ECOLOGICAL INFORMATICS 1 (2006) 179–187

available at www.sciencedirect.com

SCIENCE DIRECT®

www.elsevier.com/locate/ecolinf

Real-time 4D visualization of migratory insect dynamics within an integrated spatiotemporal system

Yi Wu[a,*], *Bronwyn Price*[b], *Daniel Isenegger*[c], *Andreas Fischlin*[b], *Britta Allgöwer*[c], *Daniel Nuesch*[a]

[a]*Remote Sensing Laboratories, Department of Geography, University of Zurich, Winterthurerstr. 190, CH-8057 Zürich, Switzerland*
[b]*Terrestrial Systems Ecology, Department of Environmental Sciences, ETH Zurich, Universitätstrasse 16, CH-8092 Zürich, Switzerland*
[c]*GIScience Center, Department of Geography, University of Zurich, Winterthurerstr. 190, CH-8057 Zürich, Switzerland*

A R T I C L E I N F O

Article history:
Received 7 November 2005
Received in revised form
9 March 2006
Accepted 12 March 2006

Keywords:
Real-time 4D visualization
Spatiotemporal analysis
Migration
Dispersion
Cloud modeling
Cloud rendering

A B S T R A C T

This paper presents a new approach of spatiotemporally visualizing the simulation output of migratory insect dynamics and resultant vegetation changes in real-time. The visualization is capable of displaying simulated ecological phenomena in an intuitive manner, which allows research results to be easily understood by a wide range of users. In order to design a fast and efficient visualization technique, a simplified mathematical model is applied to intelligibly represent migrating groups of insects. In addition, impostors are used to accelerate rendering processes. The presented visualization method is implemented in an integrated spatiotemporal analysis system, which models, simulates and analyzes ecological phenomena such as insect migration through time at a variety of spatial resolutions.

© 2006 Elsevier B.V. All rights reserved.

1. Introduction

Nowadays, ecologists are able to model and analyze dynamics of migratory insects such as larch bud moth (LBM) within efficient simulation systems (Fischlin, 1982; Fischlin and Baltensweiler, 1979; Baltensweiler and Fischlin, 1988). As a kind of migratory insects, LBM are often capable of migrating long distances and causing conspicuous defoliation on Alpine Larch (*Larix decidua*) across extensive areas (Baltensweiler and Fischlin, 1988). During peaks of population cycles, overcrowding of LBM larvae causes interrupted feeding behavior, which leads to partially eaten needles drying out such that large-scale defoliation results during the summer months. The consequences directly impact on tourism and forest succession (Baltensweiler and Fischlin, 1988; Baltensweiler and Rubli, 1999). Therefore, knowledge of the migration dynamics of such insects is of value within ecology and forest management.

However, current systems do not allow for the analysis and display of simulation results in a visual manner that may be easily interpreted by non-experts. As an example, Fig. 1 and Table 1 show two results from a LBM simulation model run. The 2D illustration in Fig. 1 describes the temporal variations of LBM larval density for the entire Upper Engadine valley aggregated to a single spatial unit, while Table 1 shows the numbers of female moths that migrate from a particular site to other sites with the resulting defoliation percentages. It is very difficult for a layperson to imagine the spatiotemporal dynamics of LBM migration while looking at Fig. 1. Even if Table 1 contains all data to describe these dynamics in space and time, the figures are detached from their spatial locations. Therefore, interpretation of results from ecological simulation systems is often limited to experts. In

* *Corresponding author*. Tel.: +41 44 635 51 63; fax: +41 44 635 68 46.
E-mail addresses: yi.wu@geo.unizh.ch (Y. Wu), bronwyn.price@env.ethz.ch (B. Price), disen@geo.unizh.ch (D. Isenegger), andreas.fischlin@env.ethz.ch (A. Fischlin), britta@geo.unizh.ch (B. Allgöwer), nuesch@geo.unizh.ch (D. Nuesch).

1574-9541/$ - see front matter © 2006 Elsevier B.V. All rights reserved.
doi:10.1016/j.ecoinf.2006.03.004

