# Part I

# Introduction

# 1 Free-Viewpoint Video

## 1.1 Motivation

Nowadays, interactive entertainment plays a major role and is starting to outperform traditional media. The computer gaming industry had higher sales than Hollywood's movie industry for three successive years now, and online games attract millions of players worldwide. With the sophistication of modern grapics hardware, lifelike dynamic scenes can be rendered, which will soon be virtually indistinguishable from their real-world counterparts recorded with cameras. Many modern computer games more and more resemble interactive movies, with a professional storyline and character development.

In contrast, the television and video experience remains unchanging. Interactivity when watching TV or a video is restricted to adjusting the volume or switching to another channel. While this is probably desireable for certain movies, where perspective and lighting were carefully chosen by a director for dramatic effect, a sports documentation would be much more exciting if one could change the viewpoint interactively, for instance to get the desired view of the last goal during a soccer broadcast. Educational documentaries would benefit as well from more interactivity, since it is much easier to visualize complex structures, e.g. molecules or engines, if the viewer can rotate them by himself.

A growing crowd of researchers is therefore pursuing 3D video and 3D television, where modern computer graphics and vision techniques are to be combined to obtain a streamable medium which gives the user more control over the aspects of playback, in particular free choice of viewpoint. This thesis aims at providing some of the necessary tools required for a 3D video system. We start with an overview of such a system in the next section, and focus on specific aspects later in the thesis. Our primary goal is to obtain a high-quality representation of the geometry visible in a scene. This is a key requirement for being able to render novel views from arbitrary perspectives. We also discuss ways to generate these views interactively from the geometry model and the source images on modern graphics hardware.

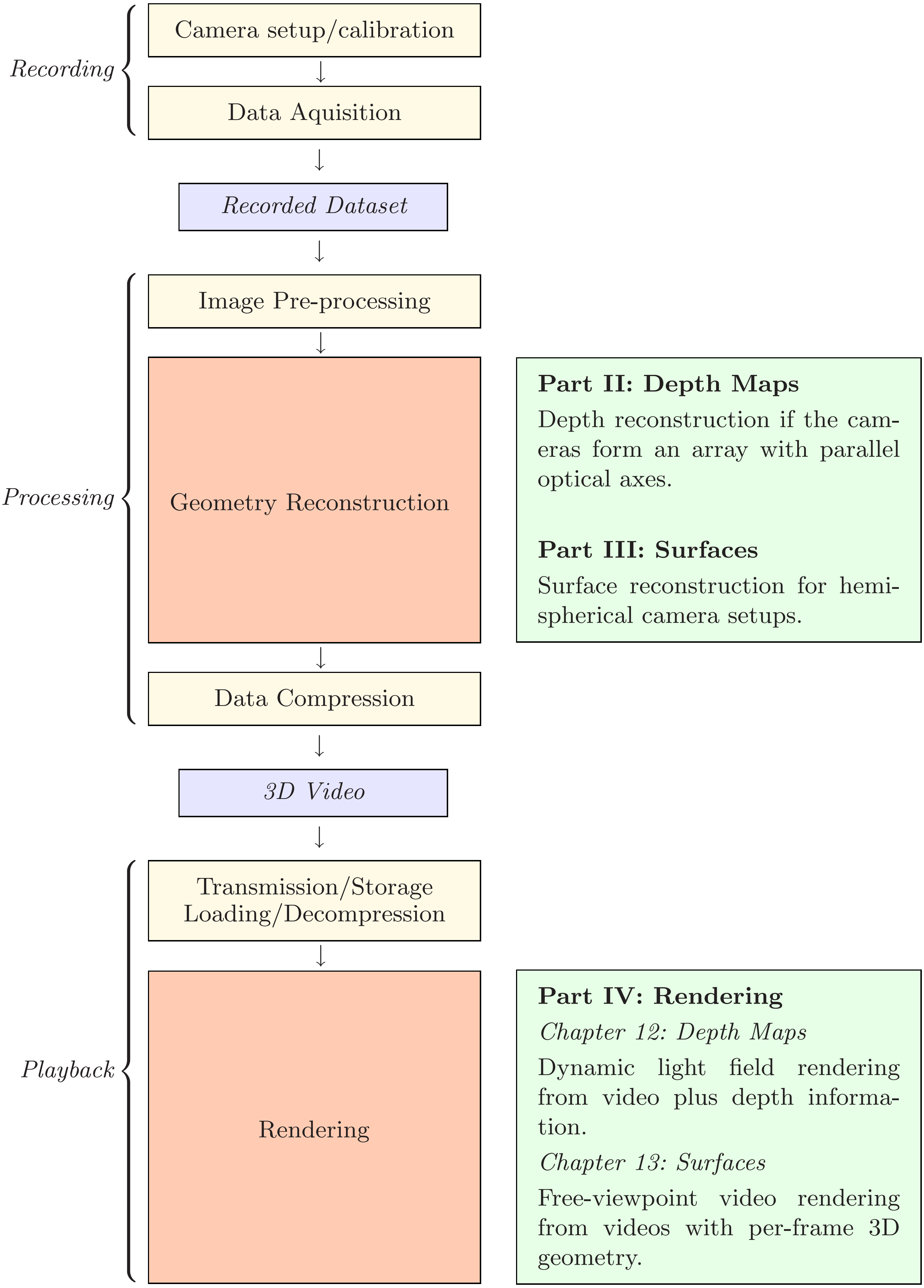
## 1.2 Components of a Free-Viewpoint Video System

