Research Statement
Matthias Zwicker

Interacting with Visual Data

My research in computer graphics focuses on the development of methods and systems to increase the efficiency and effectiveness of user interaction and manipulation of visual data. Interactivity is a fundamental characteristic of most computer graphics applications, yet user interaction processes are often extremely cumbersome and time consuming. Therefore, I believe that improving user interaction and manipulation is essential for the achievement of real progress in computer graphics. The often stunning results obtained with commercial computer graphics products can only be produced with enormous, and sometimes overwhelming, effort. Creating, modifying, and displaying three-dimensional objects with realistic shape, appearance and motion is still a challenging task even for the most highly skilled user. I believe novel algorithms and data structures that are designed to simplify the user interaction with visual data and extend the scope of available operations will impact the way we think about computer generated imagery. Computer images and interactive computer graphics systems will become even more ubiquitous; obtaining convincing results will require less user effort and experience.

I have chosen to concentrate on three research areas where interactivity has proven to be a particular challenge. First, I am attempting to develop and improve 3D geometry representations that are suitable for interactive 3D modeling and content creation. Second, I am investigating methods for intuitive manipulation of visual data. This includes not only editing the shape and appearance of three-dimensional objects, but also working with two-dimensional images. Third, I am developing interactive rendering systems that let a user visualize and explore complex data in real-time. My approach to research in these areas is interdisciplinary, combining tools and methods from mathematics, physics, and computer vision with novel applications for interacting with visual data. In the following paragraphs I describe in more detail my areas of active research, summarizing achieved results and outlining current projects and ideas for future work.

Geometry Representation and Processing. The mathematical representation of two-dimensional surfaces in three-dimensional space plays a crucial role in many computer graphics problems. For example, algorithms for rendering surfaces to a two-dimensional image are a fundamental component of any computer graphics application. The design of these algorithms largely depends on the underlying surface representation; similarly, the ability for a user to modify the shape of a surface is determined by the flexibility of its mathematical representation. To address these requirements of computer graphics applications, a variety of surface representations have been developed, ranging from parametric surfaces (e.g., Bézier surfaces, NURBS, or subdivision surfaces) and triangle meshes to implicit surfaces (e.g., using radial basis functions, level sets, or sampled distance fields). Each of these representations has its strengths and weaknesses for rendering, modeling, and other relevant operations.
In my PhD thesis, I proposed the use of point samples as a fundamental building block to represent surfaces. Due to its conceptual simplicity, this representation leads to highly efficient and flexible algorithms for interactive rendering and modeling. The investigation of the properties of point-sampled surfaces still provides a wealth of challenging research opportunities.

Surface parameterization is a fundamental operation that is useful for various computer graphics applications. I have developed algorithms for parameterizing points to planar and spherical two-dimensional domains. The extension of these methods to higher dimensions is an interesting venture that could lead to their application in computer vision and machine learning.

Automatic processing of point-sampled surfaces, such as surface smoothing, is an important step to prepare the data for interactive modeling and rendering. In collaboration with MIT colleagues, I have developed a method for smoothing surface normals that is designed to increase rendering quality.

Identifying and maintaining sharp features is a fundamental issue when processing surfaces. While several approaches have been proposed to define continuous surfaces directly from discrete sample points (e.g., moving least squares surfaces), I intend to extend these methods to preserve sharp features. Since the definition and identification of sharp features is an ill-posed problem, a user-guided approach seems promising.

**Interactive Modeling.** The wide availability of digital cameras and progress in 3D scanning and reconstruction technology has led to an increased interest in tools to process and edit such 3D data. After data acquisition, a user often needs to process and edit the raw data to obtain the desired shape and appearance of the 3D object. However, conventional systems require sophisticated preprocessing to generate consistent triangle meshes or higher order surfaces before the user can interact with the data. In addition, most tools either focus on editing the shape or the appearance of 3D objects. I have built a system for editing the geometry and appearance of 3D objects that uses a point-based surface representation, which works directly with the 3D point samples generated by the 3D acquisition process. By generalizing 2D pixels to 3D points, it provides functionality similar to image editing tools and seamlessly integrates editing of shape and appearance.

Since the original approach was constrained to small modifications of the surface geometry, I am currently working on removing these constraints and creating intuitive methods for large deformations. A promising approach is the use of local, or differential, geometry representations to solve this problem. Surface deformations are achieved by integrating the differential properties over the surface subject to some constraints that specify the deformation. However, these methods are complicated by the fact that the differential quantities are not invariant to rotations. I am proposing to solve this problem using a generalization of barycentric coordinates. I believe that the investigation of such differential surface representations will also lead to other applications such as surface smoothing and morphing.