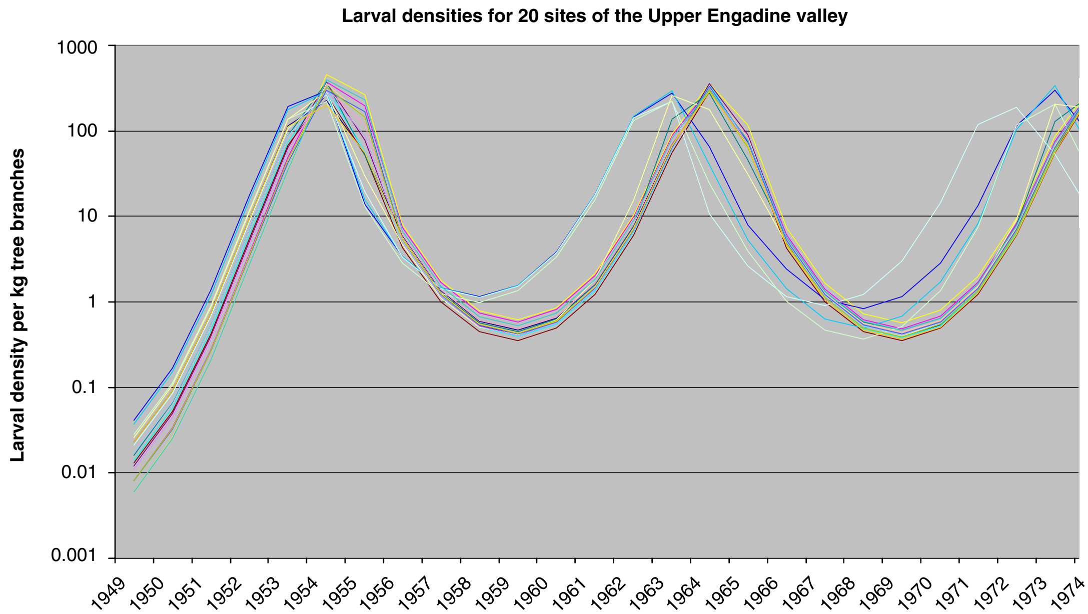

Fig. 1 – LBM larval density variation in the Upper Engadine valley results from the LBM-M9 model.

this paper, we take LBM dynamics as an example to demonstrate how a real-time 4D visualization in an integrated spatiotemporal analysis system brings easily understandable illustration of LBM migration and resultant vegetation changes to a wide range of users.

To improve the understanding of the ecological processes of migratory insects in both the temporal and the spatial dimensions, an integrated system, *Interactive, Process Oriented, Dynamic Landscape Analysis and Simulation System* (IPODLAS), is being developed. Therein, a temporally explicit ecological modeling simulation system (Temporal Simulation) is coupled with a spatially explicit geographic information system (GIS), and a real-time 4D visualization. This integrated system IPODLAS allows simulation and investigation of spatiotemporal ecological phenomena such as LBM migration and representation of simulation results in an intuitive manner.

For this case study, real-time visualization in IPODLAS plays the role of representing results from ecological systems. To interactively visualize the simulation results in a 4D virtual environment, large data sets such as a digital elevation model and terrain texture of study area must be additionally rendered in real-time. Moreover, for the visualization of insect migration dynamics, there are two alternating computing processes: 1) insects' migrating paths must be calculated according to the simulation output and terrain and 2) lively insects in the air have to be rendered dynamically. In addition, once insects have landed, the new defoliation level resulting from a change of local insect density need to be visualized by changing the appearance of vegetation on the terrain.

However, memory size and computing speed of today's common computers are insufficient to compute all these changes and photo-realistically visualize individual insect in real-time (Roettger and Ertl, 2003; Roettger Heidreich, 1998). For the convenience of a wide range of users, efficient methods are explored in this paper for adapting existing computing and rendering algorithms to visualize both insect dynamics and resultant vegetation changes in real-time. The resulting spatiotemporal visualization achieves easier understanding of interactive and explorative simulation results, whereby simulation results depicting insect dynamics and vegetation changes are visually attached to the relevant 3D spatial locations. In the following sections, we use LBM dynamics as an example to illustrate the potential of our methods.

Table 1 – Data table of LBM female migration number and defoliation percentage of every site in 1-year results from the LBM-M9 model

Site	1	2	…	20
1	4765	126	…	286
2	0	64	…	103
3	324	0	…	32
…	…	…	…	…
20	712	21	…	124
Defoliation	0.00364	0.03211	…	0.04321

2. Material and methods

2.1. *Study area*

In this paper, the Upper Engadine valley, located in the Swiss part of the European Alps, forms the study area. Herein, quantitative data on host trees, LBM larval densities, defoliation, and natural enemies has been collected for more than 50 years (Baltensweiler and Fischlin, 1988; Baltensweiler and Rubli, 1999).

For the purposes of this study, larch bud moth population dynamics, numbers of annually migrating females and defoliation levels are simulated with a coupled local dynamics and migration within the Upper Engadine valley model, known as LBM-M9 (Fischlin, 1982; Fischlin and Baltensweiler, 1979). During development of the LBM-M9 model, the Upper

