (AAR = 0.66) can be regarded as an estimation of the *ELA*. From the listed values in Table 6.2 one can conclude that in case of the first two branches nearly no dif-

Table 6.2: Estimation of ELA for *AAR* = 0.5 (left column) and *AAR* = 0.66 (right column) based on the simulated 3D branch surface.

Branch name	$h_{1:1}$ [m a.s.l.]	$h_{2:1}$ [m a.s.l.]
Piz Bianco	2520	2500
Piz Argient	2540	2500
Piz Palü W	2840	2680
Piz Cambrena	2740	2620

ference between the heights $h_{1:1}$ and $h_{2:1}$ exists and that there is a significant difference between the two branches of the Persgletscher ('Piz Palü W' and 'Piz Cambrena') and the other two branches. This difference is confirmed by observations of Maisch (1992) who estimates $h_{2:1}$ (1973) for the main branch of the Morteratschgletscher at 2695 m a.s.l. and for the Persgletscher at 2815 m a.s.l.. The simulated values of $h_{2:1}$ are therefore approximately 150 - 200 metres too low. This slight discrepancy is not surprising since the simulated accumulation area is too small as mentioned previously.

If the *ELA* lies above $h_{1:1}$ over a long period, the glacier will experience a continuous loss of ice till its final disappearance. However, when using the simulated variable $h_{1:1}$, the estimated time of glacier existence would be possibly too short since the simulated hypsography differs from reality (too small A_{acc}).

6.2.3 Simulated glacier thickness

With the method presented in Chapter 4.2 longitudinal profiles can be computed based on the glacier surface and the manually defined flow-points. In Figure 6.4 the longitudinal

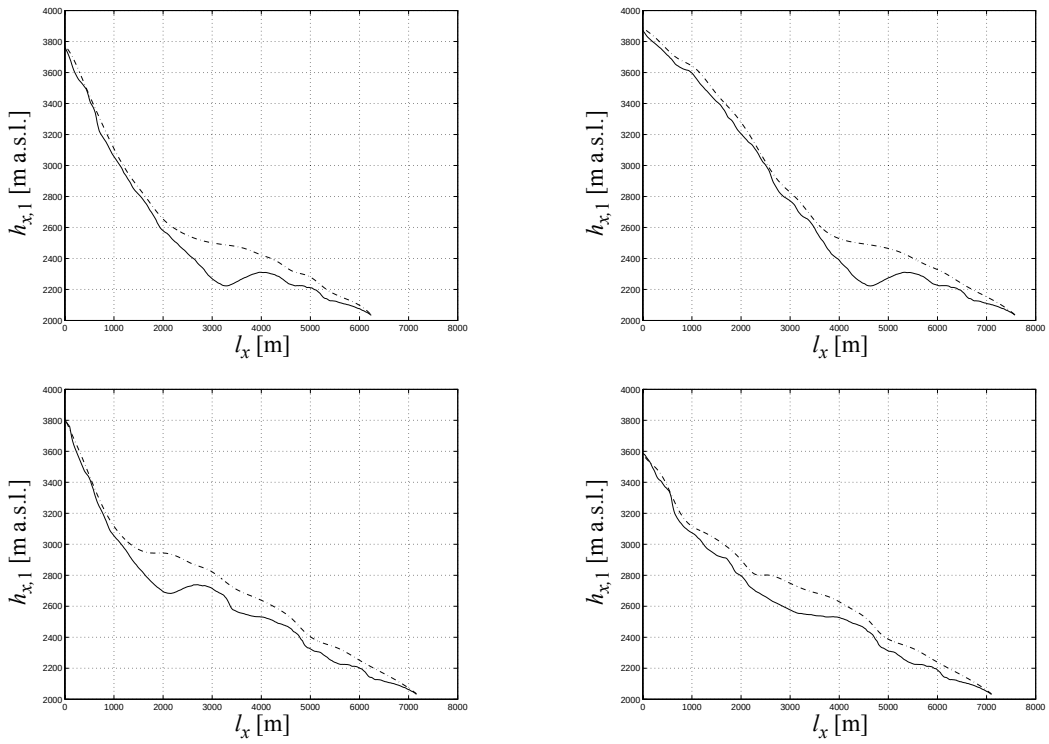


Figure 6.4: Longitudinal profiles of the branches Piz Bianco (top left), Piz Argient (top right), Piz Palü W (bottom left) and Piz Cambrena (bottom right).

profiles of 4 selected branches are presented. Due to the parametric description of the surface, all of them show a smooth shape representing the main variations in thickness. The correlation between surface slope and thickness can be well recognised in all of the pro-

files. In the profiles of the branches ‘Piz Bianco’ and ‘Piz Argient’ the location with the maximum thickness is easily recognizable due to the over-deepening of the glacier bed. The over-deepening of the bed may be caused by a) increased erosion because the glacier flows from a very steep to a flat area resulting in high water pressure within the glacier and b) strong vertical ice extension due to horizontal ice compression. These regions are important for the assessment of future hazards since they depict potential zones of subglacial water reservoirs.

The spatial distribution of the simulated glacier thickness is visualized in Figure 6.5. It can be seen that there are two regions of increased ice thickness. Both regions are located in a plane just below steep slopes where the thickness is smaller than 100 metres. The stepped pattern of the contour lines results from the discretisation of the branch surfaces which took place during bed computation. The applied spatial sampling was 25 metres according to the horizontal resolution of DHM25.

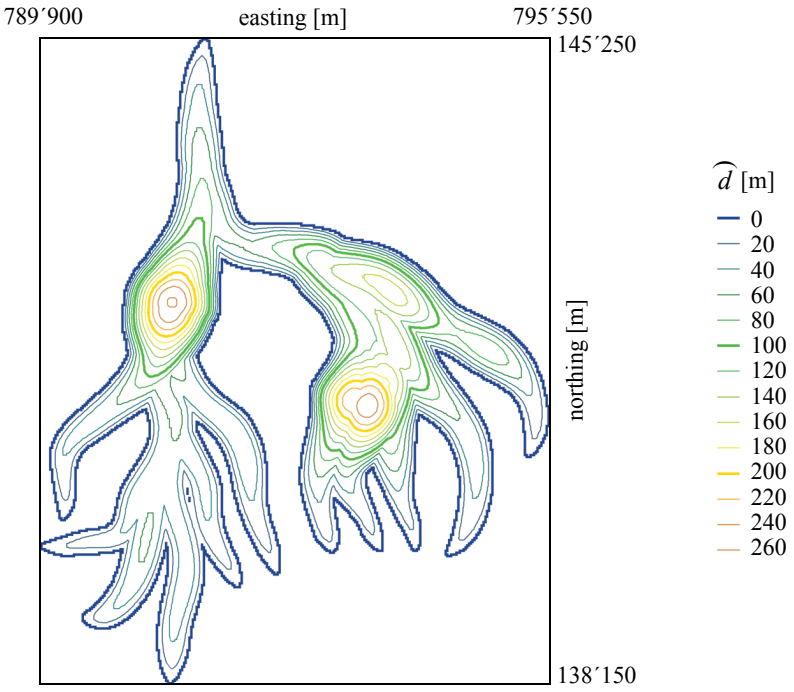


Figure 6.5: Contour plot of the thickness (1985) of the Morteratschgletscher.

Since the real *ELA* is not known it is important to study the influence of varying *ELA* on the bed computation. Changing the height of the equilibrium line has an effect on the ice cap factor and thus on the simulated glacier thickness. Differences in ice thickness between the glacier thickness computed with *ELA* = 3000 m a.s.l. and the volume with *ELA* = 2850 m a.s.l. (left) or *ELA* = 2700 m a.s.l. (right) are visualized in Figure 6.6. These values correspond to the averaged height $h_{1:1}$ (2700 m a.s.l.) and $h_{2:1}$ (2850 m a.s.l.) for the Morteratschgletscher in the year 1973 (Maisch, 1992).

A lower *ELA* causes the thickness to increase in areas below *ELA* and decrease above the *ELA*, especially in flat areas. The change of thickness is linearly correlated with the change of ELA. Regarding the right image, it can be concluded that the thickness in the area of over-deepening increases by 20 metres. Thus, by applying a more realistic size of ELA the measured maximum thickness of 290 metres is nearly achieved.

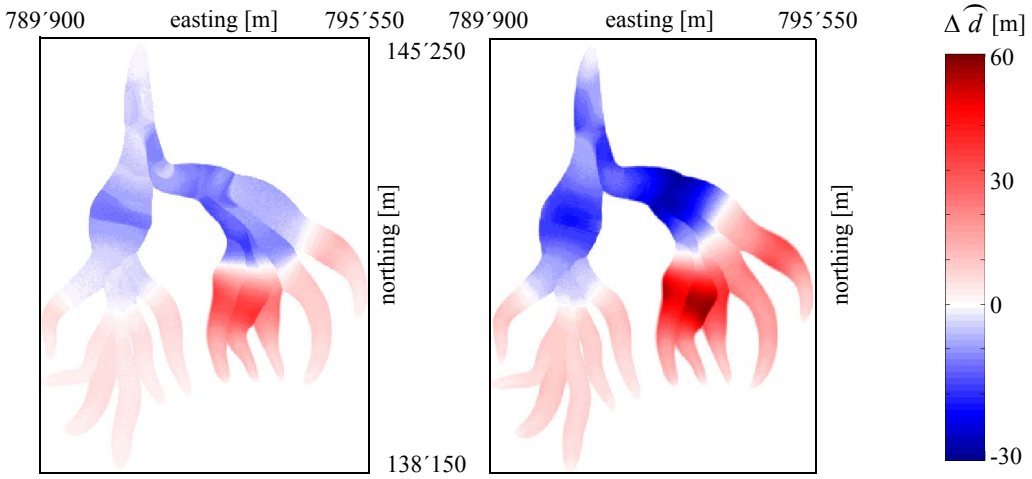


Figure 6.6: Impact of ELA on the simulated glacier thickness: differences between the glacier thickness computed with *ELA* = 3000 m a.s.l. and *ELA* = 2850 m a.s.l. (left) and computed with *ELA* = 3000 m a.s.l. and *ELA* = 2700 m a.s.l. (right).

6.3 Reconstruction of the retreat from 1850/60 to present

The model developed for the simulation of mountain glacier fluctuations should be compared with recorded reference information. Aside from determining the sensitivity of the different input parameters for the model one gets an impression of the range of uncertainty.

There are several simplifications and assumptions made within the simulation part. The relation used to estimate the mean mass balance integrates many feedback processes (e.g. albedo, accumulation) that is expressed by the variables $\delta ELA/\delta T_{Air}$ and $\delta b/\delta h$. Both of them are unknown but their size can be approximated by reconstructing a measured glacier change in a step-by-step procedure, i.e. in an experimental simulation loop (see Figure 1.2 on page 6). The computation of the change of the glacier length based on the mean mass balance lies in a range of 30%. Additionally, to estimate the thickness the slope of the surface has to be known. Therefore, for reconstruction purposes, it is estimated using paleo-information or other simplifications. Finally, the continuous description of a branch surface is achieved by a parametric fitting of the branch points $\mathbf{B}_{x,1}$. Depending

on the fitting method and settings the NURBS surface may differ 20 metres in height from $\mathbf{B}_{x,1}$ (see Chapter 4.2.3 on page 57).

6.3.1 The maximum stand 1850/60

The Morteratschgletscher reached its last steady-state in the years between 1850 and 1865, i.e. at the end of the 'Little Ice Age'. The date of the maximum stand is not clearly defined. For this case study the date of the maximum stand is set to 1850. Thus, this stand has to be simulated and geometrically modelled before computing the retreat afterwards.

There exist continuous measurements of the retreat of the Morteratschgletscher since 1878 (Beeler, 1977). These measurements are used as reference for testing the simulation model. The gap between the maximum stand and the fixed point of the measurements is approximately 100 metres.

The maximum stand is simulated using the known position of the terminus (see Figure 6.7). Since the terrain between the extensions 1985 and 1850 is nearly horizontal with a height difference of 130 metres on a distance of 2000 metres it is assumed that the slope of the terrain is equal to the slope of the ancient surface in this zone. The change of the

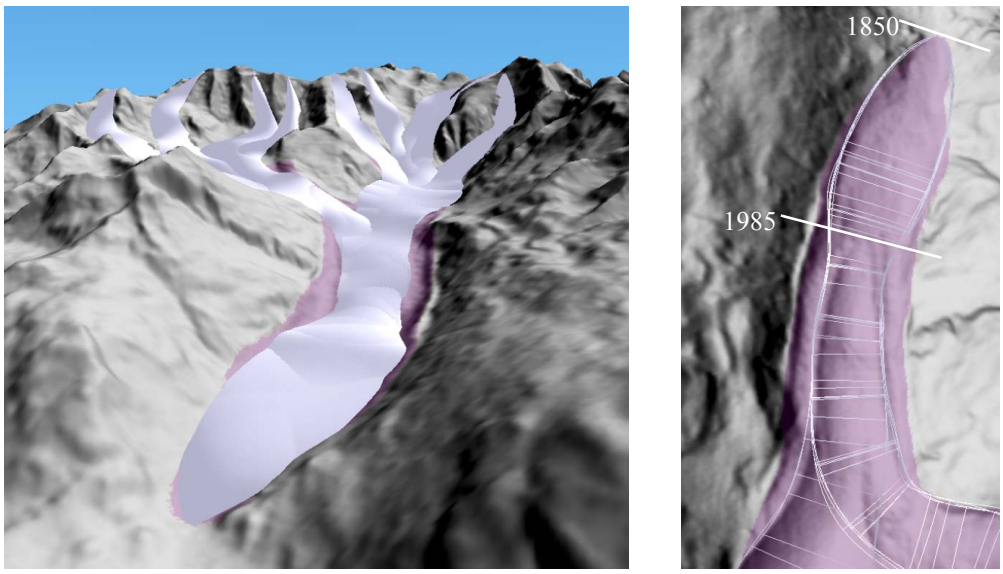


Figure 6.7: Perspective (left) and top-down (right) view of the simulated maximum stand 1850/60. The violet texture layer represents the real extension (Paul et al., 2002). In the right image only the outline of the Bézier patches are rendered. Data source: DTM DHM25 © Federal Office of Topography (BA024148).