Other promising approaches to the interactive design of graphics objects are the so-called example-based or data-driven methods. I plan to develop example-based approaches to geometric modeling where shape deformations are guided by a set of examples rather than by physical principles. Given a small number of example shapes (e.g., obtained with a 3D scanner), a user could easily generate new shapes that interpolate intuitively between the examples. At the core of such an approach would be a mechanism to measure a
distance and interpolate between shapes, and an optimization procedure that finds the closest shape when given user specified modeling constraints.

Data-driven methods rely on large collections of acquired data of a certain class of objects. As a fundamental building block, an abstract model of the specific class of objects is then built from the data. By modifying the model parameters, a user can interactively generate new instances of the class of objects and design a desired object that is different from all acquired data. Technically, each acquired object is interpreted as a sample point in a high-dimensional space. Because of the constrained class of objects that are collected, these samples only occupy a small fraction of this space. Hence, the model is built by mapping the sample points to a linear or non-linear lower dimensional manifold, which can be used as a representation of all objects in that class. Most prominently in computer graphics, this approach has been successfully applied to databases of human faces. In one of my current projects, I am developing a generative model for natural textures. This is challenging because experiments show that there is no single lower dimensional manifold that describes all natural textures. Therefore, I envision a method that eliminates the need for a global parameterization and is able to describe complex manifolds. This approach is very general and has the potential for modeling other data that is relevant for computer graphics such as motion and reflectance data.

**Interactive Rendering Systems.** Interactive rendering is a main focus of computer graphics research because it is an essential component of every interactive computer graphics system. In my work, I have investigated novel data representations and rendering algorithms for visualizing complex data sets. A major contribution of my PhD thesis was to show that point-based representations are also suitable for the purpose of rendering. Previously, point-based rendering had been plagued by low image quality and low rendering performance. In particular, aliasing was a largely unsolved problem. By applying and extending results from classical signal processing theory, I developed core methods for reconstructing, filtering, and resampling nonuniform point-sampled surfaces. Based on these techniques, I derived high quality rendering algorithms that make point rendering practical. These algorithms have already been included in a variety of applications.

In the area of volume data visualization, as well, the aliasing problem had not been solved in a satisfying manner due to a lack of theoretical considerations. By applying a more rigorous analysis, I was able to derive efficient rendering algorithms with a higher image quality.

Although sampling and aliasing has been widely studied for synthesizing two-dimensional images that are displayed on a regular computer screen, these issues have not been as rigorously explored in the case of generating three- or four-dimensional sampled images for multi-view autostereoscopic displays. I plan to develop antialiasing filters for three- or four-dimensional light field data shown on autostereoscopic displays (e.g., on a parallax-barrier display). It is well known that scene elements need to be prefiltered depending on their depth to avoid inter-perspective aliasing on such displays. With my analysis I intend to show that proper prefiltering can be achieved without explicit knowledge of depth. This technique will facilitate applications such as aliasing-free real-time display of time-dependent light field data.

Computer graphics hardware has achieved such impressive computational power because of the specific structure of rendering pipelines. They fit the framework of stream computing, which is an abstraction model for computations that are highly parallel and
require mostly local data access. As part of a collaboration between the Computer Architecture and the Computer Graphics groups at MIT, I am investigating novel architectures for graphics pipelines based on StreamIt, a compiler for stream programs, and Raw, a multi-processor with a programmable on-chip communications network, which were both developed by the Computer Architecture group. Although current graphics hardware allows a certain degree of programmability, there are still parts in the pipeline that only allow the execution of fixed functionality. In addition, the allocation of computational resources to different stages in the pipeline is static, which leads to inefficiencies that depend on the application.

In contrast, our research envisions a system that will be capable of providing optimal dynamic resource allocation and load balancing in a fully programmable pipeline. To achieve this, we intend to exploit Raw’s high-bandwidth communications network, which will be programmed to adapt to application-specific bottlenecks. Another challenge will be the configuration of Raw’s memory architecture to support graphics-specific memory access patterns, for example, efficient texture prefetching and caching. To make the implementation of such a system tractable, we will need to develop extensions of the StreamIt compiler to support operations required by a 3D rendering pipeline (e.g., dynamic data rates between stream processing elements, access to global texture memory, and message passing).