The White Rabbit put on his spectacles.
"Where shall I begin, please your Majesty ?" he asked.
"Begin at the beginning," the King said, very gravely,
"and go on till you come to the end: then stop."
(Lewis Carroll)

Chapter 1

Introduction

This chapter presents an introduction to the applications and challenges of video-segmentation and it provides an outline of the thesis structure. The motivation starts with a survey of typical application areas that apply segmentation algorithms. This includes video editing, compression, content analysis, and 3-D reconstruction applications. A particular focus is on the concept of object-oriented video coding in the MPEG-4 standard. Afterwards, the requirements are defined for a segmentation system that is compliant with the MPEG-4 video-coding approach. A proposal for such a segmentation system is made and the main components are briefly introduced. The detailed description of this segmentation system establishes the first half of the thesis (Part I). Finally, various extensions of the segmentation system are proposed which are also discussed further in the second half of the thesis (Part II and III). This introduction concludes with an overview of the individual chapters, indicating the relevant publications and contributions of the author.

1.1 Motivation

In 1966, the Artificial Intelligence pioneer Marvin Minsky directed an undergraduate student to solve *the problem of computer vision* as a summer project. Now, 40 years later, the computer-vision problem is still unsolved, despite the huge amount of joint efforts that have been undertaken in the research community. While the general video-understanding problem is widely regarded as too ambitious to be solved in the near future, various successful spin-offs for practical problems in controlled environments have been developed. Although video understanding is a very difficult area for automatization, it is worthwhile to adopt it in many applications that are currently operating mainly at the signal level without content-adaptive processing. The better we understand not only the video-signal statistics, but the semantic *meaning* of it, the better we can adapt the associated video processing and the more possibilities for interaction with the video content are made available.

The video understanding problem can be specialized into a large variety of applications. In the following sections, four application areas for video segmentation are outlined. This overview also provides references to the relevant chapters in this thesis.

1.1.1 Video editing and scene composition

Until recently, video editing was equivalent to the temporal cutting of video to create movies or documentaries. However, the production of movies is currently making the step towards an object-oriented scene composition. An increasing number of scenes are not recorded directly as a whole, but recorded object-by-object and composed later.

For the composition of video objects, not only the raw texture data is required but also the object shape in form of a mask to seamlessly insert the object into a background image. In the case of computer-generated objects, this mask is easy to obtain, but for captured real-world objects, the mask must be deduced from the image itself. For difficult cases or when no compromise on quality is allowed, this segmentation task is still done manually. Automatic segmentation is usually carried out with the chroma-key technique. Here, by providing a scene background in the designated background color, the object can be distinguished easily from the background. The disadvantage of the choma-key technique is clearly that it is only applicable in a strictly controlled environment. For the extraction of objects from general video-sequences, other segmentation techniques are required. (High-quality segmentation for editing application will be addressed in Chapter 11).

1.1.2 Object-oriented video coding

Most current techniques for video coding like MPEG-2 are non-adaptive to the content. Choosing between different coding modes for individual macroblocks allows for some adaptation to the video content, but the decision about coding modes is usually based on minimizing the resulting bit-rate. However, people started to intentionally misuse the possibilities of the video-coding tools to achieve a higher *subjective* image quality. These techniques exploit properties of the human visual system at the semantic level of attention to detect those areas in the image that are most important to the viewer. These areas are then coded with a better quality at the expense of a lower quality in other regions.

Such a system has been proposed by the author in earlier work to achieve a better visual quality for an MPEG-2 video coding system [67, 66]. In this system, the image was classified into regions of *texture*, *text/graphics*, and *smooth areas*. Since it is important to keep text at a high quality, its quality was increased by using more bits for coding text regions instead of spending the bits on the coding of texture regions. Additionally, the encoder used a scene-change detection algorithm to decrease the coding quality in a short time interval around the scene-change, based on the observation that the human perception is not sensitive to a low image quality close to global changes in the image. Compared with a system that optimizes on PSNR [64], the content-adaptive coding can achieve a better visual quality for the same bit-rate.

With the advent of the MPEG-4 video coding standard, object-oriented video coding was for the first time integrated as a substantial part of a video coding standard. The object-oriented approach offers several advantages and new possibilities. Let us present two of them by using the example case of news programs.

- Composing the visual elements of the news program into a single picture and transmitting this picture as a whole leads to low compression efficiency, because the image is composed of objects of different nature. While the anchorman is a natural video object, there is also superimposed text, graphical images like maps, and computer-generated videos like the weather chart. A better coding efficiency could be obtained by using specialized codecs for the different types of content and composing the final scene at the decoder.

- An object-oriented video representation also offers more possibilities to interact with the content. For example, the viewer could choose his favourite design of the news studio, or he may increase the text font-size if he is visually impaired. Another possibility of interaction is to

provide object-specific annotations or to make the objects *clickable*. This would make it possible to create videos with hyperlinks that can be activated by clicking on objects.

A prerequisite for the above application features is the availability of efficient object-segmentation algorithms. However, the accuracy required for the segmentation results depends on how the segmentation masks are employed in the coding process. We can identify three coding approaches that we present in the following in the order of increasing requirements on the segmentation accuracy.

- **Region-of-Interest (ROI) coding**. The image is coarsely separated into background and foreground. The background includes the content that is unimportant for the viewer whereas the foreground comprising the more important objects. The video coder can then be controlled to code the foreground with a higher quality at the expense of lower quality in the background. The ROI-coding approach is interesting, for example, for surveillance-video recording systems, since the amount of recorded video can be increased, while still keeping the high quality of the objects that matter in the analysis. Moreover, because of the static background in surveillance videos, it is also easy to define the important foreground objects.

- **Coding improvement by object-border detection**. Traditional video coding approaches mostly employ block-based transforms that do not adapt to the boundaries of objects. However, the texture usually changes suddenly at the object border. Filtering across this border consequently leads to a low decorrelation of the pixel values.

  Both problems can be eliminated by coding the interior and exterior regions independently. The MPEG-4 video-coding standard applies this approach by employing a shape-adaptive DCT to include only object pixels in the transformation, and by restricting the effect of the motion vectors to the current object area.

  For this coding approach, it is not required that the segmentation masks are semantically meaningful, as long as they serve to improve the coding efficiency.

- **Composition of video objects.** Semantically correct segmentation masks are certainly required if the purpose of the object-oriented video coding is not only to provide better compression efficiency, but also to enable the composition of new scenes from independently captured video objects. It should be noted that this area of applications

not only covers TV broadcasts, which is traditionally a more passive medium for the viewer, but it also covers especially more interactive internet applications. Numerous possible applications exist and include web-based games, product presentations (e.g., show a specific piece of furniture in various environments), virtual realities, or interactive design applications.

1.1.3 Automatic video analysis

Video-object segmentation is also an indispensable technique for an in-depth analysis of video content. Before the objects in a scene can be identified and their behaviour is analysed, video object have to be detected and separated in the image. The segmentation accuracy that is required depends on the subsequent analysis steps. In case that the object behaviour is derived from the motion trajectory of the object, an accurate segmentation mask is not required. Contrariwise, if the object shape is used to derive its pose and ultimately its behaviour, the accuracy of the segmentation mask is of significant importance.