ELA from 1850 to 1973 is estimated to be in the order of 90 metres for the Morteratschgletscher (Maisch et al., 1999). Although a small difference was noted between Persgletscher and Morteratschgletscher a uniform decrease of ΔELA = -100 metres is applied between 1985 and 1850.

The shaded branch surfaces in Figure 6.7 left show the simulated shape for the year 1850. The additional texture layer in violet refers to the reconstructed extension 1850 and allows visual detection of differences between reality[1] and simulation. It is combined in real-time with the topographic texture using multi-texturing. Comparing the simulated surface with the real extension, one can conclude that the width and height of the glacier are well reconstructed between the extensions 1985 and 1850 (see also Figure 6.7 right). The transition between different branches can be clearly recognized due to the applied Gouraud shading.

In Table 6.3 the major spatial parameters are listed and compared with values of Maisch (1992). Due to the reasons mentioned in Chapter 6.2.1, the simulated area is smaller (-5 km^2) than the reference values. The differences of the mean thickness and of the volume are smaller than for the initial state in 1985 (see Chapter 6.2.1 on page 107).

Table 6.3: Comparison between simulated parameters of the Morteratschgletscher 1850 and values of Maisch (1992).

Glacier	L [m]	A_{ortho} [km^2]	$\widehat{d}_{mean}$ [m]	V_{ortho} [km^3]
Reference 1856, Maisch (1992)	8.9	19.25	76.4	1.47
Simulated 1850	9.1	14.23	67	0.96
Difference	+0.2	-5.02	-9.4	-0.51

The simulated area and volume depend on the number of branches that are taken into account for glacier modelling. As shown in Figure 6.7, the simulated area of the Morteratschgletscher gets closer to reality as the number of branches increases. Based on a cubic extrapolation, a realistic glacier area would be achieved when using approximately 19 branches.

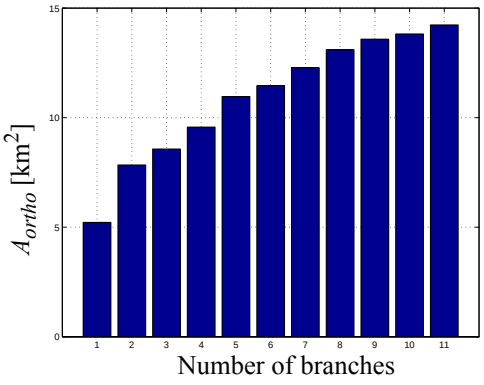


Figure 6.8: Influence of the number of branches on the simulated area of the Morteratschgletscher in 1850.

1. Although simplified it is assumed that the reference information is equal to reality.

6.3.2 Introductory experiment to simulate the retreat

The retreat of the Morteratschgletscher from the maximum stand to present serves as an interesting case study to calibrate the simulation model. An important but difficult part of this process is the estimation of the change in mass balance because it is unknown for this time period. Thus, it has to be approximated using plausible assumptions for the change in air temperature, humidity, radiation and accumulation (e.g. snow). The feedback effects between these four main components is considered by Kuhn (1990) and integrated in the variable $\delta ELA/\delta T_{Air}$ = 170 +/- 50 m/°C.

As a first estimate the air temperature is approximated by using measured data from the closest meteorological station Sils-Maria (1798 m a.s.l.) which is about 14 km away from the Morteratschgletscher. It has to be verified in this chapter wether this approach leads to plausible results.

The daily measured values of T_{Air} are averaged as MAAT and plotted in Figure 6.9 left. The red line indicates a linear interpolation of these values whereby the gradient is much bigger in the last quarter of the 20th century. This abrupt change is confirmed by considerations made in OcCC (2002) which are based on the analysis of several meteorological stations in Switzerland. According to OcCC (2002) an increase of T_{Air} by 1.0 °C between 1900 and 2000 can be assumed for the southern part of Switzerland which is 65% above the global value of 0.6 °C (OcCC, 2002). In the last three decades the warming was approximately 0.4 °C per decade in the southern part of Switzerland (OcCC, 2002). In contrast to the air temperature the plot of the measured precipitation shows no

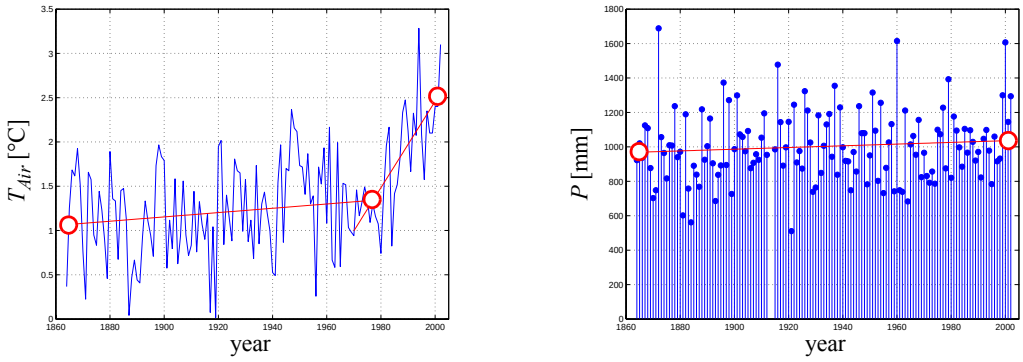


Figure 6.9: Homogenised MAAT (left) and annual precipitation (right) measured in Sils-Maria (1798 m a.s.l.) between 1864 and 2002. The linear interpolation of T_{Air} and P is shown as a red line. Data source: MeteoSchweiz.

significant trend during the time period 1864 - 2002 (Figure 6.9 right). It is uncertain if the atmospheric conditions in Sils-Maria are representative for the Morteratsch valley and for the high mountain regions above 3000 m a.s.l. but they may indicate some general trends of the region. Thus, the following values are applied as initial settings for the change of air temperature: ΔT_{Air}: +0.28 °C (1864 - 1978) and +1.21 °C (1978 - 2002). The feedback effect of increased precipitation in case of a warming is absorbed in the value of

$\delta ELA/\delta T_{Air}$. A higher mass turnover due to more mass gain can be simulated by raising the size of $\delta b/\delta h$. These long-term trends do not include any information about the seasonal distribution and changes of the atmospheric conditions. The small-term variations are not considered by the simulation model.

Now that the atmospheric changes have been estimated the mass balance change can be calculated using Equation 4-10 on page 63. According to Equation 4-10 the mass balance change Δb increases when the glacier shows a high mass turnover typically observed in areas with high precipitation (represented by a high value of $\delta b/\delta h$) and when the change of the equilibrium line $\delta ELA/\delta T_{Air}$ is high. Applying maximum values within a

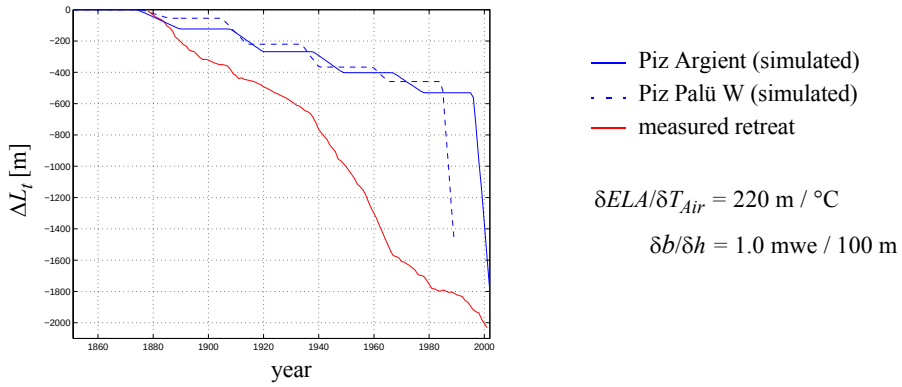


Figure 6.10: Reconstruction of the retreat since 1850 applying the linearly interpolated air temperature trend based on the measurements recorded in Sils-Maria.

reasonable range ($\delta b/\delta h$: 1.0 mwe / 100 m, $\delta ELA/\delta T_{Air}$: 220 m / °C) leads to a simulated retreat as illustrated in Figure 6.10. The retreat of two branches is shown in blue whereas the red line refers to the measured retreat. The step-wise pattern of the simulated retreat results from the reaction time where no change of the terminus occurs. Thus, both branches are taking 5 response times from 1850 till 2000 which stands in contrast to the effective number of steady-states. Regarding the measured cumulative retreat ΔL_t, two nearly-steady-states can be located at t = 1910/20 and t = 1970/80. It is obvious that the simulated retreat of both branches is underestimated, especially until 1980. Considering the overall retreat the difference becomes smaller. Based on this observation and the fact that the simulated mean specific balance $\overline{b}$ = 0.06 mwe between 1850 and 1980 is very small (even when applying upper limits for $\delta b/\delta h$ and $\delta ELA/\delta T_{Air}$) compared with other estimations (e.g. Haeberli et al., 1999a) the applied temperature trend has to be questioned. It may be more suitable to use a temperature trend of the free atmosphere in order to minimize the influence of local effects.

For the southern part of Switzerland a linear warming trend of 1.0 °C in the 20th century is derived from the analysis of past measurements (OcCC, 2002). On a global scale a warming of +0.6 °C is assumed for the time period 1900 - 2000 and +0.86 °C for 1850 - 2000 (IPCC, 2001). Based on these general trends a linear warming of 1.4 °C is applied for further investigations between 1850 - 2000.

6.3.3 Experiments to study the influence of parameter settings

In the following section three experiments are carried out in order to study the influence of the most important variables $\delta b/\delta h$, $\delta ELA/\delta T_{Air}$ and ELA. The size of these parameters is generally unknown. Therefore, it is useful to vary the settings in a reasonable range and in doing so, to study the influence of each parameter while the other parameters remain fixed. The whole process of experimental simulation is well supported by the methods for simulation steering, navigation and visual analysis implemented in VadreX. The results of the three interactively performed experiments

- **e1**: identifying the influence of $\delta ELA/\delta T_{Air}$ = {120, 170, 220} m / °C,
- **e2**: identifying the influence of $\delta b/\delta h$ = {0.5, 0.75, 1.0} mwe / 100 m, and
- **e3**: identifying the influence of ELA = {2700, 2800, 2900} m a.s.l.

are discussed in the following. The possible sizes of the variables used in the accordant experiment are indicated in curly brackets. For all experiments, the reaction time t_r is only considered at the beginning, i.e. for the maximum stand 1850/60 since no distinctive steady-state was observed later on. Additionally, it simplifies the evaluation of different gradients of ΔL_t.

The result of the experiment **e1** is plotted in Figure 6.11 and compared with the measured retreat. In the case of the first adjustment step the reaction time is clearly identifiable. Afterwards the steps are indicated by small periods of a stable position due to the parabolic ease-in/out function applied in the morphing procedure. The most important conclusions are that 1. in every case a linear retreat over time can be observed, 2. when using a high value for $\delta ELA/\delta T_{Air}$, the gradient of ΔL_t and the size of $\overline{b}_1$ increase accordingly and 3. the averaged response time is smaller for high values than for lower ones (34 years in case of 220 m / °C and 37 years in case of 120 m / °C).

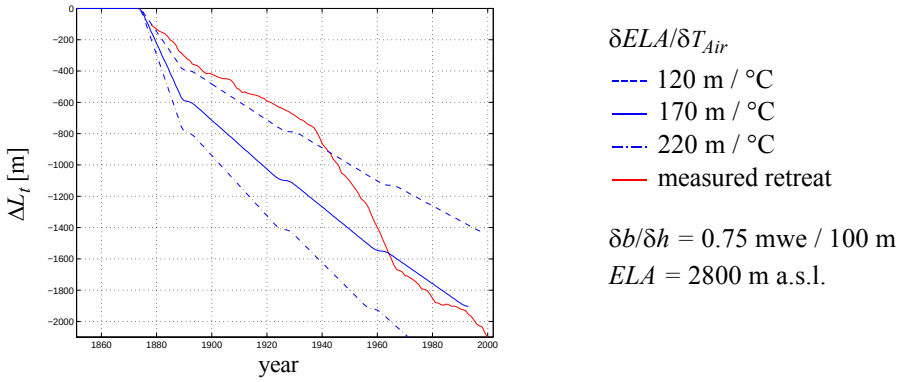


Figure 6.11: Reconstruction of the retreat since 1850 applying different increase of ELA (120, 170 and 220 m) per one degree warming.