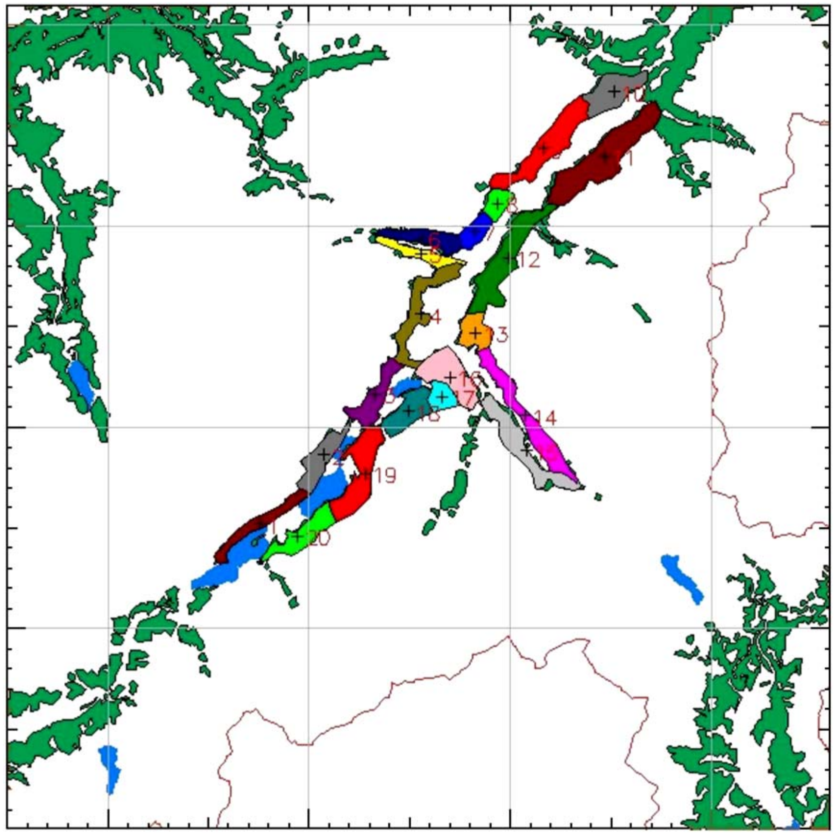

Fig. 2 – The distribution of larch forest sites in the Upper Engadine valley.

Engadine valley was divided into 20 areas, considered to be homogeneous with respect to aspect and altitude for the purposes of formulating the migration model (Fischlin, 1982). These areas, known as sites, are shown in Fig. 2. Field study-based LBM population census data cover a time series for each of the 20 sites from 1949 to 1977, thus representing an ideal study case or test bed for the dynamic visualization of spatiotemporal processes.

2.2. *Data preparation*

The local dynamics submodel of LBM-M9 is based on the food quality hypothesis, which considers population dynamics of larch bud moth to be driven by a relationship between the LBM and its host (Baltensweiler and Fischlin, 1988). The migration submodel simulates LBM migration between sites and is dependant on wind conditions, forest cover and site quality (defined by degree of defoliation) (Fischlin, 1982). All outputs of LBM-M9 are simulated within the Temporal Simulation subsystem, RAMSES (research aids for modeling and simulation of environmental systems), and then transferred via our integrated system kernel, shown in Fig. 3, to the visualization subsystem as an input.

In general, the following five data sets are necessary for the spatiotemporal visualization of LBM dynamics and resultant vegetation changes in the study area:

- A digital elevation model (DEM) [© Tydac AG, Bern] to describe the geometry of the terrain in the Upper Engadine valley. The used DEM is a raster-based elevation model with an original grid size of 50 m × 50 m. It was resampled to 10 m × 10 m using a bilinear interpolation scheme.

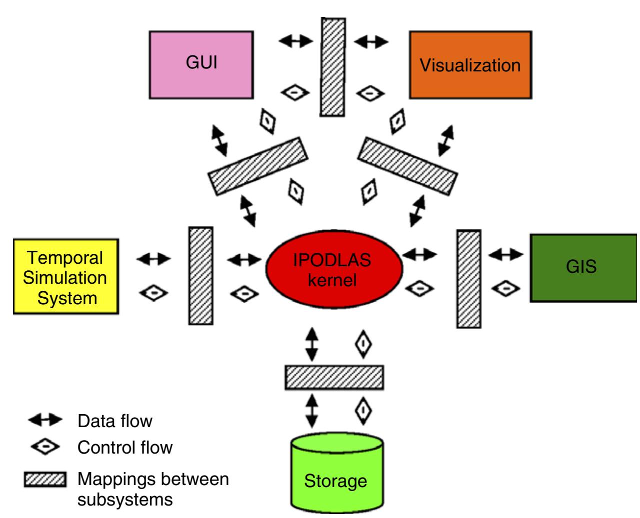

Fig. 3 – The IPODLAS kernel is the central coordination component which controls communication between the subsystems (GIS, Visualization, and Temporal Simulation), GUI, and the Storage. For performance reason, the design allows that the visualization subsystem can interact directly with the GUI, and not through the IPDOLAS kernel as required for the other subsystems.

Table 2 – Defoliation levels for categories of simulated defoliation percentage and the corresponding visualized forest color

Defoliation percentage	Defoliation level	Forest color
0.00–0.01	None	Green
0.011–0.33	Light	Yellow
0.331–0.66	Medium	Orange
0.661–1.0	Heavy	Brown

- A satellite image [© ESA/Eurimage, CNES/Spotimage, Swisstopo/National Point of Contact (NPOC)] to describe the appearance of the terrain in a photo-realistic manner. A mosaic of Landsat Thematic Mapper imagery with a raster size of 25 m × 25 m was used as a basic data set. Its spatial resolution was improved to 10 m × 10 m by image fusion with panchromatic SPOT data [15].
- Geographical locations and extents of the 20 spatially homogenous LBM study sites within the Upper Engadine valley. This information was generated within the GIS subsystem and has the same resolution as the DEM.
- Annual number of simulated female LBM migrating from each site to all other sites.
- Simulated defoliation percentage of the Larch trees following insect migration for each site each year. This simulation was performed within RAMSES. Defoliation was categorized to 4 different levels following Baltensweiler and Rubli (1999), colors for visualized forests were chosen accordingly to represent a gradient from green, for no defoliation, to brown, for heavy defoliation, shown in Table 2.