Automatic video analysis is relevant for a broad range of applications. In the following, we provide only a few examples:

- **Surveillance**. One application area that is quickly growing is the automatic analysis of surveillance videos. Currently, surveillance systems are still non-intelligent video recording systems, often comprising a larger number of cameras. The analysis is primarily still performed by humans watching the videos either in real-time or from the recording. Automatic surveillance systems can help in this situation by either doing the analysis completely automatic, or by providing a pre-alarm indicating situations that require a closer look by a human observer.

  Surveillance is also attractive for automatic analysis from a technical point of view because the input video is relatively easy to analyze. Often, the cameras are statically mounted, such that the environment is well-defined. Moreover, the objects that should be observed can usually be well defined. (Related to Appendix F.)

- **Sports**. Another application that is, from an algorithmic point of view, similar to the surveillance application is the automatic analysis of sport events like tennis or soccer games. Well-known examples are offside analysis in soccer games or court-line checks in tennis. Automatic sports analysis can extract statistical information about the game or for individual players. This information can subsequently

be used to enrich the sports transmission with additional information about the player performance, which is for the entertainment of the viewers, but which can also provide valuable information for the coaches to analyze the strengths and weaknesses of their own athlets or the competitors. From the technical point of view, sports analysis is also interesting because the variety in the scenes is rather limited, thereby enabling a more detailed analysis. The playfield is usually well defined by the markers that are drawn on the playfield. Moreover, the behaviour of the players is well defined and can be described by the rules of the game. (Related to Chapter 13.)

- **Video databases**. Storage costs for video data become increasingly lower and large amounts of video data are already collected in professional archives and at home. The search and retrieval in these media databases poses the new problem of efficiently searching in video data. Manual annotation of the videos with meta-data is often not feasible, so that the search must be carried out either on the raw video data or on automatically generated meta-data. This again requires detailed video analysis, since the queries have typically a high semantic level. An optimal query system should be able to transfer a linguistic description of the scene to a suitable query into the video data. For specific applications like surveillance, this is easier to accomplish, since the nomenclature is well defined. (Related to Chapters 9 and 10.)

  Another problem specific to video databases is the quick browsing in the archive. Since video is a medium that takes place also in time, it cannot be understood quickly from a static snapshot. However, the computer can help to reduce the amount of video that should be viewed by preselecting the most important scenes, or the scenes which are most characteristic to deduce an impression of the full video. Current algorithms in this area usually only consider the global appearance of the image, but the systems can be extended to more in-depth analysis by detecting specific objects of interest [104] and extracting preferably those scenes where these objects occur. (Related to Appendix A.)

Further applications of video analysis which we do not consider further are medical applications, industrial image processing for, e.g., quality control, robotics, or remote-sensing. Even though such applications may be very different, the basic video-analysis techniques are comparable and their principles can be reused. For example, semi-automatic segmentation algorithms as described in Chapter 11 for natural images are also

very popular in medical applications, e.g., for defining tumor areas on CT scans. Industrial image processing often employs of simple color segmentation (Appendix E), because the environment can be controlled easily. Mobile robots require video-object segmentation for collision prevention or for the interaction with the objects. Finally, remote-sensing applications apply the same change-detection algorithms as the ones in the foreground extraction presented in this thesis, but for remote-sensing, their usage is to identify changes in vegetation.

1.1.4 3-D analysis and reconstruction

Video-object segmentation as discussed up to now was related to the extraction of a two-dimensional mask of the object in the input image. However, the input image itself is only a projection of the 3-D world onto a flat image. More information about the scene can be obtained if the analysis system is successful in recovering the 3-D geometry and motion of objects. The 3-D reconstruction approaches can be coarsely classified in techniques generating volumetric models and techniques reconstructing surface models. A volumetric model is obtained with reconstruction-from-projections approaches as they are known, e.g., from computer tomography. These volumetric models are beyond the scope of this thesis. In a surface model, the objects are represented using only their textured surface. Apart from the object geometry, 3-D reconstruction also includes estimating the 3-D motion of objects as well as the motion of the camera.

Even though general 3-D reconstruction is out of the scope of this thesis, there is a gradual transition between video analysis and 3-D reconstruction. For example, to derive an appropriate model for camera motion, we have to consider the 3-D motion of the camera. It turns out that the depth of a scene is insignificant as long as the camera motion is restricted to rotational motion around a fixed optical center. This type of camera motion plays a central role in the thesis and is therefore examined in more detail. Especially, it is discussed how the physical camera-movements can be recovered from the observed camera motion (Chapter 12). This information establishes a link between the 2-D video image and the 3-D real-world geometry. Knowing this relation, techniques to augment the input video with computer-generated objects are made possible. Thereby, the virtual camera for the generation of the computer images is controled by the parameters extracted from the input video. This has the effect that the virtual camera follows the motion of the real camera, which enables a seamless integration of the virtual objects into the original scene.

The physical camera-parameters are also helpful in the video-content analysis, since the type of camera motion often provides information about

the intention of the editor. For instance, a camera zoom onto a face indicates that this person plays a major role in the scene.

Another case considered in the thesis is the calibration of the camera for sports sequences (Chapter 13). In this application, the calibration establishes the link between the 2-D image coordinate system and a real-world coordinate system. The transformation to absolute coordinates is required for in-depth analysis of the content, since in sports sequences, the position of the players on the playfield is important, but not their position in the image.

Although the thesis does not cover general 3-D reconstruction, the segmentation algorithm employs synthesized background images as a representation of the scene. These background images can also be considered as 360-degree panoramic images, which, when unwrapped, are also rectangular flat images, but covering a full 360-degree panoramic view. In this context, the question arises how these panoramic images can be best visualized to the user. The proposed solution from Chapter 14 is a simplified semi-automatic 3-D reconstruction which recovers the global room geometry to give coarse orientation hints to the viewer.

1.2 The video-object segmentation problem

The task of video-object segmentation is to identify and separate the important *objects* in a video scene from the scene background. Clearly, to approach this problem, it is necessary to define what is exactly meant with *important objects* and how the correct object masks should look like. However, in practice, it turns out that even an unambiguous definition of video objects is a fundamental problem. In the following, the involved definition problems are addressed and grouped into *physical* problems, being a consequence of the image formation, and *semantic* problems. The physical problems are as follows.

- **Shadows.** Objects cast a shadow onto the ground or background. Since this shadow moves with the object, it can be considered part of the object. However, this might be undesired in video analysis applications. For example, if a subsequent object recognition is based on shape information, this can lead to an erroneous object classification. For this reason, special algorithms for shadow identification and removal (Fig. 1.1(a)) have been proposed [72, 162]. In video editing applications, it may be desired instead to reproduce a similar shadow on a new background [27, 20].

- **Reflections.** The problem of handling reflections is actually similar to object shadows. However, reflections are more difficult, because the appearance of the reflected image depends on the physical properties of the reflecting surface (Fig. 1.1(b)) and because the reflection is not necessarily attached to the object.

- **Occlusions.** The object shape can also change because of occlusions. It depends on the application whether the masks of occluded objects should be extended to their original shape.