In experiment **e2** the influence of the mass balance gradient is evaluated. The corresponding results are shown in Figure 6.12. The most interesting fact is that there is a very

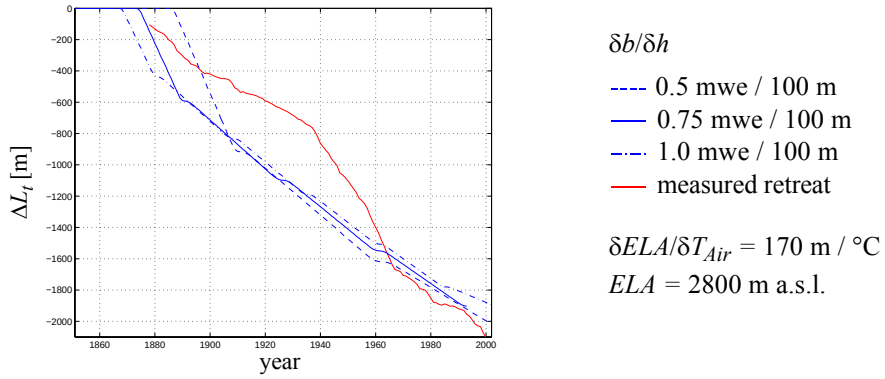


Figure 6.12: Reconstruction of the retreat since 1850 applying different mass balance gradients (0.5, 0.75 and 1.0 m / 100 m). The reaction time t_r is considered only at 1850/60.

small discrepancy in retreat when applying different values of $\delta b/\delta h$. However, a main difference can be identified by considering the response time. An inverse correlation between the average size of t_A and the mass balance is observed ($\delta b/\delta h$ = 0.5 mwe / 100 m: t_A = 55 a; $\delta b/\delta h$ = 0.75 mwe / 100 m: t_A = 36.6 a; $\delta b/\delta h$ = 1.0 mwe / 100 m: t_A = 27.5 a). Thus, when applying a small mass balance gradient the response time becomes larger, leading to a higher ΔT_{Air} and Δb. But the mean specific balance $\overline{b}$ remains constant in the range of 0.2 mwe for all three cases. This explains why $\delta b/\delta h$ has no effect on the retreat when a linear gradient of air temperature is assumed.

In the last experiment **e3**, the impact of *ELA* with regards to ΔL_t is studied. Regarding Figure 6.13 it can be stated that the lower the equilibrium line lies, the bigger is the simulated retreat. This observation can be explained by using Equation 4-11, Equation 4-12 and Equation 4-13 on page 63. It is the ratio between Δb and b_t which determines the size of Δl_t for a given glacier length. A lower *ELA* leads to a smaller ablation b_t at the terminus but also to an increased size of t_A and Δb. In this specific experiment the change of Δb is just a function of ΔT_{Air} and bigger than the change of b_t. The ratio Δb /b_t therefore increases for lower values of ELA.

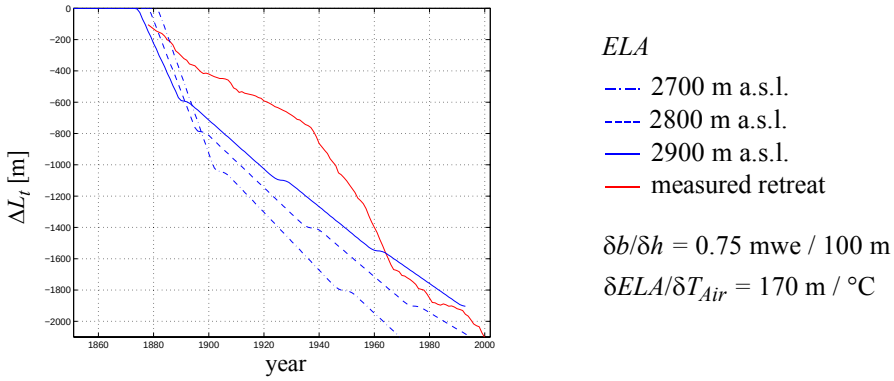


Figure 6.13: Reconstruction of the retreat since 1850 applying different values of ELA (2700, 2800 and 2900 m a.s.l.).

Based on the conclusions drawn from the results of the three experiments it seems to be difficult to simulate a non-linear change of ΔL_t as observed at the Morteratschgletscher using constant values of ΔT_{Air}, $\delta b/\delta h$, $\delta ELA/\delta T_{Air}$, and ΔELA.

6.3.4 Optimization of the simulation parameters

In experiment **e4**, a different approach is tested by adjusting the input variables in such a manner that the measured (ΔL_m) and simulated (ΔL_s) values of ΔL_t agree. In system analysis this procedure is called optimization and is the inverse case of the simulation approach applied in the previous three experiments. The optimization is carried out in two phases:

- reconstruction of temporal behaviour: optimizing the mass balance gradient to achieve observed response time (given predefined *ELA*) and
- reconstruction of spatial behaviour: adjusting the size of $\delta ELA/\delta T_{Air}$ and ΔT_{Air} in order to fit ΔL_s with ΔL_m.

Based on Maisch (1992) the *ELA* is set to 2700 m a.s.l. for the year 1850. To achieve a response time of approximately 60 years (1850 - 1910) the mass balance gradient is experimentally defined to be 0.68 m / 100 m which indicates a smaller mass turnover than assumed by the initial settings. This is a reasonable value since the Upper Engadin is characterised by inner-alpine continental climate. The first step in the second optimization phase is to reconstruct the total retreat by optimizing $\delta ELA/\delta T_{Air}$ which is considered to be constant over this time period. The optimized value of $\delta ELA/\delta T_{Air}$ is 160 m / °C assuming a warming of 1.4 °C between 1850 and 2000. This value is between the standard size of 170 m / °C and an empirically derived size of 150 m / °C (Maisch et al., 1999) which is based on a study on a large set of glaciers in Switzerland. The second step consists of reconstructing the non-linear retreat which is achieved by applying a non-linear temperature trend. The trend of T_{Air} is estimated by adjusting the curve at each quasi-steady-state resulting in the following values: 1850: 0.0 °C, 1900: 0.2 °C, 1970: 0.9 °C and 2000 1.4 °C.

In Figure 6.14 the final result of the iterative process is shown. In the left image the simulated retreat is shown twice: in case of the dash-dotted line the reaction time t_R is set to 0 whereas for the solid line t_R is considered. A good agreement of the temporal behaviour between simulation and reality can be detected since each simulated steady-state corresponds with a slowed retreat phase, i.e. quasi-steady-states (1910 and 1970). Furthermore, the measured retreat lies in-between the two simulated curves. On the right side a comparison of the retreat behaviour between two branches is made. It shows that each branch has an individual behaviour, especially concerning the temporal component.

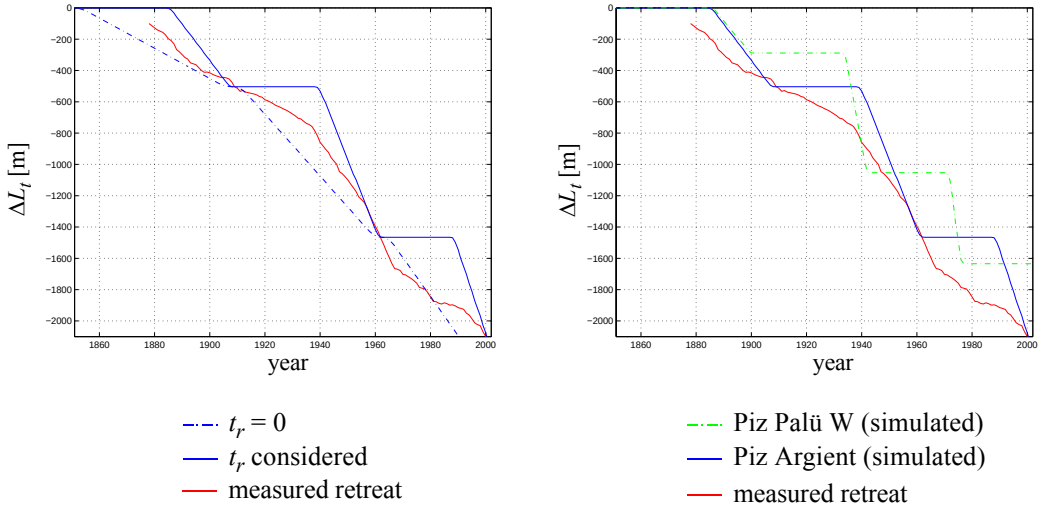


Figure 6.14: Optimized reconstruction of the retreat since 1850 applying experimentally evaluated values for $\delta b/\delta h$ (0.68 mwe / 100 m), $\delta ELA/\delta T_{Air}$ (160 m / °C) and T_{Air} (1850: 0.0 °C, 1900: 0.2 °C, 1970: 0.9 °C, 2000: 1.4 °C). Left: The simulated retreat of the branch Piz Argient considering the reaction time (line) and t_r = 0 (dash-dot). Right: comparison of the retreat behaviour between the branches Piz Argient (blue) and Piz Palü W (green).

In order to validate visually the 3D geometry of the reconstructed glacier surface it is compared with a historical photograph. In Figure 6.15 left a photograph shows the real extent of the Morteratschgletscher in the year 1910. On the right side a perspective view of the simulated counterpart is shown. For the simulation the settings evaluated in experiment e4 are applied. The position of the terminus and the dimensions of the tongue (width and height) are nearly identical.

Figure 6.15: Reconstruction of the extension 1910. Left: photography of the Morteratschgletscher 1910 (© Eidgenössisches Archiv für Denkmalpflege, Hallwylstrasse 15, 3003 Bern, from collection Wehrli). Right: perspective view of the simulated glacier applying the experimentally evaluated values of $\delta b/\delta h$ (0.68 mwe / 100 m), $\delta ELA/\delta T_{Air}$ (160 m / °C) and ΔT_{Air} (+0.29 °C since 1850). Data source: DTM DHM25 © Federal Office of Topography (BA024148).

The four experiments e1- e4, discussed in this and the previous chapter, to reconstruct the retreat of the Morteratschgletscher since 1850 are carried out in the same interactive session. This was possible because of the implemented functionality for the visually supported experimental simulation. The presented reconstructions illustrate the potential of the developed simulation method.

Within this study it is not intended to evaluate the accuracy of the simulation model in all details by performing tests with a large number of different settings, or by comparing and analysing the results with other glaciological information, or by investigating glaciers all over the world. This has to be done by glaciologists using the developed system.

6.4 Simulation of future glacier fluctuations

To simulate and visualize the future evolution of the Morteratschgletscher several assumptions are made. First, possible future scenarios for climate change have to be outlined that are based on predefined emission scenarios and climate models. IPCC (2001) presents a suite of projected future climate changes that illustrate the possibilities that could lie ahead on a global scale. Then, these possible global changes should be downscaled to the regional and local level. The sensitivity analysis performed by Gyalistras (2000) for estimating the future trends of air temperature and precipitation in the Alps and Switzerland lead to controversial results. Nevertheless, some tendencies became apparent such as air temperature averaged over the Alps possibly rising more than on the global scale and precipitation in the Alps seeming to increase more than on a global scale. However, due to the increased uncertainty on small spatial scales, regional and local effects are neglected and just a set of three IPCC scenarios is taken into consideration to illustrate the possible evolution of the Morteratschgletscher in the 21th century. The used scenario en-

semble consists of the scenarios B1, A1B and A1FI (IPCC, 2001) and encompass the whole range of potential air temperature changes.

The impact of the applied climate scenarios on the Morteratschgletscher is visualized in Figure 6.16. The optimized values evaluated in experiment e4 are used. For each scenario three states are shown with a time interval corresponding to the current response time of the glacier. The most obvious changes can be recognized at the tongue whereas the accumulation area remains more or less identical. The date when the two main branches Persgletscher and Morteratschgletscher will be separated varies approximately between 2040 (A1FI) and 2070 (B1). In the year 2100 each branch formerly contributing to the Morteratschgletscher might become isolated when assuming an increase of the air temperature of more than 4 °C (scenario A1FI). To compare visually the simulated results between two different scenarios, dynamically captured textures of the corresponding A1B extent are blended when experimenting with the scenario A1FI (see blue texture layer in the right images of Figure 6.16).

In contrast to the simulated behaviour, a different retreat could result when the terminus of the Morteratschgletscher reaches the zone where an over-deepening of the glacier bed is assumed. The glacier would build a calving front due to the potential small lake with possibly accelerated retreat.

A comparison of the simulated future change of length ΔL_s, area ΔA_{ortho} and volume ΔV_{ortho} is presented in Table 6.4. The changes of the area over time are smaller than the

Table 6.4: Comparison of simulated retreat ΔL_s, area A_{ortho} and volume V_{ortho} for the time period 1985 - 2098. Reference values (1985): L (Piz Argient) = 7596 m, A_{ortho}= 12.74 km^2, V_{ortho}= 0.68 km^3.

Year	B1			A1B			A1FI		
	ΔL_s [m]	ΔA_{ortho} [km^2]	ΔV_{ortho} [km^3]	ΔL_s [m]	ΔA_{ortho} [km^2]	ΔV_{ortho} [km^3]	ΔL_s [m]	ΔA_{ortho} [km^2]	ΔV_{ortho} [km^3]
2030	-1049 (-14%)	-1.08 (-8%)	-0.12 (-18%)	-1157 (-15%)	-1.15 (-9%)	-0.13 (-19%)	-1157 (-15%)	-1.15 (-9%)	-0.13 (-19%)
2066/ 2067	-1933 (-25%)	-1.61 (-13%)	-0.20 (-29%)	-2609 (-34%)	-2.47 (-19%)	-0.29 (-43%)	-3364 (-44%)	-3.75 (-29%)	-0.42 (-62%)
2091/ 2098	-2299 (-30%)	-1.76 (-14%)	-0.21 (-31%)	-3114 (-41%)	-3.22 (-25%)	-0.37 (-54%)	-4176 (-55%)	-5.34 (-42%)	-0.52 (-77%)

volume changes and retreat regardless of the applied scenario. Comparing the result of the two scenarios B1 and A1FI it can be concluded that the retreat is nearly 2 times, the change of the area 3 times, and the change of the volume 2.5 times bigger when applying scenario A1FI.