2.3. *The integrated spatiotemporal system*

The integrated spatiotemporal system, IPODLAS, uses data and functionalities of three subsystems, GIS, Temporal Simulation, and Visualization in a transparent and seamless way to profit from the individual strengths of the respective subsystems. Fig. 3 shows the software architecture of IPODLAS, which supports the smooth interaction of the subsystems.

IPODLAS is designed as a user-driven system, which means that actions of a user in the graphical user interface (GUI) trigger the GUI to invoke activities, possibly in all other subsystems. These activities are coordinated by a synchronization mechanism, which is one of the main functionalities of the IPODLAS kernel. For example, when the IPODLAS kernel receives a user request from the GUI, it gives primary feedback to the user and breaks the request down into subtasks, which are sent to the appropriate subsystems. After completion of subtasks, the subsystems send the results via the IPODLAS kernel to the GUI and user.

The IPODLAS kernel not only supports the interoperability of the subsystems, it also allows a subsystem to be easily exchanged. For example, if we wanted to use another Temporal Simulation system, it needs only to be registered in the IPODLAS kernel. Since all the subsystems communicate through the IPODLAS kernel, different protocols map the data models of the subsystems to the data model of the IPODLAS kernel. To support seamlessly accessing the required functionalities of the subsystems from the GUI, the mapping of the functions of the subsystems to the IPODLAS kernel is also mandatory (Leclercq et al., 1996). However, in order to speed up data exchange, some interactions between the GUI and Visualization do not go via the IPODLAS kernel.

When using IPODLAS to model and simulate migratory insect dynamics, we use both public domain GRASS (geographic resources analysis support system) and ESRI's ArcGIS/ArcInfo to obtain geographical locations and extents of the study area. Spatial data and functions from the GIS subsystem are used within the Temporal Simulation subsystem, RAMSES in this study, to simulate LBM spatiotemporal dynamics. Then the visualization subsystem functions as an interface to users, displaying the simulation results from RAMSES.

In IPODLAS, an efficient technique is essential for terrain rendering. To visualize spatiotemporal migratory insect dynamics, namely allowing extension of existing functions, the source code and access to the low level rendering application programming interface (API) must be available. The virtual terrain project (VTP) satisfies these requirements and is selected as a software foundation (Roettger and Heidreich, 1998; Doellner and Kersting, 2000; Losasso and Hoppe, 2004).

VTP is a creative project for easy construction of virtual terrain in an interactive, 3D digital format. One of its main advantages is that its run-time environment makes use of continuous level of detail (CLOD) algorithms for terrain rendering (Roettger and Heidreich, 1998). The work of Biegger (2004) shows that VTP can act as a suitable base for implementation of a visual system dedicated to simulation and analysis of a dynamic process, i.e. glacier fluctuations. Therefore, visualization of migratory insect dynamics and resulting vegetation changes can be developed as an embedded module within VTP.

2.4. *Visualization of LBM dynamics*

According to the simulation results from RAMSES, many thousands and even millions of LBM may migrate from one site to others (Baltensweiler and Rubli, 1999; Fischlin, 1982, 1983). Since the number of migratory female LBM from each site each year is given statistically, we simply assume that moths from each site within the Upper Engadine valley move together and thus can be thought of as a 'cloud' with each small group of moths representing an internal particle of the cloud.

LBM researchers found that LBM migrating from each site normally land in several sites separately. Hence, during migration, some LBM may land in a particular site, while the others continue flying. Accordingly, LBM cloud can be considered split into several small clouds flying to different target sites after taking off from each site. If a user views a visualization of migration from several different sites in a given time period, he may find it difficult to distinguish from which site a migrating cloud originates. For convenience, during visualization a different color is assigned to individual insect groups originating from different sites.

However, visualization of animated clouds remains a challenge. On one hand, the difficulties in computing velocity fields are apparent. Velocity fields governing the dynamics of clouds are influenced heavily by wind and other disturbance forces in the air. Our LBM clouds need to be visualized as dynamic clouds because they are composed of dynamic creatures that do not keep to the relative static positions within the clouds during migration. On the other hand, rendering a cloud requires consideration of both the effects of optical properties along light paths through the cloud volume and the complex multiple scattering of light within the medium before it reaching the viewer. Previous work related to cloud visualization within the computer graphics field could be divided into two parts: cloud modeling and cloud rendering (Dobashi et al., 2000). Cloud modeling deals with methods used to represent construction and cloud dynamics, while cloud rendering deals with the techniques of efficient and realistic rendering.

2.4.1. *LBM cloud modeling*

Since our goal is to render real-time dynamic LBM clouds in large numbers and at the same time the main task of central processing unit (CPU) and graphics processing unit (GPU) of computer is to render a large terrain data set with a high-resolution satellite image, a simple and efficient modeling method is required to construct and simulate the dynamics of LBM clouds in a visually convincing way. Fluid dynamics is a straightforward method to calculate velocity fields for realistic dynamic clouds with an arbitrary initial structure (Fedkiw et al., 2001; Foster and Metaxas, 1997; Stam, 1999). However, this is impractical since it is computationally too expensive. The heuristic approach is computationally inexpensive and much easier to implement (Ebert, 1997; Roettger and Ertl, 2003). A third approach which lies between the above two approaches uses cellular automation to simplify dynamic cloud motion (Dobashi et al., 2000; Harris and Lastra, 2001).