- **Translucent objects.** Objects can appear partially translucent since they are made of translucent materials, or because thin structures like hair or cloth appear translucent. Moreover, pixels along object boundaries are always a mixture of foreground color and background color. To model the translucency, the segmentation algorithm has to compute an *alpha-channel* mask which identifies the translucency factor for each pixel instead of only computing a binary object mask. Accurate alpha-channel information cannot be obtained from a single image, but algorithms using heuristic approaches have been proposed [26, 161].

Apart from the physical problems, there are semantic definition problems, like the following.

- **Objects of interest (foreground objects).** The first and obvious question of video segmentation is what parts of an image constitue the *foreground objects*. This issue is already surprisingly difficult, since the intuitive human understanding of foreground objects is strongly depending on the scene context. Mostly, human intuition expects that this should be the main acting objects. For example, in a sports broadcast, the players are usually considered foreground and the audience is considered background, even if the audience is moving (see Fig. 1.1(c) for an example). This distinction is on a very high semantic level, since it assumes knowledge about the meaning of the scene. Note that the object definition can also vary with the application. A surveillance system in a sports stadium will be interested in other objects than a system for automatic analysis of the sports game.

- **Small background movements.** When taking a more detailed view on the last point, it can be observed that the distinction between foreground and background is in fact gradual. The question is to what extent a background should change such that it is considered part of the foreground. For example, trees may occur in the background with

(a) Segmentation without and with shadow removal.

(b) Semantically difficult segmentation with reflections and transparencies.

(c) Background objects: not only the players are detected, but also the referee on the right side.

(d) Occlusion: the car passes behind a pole, which separates the car into two parts.

Figure 1.1: *Various segmentation problems, such as shadows, reflections, and occlusion.*

leaves moving slightly in the wind, or there may be a clock on a wall at the back of the room.

- **Object-status change.** Objects can also change their classification over time. For example, most people would consider a car that drives along a street as an important object. But how to define the object status when the car stops and parks at the side of the street? Alternatively, the opposite case may occur that a car that was parked for a long time suddenly drives away. Note that it is practically impossible to separate all objects, including the static ones, into independent objects, since this would imply that all future actions would have to be predicted.

- **Multi-body objects.** Objects may be separated into several non-connected regions in the image. One reason for this can be that an occluding object cuts the foreground object into pieces. For example, see Fig. 1.1(d). Another complex example are objects that are really composed of several parts but still belonging together like flocking birds.

- **Hierarchical objects.** Additional to multi-body objects, there can also exist a hierarchical relationship between objects. One example is a *car* object that contains a *driver* object.

When considering all of these problems simultaneously, it can only be concluded that a general-purpose segmentation of video objects is virtually impossible, since the definition of the expected output from the algorithm depends largely on the scene context and the application that we have in mind. However, despite all the mentioned problems, it is still possible to design algorithms that cover a multitude of specific applications and that work well in many practical cases.

1.3 Object-oriented video coding in MPEG-4

The MPEG-4 video-coding standard is the first and to date only video-coding algorithm that supports the coding of arbitrarily-shaped video objects. In the terms of MPEG-4, a video is composed of several independent Video Objects (VOs) that can be placed in front of a background image (*sprite*). This background sprite image can be larger than the display size, such that only part of the sprite is shown at a time. This concept enables an efficient transmission of video sequences with camera motion, since the current background view can be obtained from the sprite buffer. Additionally, only the foreground objects have to be transmitted to the decoder. After decompression, the decoder superimposes the foreground objects onto the background image.

This coding architecture has two pronounced advantages. First, it provides for a potentially higher compression ratio, since the background is only transmitted once and the foreground objects are considerably smaller than the complete picture. Moreover, the video quality can be regulated on a per-object basis, assigning a larger bit-rate for the important objects and a smaller bit-rate to the background [136, 188]. Second, the separation of objects allows for new possibilities to interact with the content. Video objects can be extracted from one sequence and placed into a different scene. Note that it is also possible to remove objects from the scene. This does

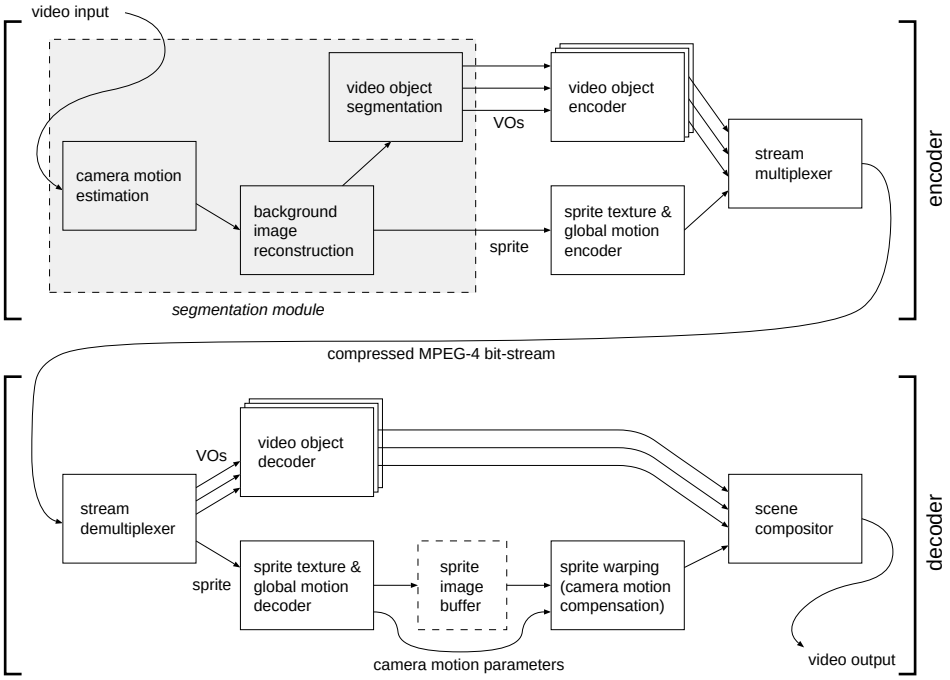


Figure 1.2: *Overview of an object-oriented MPEG-4 video coding system.*

not result in an undefined hole in the image because the background can be obtained from the sprite image.

An overview of a typical object-oriented MPEG-4 video coding system is depicted in Figure 1.2. At the encoder side, the input video is analyzed and split into independent video objects (VOs) in the segmentation module. Instead of the indicated automatic segmentation module, the video objects can also be generated by other techniques, like synthetic content, or they may be recorded separately and segmented using a blue-screen technique. The output of the segmentation module is formed by several video objects that comprise the texture data and also the shape of the independent objects. Moreover, the segmentation module also generates a sprite image of the scene background and suitable camera-motion parameters to reconstruct the original camera motion.

Video objects and sprite data are encoded independently into separate elementary bitstreams by an MPEG-4 compliant encoder and multiplexed into a single bitstream. At the decoder side, the received bitstream is split into the elementary bitstreams and passed to the texture and VO-shape

decoders. Note that the sprite image is only transmitted at the beginning of the scene and then stored in a sprite image buffer. The scene background image is then reconstructed by displaying a geometrically transformed part of the sprite-buffer content (using a projective transformation). Finally, the video objects are superimposed onto the scene background in the scene compositor using the VO shape masks.