Instead of a positive change of the air temperature a cooling scenario could also be simulated. But due to the very low probability no such experiment is presented.

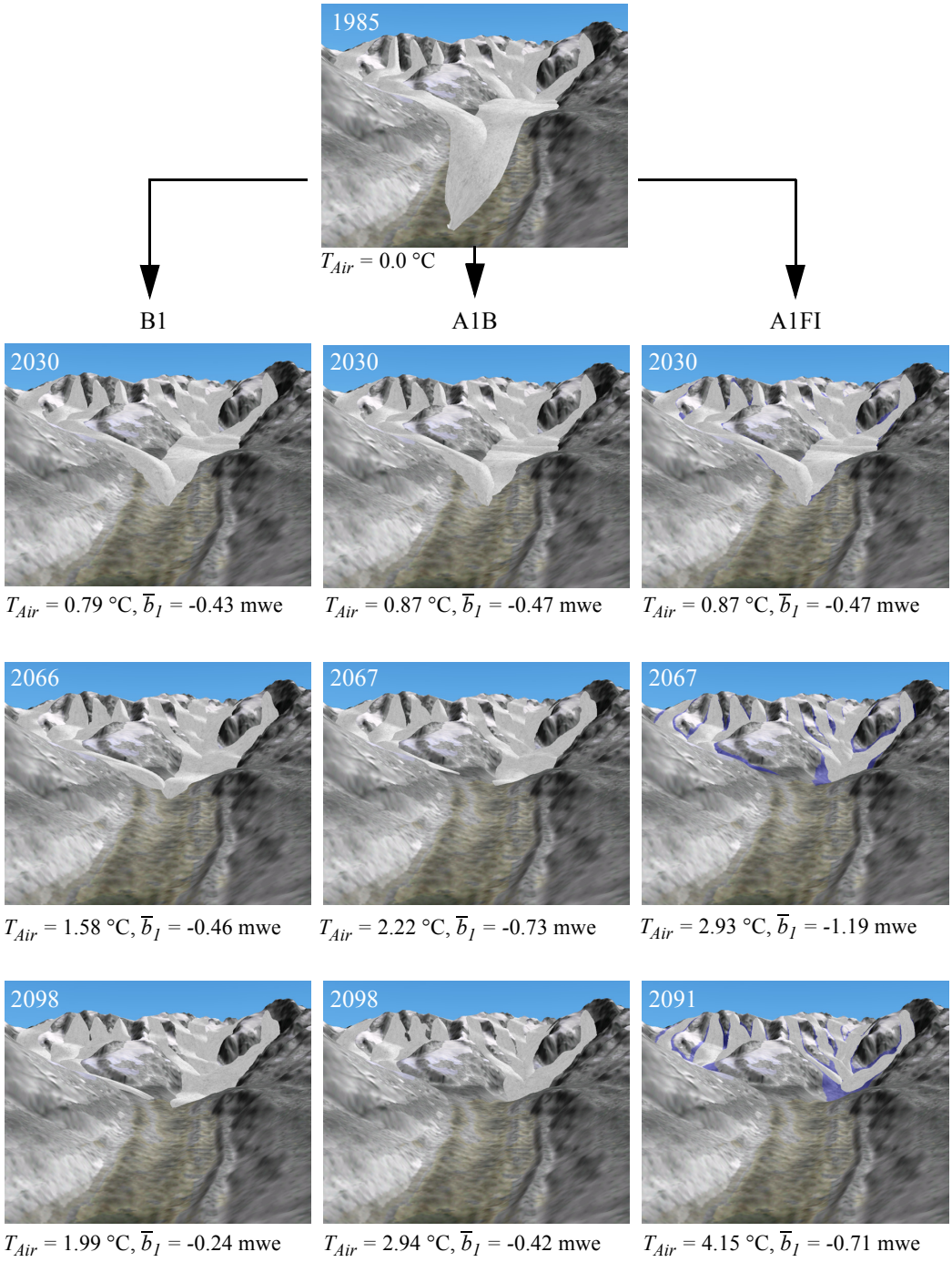


Figure 6.16: Simulated fluctuations of the Morteratschgletscher over hundred years assuming the IPCC climate scenarios B1, A1B and A1FI.

For simulation of the glacier fluctuations all ice-flow models including the developed model assume that the glacier is able to adjust its geometry dynamically to the changed atmospheric conditions by moving forward or backward. For projection of the future evolution this assumption may be wrong. When the air temperature rises very fast the glacier reacts no longer by changing its length but by changing primarily its vertical dimension. In this case a mass balance model would describe the process of down-wasting in a more appropriate way.

6.5 Aspects of the visual representation

Aside from the simulation capabilities a few visual aspects are discussed in the last part of the case study. To illustrate the task-oriented change of the appearance and LOA two views of the Morteratsch valley are presented in Figure 6.17. On the left side the user is interested in analysing the terrain including the glacier bed. For this, the user is able to interactively modify the LOA of the terrain appearance and to apply a previously generated topographic texture. In Figure 6.17 right the Morteratschgletscher is visualized using a reduced LOA to achieve a more geotypical appearance. Doing so, the user is changing the focus of exploration towards the glacier surface and its fluctuations. The LOA has to be similar for both objects, the environment and the glacier. The LOA applied to the terrain is therefore lowered by applying a modified geotexture. This texture is generated accord-

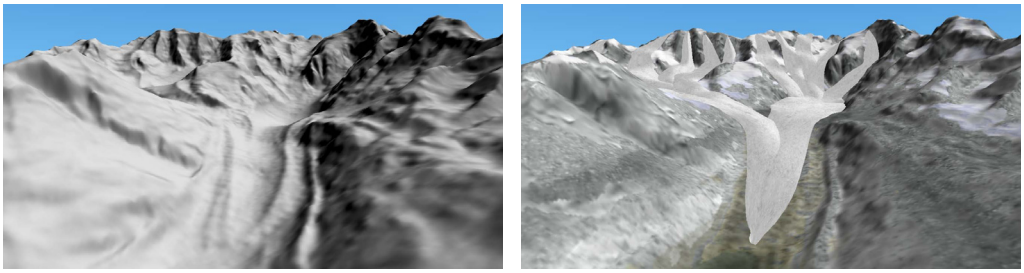


Figure 6.17: Perspective view of the simulated bed and surface of the Morteratschgletscher in the year 1985. Left: reconstructed glacier bed with topographically textured terrain. Right: the 11 branch surfaces, Gouraud shaded and with geotypical appearance.

ing to the approach presented in Chapter 4.4.3 on page 71 using the freely available software T2 (2003). A critical point in lowering the LOA is that although the glacier may change due to warming or cooling, the user also expects a change of the terrain appearance since the snow level is changing. Thus, he may be irritated by the unaltered terrain texture.

For more specific analysis of the glacier bed and glacier volume, branch-specific cross-sections can be selected by pointing to a position on a branch surface. The left image in Figure 6.18 shows the interactively picked cross-section of the branch ‘Crast Agüzza’ at the selected position. It points out the well-formed parabolic shape and the horizon-

tal surface. Due to the properties of the applied NURBS surfaces a cross-section is defined efficiently by just 3 points: the central point and the two side points on the terrain. On the right side a cross-section of the branch ‘Piz Palü W’ taken at another position il-

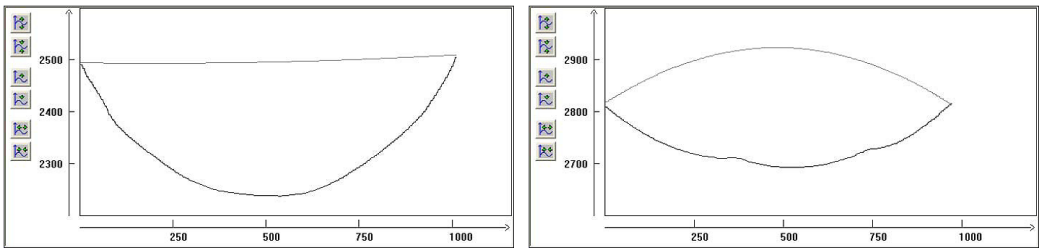


Figure 6.18: Interactively picked cross-sections of the branch ‘Crast Agüzza’ (left) and ‘Piz Palü W’ (right). The x- and y-axis refer to w_x and h_x, respectively.

lustrates the effect of combining the volume of several branches. Here, three branches contribute to the final shape of the bed as is clearly recognisable by the two ridges on the bed. Since the width of the bed is larger than the width of the branch, the resulting branch surface is highly convex (in contrast to the example in Figure 6.18 left).

6.5.1 Specific features of the branch shape

Regarding the shape of the terminus three different aspects can be pointed out. In Figure 6.19 left the parabolic terminus of the front branch (in this case ‘Piz Bianco’) can be clearly recognised. Its dimension is varied according to the state of evolution by modify-

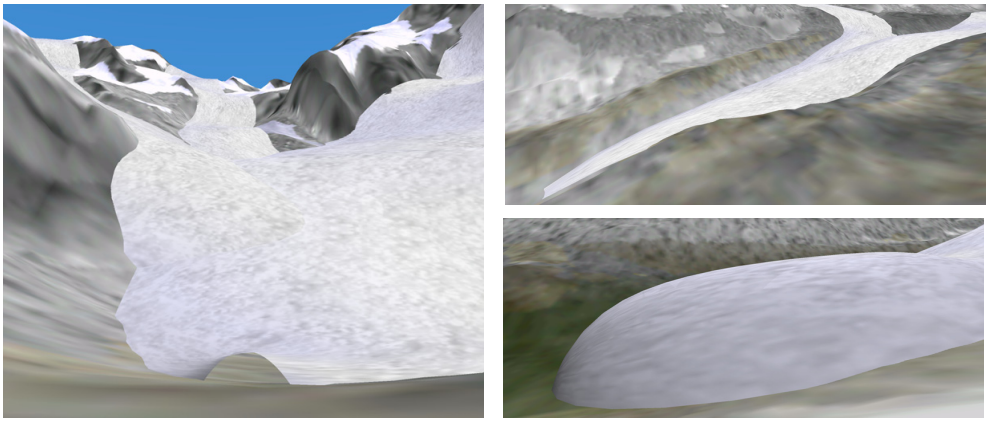


Figure 6.19: Modelling of the glacier terminus. Left: parabolic terminus, right top: typical flat front during glacier retreat, right bottom: steep front indicating an advance.

ing the height of the snout. If the glacier is advancing the height of the snout is set to zero whereas in the inverse case the height is set to a predefined value. Since in this example a noise texture with a repeating pattern is applied for branch texturing it is almost impossible to separate the branches in the confluence zone. The images on the right side illustrate typical front shapes during a retreat (top) or advance (bottom). The steep front is geometrically well modelled by applying NURBS surfaces.

Another specific feature is the visual appearance of several branches which meet to form a bigger united branch. An example is presented in Figure 6.20 showing five branches meeting viewed from below (left) and from above (right). Due to independent rendering of the branches and the enabled Z-buffering just the closest part of a branch surface is visible. Surface patches that are below other branch surfaces are rendered but do not pass the Z-buffer test. This fact can be easily recognized in the left image where an interactively selected branch is highlighted by changing the ambient colour. Additionally, it can be detected visually which branch shows the largest vertical dimension at a specific location. Although each branch is based on a distinctive flow-line all of them have a similar thickness (see also right image). Regarding the right image one can observe that the branches meet in a fluid-like manner.

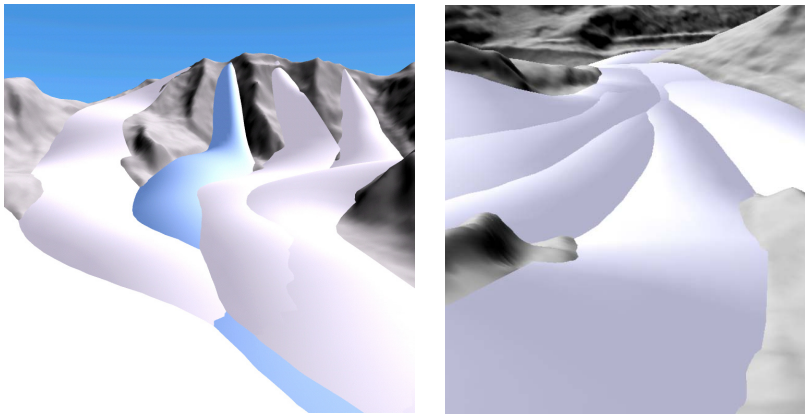


Figure 6.20: Confluence and superposition of several branches studied from below (left) and above (right). In the left image an interactively picked branch is indicated by the blue ambient colour.

Considering an animated sequence of dynamic glacier changes, the method developed to describe continuously the transition between two static steady-states leads to a smooth morphing along the terrain without any ‘popping’ artifacts. Due to slow motion with nearly constant velocity and laminar flow behaviour no problems arise from applying a geometrically-based approach. Furthermore, the animated branch texture imitates the ice flow with low computation costs. The influence of the exponential morphing factor depends on the number of used control points n_j. When n_i is too small, i.e. smaller than approximately 15 depending on the size of L, a poor spatial sampling in longitudinal direction results and thus, the dynamic swelling is modelled insufficiently.