Since the primary task for this study is to render LBM clouds and terrain, the heuristic approach is applied to construct and simulate dynamic LBM clouds in order to save computation time. In this paper, a two-level model is applied to construct the initial shapes: the cloud macrostructure and the cloud microstructure (Ebert, 1997). Implicit primitives such as ellipsoids are assigned locations, sizes, and weights to construct the macrostructure. A Gaussian distribution combined with a random noise function is applied to create the microstructure, which describes the distribution of particles within each ellipsoid. Moreover, the Gaussian center is irregularly located inside the ellipsoid to introduce more randomness to the particle distribution. Each internal particle is represented by a sphere (blob) with a density function identifying the distribution of a LBM subgroup within the sphere. To reduce the computation cost and accelerate rendering process, blob centers are supposed to have the local highest density of LBM and the density decreases according to a Gaussian distribution.

Both ellipsoids and blobs in a cloud have different sizes and densities. The volume of total ellipsoids for a cloud decides its size. Meanwhile, the number of the blobs also increases or decreases accordingly to fill in the volume. As an example, Fig. 4 shows two cloud models constructed by 4000 and 1000 blobs, respectively, in 3 ellipsoids.

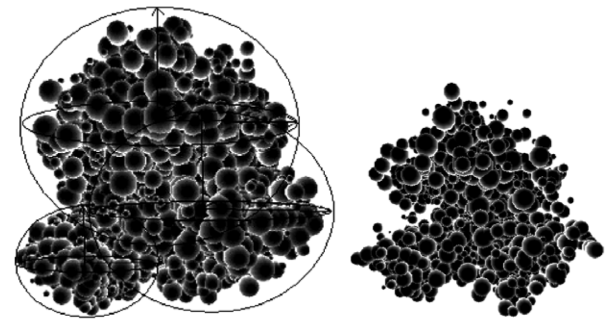

Fig. 4 – Two clouds shaped by 4000 and 1000 blobs, respectively, in 3 ellipsoids.

With the above approach, every LBM cloud has different structure. Then to simulate dynamic LBM clouds, we assume moths randomly flight inside a LBM cloud. A straightforward method is to store every blob's position and continuously update it according to some random rules in real-time. However, this approach can cause a large burden on CPU and main memory since each LBM cloud may have hundreds or thousands of blobs. Hence, we only store macrostructure and Gaussian center of blob distribution for each LBM cloud. Consequently, blobs are dynamically generated in each frame with random function under constraints of Gaussian distributions for LBM clouds. Reader may think that LBM clouds will change appearance unreasonably quickly since every blob has a new position in each frame. However, this is not true since statistically blob distributions have been decided by macrostructures and Gaussian centers. After rendering, discontinuous changes can only be distinguished at places having few blobs. Therefore, the above simulation method exactly reflects our assumption of the unordered flight of moths inside LBM clouds.

2.4.2. *LBM cloud rendering*

"Rendering clouds is difficult because realistic shading requires the integration of the effects of optical properties along paths through the cloud volume, while incorporating the complex scattering within the medium" (Harris and Lastra, 2001). In recent research within the computer graphics field, one of the most common approaches is to use 3D textures to render amorphous phenomena. Fedkiw et al. successfully rendered smoke with 3D texture mapping hardware (Fedkiw et al., 2001). Although it is simple to be used by programmers, GPU does many background computations to support it. Moreover, to use this technique, standard APIs such as OpenGL have strict constraints on the size of 3D texture data. Therefore, we use 2D texture mapping to render half-transparent property of LBM clouds. However, it is important to note that the algorithms based on 2D texture mapping are applicable to 3D texture mapping hardware as well.

In order to display visually convincing cloud images, we need to take into account light absorption and multiple scattering caused by blobs inside clouds. Absorption is decided by extinction coefficient, which varies according to the properties of internal blobs. In this study we choose a constant for LBM in order to simplify the computation. Since clouds are composed of many tiny blobs, multiple scattering among them is too complicated to be computed in real-time. Fortunately, the work of Nishita et al. demonstrated that the contribution of multiple scattering is dominated by the first two orders (Nishita et al., 1996). Hence, multiple forward scattering is often used nowadays to approximate multiple scattering for cloud rendering.

With the help of hardware acceleration method, Dobashi et al. applied an approximation of isotropic single forward scattering to render clouds (Dobashi et al., 2000). Later, Harris and Lastra extended Dobashi's method with multiple forward scattering and anisotropic first order scattering towards the eye to obtain more realistic results (Harris and Lastra, 2001). Both methods considered shading and self-shading when light passes through a cloud. Moreover, both work applied hardware acceleration method to accomplish the computation of light intensities on blobs in their implementations. Although photo-realistic results can be obtained with their methods, it is still too time-consuming for our purposes, especially during the crucial process of reading pixels back from the frame-buffer in GPU. Our method is a simple approximation of their work, which is fast enough to achieve real-time rendering without losing too much visual quality.