1.4 Automatic video segmentation system (Thesis Part I)

Part I of the thesis (Chapters 2-8) describes a generic, automatic segmentation system. The motivation for this segmentation system is to implement a video-object segmentation module that can be integrated into an MPEG-4 encoding framework, as depicted in Figure 1.2. The requirement that the segmentation system should be compliant to MPEG-4 imposes restrictions onto the algorithm design. These design constraints are described in the subsequent section. Afterwards, an overview of the segmentation system is presented, briefly summarizing the processing carried out in each algorithm step.

1.4.1 Design goals

A principal design criterion for our segmentation system is to build a system that is compatible with the object-oriented video-coding tools as defined in the MPEG-4 video-compression standard [93]. It was discussed previously that it is not feasible to target an automatic segmentation system operating unambiguously for every possible input video. Consequently, we restrict ourselves to a limited, well-defined case which nevertheless enables a broad range of applications. More specifically, the proposed system is based on the following two fundamental assumptions.

- **Static background.** We assume that the scene background is static. Camera motion is allowed (see next point), but every object that changes its appearance relative to a static background is considered foreground.[1]

- **Rotational camera motion.** We assume that the recording camera is a pan/tilt/zoom camera. This means that the camera is allowed to rotate around its optical center in any direction, and it may also

[1]Note that this definition implies that shadows and reflections are extracted as part of the object and that "background objects" like the audience in a sports broadcast are also considered as foreground when they are moving.

change the focal length (zoom). However, translatorial camera motion is not allowed because in such a case, the parallax effect would make it impossible to synthesize a static background image.

These assumptions are sensible for many practical applications. For example, most surveillance cameras are pan/tilt/zoom cameras observing objects in a static environment. Moreover, in many typical television scenes like interviews or sport transmissions, several cameras are used at a fixed position and the operators switch only between the views. The restriction to rotational camera motion is also required in order to be compatible with the MPEG-4 video-compression standard, which only supports this type of camera motion for the background sprites.

1.4.2 Segmentation-algorithm overview

The segmentation system developed in this thesis is based on the background-subtraction approach. In this technique, the segmentation algorithm compares the input images to a background image. The background image is a synthesized view of the scene background in which all foreground objects have been removed. Regions in the input image that differ from the background image are marked as foreground objects (Fig. 1.3).

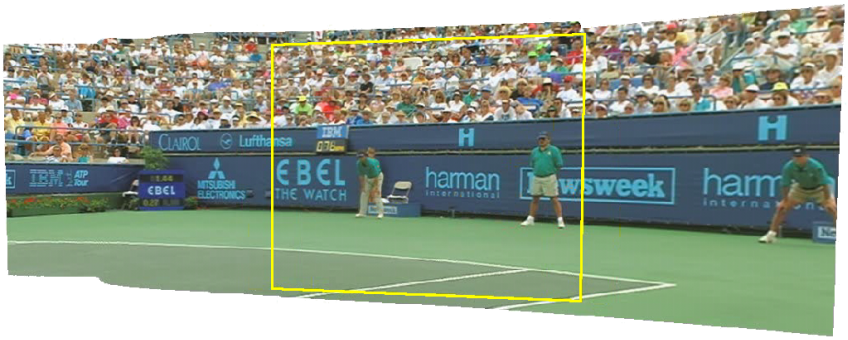
To compute the background image, the camera motion is estimated and by compensating this motion, the input images are composed into a background image. If the camera is panning across the scene, this results in panoramic background images which are larger than the input image. Finally, the background image is reconstructed from the input sequence by integrating all frames of the sequence such that non-static foreground objects are removed from the image.

The synthesized background images together with the extracted foreground objects match the input that is required for the MPEG-4 object-oriented video coding tools. Hence, the output of our segmentation algorithm can be directly used as input for an object-oriented MPEG-4 video coder.

1.4.3 Framework of the segmentation algorithm

This section briefly describes the processing steps of the developed segmentation system (see Figure 1.4) and introduces the core algorithms that are used.

(a) Input image. (b) Background view. (c) Segmented foreground.



(d) Sprite image.

Figure 1.3: *Principle of the segmentation algorithm. All input images (a) are combined into a static background sprite (d). The current camera view (marked quadrilateral) is extracted and dewarped (b). By comparing the input (a) and the background (b), the foreground object (c) is obtained.*

Camera-motion estimation

The largest part of the system is the camera-motion estimation. The difficulty of the camera-motion estimation is that it must be robust to foreground object motion, and that it also has to give very accurate motion parameters such that all input frames can be combined seamlessly into a background image. To achieve this, we apply a combination of a feature-based motion estimator and a direct estimation algorithm.

Chapter 3 starts the presentation of the feature-based motion estimator with the detection of feature-points that can be retrieved reliably in a subsequent frame. In the next step, correspondences between matching points are established. Each feature-correspondence can be viewed as the motion of this point between the two images. Chapter 3 describes several feature-point detectors (SUSAN, Harris, Shi-Tomasi, and Moravec), which are evaluated for their accuracy and robustness, for a wide variety of

video sequences. Moreover, the chapter describes an algorithm to compute feature-correspondences across pairs of images. The algorithm features prediction of the matching feature-points and a fast neighborhood-search to achieve a low computation time.

Chapter 4 discusses the estimation of the global-motion parameters from the previously computed feature-correspondences. First, the problem is considered on sequences with pure camera motion without foreground objects. Subsequently, the estimation problem is considered for the general case that the observed motion is a mixture of camera motion and object motion. Robust estimation algorithms are introduced to estimate the parameters of the dominant motion. Chapter 4 concentrates particularly on the RANSAC algorithm, while other estimators are considered in Appendix C. Finally, the chapter presents why the robust estimator breaks down earlier in practice than predicted from theory. Based on the discussion of this phenomenon, the RANSAC algorithm is modified to increase its robustness to about the theoretically predicted performance.

The first part of **Chapter 5** presents the direct motion-estimation algorithm to refine the motion parameters obtained from the feature-based motion estimator. The obtained motion parameters are a good estimate for the inter-frame motion, but they are not accurate enough to build a global background image. The direct motion estimator is a gradient-based global-motion estimator, which computes long-term motion parameters between each input frame and the background sprite. Compared to the short-term motion parameters obtained from the feature-based motion estimator, the long-term parameters have a higher accuracy, since there is no accumulation of errors as would happen with a concatenation of inter-frame motion parameters. However, the gradient-based estimator should be initialized with the result of the feature-based motion estimator, because the area of convergence is smaller and it thus requires a good initialization.

Background reconstruction

The second part of **Chapter 5** discusses the synthetization of static background-sprite images with all foreground objects removed. Knowing the camera-motion parameters, a background image can be synthesized by stitching the input images to a common background image using the obtained parameters. However, the essential problem in this step is to remove the foreground objects from the background image. The chapter presents a new algorithm based on the observation that each region in the image can be classified into one of three states: *static background*, *moving foreground*, *non-moving foreground*. The difficult case in this classification is the distinction between static background and temporally non-moving fore-

ground. This classification problem is solved by building clusters of regions with stable content in temporal direction, and by considering that the times in which foreground appears in neighboring regions are similar. Finally, the proposed algorithm is compared to other algorithms, especially the median algorithm, which is the best previously known algorithm.