6.5.2 Visual quality of the branch surface

The visual quality of a branch representation depends heavily on the subdivision method applied and the flatness criteria. In order to evaluate the most suitable and efficient method a uniform and two adaptive subdivision methods are compared visually. The uniform criteria is defined by the number of sample points per unit length whereas with the adaptive approach, the edge length in screen space and the maximum distance in screen space are applied (OpenGL ARB, 1999). The Windows distribution of OpenGL includes all functionality of the GLU library version 1.2 and thus, no subdivision schemes based on object space metrics are included.

As the test-site the transition zone between the steep slopes at 'Piz Palü' and the flatter region just below is chosen (see Figure 6.21) in order to assess the grade of the Sur-

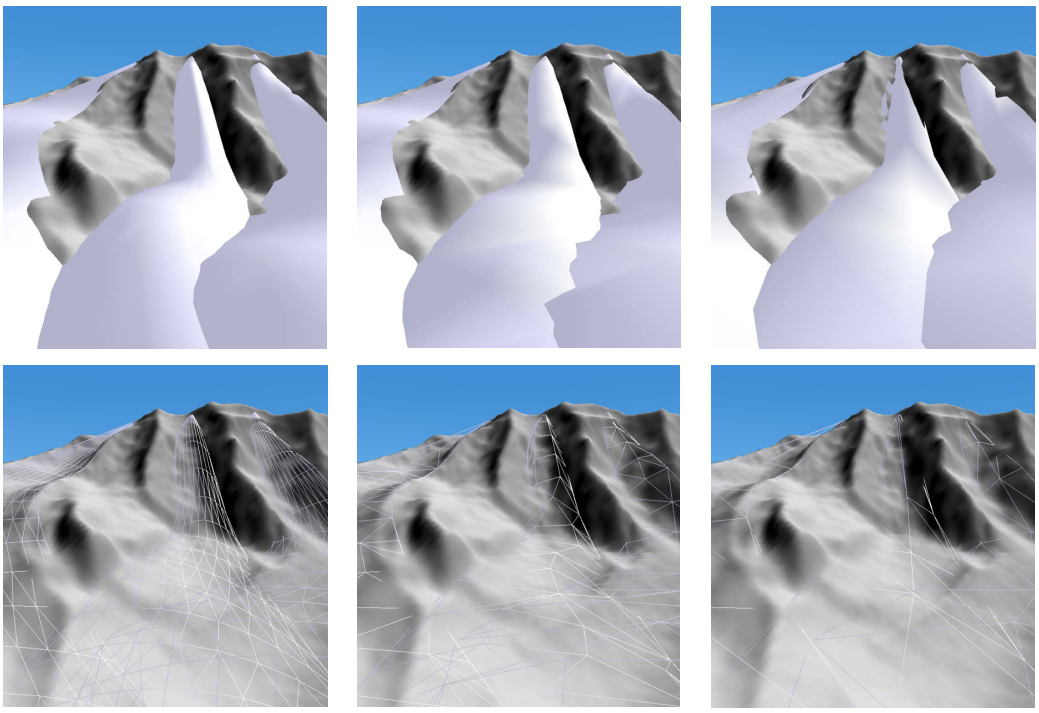


Figure 6.21: Visual evaluation of different subdivision schemes concerning the quality of STI, SSI and surface shading: sample points in unit length (left column), length of the polygon edges (centre column) and distance between tessellated polygons and surface (right column). Bottom row: the corresponding surfaces rendered as wire-frame.

face/Terrain Intersection (STI), the smoothness of the Surface/Surface Intersection (SSI) and the quality of shading. The rendering costs of the given examples are similar. Because of the efficient surface tessellation using the uniform subdivision scheme, a much higher number of vertices can be used than in the other two cases (see bottom row). This is the

main reason why the uniform approach leads to visually better results concerning STI, SSI and shading quality. The adaptive approach presented in the centre column (maximum length of polygon edges) lacks in the visual appearance of the SSI and gives poor quality of the Gouraud shading which depends on the geometrical representation of the surface. The major problem with the example in the right column can be identified on the steep slopes where the low mesh resolution leads to easily recognizable artifacts in case of STI.

Together with the subdivision scheme, the number of control points n_i used for the surface fitting influences the representation of a branch on the terrain. In Figure 6.22 the upper part of the branch surface 'Piz Argient' lying on rugged terrain is visualized four times using different sizes of n_i but identical subdivision settings. Additionally, the visible parts of the corresponding control net of the NURBS surface are shown as grey lines. In

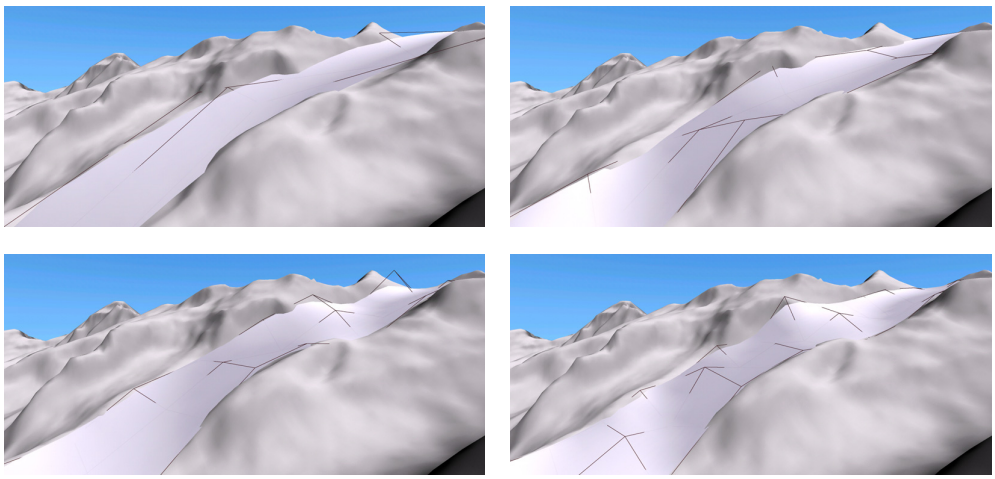


Figure 6.22: Influence of n_i on surface-terrain alignment. Top right: n_i = 10 (*fps* = 19), top left: n_i = 20 (*fps* = 13), bottom left: n_i = 30 (*fps* = 9.3), bottom right: n_i = 50 (*fps* = 6.3).

the first image (top left) just 10 control points are used for the fitting in the longitudinal direction. Visual gaps between terrain and surface are recognizable. Increasing the size of n_i ($n_i \in [4, n_x - 2]$) leads to a gradually better alignment of the surface as illustrated in the other three images. However, to achieve a more detailed shape, higher rendering costs have to be accepted. Moreover, the optimal size of n_i depends not only on the size of the branch (see Figure 4.9 on page 60) but also on the roughness of the terrain.

6.5.3 Rendering performance

All performance tests were carried out on a standard Laptop (Dell Inspiron 8000) with a 1.0 GHz processor and 256 MByte RAM. A Nvidia GeForce2Go graphic chip is used with 16 MByte memory. As software WindowsXP is combined with a display driver of

Nvidia (version 44.25), OpenGL 1.1 and GLU 1.2. The window size is 800 x 600 pixels with 32 Bit colour depth. No anti-aliasing is applied. An average number of n_{tris} = 15100 is used for the rendering of the terrain.

In Figure 6.23 left the achieved frame rate is plotted during the exploration of different numbers of branches. The uniform subdivision method is applied with 10 and 4 sample points taken for the sampling of the NURBS surfaces in longitudinal and transverse directions, respectively. No significant difference can be recognized when comparing the performance of rendering no branches (n_b = 0) and one branch (n_b = 1), i.e. the rendering time for one branch is smaller than the idle time of the graphics pipeline when just rendering the terrain. When increasing the size of n_b the performance lowers accordingly to a minimum of less than 20 fps. If not all of the branches are within the viewing frustum, intermediate frame rates are observed.

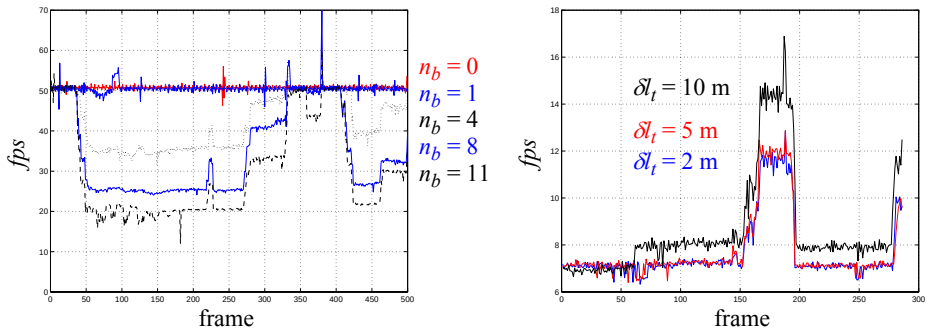


Figure 6.23: Achieved rendering performance by varying the number of branches (left) and the length δl_t applied for one morphing step (right).

The size of the terminus change δl_t per interpolation step influences the number of branches to be animated per frame (*bpf*) and thus the rendering performance. To illustrate the relation between δl_t and *fps* measured values of *fps* are presented in Figure 6.23 right. Due to the increased tessellation quality and n_b = 11 the overall performance is lower than in the left image. The effect of increasing δl_t is smaller than expected: just 1 - 3 fps between δl_t = 2 m and δl_t = 10 m. The reason may be the very small morphing costs since the morphing procedure takes less then 10 milliseconds per branch (with n_i = 20).

The time required for branch deformation depends on the number of cross-sections n_x (linearly) and the surface approximation method. Using the global least squared surface approximation with n_x = 40 and n_i = 20 the deformation of one branch requires 440 milliseconds on average.

6.5.4 Visual analysis of glacier fluctuations

Two examples illustrate the use of analysis functionality provided by VadreX for the visual study and comparison of simulated glacier changes in 3D. The first example shows a perspective view of the simulated tongue of the Morteratschgletscher during a reconstruction experiment at time t = 1945 (see Figure 6.24). Here, the maximum stand 1850/60 is

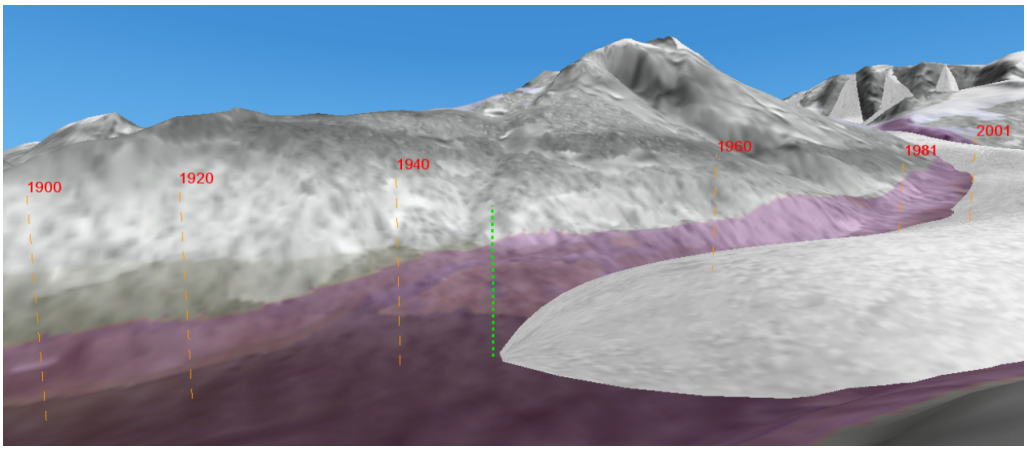


Figure 6.24: Visual analysis of the animated result of a reconstruction experiment (see text for further explanations).

added as semi-transparent violet texture layer and allows to easily carry out a spatial analysis of the retreat. The dynamic correlation between the simulated (ΔL_s) and measured (ΔL_m) retreat is performed by adding static glyphs which represent the measured retreat and a dynamic glyph in green indicating the measured location of the terminus at the current simulation time.

The second example (see Figure 6.25) shows the combined use of dynamic capturing and the measuring bar to evaluate the future evolution of the volume of the Morteratschgletscher. The investigation of the volume changes is focused on the position where the bed shows over-deepening because a maximum loss of ice and reduction in the vertical dimension is expected. A sequence of four images documents the decrease of the Morteratschgletscher from 1985 to 2098, assuming a warming of 2.94 °C (IPCC scenario A1B). The measuring bar is positioned at the deepest location and remains fixed during the whole experiment. In contrast to the visual extraction of the thickness using the measuring bar, the procedurally derived size of the vertical branch dimension is neither affected by the subdivision scheme nor the tessellation quality. It is given in the centre just below the corresponding image. The simulated glacier extension of 1985 is represented as a blue texture layer that is generated and blended dynamically using dynamic capturing. One can recognize that the simulated thickness at the measuring bar will decrease by more than 80% whereas the changes in width and especially in length (41%, see Table 6.4 on page 122) are less dramatic.