If we consider a light ray traversing a cloud, starting from outside, the intensity I_k on any blob p_k is equal to light intensity scattered to p_k from other blobs plus the intensity transmitted to p_k through p_{k-1} (as determined by its transparency), which is closer to light source than p_k along ray direction. Hence, Blobs need to be queued according to ascendant distance to light source. Multiple forward scattering assumes the light intensity scattered by other blobs on p_k is determined by the light intensities on the blobs closer to light source than p_k with an isotropic albedo. Then, light intensity on each blob can be computed from the beginning of the queue to the end.

Although the above process mimics light traversing cloud well, it is computationally expensive to achieve real-time LBM clouds rendering with a common computer. Hence, instead of precise computation, we imitate the result by assuming light intensity for each big ellipsoid in a cloud can be mapped onto an elliptic figure through its centre and facing the viewer. The intensities on blobs inside this ellipse have an anisotropic Gaussian distribution. That means if we place a Gaussian center at the centre of the ellipse, light intensities will isotropically increase from the centre of the ellipse to its border. Here, light intensities have inverse relationship to LBM densities in ellipsoids. It coincides with the assumption in the last section.

Besides the approximation of light multiple forward scattering inside clouds, single scattering towards the viewer must also be considered to re-adjust each blob color in eyes. We simplify light scattering towards the viewer by giving constant values to the albedo and extinction coefficient. Assuming $\vec{l}$ is view direction from blob p_k to the viewer, and $\vec{w}$ is light direction; the intensity E_k scattered from the position of p_k towards the viewer is computed:

$$E_k = \alpha p(\vec{l}, \vec{w}) I_k + (1 - e) E_{k-1}, \qquad 1 \le k \le N \tag{1}$$

where α is albedo and e is extinction coefficient, $p(\vec{l}, \vec{w})$ is the phase function, which determines the extent of light incident from direction $\vec{w}$ scattered by cloud to direction $\vec{l}$, N is the number of blobs. This equation simply says that light scattered from the position of p_k towards the viewer is equal to the intensity scattered to the viewer by p_k, plus the intensity scattered by other blobs to the viewer through p_k but not absorbed by p_k. Then cloud image is rendered by the portion of light passed through cloud towards the viewer and not absorbed by internal blobs. Here, the simple Rayleigh scattering phase function is used:

$$p(\theta) = 3/4(1 + \cos^2\theta) \tag{2}$$

where θ is the angle between the incoming and scattered light directions. This phase function was proved to be efficient and practical for cloud rendering by Harris and Lastra (2001).

In rendering process, 2D polygons textured with a Gaussian function are used to represent blobs. The colors of polygons are determined by light intensities on blobs, albedo and angles between incident light and view directions. Polygons are blended according to Eq. (1) with alpha equaling to extinction coefficient using alpha blending technique. This technique is a standard capability of today's common GPU. Hence, all blending processes are executed automatically and intermediate results are saved as image in GPU, which largely alleviate the storage burden of the main memory.

Since all the blobs of a LBM cloud are finally blended into one cloud image, dynamic impostor is used to insert the LBM cloud to the 3D virtual landscape. An impostor is a semi-transparent quadrangle textured with an image of the object which is replaced by the impostor in the virtual world. Since our virtual landscape allows free navigation, impostors must always be placed on planes facing the viewer. The direction from the centre of an impostor to the viewer is always the normal direction of the impostor. Finally, we achieve rendering of LBM clouds in real-time with promising results.

2.4.3. *LBM migration path*

Migration paths of LBM in the real world are complicated and difficult to determine. To simplify the computation, we assume they migrate in the following manner:

When moths start migrating from each site, they gradually fly to a certain height directly above terrain. As the number of migrating LBM varies largely in different sites and years, we suppose that migration starts from a region near the centre of a site. As migrating LBM number becomes larger, the region extends gradually to the whole site.

Once moths reach the assumed height above terrain, they split into smaller LBM clouds, which fly directly to their respective destination sites. To enhance visual effects, we assume migration paths moving up and down according to terrain. Since every LBM cloud heads to its own destination site, there is a computable shortest path on terrain between the projection point of a LBM cloud center and the center of its destination site. We further assume that every LBM cloud flies along the shortest path and keeps the height above terrain. As soon as the projection point of a LBM cloud center exceeds the centre of its landing site, the LBM cloud is ready to land.

The landing process is also a dispersive process. During the landing of a LBM cloud, its density is reduced, while its vertical extension becomes shorter and its cross section becomes gradually larger. Finally, the LBM cloud covers a certain region of the landing site and moths disappear from the users view. Assumptions about the landing region are the same as the assumptions for the take-off region described above. The landing paths are implemented as individual straight lines

Fig. 5 – Clouds fast rendered by simplified method.

between some sample points in the LBM cloud and points inside the landing region.