Our approach of generating background images uses the same motion models as those that have been defined in the MPEG-4 and MPEG-7 standards to describe camera motion. However, it is shown in **Chapter 6** that this approach does not work for all kinds of camera motion. This problem has not been considered in previous work, even though it leads to major problems when camera rotations of large angles are present. For example, the MPEG-4 sprite-coding approach becomes inefficient for camera rotation angles larger than approximately 25 degrees. Moreover, *ad-hoc* implementations for sprite generation usually do not consider that camera zooming changes the image resolution and, if the higher resolution is not reflected in the sprite, the reconstructed view from the sprite misses small details. Chapter 6 first shows theoretically that all of these problems can be solved by computing a set of independent sprites (a *multi-sprite*) instead of trying to compute a single sprite representation. Afterwards, a novel *multi-sprite partitioning* algorithm is presented, which partitions the video sequence into a number of segments, for which independent sprites are synthesized. The partitioning is computed in an optimal way, such that the total area of the resulting sprites is minimized. Furthermore, the algorithm can incorporate constraints, such as a limited sprite-buffer size at the decoder, or the restriction that the image resolution in the sprite should never be lower than the input image resolution. The described multi-sprite approach is compatible to the MPEG-4 standard, and yet provides several advantages: any arbitrary rotational camera motion can be processed, the coding-cost for transmitting the sprite images is lower, and the quality of the decoded sprite images is better. In Figure 1.4, the multi-sprite algorithm is depicted as an extension to the baseline segmentation algorithm.

Foreground object segmentation

Chapter 7 describes the actual foreground-object segmentation, which is based on a background-subtraction technique. Input images are compared with the corresponding camera-motion compensated view from the synthesized background sprite and areas that deviate are marked as foreground. At first, a classification of independent pixels is considered, where the influence of the color-space and the difference measure is evaluated. Subsequently, multi-pixel based tests and Markov random fields are used to derive the foreground mask with improved accuracy. Moreover, the concept

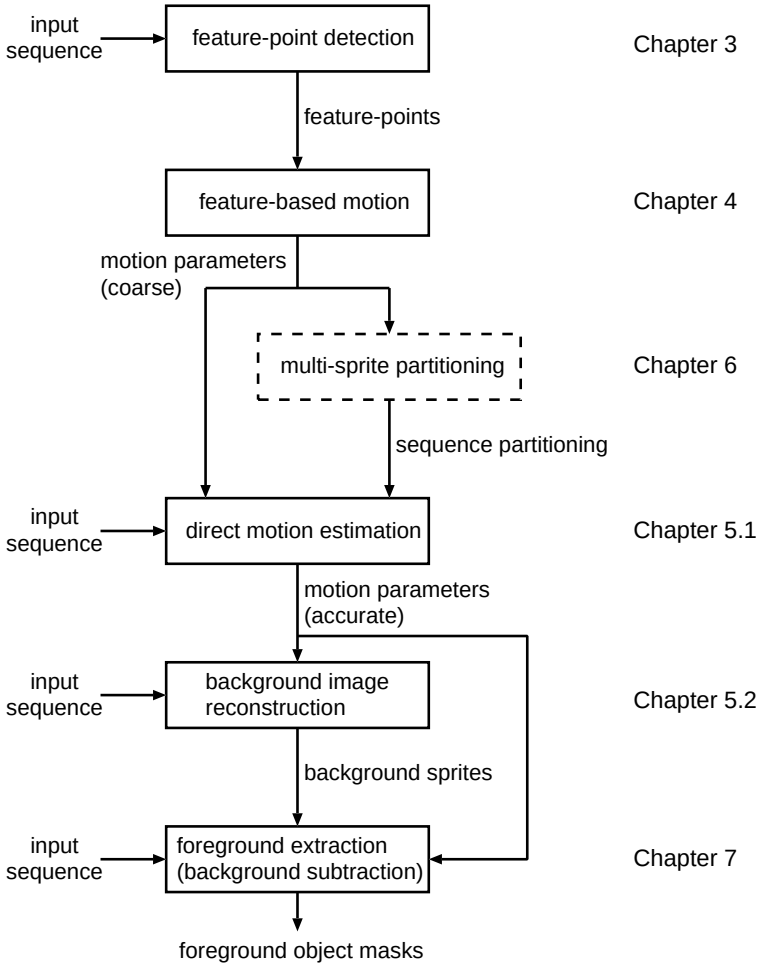


Figure 1.4: *Overview of the segmentation system.*

of *risk maps* is introduced to account for the problem that the background image may not be perfectly aligned to the input images, e.g., because of inaccuracies in the motion-estimation. These risk maps significantly reduce the errors caused by misregistration and by blurring of the background image that occurs in the image warping. Finally, post-processing filters are described that remove clutter regions from the segmentation mask.

System architecture

Chapter 8 discusses that a complete segmentation system comprises all

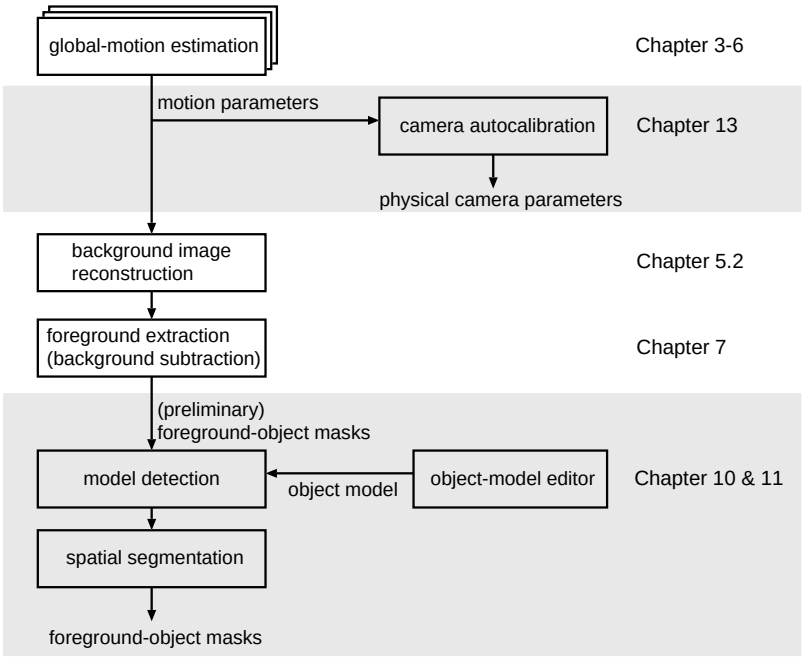


Figure 1.5: *Extensions to the segmentation system.*

or a selection of the processing steps outlined above. Depending on the application and type of sequences to be processed, the system can be implemented in different variations. For example, the algorithm can be simplified for a surveillance application with static cameras, or the application may require an online real-time segmentation. Furthermore, this chapter provides results of the segmentation algorithm on a wide variety of input sequences and typical effects and problems of the segmentation algorithm are discussed. Finally, example applications for the described segmentation algorithm are presented. This includes MPEG-4 video coding, for which the gain in the compression ratio is discussed, object-based video editing, pseudo 3-D video generation, or object recognition.