The full pipeline for a system capable of recording, encoding and high-quality playback of 3D video is very sophisticated and consists of a lot of different steps, each of which is a small challenge in itself. In this work, we can only adress a small subset of them in more detail. We chose two topics which are in our opinion the most interesting ones, 3D reconstruction and rendering, and we discuss them in the remaining parts of this thesis. For the sake of completeness, we briefly outline the other necessary steps here. Fig. 1.1 shows the whole 3D video pipeline at a glance.

**Camera Setup and Calibration.** Naturally, the first and foremost task is to acquire data. For our reconstruction algorithms, we require the internal and external camera parameters of each of the imaging devices to be available. In particular, we need to know the exact mapping of 3D points to 2D image coordinates. To achieve this, before recording any actual data, a calibration procedure is applied. A suitable one has to be chosen depending on the setup of the cameras. We review some choices when we present different recording setups in the following two parts.

**Data Aquisition.** To record multi-video sequences of a temporally varying scene, the necessary hardware effort is considerable. Multiple synchronized video cameras are needed to capture the scene from different viewpoints. Aside from the physical construction of the calibrated studio, the cameras have to be triggered simultaneously, and the huge amounts of data they produce must be stored on hard drives in real time.

With recording hardware becoming cheaper and cheaper, nowadays, several research labs around the world feature studios capable of recording multi-video sequences [137, 16, 46, 79, 84, 100, 136]. In this work, sequences from two different systems were used. The first is the Stanford Multi-Camera Array [137]. Its cameras are densely packed in one or more arrays, and the data is ideally suited for the depth reconstruction algorithms and light field rendering techniques investigated in Part II of this thesis. The second one is our own custom-made studio available at the MPI Informatik [130]. It captures wide-baseline multi-video sequences in a hemispherical setup around the scene. Thus, it is ideal to aquire data for the surface reconstruction techniques presented in Part III. Both capturing systems are described in the respective



**Part II: Depth Maps**

Depth reconstruction if the cameras form an array with parallel optical axes.

**Part III: Surfaces**

Surface reconstruction for hemispherical camera setups.

**Part IV: Rendering**

*Chapter 12: Depth Maps*

Dynamic light field rendering from video plus depth information.

*Chapter 13: Surfaces*

Free-viewpoint video rendering from videos with per-frame 3D geometry.

**Fig. 1.1.** The components forming the pipeline of a free-viewpoint video system, and where they are discussed in detail in this thesis.

part of the thesis where they are used, the Stanford Array in Sect. 4.2, and our in-house system in Sect. 7.4.

**Image Processing.** Before the reconstruction algorithms can be applied to the data, some image processing usually has to be performed. Colors have to be corrected in order to account for differences in the cameras' sensors, and for some vision techniques background subtraction is necessary. We do not discuss image processing techniques in more detail in this work. Instead, we assume that the input to our algorithms consists of sequences already pre-processed in a suitable manner.