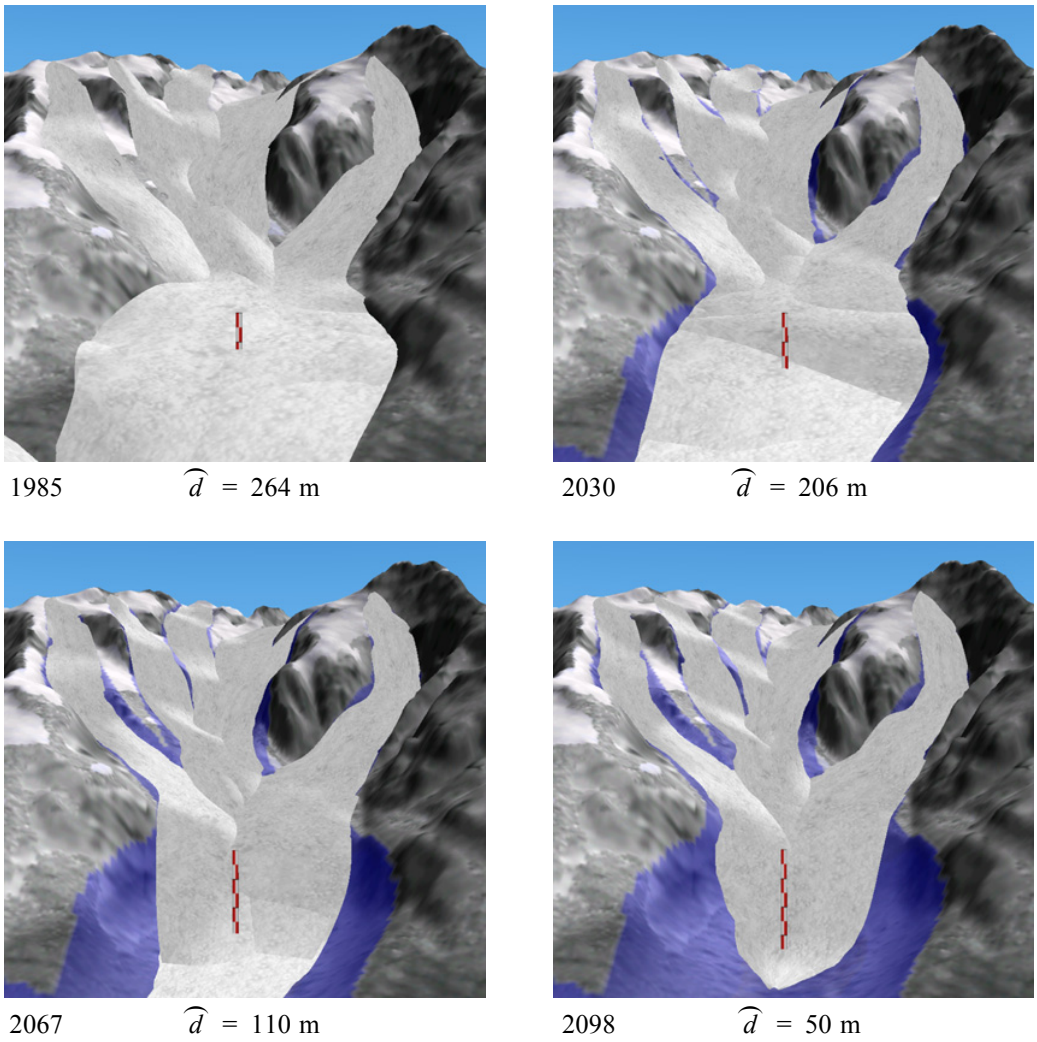


Figure 6.25: Visual analysis of the simulated volume changes of the Morteratschgletscher from 1985 to 2098. Applied climate scenario: A1B. Height of the measuring bar: 400 metres. During the whole experiment the position of the measuring bar remains fixed.

Finally, it is important to realise that all simulations of the future evolution presented in Chapter 6 represent a possible behaviour of the Morteratschgletscher. They do not contain any predictive character since too many processes and parameters are unknown. The simulations are based on simplified assumptions about the global population and economic growth (emission scenarios), the climate system (climate models), the effects on a regional scale (down-scaling), the response of the glacier (glacier model) and the visual representation of the simulated glacier fluctuations.

Chapter 7

Conclusions and perspectives

7.1 Conclusions

Based on the case study carried out in Chapter 6, conclusions can be drawn referring to the developed simulation method, the visual representation of a mountain glacier and the visual system VadreX, including functionality for the visual analysis and experimental simulation.

7.1.1 Discussion of developed methods

The developed method for the simulation of glacier fluctuations describes a glacier as a combination of several branches. Although only a limited amount of physics was applied to generate a 3D image of the glacier bed, the presented comparison with reference data of the Morteratschgletscher is promising. The simulated bed shows an over-deepening where a maximum thickness of 270 ± 20 metres is assumed. Improvements are possible at the accumulation area, due to the underestimation of this zone when using the developed branch-based approach. An evaluation of the bed was not possible since no in-situ measurements are yet available.

By combining the experiential knowledge derived from the three experiments e1 - e3 carried out in the case study, it is possible to reconstruct the historic retreat of the Morteratschgletscher since 1850 in a step-wise manner. First, the mass balance gradient is adjusted to achieve the observed response times (temporal behaviour). In a second step, the spatial behaviour is reconstructed by optimizing $\delta ELA/\delta T_{Air}$ and the trend of the air temperature. However, further investigations are required to verify wether this step-wise optimisation approach is appropriate to reconstruct glacier and climate parameters from measured glacier retreats.

For simulations of the future evolution of the Morteratschgletscher, a step-wise change of the air temperature and the ELA is assumed that incorporates changes of humidity, radiance and albedo. The specific hypsography of the glacier is not considered, al-

though an influence on the change of mass balance has been observed. For a reliable assessment of the simulation quality, further tests with other glaciers and comparisons with the results of other glacier models would be indispensable. Additionally, systematic observations of glaciers have to be continued and extended to provide glaciological data for validation purposes.

The simulated glacier is represented visually by a combination of several NURBS surfaces, each referring to an individual branch. The NURBS surfaces used are very well adapted for visualizing the parabolic and smooth shape of a branch, and their rendering is explicitly supported by OpenGL. Good results were achieved by using animated textures to visually describe the flow of ice at the branch surface. A procedural approach is used for morphing between two steady-state surfaces. Since the transition of the NURBS is performed by moving the control points of the parametric form along the terrain, a consistent surface deformation with low computing costs is ensured. The quality of the non-linear morphing of specific parts of the surface depends on the number of control points in the longitudinal direction.

The branch surfaces are rendered in immediate mode. Thus, the performance depends heavily on the tessellation costs that are a function of the complexity of the surface and on the subdivision method applied and required accuracy. The most suitable approach proved to be a uniform sampling of the surface in the parametric space due to lower computational costs. High visual quality of the glacier surface results, combined with acceptable frame rates. Shading of the branch surfaces is indispensable to clear perception of the shape.

The use of NURBS surfaces for continuous description of glacier volumes has been proven as an efficient approach that implicitly ensures a smooth shape by minimizing the elastic energy. There are three reasons why this surface-based approach is preferred over classical viscous fluid models or finite element models. First, only very little glaciological information and a DTM are required to simulate glacier fluctuations. Second, this approach uses the available slope information of the real glacier surface to estimate the basal shear stress and thickness. Third, only this approach provides the possibility to estimate a glacier bed without field measurements and thus is able to be applied for a large set of glaciers.

Limitations exist on the use of the simulation method. It assumes steady-state conditions and temperate ice, does not consider any dynamic effects, and is applicable only in the time scale of the response time of the investigated glacier. Since knowledge of the glacier surface is required, paleo-information must be available for reconstruction of glacier surfaces in the past. Additionally, if the simulated glacier consists just of a small number of branches, the estimated hypsography of the glacier can differ significantly from reality due to the too small accumulation area.

7.1.2 The visual system VadreX

The concept of the experimental simulation of mountain glacier fluctuations in a visual environment is successfully implemented in the prototype system VadreX. Using the functionality of VadreX, it was possible to conduct several experiments in order to reconstruct

the 3D retreat of the Morteratschgletscher from 1850 until present, and to project possible future scenarios for the glacier's evolution.

The implemented system VadreX is built on top of several lower-level software layers. The integration of new methods was only possible because each layer provides an open and extensible interface. Access to low-level rendering methods is fundamental and necessary. Use of the open source software provided by Discoe (2004) made it possible to focus the effort on the implementation of new ideas and methods. Another advantage is that VadreX does not require a high-end graphic system and even runs on a Laptop, enabling to work in the field.

The navigation metaphors defined in this study are suitable to interactive study of glacier fluctuations. The selection of a navigation style depends on the intended task. Therefore, it is necessary to provide several ways to change viewing position and direction for an efficient and non-confusing navigation. The locked-view metaphor is the most suitable navigation mode when observing dynamic changes of the glacier surface because, otherwise, the absolute changes would result in changes relative to the observer that are difficult to interpret.

A set of different approaches to analyse the simulations is implemented in VadreX. It has been proven that only the combined use of several analysis tools provides an appropriate image of the simulations. Although it is possible to show a major part of the whole complexity of glacier fluctuations in a single view, it is nearly impossible to analyse the simulations in detail. Therefore, approaches for the task-dependent variation of information complexity, e.g. changing LOA, scene complexity or dimensionality, are fundamental for the investigation of simulations in a virtual reality system.

In order to validate the simulated results, the implemented visual methods as dynamic capturing, interactive profiling and multi-texturing were successfully applied in the case study. Recording the most important parameters of the conducted experiments in an experiment repository is helpful for post-analysis of the computed results.

Although the system is intended for different user-groups (scientists, students and non-experts) no special adjustments were applied in order to provide a user-specific system behaviour and GUI. Practical tests would have to show if any simplifications made would be appropriate. Based on first experiences made in collaboration with under-graduate students one can conclude that basic instructions concerning navigation and simulation steering are mandatory.

7.2 Perspectives

7.2.1 Improving the simulation and rendering

Further experiments with other glaciers are required to gain more experience for more reliable statements about the quality of the simulation model. Comparisons with other glacier models are indispensable to estimate the model's skill of projecting future fluctuations. For simulating an accelerated decomposition of a mountain glacier given increased warming, a mass balance model should be included. The hypsography has to be integrated

into the simulation in order to consider local circumstances and incorporate varying area-elevation distribution over time.

The use of Hierarchical B-Splines or the more general form of pasting spline surfaces (Barghiel et al., 1999) could improve the visual representation, since they allow addition of areas in local detail, e.g. terminus or tongue, to a tensor product B-Spline surface. If a more realistic appearance is desired, e.g. for presentation purposes, it could be achieved by 1. applying bump-mapping for visually simulating the surface roughness, 2. considering different flow behaviours on the surface using a more sophisticated texture mapping, 3. modelling crevasses in cases of extending flow using procedural textures and 4. incorporating BRDF measurements for highly realistic shading and reflection effects.

The glacier is modelled visually as a surface. To get a volumetric description of the glacier the use of implicit surfaces would be a more appropriate approach.

Aside from visualizing dynamic glacier changes, it could be required to represent visually the change in the environment accordingly. To describe these changes, precomputed geotypical terrain textures could be combined dynamically using appropriate blending operations. A procedural texturing method that combines given temperature and precipitation trends with basic assumptions on vegetation, soil and snow distribution would be a more time-consuming approach.

7.2.2 Development of a generic visual simulation system

Improvements are required in the field of information management. A more generic approach has to be developed that accounts for the intended task and displays only as much information as required. It should be possible to dynamically configure the features of interest that are to be plotted or sent to the experiment repository.

To enhance the performance and independence of the simulation and the rendering components, they could be separated into asynchronous threads and run on different CPU's.

Further technical effort and promotion activities would be required to improve system's user-friendliness and to make the developed system available to the public.

A main attribute of all geoscientific simulation models is their simplification of complex systems characterised by their open boundary. Due to this abstraction gaps between observed and simulated processes are common. Thus, an explicit visual representation of the uncertainties would contribute to a better evaluation to simulation results although it would be ambitious to try to quantify all the introduced errors.

Further investigations have to be done to allow creative amalgamation of low level functionality into a more complex model, since modifications of the simulation model still need additional code development in VadreX.

Since this study was focused on the development of a visual system for a particular purpose, i.e. simulating and studying glacier fluctuations, VadreX was designed as a closed system. A general approach that allows coupling any geoscientific simulation to a visual system claims for an open system. It may consists of a high-level software layer at the top of three interoperable subsystems, i.e. a generic computer graphics system, interoperable simulation and GIS components for geospatial analysis. The most important

and challenging tasks in the development of such a generic geoscientific simulation and visualization system are identified in two areas. Within the methodological scope, the definition of 1. an ontology of simulated geoprocesses and 2. concepts for the automatic generation of a visual representation of any geoprocess based on the previously developed ontology are of primary interest. Software-related aspects are 1. specification of the interfaces provided by the high-level layer and between the subsystems, as they define the extensibility and interoperability of the system, 2. coherent management of all dimensions (space and time) on varying scales and 3. efficient and persistent data storage.

Appendix

Data sources

A.1 Digital terrain model

DHM25: © Swiss Federal Institute of Topography (DHM25 © Bundesamt für Landestopographie (BA024148)) (Swiss Federal Office of Topography, 2001). The DHM25 was generated from vectorized contour lines, vectorized lake perimeters and digitized spot heights of the topographic map 1:25'000. The basic vector model was interpolated to a grid of 25 metres resolution. The height values have a resolution of 0.1 metres. A subset of the data set has been used for the case study in the Upper Engadin. This part of the DHM25 was updated in 1985 (point measurements) and 1991 (linear features).