1.5 Extensions to the segmentation system

The segmentation system outlined in the previous section can be regarded as the core framework, which can be adapted to many specific applications. Some of these possibilities are the subject of Part II and Part III of the thesis. These two parts relate to the two research directions *object models* and *camera models* that are considered as particularly interesting.

1.5.1 Segmentation using object models (Thesis Part II)

A first possibility is the integration of *model knowledge* about the objects to be segmented into the segmentation algorithm. The segmentation algorithm outlined so far has no explicit knowledge about the objects to be extracted. However, if object models should be added to the segmentation system, the central question is how the object description should be defined. It is important to balance between an object definition that is accurate enough to uniquely identify the object, and a definition that allows for enough freedom to recognize the object in different views.

Graph-based object models

In **Chapter 9**, a graph-based object model is presented. In this model, the main regions of the object and the region features are summarized in the graph nodes, and the spatial relations between these regions are expressed with the graph edges. The approach is first described in Chapter 9 for the special case of cartoon sequences, since for this type of sequence, the object regions can be obtained easily with color segmentation. The graph-based object-detection system consists of two parts. First, the user defines the object model based on an example image of the object. At the detection stage, the algorithm applies an automatic color segmentation onto the input image to obtain a similar, but much larger graph of the input image. Using an efficient sub-graph matching algorithm, the object is identified in the input image.

Chapter 10 extends this concept to the detection of objects in natural video sequences. A similar graph-model is used with the only difference that the region shapes, which cannot be extracted easily using color segmentation, are approximated using ellipses. The concept of the object-detection algorithm is similar to the algorithm for the cartoon sequences, except that for each model region, a set of candidate regions is first extracted from the input image. This set comprises several possible placements of ellipses in the input image to cover areas that have similar color and size as indicated in the object model. Additionally, the algorithm can integrate change-detection masks providing a coarse hint about the location of the objects. The final object segmentation is carried out with a color segmentation algorithm that is modified such that the object regions are restricted to the areas covered by the detected object position.

The graph-based object models can be integrated into the core segmentation system as depicted in Figure 1.5. In this configuration, the segmentation result of the core segmentation system is applied only as a coarse indication of object location. Note that this indication can be incomplete

if only a part of the object is detected in the segmentation system (e.g., see Fig. 8.14). The subsequent *model detection* step uses this first indication of the object location and the object model that has been created manually with the *object-model editor* to determine the image area that comprises the object. Accurate pixel-level object boundaries are computed in the final *spatial segmentation* step.

Object signatures for tracking

Chapter 11 presents a different approach to describe specific objects. The segmentation problem is approached from a different perspective. For applications requiring highly accurate segmentation masks, the quality provided by an automatic segmentation may not be sufficient. For these cases, it should considered to use a semi-automatic segmentation algorithm where the user controls the segmentation, but the computer relieves him from working at the pixel level. To this end, an advanced algorithm based on the concept of the Intelligent Scissors algorithm [130] is presented. This is an edge-based segmentation algorithm, in which the user traces along the object edge to define the boundary. This approach is generalized to search the object contour in a user-drawn *corridor* that he draws along the object boundary. The exact object boundary is obtained using a newly developed shortest circular-path search algorithm.

Apart from the semi-automatic segmentation algorithm, the chapter deals with the novel concept of object signatures. The object signature is defined as the image texture along the object boundary. Once this object signature is known, for example from a segmentation of the first frame of a sequence, its information can be integrated into the segmentation of the successive frames. This enables an automatic tracking of the object through the sequence without manual intervention. The tracking step can also be added to the core segmentation system in order to carry out the computationally expensive segmentation only for one frame and then switch to the more efficient object tracking.

1.5.2 From camera motion to 3-D models (Thesis Part III)

During the work on camera-motion estimation, it was observed that there is a close connection between the camera-induced motion in the image and the scene geometry. This relation between the 2-D image and the 3-D world becomes especially important when the segmentation result is used to analyze the video content. Whenever object motion is used for the analysis, the object motion in the image has to be translated to motion in the real-world, because the motion in the image is composed of the object

motion as well as the camera motion. Different techniques are explored to derive information about the physical 3-D world for the special case of rotational camera motion.

The core segmentation system employs camera-motion estimation to compensate for any camera motion. When we derived the camera-motion model in Chapter 2, we started with a 3-D model of the environment and the image-formation process of a camera. Based on this physically motivated model, the projective motion model was derived, which was subsequently used in the camera-motion estimation. However, although the motion model was derived from a physical description including rotation angles and the focal length of the camera, it is not easily possible to recover these parameters from the parameters of the estimated projective motion model.

Chapter 12 adresses the inverse problem of factorizing the motion parameters into physically meaningful parameters using camera autocalibration techniques. Our approach uses first a linear estimation approach based on the concept of the *image of the absolute conic*. To refine the motion parameters, the accuracy is further increased with a non-linear optimization similar to *bundle-adjustment* techniques. The speciality of the new algorithm is that it can integrate camera motion that spans several sprites according to the earlier introduced multi-sprite technique. Consequently, the algorithm can be applied to arbitrary unrestricted rotational camera motion.

For the analysis of video sequences, it is often required to know and follow the position of the objects. Clearly, the object position in terms of image coordinates provides little information as long as the viewing direction of the camera is not known. In some application, like sport videos, the camera view can be determined from markings on the ground of the playing field. **Chapter 13** provides a new algorithm to deduce the transformation between the image coordinates and the real-worlds coordinates, based on the lines defining the playfield. The theory behind this approach is closely related to the estimation of camera motion, since also the mapping between two image planes (the image and the flat real-world ground plane) is estimated. However, the difference is that in this case, a mapping onto absolute coordinates is obtained.

The camera calibration for sports sequences employs a special model for the arrangement of lines in the playfield, which is usually defined in the rules of the game. After detecting lines in the input image, a combinatorial search is carried out to establish correspondences between lines in the input image and lines in the model. Comparable to the feature-based motion estimator, motion parameters are deduced from a set of corresponding lines. To reduce

the overall computation time, a tracking step is additionally presented that updates the transformation parameters during camera motion with reduced computational complexity.

Chapter 14 describes a step towards the reconstruction of 3-D models from video images. During the work on background sprites and their generalization to multi-sprites, the question arose how the sprite images, which are informative pictures by themselves, can be best presented to the user. Usually, very wide-angle images are presented in the form of panoramic images, which are actually a mapping of the environment on a cylinder instead of a plane, as it is the case for MPEG-4 compliant background sprites. However, the disadvantage of panoramic images is that complete 360-degree views are unwrapped into one rectangular image with the consequence that all straight lines in the image become bent and, more importantly, the viewer has no good orientation in the image because he looks into all directions at the same time. In order to provide a more intuitive presentation for wide-angle views, a visualization technique is developed which is specialized for the case of indoor environments. The visualization program recomputes the 3-D shape of the room in which the image was captured and projects the panoramic image onto these virtual room walls. The advantage of this presentation is that the room shape helps the user in the orientation, making it clear which part of the image corresponds to which wall. This concept is further generalized with an algorithm to reconstruct the complete floor plan from several panoramic images. This enables to conduct virtual walk-throughs in the reconstructed rooms.

1.6 Contributions of the author

Most parts of the chapters in this thesis have been published in conference proceedings or scientific journals. An overview of which chapters are covered by corresponding publications is summarized in Table 1.1.