2.5. *Modeling dynamic vegetation change*

Depending on LBM larval density, larch forests experience different degrees of defoliation. This results in significant color changes of the larch foliage during the summer, changing the appearance of the forested landscape within each LBM site considerably (Baltensweiler and Fischlin, 1988; Baltensweiler and Rubli, 1999). However, satellite images are not available for all the investigated years (1949–1977), and thus cannot fulfill our requirements. Moreover, since a large raster-based satellite image is used as a texture layer for our virtual landscape, it is time-consuming to change some parts of the image and reload back to texture cache to display the yearly resultant vegetation changes of the whole study area in real-time. Therefore, we import geographical information about the larch forest sites with the same resolution as the DEM used to form the virtual terrain from the subsystem GIS in advance and build polygon-based representations of these sites slightly above the virtual terrain. By doing so, the highlighted resultant vegetation areas can easily be distinguished by the viewers.

With the alpha blending technique once again, the colors of the sites can be combined with the underlying satellite image. Therefore, dynamic vegetation changes in real-time is possible to be visualized by changing only the colors of these sites. This method illustrates apparently annual resultant vegetation changes to users with low computation costs.

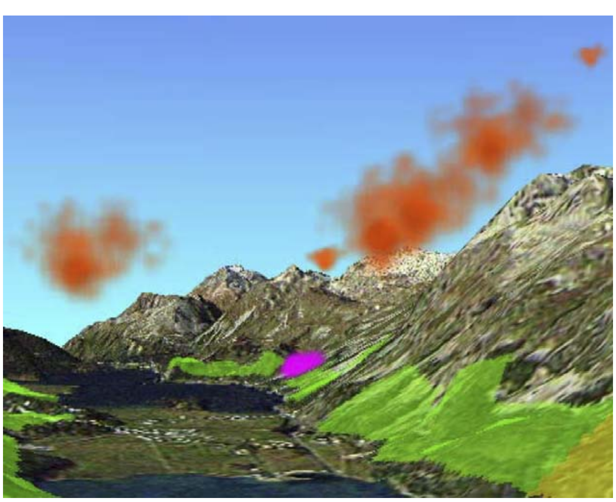

Fig. 6 – Several orange LBM clouds beginning from one site and a landing purple LBM cloud beginning from another site. The height of the clouds is determined by the terrain.

3. Results

To present spatiotemporal simulation results in a visually intuitive manner, an approach for 4D real-time visualization of migratory insects and resultant vegetation changes was developed. Standard digital terrain models and satellite images are used to render virtual pseudo-realistic landscapes. Against this background, groups of migrating insects are represented as continuously animated clouds using a GPU accelerated technique. A simplified mathematical model is applied to make cloud modeling and rendering fast and efficient. Thus, the analysis outputs of LBM migration, data tables which detached from spatial locations, from the temporal ecological simulation system can be visually approximately expressed by the dynamic processes of ascending, splitting, flying, descending and dispersing of LBM

(a)

(b)

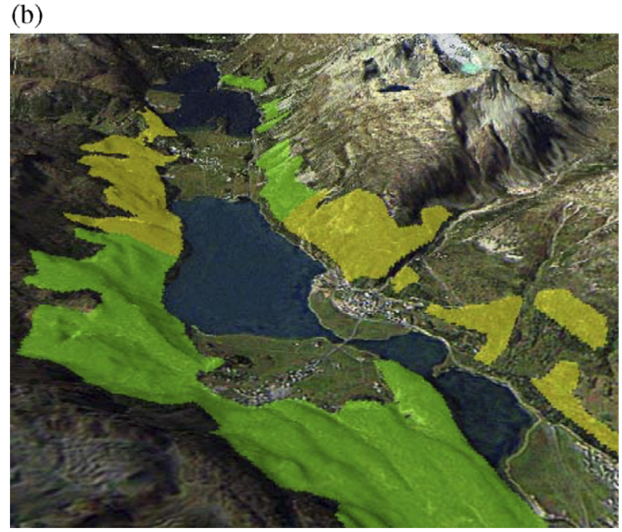

Fig. 7 – (a, b) The appearances of forest sites following larval feeding in two successive years.

clouds. Simultaneously, resulting from varied larch forest defoliations caused by annually different LBM densities, the appearance of the landscape is changed by dynamically blending colors of influenced areas with the underlying terrain texture.

As an example, Fig. 5 shows three clouds, with a little more complicated macrostructure, fast rendered by our simplified method.

Fig. 6 shows LBM clouds in migration. All orange LBM clouds originate from a same site while the small purple one comes from another site. At this moment, the purple LBM cloud is landing and the orange LBM clouds are still heading to their target sites. Because LBM cloud migration paths are moving up and down according to the terrain, the orange clouds are flying at different heights.

Once each LBM cloud has landed, the simulated resultant level of defoliation in each larch forest site is displayed by changing the site colors according to the output of the temporal ecological simulation system. In Fig. 7(b), two originally green sites in the Upper Engadine valley turn into yellow (light defoliation).

4. Discussion and future work

Using the simplified models presented in this paper, we can visualize the simulation results of migratory insects in virtual landscape in real-time. The 4D dynamic scenarios give users a better understanding of insect migration and its influence on the vegetation. Hence, simulation results are more understandable and directly perceivable by users than a table or a 2D picture.

Animated clouds are rendered by taking advantages of color blending capability of GPU and dynamic impostors. It allows real-time visualization of a few animated clouds with our simple cloud models at the same time.