**Geometry Reconstruction.** 3D video requires approximate scene geometry to be known for every frame of the sequence. Geometry reconstruction is one of the two major topics of this work. For setups with parallel optical axes, estimation of depth maps is the best one can hope for. We refer to these approaches as $2\frac{1}{2}$D. They are explored in Part II. In general camera setups, the recovery of true 3D surface geometry is desireable, as described in Part III.

**Data Compression.** Multi-camera video recordings produce huge amounts of data. Any useful 3D video format must include a very efficient way to reduce the redundancies in this data. It must be possible to load and decompress the video data in real-time for the rendering stage.

**3D Video Rendering.** Once the video data is uncompressed in main memory, it can be played back with the original frame rate. Ideally, the user is able to choose an arbitrary viewpoint without impacting the visual quality of the result. Real objects can be placed in virtual 3D environments, the environment can interact with them and vice versa. Rendering is the second main topic covered in this thesis, and discussed in detail in Part IV.

## 1.3 Outline of the Thesis

The focus of this work are methods for the reconstruction of $2\frac{1}{2}$D and 3D geometry in different camera setups, and photo-realistic rendering of the geometry from novel viewpoints, using the input images as texture. Thus, we only pursue the steps *Geometry Reconstruction* and *3D Video Rendering* of the free-viewpoint video pipeline.

In the next chapter, we introduce basic concepts and mathematical notation, which we use throughout the rest of the thesis. Afterwards, we present related work in Chapter 3.

This is followed by the analysis of the first kind of camera setups in Part II, where all cameras have parallel or near-parallel optical axes, as for instance in case of the Stanford aquisition system. For these setups, it is usually only possible to recover a $2\frac{1}{2}$D model of scene geometry in form of a *depth map* for each camera image. We present an algorithm which simultaneously estimates both a full set of dense depth maps for all cameras, as well as a segmentation into moving foreground and static background, Chapter 5. Temporal coherence can also be exploited in order to improve the accuracy of the estimate, Chapter 6.

Another kind of camera setup, which requires a substantially different representation of scene geometry as well as reconstruction and rendering techniques is the (hemi-)spherical setup, where the cameras are more or less evenly distributed around a moving object. This setup is analyzed in Part III. We present a space-time isosurface reconstruction technique in Chapter 9, which is based on the mathematical analysis of weighted minimal surfaces in Chapter 8. With a sophisticated choice of energy functional, the weighted minimal surface approach can also be employed to reconstruct solid transparent objects of homogenous index of refraction, Chapter 10.

The final Part IV is devoted to rendering techniques suitable for both kinds of setups. Rendering of dynamic light fields with depth information, which relies on the kind of data estimated in Part II, is presented in Chapter 12. Surfaces recovered from the 3D reconstruction techniques in Part III can be rendered in real-time and textured with the video data, as shown in Chapter 13.

# 2

# Basic Concepts and Notation

## 2.1 Scene Geometry

The goal of every reconstruction algorithm is to obtain an approximation to the exact scene geometry. At a single time-step $t \in \mathbb{R}$, we require the geometry to be a closed, piecewise differentiable two-dimensional manifold $\Sigma_t \in \mathbb{R}^3$. In particular, in this work we only deal with well-behaved, entirely solid objects with an open interior. Phenomena like smoke and foam, which consist of many small particles or are more of a fractal-like nature, are not being considered. However, we can work with large bodies of transparent, refractive materials, see Chapter 10.

The surface manifold is allowed to vary smoothly over time. If such a time-varying scene with moving objects is considered, the scene geometry $\Sigma_t$ evolves over time and traces a three-dimensional manifold $\mathfrak{H}$ in space-time, Fig. 2.1. The intersections of $\mathfrak{H}$ with planes of constant time $t$ yield the surface geometry $\Sigma_t$ at this moment.

The motion of surface points is described by the scene flow, which yields correspondences between points in surface manifolds at different time steps. It is a time-dependent vector field $\mathbf{v}_t : \Sigma_t \to \mathbb{R}^3$ which assignes to each point on the surface its current speed. Thus, the integral curves of the scene flow are the trajectories of surface points.

## 2.2 Multi-View Geometry

An essential idea of most 3D reconstruction schemes is that the location of 3D scene points can be computed from their projections. Indeed, in theory, if only two projections of the same 3D point $p$ in two different views are