Part I - An Automatic Video-Object Segmentation System

The concept of Part I of the thesis is to provide the reader with a complete discussion of a segmentation system in every detail. Because of this, some chapters also comprise background information, additional to the contributions of the author. In particular, Chapter 2 provides an introduction to projective geometry, which can be skipped by the reader who is familiar with this topic.

The implementation of a complete segmentation system is a difficult task because of the complexity of the system. Various algorithm types and

techniques covering many research areas have to be combined, like global-motion estimation, feature extraction, statistics (Markov Random Fields), and various linear, non-linear, and combinatorial optimization techniques. Each of the processing steps has to be designed carefully, since a low accuracy in one step can lead to a complete failure of the system. Consequently, it was of significant importance to evaluate different alternatives for each processing step and select those algorithms that provide the most robust result when combined. For example, only the combination of the feature-based motion estimator with the direct estimation leads to high-accuracy parameters as well as robustness against fast camera motion, and the accuracy limitations of the motion estimator requires the adaptation of the change-detection algorithm with the risk-map approach.

An important algorithmic invention is the use of multi-sprites as a replacement for single static background images (Chapter 6). This technique has made it possible for the first time to process arbitrary camera motion. Although this is a crucial part to enable a practical implementation, this problem has been overlooked in the literature. The attractivity of our multi-sprite approach is that the problem is solved in an optimal way, also minimizing the MPEG-4 sprite-coding cost.

The two papers [59, 55] about multi-sprite coding both received the *Best Student Paper* award at the SPIE Visual Communications and Image Processing conference 2004, and at the 24th Symposium on Information Theory in the Benelux, 2003.

Another new development is the algorithm for background synthetization (Section 5.3). Compared with previous algorithms, this new algorithm also succeeds in reconstructing the background if it is visible for only short periods of time. This is important, because for a given video-sequence, the total observation time can be short (only one camera-pan). Because of this reason, existing algorithms that were primarily designed for background reconstruction in surveillance video, where the same scene is observed for a long time, cannot be applied.

The segmentation system was summarized in a book chapter [62] and it will also be presented by the author as a tutorial at the IEEE International Conference on Consumer Electronics 2006. Furthermore, the research on segmentation and object models (see below) has led to the organization of a special session about content analysis at the same conference.

Part II - Segmentation using object models

Because we made the observation that a semantically meaningful segmentation requires pre-knowledge about the object to be extracted, we explore in Chapters 9 and 10 how this object description can be specified.

We combined ideas of image databases supporting region-based queries [23, 17, 112] with articulated-object models as they are used for object-tracking applications [154, 70]. This is implemented in a new integrated framework supporting the creation of object models from sample images, as well as two algorithms for detecting the objects in real-world or cartoon images. We extended the concept of a 1 : 1-matching as it is used for tracking to an $1 : N$-matching to enable a complete coverage of the object to be segmented. The object-detection algorithm uses a cascade of steps (candidate-region detection, skeleton-tree based graph-matching, extension of the mapping from an isomorphism to an homomorphism) to be computationally efficient. Finally, the object detection is combined with a color segmentation to obtain accurate object boundaries.

In Chapter 11 a new approach for semi-automatic segmentation is developed. Based on the Intelligent Scissors algorithm, we present an interactive segmentation tool that is easier to use and which also comprises a tracking component. Our tool replaces the shortest-path search with a shortest circular-path search. This not only provides a more intuitive user-interface, but it is also used in the tracking step. The tracking step is special because it uses a model of the object that is derived automatically from a previously segmented image. A main innovation in this chapter is the development of the circular-path search algorithm. It is the fastest algorithm currently known for planar graphs, with a typical computation complexity equal to the ordinary shortest-path search. This algorithm is generic and can be used for many applications apart from our manual segmentation tool, such as shape matching.

Part III - From camera motion to 3-D models

The contribution of Chapter 12 is the integration of the multi-sprite concept into the autocalibration for rotational cameras. This enables the recovery of physical camera parameters from projective-motion parameters for unrestricted camera motion.

In Chapter 13 discusses a special kind of model: the model of a sports court to compute camera-calibration parameters. In order to obtain a robust court detection, the algorithm is based on line features. Similar to the graph-based object models, we use a combinatorial optimization to establish the correspondences between image features and the model. Our results show that a specialized model can lead to highly robust object detection, invariant to observation conditions like illumination or court colors. It is also interesting to note that the robustness of the algorithm allows to adapt it to various kinds of sport by simply exchanging the court model.

Later, the work on camera calibration for sport sequences was integrated

into the Philips Cassandra demonstrator [133] that was presented at the IEEE International Conference on Multimedia and Expo (ICME), 2005. The research in this area has also led to a special session on that topic at the same conference, co-organized by the author and Xinguo Yu. The calibration algorithm also builds the basis of ongoing research on sports analysis, extending it to a complete analysis system [81]. Furthermore, several international research groups employed our algorithm in their tennis or soccer analysis systems [89, 116].

Due to the work on background sprites, the author was invited in 2003 to stay at the *Stanford Center for Innovations in Learning* (SCIL) in the context of the *Diver* project [142], in which the human interaction with panoramic-video content was studied. In this time, the author developed the room reconstruction algorithm for rectangular rooms, presented in Chapter 14. Later, the algorithm was extended for general floor plans. A main contribution is the new approach of combining pre-knowledge about the room shape with measurement data from the panoramic image. Using the panoramic image for obtaining measurements is convenient since internal camera parameters like the focal-length are easily computed from the panoramic image instead of being estimated a difficult process.

Appendices

Apart from the work on segmentation, the author contributed to the German BMBF project "L3-Lifelong Learning". In this project, the author contributed a video-database application featuring an automatic video-summary generation (Appendix A). Furthermore, the segmentation system has also been integrated into a video-abstracting system that was developed in the context of the European ECHO project (European CHronicles Online) [104]. In this project, a video-archiving system for historical films was established.

Appendix E describes early work about color-segmentation that has later been integrated into the model-based object detection. This color segmentation features a new speed-improved variant of the region-merging algorithm and a multi-stage approach, in which the merging criterion is switched during the segmentation. The paper about multi-stage segmentation [12] also received a best student paper award at the 22nd Symposium on Information Theory in the Benelux, 2001.