Assumptions are made in order to visualize migration behavior of LBM using limited computing time. Nevertheless, the goal of this visualization is to efficiently demonstrate simulation results of a modeled ecological system in real-time so that they are easily understandable and perceivable by the layperson, as opposed to other researchers or experts, who might ask for high quality of realistic images, for example animated movies. A more realistic visualization would allow for greater visual feeling, however, the understanding of simulation outputs would not be improved at the cost of computation efficiency.

Although the methods developed are tested in the Upper Engadine valley, they could be also applied to other regions. Moreover, they are suitable for the visualization of the dynamics of other migratory insects, which have similar migration properties as LBM.

Our visualization approach treats the LBM simulation procedure as a 'self-contained' process and does not take further physical parameters into account while visualizing LBM migration. For example, wind speed and direction could change cloud geometries and dynamics so heavily that LBM cloud models should be accordingly reconstructed for each frame. Therefore a more efficient method is necessary to complete this task.

Instead of static resultant vegetation geometries, new CLOD techniques should be applied to those irregular geometries to make the rendering process faster.

In addition, improvement in the interaction between visualization and the other subsystems of IPODLAS could significantly support users in studying the simulation results. For example, the geographical information of a selected area on the virtual terrain will help the layperson know where the insect migration takes place. Such problems could be avenues of our future work.

We hope our work can also serve as an example for other researchers to apply computer graphics to illustrate analysis results in ecological research.

Acknowledgements

This work is funded by the Swiss National Science Foundation within the National Research Program 48 'Landscapes and Habitats of the Alps' (Grant No. 4048-064432/1).

REFERENCES

Baltensweiler, W., Fischlin, A., 1988. The larch bud moth in the Alps. In: Berryman, A.A. (Ed.), Dynamics of Forest Insect Populations: Patterns, Causes, Implications. Plenum Publishing Corporation, New York, pp. 331–351. a.o.

Baltensweiler, W., Rubli, D., 1999. Dispersal: an important driving force of the cyclic population dynamics of the larch bud moth. *Zeiraphera diniana* Gn. Forest Snow Landsc. Res. 74 (1), 3–153.

Biegger, S., 2004. A visual system for the interactive study and experimental simulation of climate-induced 3D mountain glacier fluctuations. Remote Sensing Series, vol. 43. University of Zurich. 150 pp.

Dobashi, Y., Kaneda, K., Yamashita, Okita, H.T., Nishita, T., 2000. A simple, efficient methos for realistic animation of clouds. Proc. Of SIGGRAPH'2000, pp. 19–28.

Doellner, J., Kersting, O., 2000. Dynamic 3D maps as visual interfaces for spatiotemporal data. ACMGIS 2000. ACM Press, pp. 115–120.

Ebert, D.S., 1997. Volumetric modelling with implicit functions: a cloud is born. Visual Proc. of SIGGRAPH'97, pp. 147–156.

Fedkiw, R., Stam, J., Jensen, H.W., 2001. Visual simulation of smoke. Proc. of SIGGRAPH'2001, pp. 15–22.

Fischlin, A., 1982. Analyse eines Wald-Insekten-Systems: Der subalpine Lärchen-Arvenwald und der graue Lärchenwickler Zeiraphera diniana Gn. (Lep., Tortricidae). Diss. ETH No. 6977, Swiss Federal Institute of Technology: Zürich, Switzerland, pp. 294–318.

Fischlin, A., 1983. Modelling of Alpine valleys, defoliated forests, and larch bud moth cycles: the role of migration. In: Lamberson, R.H. (ed.), Mathematical Models of Renewable Resources. Humboldt State University, Mathematical Modelling Group, University of Victoria, Victoria, B.C., Canada, pp. 102–104.

Fischlin, A., Baltensweiler, W., 1979. Systems analysis of the larch bud moth system: Part I. The larch–larch bud moth relationship. Mitt. Schweiz. Entomol. Ges. 52, 273–289.

Foster, N., Metaxas, D., 1997. Modeling the motion of a hot turbulent gas. Proc. of SIGGRAPH'97, 181–188.

Harris, J.M., Lastra, A., 2001. Real-time cloud rendering. Eurographics 20 (3), 76–84.

Leclercq, E., Benslimane, D., Yetongnon, K., 1996. A distributed object architecture for interoperable GIS. IEEE Knowledge and Data Engineering Exchange Workshop, pp. 73–80.
Losasso, F., Hoppe, H., 2004. Geometry clipmaps: terrain rendering using nested regular grids. Proc. of SIGGRAPH'2004, pp. 769–776.
Nishita, T., Dobashi, Y., Nakamae, E., 1996. Display of clouds taking into account multiple anisotropic scattering and sky light. SIGGRAPH, pp. 379–386.
Roettger, S., Ertl, T., 2003. Fast volumetric display of natural gaseous phenomena. Comput. Graph. Int. 74–86.
Roettger, S., Heidreich, W., 1998. Ph. Slusallek and H. Seidel, Real-time generation of continuous levels of detail for height fields. Proc. WSCG'98, pp. 315–322.
Stam, J., 1999. Stable fluids. Proc. of SIGGRAPH'99, pp. 121–128.