A.2 Geographical data used for texturing purposes

Enhanced Satellite Map of Switzerland: © ESA 1990 - 1994 / Eurimage, © CNES 1992 - 1999 / Spot Image, Swiss Federal Institute of Topography, NPOC. The spatial resolution of the Thematic Mapper (TM) based Satellite Map of Switzerland (30 x 30 m^2) was enhanced by merging the SPOT HRV data (mainly representing the spatial information) with the low-frequency data of the TM images, providing spectral (or colour) information (Graf, 1995). The resulting mosaic covers Switzerland at a spatial resolution of 10 x 10 m^2.

Pixel maps PK50 and PK200: © Swiss Federal Institute of Topography (DHM25 © Bundesamt für Landestopographie (BA024148)) (Swiss Federal Office of Topography, 2000). The pixel maps PK50 (scale 1:50'000) and PK200 (scale 1:200'000) of the Swiss Federal Office of Topography (2000) are a rasterized representation of the corresponding analog topographic maps.

Bibliography

Acevedo D., Vote E., Laidlaw D.H. and Joukowski M.S.: Archaeological data visualization in VR: analysis of lamp finds at the Great Temple of Petra, a case study. *Proc. IEEE Visualization '01*, 493-496, 2001.

Andrey J. and Mortsch L.: Communicating about climate change: challenges and opportunities. *Proc. of the International Climate Change Communication Conference*, Welcome Plenary, 1-11, 2000.

Bar-Zeev A.: Scenegraphs: past, present and future. *RealityPrime*, white paper, http://www.realityprime.com, 15 pages, access: November 11, 2003.

Baraff D. and Witkin A.: Dynamic simulation of non-penetrating flexible bodies. *ACM Computer Graphics*, **26**(2):303-308, 1992.

Barghiel C., Bartels R. and Forsey D.: Pasting spline surfaces. In: Mathematical Models for Curves and Surfaces. *Vanderbilt University Press*, 31-40, 1999.

Barnhill R.E.: Surface/surface intersection. *Computer Aided Geometric Design*, **4**(1/2):3-16, 1987.

Barr A.H.: Global and local deformations of solid primitives. *ACM Computer Graphics*, **18**(3):21-30, 1984.

Baumann K., Döllner J., Hinrichs K. and Kersting O.: A hybrid, hierarchical data structure for real-time terrain visualization. *Proc. Computer Graphics International 1999*, 85-92, 1999.

Becker L., Bernard L. and Döllner J.: Integration dynamischer Atmosphärenmodelle mit einem (3+1)-dimensionalen objektorientierten GIS-Kern. *Proc. Umweltinformatik '99*, 429-442, 1999.

Beeler F.: Geomorphologische Untersuchungen am Spät- und Postglazial im Schweizerischen Nationalpark und im Berninagebiet (Südrätische Alpen). Ergebnisse wiss. Unters. Schweiz. Nationalpark, **XV**(77), 276 pages, 1977.

Belleman R.G., Kaandorp J.A., Diykman D. and Sloot P.M.A.: GEOPROVE: Geometric probes for virtual environments. *Proc. 5th annual conference of the Advanced School for Computing and Imaging ASCI*, 15-17, 1999.

Belleman R.G. and Sloot P.M.A.: The design of dynamic exploration environments for computational steering simulations. *Proc. of the SGI User' Conference 2000*, 57-74, 2000.

Beniston M., Haeberli W., Hoelzle M. and Taylor A.: On the potential use of glacier and permafrost observations for verification of climate models. *Annals of Glaciology*, **25**:400-406, 1997.

Bernard L., Schmidt B., Streit U. and Uhlenküken C.: Managing, modeling and visualizing high-dimensional spatio-temporal data in an integrated system. *Geoinformatica*, **2**(1):59-77, 1998.

Bernard L. and Krüger T.: Integration of GIS and spatio-temporal simulation models: Interoperable components for different simulation strategies. *Transactions in GIS*, 4(3):197-215, 2000.

Bishop I.: The role of visual realism in communicating and understanding spatial change and process. In: Visualization in Geographic Information Systems, edited by Hearnshaw H.M and Unwin D.J., *Wiley Publishers*, 60-64, 1994.

Bloomenthal J.: Introduction to implicit surfaces. *Morgan Kaufmann Publishers*, 332 pages, 1997.

Bray D.: Speaking truth to power revisited: Science, policy and climate change. *Proc. of the international Climate Change Communication Conference*, Session A1, 1-15, 2000.

Bryson S.T and Johan S.: Time management, simultaneity and time-critical computation in interactive unsteady visualization environments. *IEEE Visualization '96*, 255-262, 1996.

Burns D. and Osfield R.: OpenSceneGraph. http://www.openscenegraph.org. Last access: 2 February, 2004.

Cani M.: Implicit representations in computer animation: a compared study. *Proc. Implicit Surface '99*, 8 pages, 1999.

Cani-Gascuel M. and Desbrun M.: Animation of deformable models using implicit surfaces. *IEEE Transactions on Visualization and Computer Graphics*, **3**(1):39-50, 1997.

Carlson D.A. and Hodgins J.K.: Simulation levels of detail for real-time animation. *Proc. of Graphics Interface '97*, 1-8, 1997.

Cerezo E., Pina A. and Seron J.: Motion and behaviour modelling: state of the art and new trends. *The Visual Computer*, **15**(3):124-146, 1999.

Chiba N., Muraoka K. and Fujita K.: An erosion model based on velocity fields for the visual simulation of mountain scenery. *The Journal of Visualization and Computer Animation*, **9**(4):185-194, 1998.

Cline D. and Egbert P.K.: Terrain decimation through quadtree morphing. *IEEE Transactions on Visualization and Computer Graphics*, **7**(1):62-69, 2001.

Dana K.J., van Ginneken B., Nayar S.K. and Koenderink J.J.: Reflectance and texture of real-world surfaces. *ACM Transactions on Computer Graphics*, **18**(1): 1-34, 1999.

Discoe B.: The Virtual Terrain Project: http://www.vterrain.org. Release 02-09-11. Last access: February 12, 2004.

Döllner J. and Hinrichs K.: Object-oriented 3D modelling, animation and interaction. *The Journal of Visualization and Computer Animation*, **8**(2):33-64, 1997.

Döllner J., Kersting O., Hinrichs K. and Baumann K.: Konzepte und 3D-Visualisierung interaktiver, perspektivischer Karten. *Angewandte Geographische Informationsverarbeitung XI*, 128-139, 1999.

Döllner J. and Hinrichs K.: An object-oriented approach for integrating 3D visualization and GIS. *Computers & Geosciences*, **26**(1):67-76, 2000.

Döllner J. and Hinrichs K.: Dynamic 3D maps and their texture-Based Design. *Proc. Computer Graphics International*, 325-334, 2000.

Döllner J.: Software-Architektur computergraphischer Systeme. Habilitationsschrift. *Westfälische Wilhelms-Universität Münster*, 145 pages, 2000.

Döllner J., Baumann K., Kersting O.: LandExplorer - ein System für interaktive 3D-Karten. *Kartographische Schriften*, Kirchbaumverlag, **7**:67-76, 2003.

Duchaineau M., Wolinsky M. and Sigeti D.: ROAMing terrain: Real-time Optimally Adapting Meshes. *Proc. IEEE Visualization '97*, 81-88, 1997.

Ebert D.S., Musgrave F.K., Peachey D., Perlin K. and Worley S.: Texturing and modeling: a procedural approach. *AP Professional*, 3rd edition, 687 pages, 2003.

Ervin S.M.: Digital landscape modeling and visualization: a research agenda. *Landscape and Urban Planning*, **54**:49-62, special issue, 2001.

Farin G.E.: Curves and surfaces in computer aided geometric design. *Academic Press*, 5th edition, 499 pages, 2002.

Feibush E. and Gagvani N.: Visualization for situational awareness. *Computer Graphics and Applications*, **20**(5):38-45, 2000.

Foley J.D., van Dam A., Feiner S.K. and Hughes J.F.: Computer graphics principles and practices in C. *Addison Wesley*, 2nd Edition, 1200 pages, 1995.

Foster N. and Metaxas D.: Realistic animation of liquids. *Graphical Models and Image Processing*, **58**(5):471-483, 1996.

Fournier P., Habibi A. and Poulin P.: Simulating the flow of liquid droplets. *Proc. Graphics Interface '98*, 133-142, 1998.

Fowler M.: UML Distilled. *Addison-Wesley*, 3rd edition, 175 pages, 2004.

Fujishiro I. and Aoki E.: Volume graphics modelling of ice thawing. *Proc. of Volume Graphics*, 45-50, 2001.

Funk M., Gudmundsson G.H. and Hermann F.: Geometry of the glacier bed of the Unteraarglacier, Bernese Alps, Switzerland. *Zeitschrift für Gletscherkunde und Glazialgeologie*, **30**:187-194, 1994.

Funkhouser T.: Curved Surfaces. *Course Notes COS 426*, Princeton University, 1999.

Garcia M.A.: Terrain modelling with uncertainty for geographic information systems. *Proc. ISPRS Commission III Symposium*, 273-280, 1994.

Gascuel J.D., Cani-Gascuel M.P. and Desbrun M.: Simulating landslides for natural disaster Prevention. *Proc. 9th EG Workshop on CGAS*, 1998.

Gibson S. and Mirtich B.: A survey of deformable modelling in computer graphics. Tech. Report No. TR-17-97, *Mitsubishi Electric Research Lab*, 33 pages, 1997.

GRAF, K.C.: Realistic landscape rendering using remote sensing images, digital terrain models and 3D objects, *Remote Sensing Series*, **25**, 110 pages, University of Zurich, 1995.

Greuell W.: Hintereisferner, Austria: mass balance reconstruction and numerical modeling of historical length variation. *Journal of Glaciology*, **38**(129):233-244, 1992.

Gudmundsson G.H.: A three-dimensional numerical model of the confluence area of Unteraargletscher, Bernese Alps, Switzerland. *Journal of Glaciology*, **45**(150):219-230, 1999.

Gyalistras D.: Zur zukünftigen Entwicklung des Klimas im Alpenraum und in der Schweiz. In: Wanner H., Gyalistras D., Luterbacher J., Rickli R., Salvisberg E. and Schmutz C. (editors): Klimawandel im Schweizer Alpenraum. Abschlussbericht NFP31. *vdf Hochschulverlag AG an der ETH Zürich*, 161-235, 2000.

Häberling C.: Benutzerbedürfnisse und Anforderungen zu neuartigen Gletscherdarstellungen. *Wiener Schriften zur Geographie und Kartographie*, **11**:81-93, 1998.

Haeberli W. and Schweizer J.: Rhonegletscher 1850: Eismechanische Überlegungen zu einem historischen Gletscherstand. *Mitteilung der Versuchsanstalt für Wasserbau, Hydrologie und Glaziologie an der ETH Zürich*, **94**:59-70, 1988.

Haeberli W.: Glacier fluctuations and climate change detection - operational elements of a worldwide monitoring strategy. *Bulletin of World Meteorological Organization*, **44**(1):23-31, 1995.

Haeberli W. and Hoelzle M.: Application of inventory data for estimating characteristics of and regional climate change effects on mountain glaciers - a pilot study with the European Alps. *Annals of Glaciology*, **21**:206-212, 1995.

Haeberli W., Kääb A., Hoelzle M., Bösch H., Funk M., Vonder Mühll D. and Keller F.: Eisschwund und Naturkatastrophen im Hochgebirge. Schlussbericht NFP31. *vdf Hochschulverlag AG an der ETH Zürich*, 190 pages, 1999.

Haeberli W., Frauenfelder R., Hoelzle M. and Maisch M.: On rates and acceleration trends of global glacier mass change. *Geografiska Annaler*, **81**:585-591, 1999.

Haeberli W., Cihlar J. and Barry R.G.: Glacier monitoring within the global climate observing system. *Annals of Glaciology*, 31:241-246, 2000.

Haeberli W.: Personal communication, 2003.

Hay K.E., Marlino M. and Holschuh D.R.: The virtual exploratorium: foundational research and theory on the integration of 5-D modeling and visualization in undergratduate geoscience education. *Proc. 4th International Conference of the Learning Sciences*, 24-220, 2000.

Hehl-Lange S. and Lange E.: Visualization of brown coal surface mining in the Niderlausitz from 1940 – 1995 – 2010 – 2030. *Proc. Data Visualization '97*, 1997.

Hirtz P., Hoffmann H. and Nüesch D.: Interactive 3D landscape visualization: improved realism through use of remote sensing data and geoinformation. *Proc. Computer Graphics International*, 101-108, 1999.

Hirtz P.: A framework to interactively compose realistic 3D landscape visualizations. PhD thesis, *Remote Sensing Series*, University of Zurich, 149 pages, 2003.

Hoelzle M.: Permafrost und Gletscher im Oberengadin. *Mitteilung der Versuchsanstalt für Wasserbau, Hydrologie und Glaziologie an der ETH Zürich*, **132**, 121 pages, ETH Zurich, 1994.

Hoelzle M., Haeberli W., Dischl M. and Peschke W.: Secular glacier mass balances derived from cumulative glacier length changes. *Global and Planetary Change*, **36**(4):295-306, 2003.

Hoelzle M.: Personal communication, 2004.

Hoppe H., DeRose T., Duchamp T., McDonald J. and Stuetzle W.: Piecewise smooth surface reconstruction. *Proc. SIGGRAPH 94*, 295-302, 1994.