Chapter	Publication title and contribution
	Early work about content-adaptive MPEG-2 encoding
1	**"A Software-Based High-Quality MPEG-2 Encoder Employing Scene Change Detection and Adaptive Quantization", IEEE Trans. on Consumer Electronics, 2002, [67]**
	and
1	**"SAMPEG, a Scene Adaptive Parallel MPEG-2 Software Encoder", SPIE VCIP, 2001, [66]**
	Preparatory work: implementation of a parallel MPEG-2 encoder. Includes video content analysis to control the adaptive quantization separately for different types of content. Also includes scene-change detection for adapting the GOP pattern.
1	**"Rate-Distortion Optimal Adaptive Quantization and Coefficient Thresholding for MPEG Coding", 23rd Symposium on Information Theory in the Benelux, 2002, [64]**
	Development of a theoretically optimal encoder for MPEG-2 I-frames, yielding the highest possible PSNR for quality comparison to adaptive quantization approaches.
	Part I - An Automatic Video-Object Segmentation System
2–8,F	**Book chapter "Segmentation and Classification of Moving Video Objects", in "CRC Handbook of Video Databases", 2003, [62]**
	Overview of the core segmentation system, excluding multi-sprite segmentation. Presents different motion-models, feature-based and dense motion-estimation, Markov Random Field segmentation. Also discusses object recognition based on the object shape.
3–8	**"Video-Object Segmentation using Multi-Sprite Background Subtraction", IEEE ICME, 2004, [60]**
	Overview of the segmentation system with special consideration of using multi-sprites.
3,4	**"Evaluation of a Feature-Based Global-Motion Estimation System", SPIE VCIP, 2005, [50]**
	Evaluation of different feature-point detectors. Efficient feature-correspondence algorithm. Comparison of linear vs. non-linear parameter estimation. Improvement of the robustness of the RANSAC parameter estimation algorithm.
5	**"Robust Background Estimation for Complex Video Sequences", IEEE ICIP, 2003, [57]**
	Algorithm for synthesizing background images in which moving foreground objects are removed. The algorithm is particularly robust even if the background is only visible for short periods of time.

Chapter	Publication title and contribution — continued
6	**"Optimal Partitioning of Video Sequences for MPEG-4 Sprite Encoding", 24th Symposium on Information Theory in the Benelux, 2003,** [**55**] Observation that MPEG-4 sprite-coding is not applicable to wide camera pans. Proposal of an algorithm that removes this limitation and which provides a significant reduction of bit-rate for the transmission of sprite images. **(Award paper.)**
6	**"Minimizing MPEG-4 Sprite Coding-Cost Using Multi-Sprites", SPIE VCIP, 2004,** [**59**] Extended paper of [55]. Enhanced algorithm now supports additional constraints like a maximum decoder sprite-buffer size. Also provides a better approximation of sprite-coding cost based on the sprite area. **(Award paper.)**
6	**"Enabling Arbitrary Rotational Camera-Motion Using Multi-Sprites with Minimum Coding-Cost", IEEE Trans. on Circuits and Systems for Video Technology, accepted for publication,** [**54**] See above.
6,8	**"Automatic Video-Object Segmentation Employing Multi-Sprites with Constrained Delay", IEEE ICCE, 2006,** [**53**]**, (will also be submitted as journal paper)** Overview of the segmentation system with special consideration of the real-time implementation of multi-sprites.
7	**"A New Similarity Measure for Sub-Pixel Accurate Motion Analysis in Object-Based Coding", ISAS SCI, 2001,** [**47**] Improved distance metric to reduce matching error in the presence of aliasing and misregistration. **(Invited paper.)**
7	**"Misregistration Errors in Change Detection Algorithms and How to Avoid Them", IEEE ICIP, 2005,** [**51**] Change-detection algorithm is improved by estimating a map indicating regions with risk of misregistration errors. Risk map is integrated into change-detection algorithms to improve robustness against inaccuracies in the motion estimation and aliasing noise.
	Part II - Segmentation Using Object Models
9	**"Recognition of User-Defined Video Object Models using Weighted Graph Homomorphisms", SPIE IVCP, 2003,** [**56**] Specification of objects by attributed graph models. Detection of the user-supplied object model by detection of graph homomorphisms ($1 : N$ mapping). Efficient implementation using a matching algorithm based on dynamic programming. Algorithm is specific to cartoon sequences.

Chapter	Publication title and contribution — continued
10,E	**"A Segmentation System with Model Assisted Completion of Video Objects", SPIE VCIP, 2003,** [**58**] Graph-based object models are applied to the detection of objects in natural video sequences. Model matching yields approximate location of object. Accurate boundaries are computed by combining the positional hint with pixel-level color segmentation.
E	**"Multi-Stage Region Merging for Image Segmentation", 22nd Symposium on Information Theory in the Benelux, 2001,** [**12**] A color-segmentation algorithm based on region-merging that features a multi-stage segmentation, where the segmentation criterion is changed during the segmentation to adapt it to the typical signal characteristics at the respective stage. **(Award paper.)**
E	**"Towards Real-Time MPEG-4 Segmentation: A Fast Implementation of Region-Merging", 21st Symposium on Information Theory in the Benelux, 2000,** [**46**] Fast implementation of region-merging color segmentation.
11	**"Corrisor Scissors: A Semi-Automatic Segmentation Tool Employing Minimum-Cost Circular Paths", IEEE ICIP, 2004,** [**68**] Development of the *Corridor Scissors* segmentation algorithm as an extension of the *Intelligent Scissors* algorithm. A first fast algorithm for shortest circular paths is presented.
11	**(drafted as journal paper)** Enhancement of the circular-path search algorithm to remove special cases and achieve a high computation speed for any input data.
	Part III - From Camera Motion to 3-D Models
12	**"Estimating Physical Camera Parameters for 3DAV video coding", 25th Symposium on Information Theory in the Benelux, 2004,** [**48**] and **"Estimating Physical Camera Parameters based on Multi-Sprite Motion Estimation", SPIE IVCP, 2005,** [**49**] Auto-calibration of rotational cameras. Rotation angles and focal-length are estimated from the projective-motion parameters obtained in the segmentation system. Speciality of this algorithm is that it supports the estimation also in the case of unrestricted camera motion, since the multi-sprite technique is integrated.

Chapter	Publication title and contribution — continued
13	**"Robust Camera Calibration for Sport Videos using Court Models", SPIE Storage and Retrieval Methods and Applications for Multimedia, 2004, [65]** Estimation of mapping between image coordinates and real-world coordinates for sport sequences, based on a model of the playfield. Lines are estimated and a set of four corresponding lines between the image and the model are determined. Also supports a tracking step to reduce the computational complexity after initial calibration.
13	**"Fast Camera Calibration for the Analysis of Sport Sequences", IEEE ICME, 2005, [63]** Enhancement of the camera calibration algorithm that reduces the computation time by only requiring the matching of two line-segments for the calibration instead of four correspondences.
13	**"Current and Emerging Topics in Sports Video processing", IEEE ICME, 2005, [199] (co-work with Xinguo Yu)** Overview of techniques and applications for sport-video analysis.
14	**"Reconstructing Virtual Rooms From Panoramic Images", 26th Symposium on Information Theory in the Benelux, 2005, [52]** Geometry reconstruction of rectangular rooms from panoramic images. Generation of 3-D room-models where the wall-textures are extracted from the panoramic image.
	Appendices
A	**"Robust Clustering-Based Video-Summarization with Integration of Domain-Knowledge", IEEE ICME, 2002, [61]** Development of a video-summarization algorithm that can incorporate pre-knowledge about scenes that should be excluded from the summary.
A,2-8	**"Automatic generation of video summaries for historical films", IEEE ICME, 2004, [104], (co-authored with Stephan Kopf)** and
A,2-8	**"Automatic Generation of Summaries for the Web", SPIE Storage and Retrieval for Media Databases, 2004, [103], (co-authored with Stephan Kopf)** Video-abstracting algorithm that incorporates the described segmentation system and object recognition to improve the detection of significant scenes.

Table 1.1: Contributions of the author in the respective chapters. If not noted otherwise, the thesis author is also the first author of the paper.