Hoppe H.: Progressive meshes. *Computer Graphics*, **30**:99-108, 1996.

Hoppe H.: Smooth view-dependent level-of-detail-control and its application to terrain rendering. *Proc. IEEE Visualization*, 35-42, 1998.

IAHS (ICSI)/UNEP/UNESCO/WMO: In: Haeberli W., Frauenfelder R. and Hoelzle M. (editors), Glacier mass balance bulletin No. 5 (1998-1999). *World Glacier Monitoring Service*, 93 pages, 2001.

IEEE: IEEE standard computer dictionary: A compilation of IEEE standard computer glossaries, 1990.

Intergovernmental Panel on Climate Change: Climate change: The scientific basis. Contribution of working group I to the third assessment report of the IPCC. *Intergovernmental Panel on Climate Change*, 882 pages, 2001.

Jóhannesson T., Raymond C. and Waddington E.: Time-scale for adjustment of glaciers to changes in mass balance. *Journal of Glaciology*, **35**(129):355-369, 1989.

Jóhannesson T.: The response of two Icelandic glaciers to climate warming computed with a degree-day glacier mass balance model coupled to a dynamic glacier model. *Journal of Glaciology*, **43**(144):321-327, 1997.

Juhasz I.: Weight-based shape modification of NURBS curves. *Computer Aided Geometric Design*, **16**(5):377-383, 1999.

Kaufmann V. and Ploesch R.: Mapping and visualization of the retreat of two cirque glaciers in the Austrian Hohe Tauern national park. *International Archives of Photogrammetry and Remote Sensing*, **13**:446-453, 2000.

King S.A. and Parent R.E.: A 3D parametric tongue model for animated speech. *Journal of Visualization and Computer Animation*, **12**(3):107-115, 2001.

Kirsch F. and Döllner J.: Rendering techniques for hardware-accelerated image-based CSG. *Journal of WSCG*, **12**(2):221-228, 2004.

Kleffner D.A. and Ramachandran V.S.: On the perception of shape from shading. *Perception and Psychophysics*, **52**(1):18-36, 1992.

Kraak M.J.: Visual exploration of virtual environments. In: Virtual Reality in Geography, edited by Fisher P and Unwin D., *Taylor & Francis*, 59-67, 2002.

Kreuseler M.: Visualization of geographically related multidimensional data in virtual 3D scenes. *Computers & Geosciences*, **26**(1):101-108, 2000.

Kuhn M.: The effects of long-term warming on Alpine snow and ice. Discussion Report on Alpine regions. *European Conference on Landscape-ecological impact of climate change*, 10-20, 1989.

Kuhn M.: Energieaustausch Atmosphäre - Schnee und Eis. Schnee, Eis und Wasser der Alpen in einer wärmeren Atmosphäre. *Mitteilungen der Versuchsanstalt für Wasserbau, Hydrologie und Glaziologie der ETH Zürich*, **108**, 135 pages, 1990.

Lane J.M., Carpenter L.C., Whitted T. and Blinn J.T.: Scan line methods for displaying parametrically defined surfaces. *Communications of ACM*, **23**(1):23-34, 1980.

Lavoie P.: NURBS++. http://libnurbs.sourceforge.net/index.shtml. Release 3.0.11. Last access: February 2, 2004.

Layton A.T. and van de Panne M.: A numerically efficient and stable algorithm for animating water waves. *The Visual Computer*, **18**(1):41-53, 2002.

Lazarus F. and Verroust A.: Three-dimensional metamorphosis: a survey. *The Visual Computer*, **14**(8/9):373-389, 1998.

Lindstrom P., Koller D., Ribarsky W., Hodges L.F., Faust N. and Turner G.: Real-time, constinuous level of detail rendering of height fields. *Computer Graphics*, 109-118, 1996.

Lindstrom P., Koller D., Ribarsky W., Hodges L.F., Op den Bosch A. and Faust N.: An integrated global GIS and visual simulation system. *Technical Report GIT-GVU-97-07*, 9 pages, 1997.

Lopez A.M., Lumbreras F., Serrat J. and Villanueva J.J.: Evaluation of methods for ridge and valley detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **32**(4):327-335, 1999.

Maisch M.: Die Gletscher Graubündens. Rekonstruktionen und Auswertung der Gletscher und deren Veränderungen seit dem Hochstand von 1850 im Gebiet der östlichen Alpen. Part A and B. *Physische Geographie*, University of Zurich, **33**, 440 pages, 1992.

Maisch M., Wipf A., Denneler B., Battaglia J. and Benz C.: Die Gletscher der Schweizer Alpen. Schlussbericht NFP31. *vdf Hochschulverlag AG an der ETH Zürich*, 373 pages, 1999.

Meier M.F.: The contribution of small glaciers to global sea level. *Science*, **226**:1418-1421, 1984.

Möller T. and Haines E.: Real time rendering. *AK Peters*, 2nd Edition, 889 pages, 2002.

Neyret F.: Animated texels. *Proc. of 6th Workshop on Computer Animation and Simulation*, 18 pages, 1995.

Nishita T., Sederberg T.W. and Kakimoto M.: Ray tracing trimmed rational surface patches. *ACM Computer Grahpics*, **24**(4), 337-345, 1990.

Nixon D. and Lobb R.: A fluid-based soft-object model. *IEEE Computer Graphics and Applications*, **22**(4):68-75, 2002.

Nye J.F.: The flow of a glacier in a channel of rectangular, elliptic or parabolic cross-section. *Journal of Glaciology*, **5**(41):661-690, 1965.

Organe consultatif sur les changements climatiques (OcCC): Das Klima ändert - auch in der Schweiz. Die wichtigsten Ergebnisse des Wissenstandsberichts des IPCC aus der Sicht der Schweiz. *Organe consultatif sur les changements climatiques*, 48 pages, 2002.

Oepen M.: Environmental communication in a context. In: *Environmental Education, Communication and Sustainability*, edited by Filho W.L., **7**:41-61, 2000.

Oerlemans J., Anderson B., Hubbard A., Huybrechts P., Jóhannesson T., Knap W.H., Schmeits M., Stroeven A.P., Van de Wal R.S.W., Wallinga J. and Zou Z.: Modelling the response of glaciers to climate warming. *Climate Dynamics*, **14**:267-274, 1998.

Oerlemans J.: Glaciers and climate change. *A.A. Balkema Publishers*, 148 pages, 2001.

OpenGL ARB, Woo M., Neider J., Davis T. and Shreiner D.: OpenGL, programming guide. 3rd Edition. *Addison-Wesley*, 730 pages, 1999.

Pajarola R. and Widmayer P.: Virtual geoexploration: concepts and design choices. *International Journal of Computational Geometry and Applications*, **11**(1):1-14, 2001.

Pang A.T., Wittenbrink C.M. and Lodha S.K.: Approaches to uncertainty visualization. *The Visual Computer*, **13**(8)370-390, 1997.

Parent R.: Computer animation, algorithms and techniques. *Morgan Kaufmann Publishers*, 527 pages, 2002.

Paterson W.S.B.: The physics of glacier. *Pergamon Press*, 3rd Edition, 480 pages, 1994.

Paul F., Kääb A., Maisch M., Kellenberger T. and Haeberli W.: The new remote-sensing-derived Swiss glacier inventory: I. Methods. *Annals of Glaciology*, **34**:355-361, 2002.

Perlin K.: An image synthesizer. *Computer Graphics*, **19**(3):287-296, 1985.

Piegl L.: Modifying the shape of rational B-splines. Part 2: surfaces. *Computer-Aided Design*, **21**(9):538-546, 1989.

Piegl L. and Tiller W.: The NURBS book. *Springer*, 2nd Edition, 646 pages, 1997.

Piegl L. and Tiller W.: Surface approximation to scanned data. *The Visual Computer*, **16**(7):386-395, 2000.

Preston M. and Hewit W.T.: Animation using NURBS. *Computer Graphics Forum*, **13**(4):229-241, 1994.

Rodger J.C. and Browse R.A.: Choosing rendering parameters for effective communication of 3D Shape. *Computer Graphics and Applications*, **20**(2):20-28, 2000.

Roettger S., Heidreich W., Slusallek Ph. and Seidel H.: Real-time generation of continuous levels of detail for height fields. *Proc. WSCG'98*, 315-322, 1998.

Rogers D.: Z-buffering, interpolation and more W-buffering. Nvidia white paper, 7 pages, http://www.nvidia.com, access: February 12, 2003.

Rost R.J.: OpenGL Shading Language, *Addison Wesley Professional*, 608 pages, 2004.

Rumbaugh J., Blaha M., Premerlani W., Eddy F. and Lorensen W.: Object-oriented modeling and design, *Prentice Hall*, 1991.

Sánchez-Reyes J.: A simple technique for NURBS shape modification. *IEEE Computer Graphics and Applications*, **17**(1):52-59, 1997.

Scheib V., Haber J., Lin M.C. and Seidel H.: Efficient fitting and rendering of large scattered data sets using subdivision surfaces. *Computer Graphics Forum*, **21**(3):353-362, 2002.

Schneeberger C., Albrecht O., Blatter H., Wild M. and Hock R.: Modelling the response of glaciers to a doubling in atmospheric CO_2: a case study of Storglaciären, northern Sweden. *Climate Dynamics*, **17**:825-834, 2001.

Sederberg T.W.: Free-form deformation of solid geometric models. *ACM Computer Graphics*, **20**(4):151-159, 1986.

Shreiner D.: OpenGL Reference Manual. *Addison Wesley*, 3rd Edition, 704 pages, 1999.

Simpson J. and Weiner E.: The Oxford English Dictionary. *Oxford University Press*, 2nd Edition, 22'000 pages, 1989.

Spence R.: Information visualization. *Addison Wesley*, 224 pages, 2001.

Stora D., Agliati P., Cani M., Neyret F. and Gascuel J.: Animating lava flows. *Proc. Graphics Interface '99*, 203-210, 1999.

Storch H. Von: Models in environmental research. *Springer Verlag*, 272 pages, 2001.

Summer R.W., Brien J.F. and Hodgins J.K.: Animating sand, mud, and snow. *Computer Graphics Forum*, **18**(1):17-26, 1999.

Surazhsky T. and Elber G.: Shape blending - matching free-form surfaces. *Computers & Graphics*, **25**(1):3-12, 2001.

Suter M.: Aspekte der interaktiven real-time 3D-Landschaftsvisualisierung. *Remote Sensing Series*, **29**, 141 pages, University of Zurich, 1997.

Swiss Federal Office of Topography: Pixel map PK50 (1:50'000) and PK200 (1:200'000), product description, *Federal Directorate of Cadastral Surveying*, 2000.

Swiss Federal Office of Topography: Digital Height Model DHM25, product description, *Federal Directorate of Cadastral Surveying*, 2001.

T2: Texture generation program. http://www.petra.demon.co.uk/Games/texgen.html. Last access: Decembre 17 , 2003.

Takatsuka M. and Gahegan M.: GeoVISTA Studio: a codeless visual programming environment for geoscientific data analysis and visualization. *Computers & Geosciences*, **28**(10):1113-1144, 2002.

Terzopolus D. and Fleischer K.: Deformable models. *Proc. of SIGGRAPH 88*, 269-278, 1988.

Terzopolus D. and Qin H.: Dynamic NURBS with geometric constraints for interactive sculpting. *ACM Transactions on Graphics*, **13**(2):103-136, 1994.

Technische Universität Wien: Powerpoint slides “Freeform surfaces”. http://www.geometrie.tuwien.ac.at. Access: November 21, 2002.

Uhlenküken C., Schmidt B. and Streit U.: Visual exploration of high dimensional spatial data: requirements and deficits. *Computers & Geosciences*, **26**(1):77-85, 2000.

Van Dam A., Forsberg A.S., Laidlaw D.H., LaViola J.J. and Simpson R.M.: Immersive VR for scientific visualization: A progress report. *IEEE Computer Graphics and Applications*, **20**(6):26-52, 2000.

Verbree E., van Maren G., Germs R., Jansen F. and Kraak M.-J.: Interaction in virtual world views - linking 3D GIS with VR. *International Journal of Geographical Information Science*, **13**(4):385-396, 1999.

Warmerdam F.: Geospatial data abstraction library. http://www.remotesensing.org/gdal. Release 1.1.7. Last access: March 14, 2004.

Warren J. and Weimer H.: Subdivision methods for geometric design. *Morgan Kaufmann Publishers*, 310 pages, 2001.

Watt A.: Computer graphics. *Addison Wesley*, 3rd edition, 610 pages, 2002.

Wei X., Wei L. and Kaufman A.: Melting and flowing of viscuous volumes. *Proc. Computer Animation and Social Agents*, 54-59, 2003.

Wittmann J.: Simulationsmodell und Geographisches Informationssystem: Kopplungsalternativen am praktischen Beispiel. *Proc. Umweltinformatik 2000*, 45-48, 2000.

WordNet: A lexical database for the english language. http://www.cogsci.princeton.edu/~wn/index.shtml, Princeton University, 2003.

wxWindows: http://www.wxwindows.org. Last access: February 2, 2004.

Zhou X. and Li S.: Measurement of BRDF of snow and sea ice in the Ross and Amundson Seas by the visible and near-Infrared channels of MODIS. *Proc. IEEE International Geoscience and Remote Sensing Symposium*, 1567-1569, 2000.

Zöckler M., Stalling D. and Hege H.: Fast and intuitive generation of geometric shape transitions. *The Visual Computer*, **16**(5):241-253, 2000.

ISBN: 3-03703-009-7