Technische Universität München
Lehrstuhl für Medientechnik

# Spatio-temporal Analysis of Multiview Video

Dipl.-Ing. Univ. Florian Schweiger

Vollständiger Abdruck der von der Fakultät für Elektrotechnik und Informationstechnik der Technischen Universität München zur Erlangung des akademischen Grades eines

Doktor-Ingenieurs (Dr.-Ing.)

genehmigten Dissertation.

| | | |
|---|---|---|
| Vorsitzender: | | Univ.-Prof. Dr. rer. nat. habil. Bernhard Wolf |
| Prüfer der Dissertation: | 1. | Univ.-Prof. Dr.-Ing. Eckehard Steinbach |
| | 2. | Univ.-Prof. Dr.-Ing. habil. Gerhard Rigoll |

Die Dissertation wurde am 17.04.2013 bei der Technischen Universität München eingereicht und durch die Fakultät für Elektrotechnik und Informationstechnik am 10.10.2013 angenommen.

# Spatio-temporal Analysis of Multiview Video

Dipl.-Ing. Univ. Florian Schweiger

2. Dezember 2013

Für *H*

# Acknowledgments

At this point, having compiled and submitted my dissertation, I would like to express my gratitude to all individuals directly and indirectly involved in the process that has led up to this moment.

First of all, I feel obliged to thank my doctoral adviser Professor Eckehard Steinbach for the opportunity to carry out this project under his supervision. During the years at his institute, I benefited from a highly scientific atmosphere, from many fruitful discussions and, above all, his willingness to always lend a sympathetic ear. I would like to take the opportunity to thank Professor Steinbach for the offered guidance, his patience and his commitment towards my work.

I would also like to thank Professor Rigoll for accepting to serve as my second examiner, as well as Professor Wolf for chairing the examination committee.

A special thanks goes out to my coauthors, colleagues and friends for their invaluable support, the many helpful discussions I had with them, and all the inspiration they gave me. First and foremost, I want to thank Georg Schroth, Michael Eichhorn, Anas Al-Nuaimi, Robert Huitl, Burak Cizmeci and Julius Kammerl. On the same note, I want to thank my dear friend Christoph Promberger.

Further, I'd like to extend thanks to my project partners at DOCOMO Euro-Labs, Prof. Kellerer and Dr. Fahrmair, for three years of prolific cooperation.

Last but not least, I want to acknowledge the contributions made by the students I have supervised over the years, and especially thank Bernhard Zeisl, whom I co-supervised together with Pierre Georgel, Engin Gümüsel, Yasir Latif and Serkan Türker.

München, November 2013 *Florian Schweiger*

# Abstract

Multiview video increasingly finds its way into our everyday lives. While only ten years ago the practical relevance of this kind of data was low, the availability of mobile commodity hardware capable of recording high quality video has led to an explosion of produced videos, calling for a intensified examination of multiview video.

The glut of video data being acquired every day and partially being made available online entails technological challenges regarding efficient representation, transmission and storage. Then again, a variety of hitherto unthinkable opportunities arise for novel, highly innovative applications around multiview video. By all means, adequate and effective means for data analysis become necessary.

In this thesis, different approaches and algorithms are presented which aim at enabling and facilitating the analysis of multiview video. The temporal and spatial aspects of video are treated separately. The specific contributions are a novel, fully automatic video synchronization algorithm that is very robust against disturbing influences and highly flexible. Furthermore, the thesis introduces a universal framework to evaluate the precision of multiscale image features. Lastly, a new class of visual markers is presented and, based thereon, two novel and highly efficient multiscale feature detectors.

The thesis hence makes theoretical, but also practically relevant contributions in several areas of spatio-temporal analysis of multiview video.

# Kurzfassung

*Multiview Video*, also aus mehreren Perspektiven gleichzeitig aufgenommenes Video, gewinnt zunehmend an praktischer Bedeutung. Während noch vor zehn Jahren relativ wenig alltäglicher Bedarf bestand, sich mit dieser Art von Daten zu befassen, ergibt sich, getrieben durch die zunehmende Verfügbarkeit erschwinglicher mobiler und videofähiger Geräte und der damit verbundenen rasanten Zunahme an produziertem Videomaterial, mehr und mehr die Notwendigkeit, sich mit diesem Thema auseinanderzusetzen.

Die enorme Anzahl an Videos, die täglich aufgenommen und teilweise auch online verfügbar gemacht werden, stellt einerseits eine technologische Herausforderung dar in Bezug auf die effiziente Repräsentation, Übertragung und Speicherung dieser Daten. Zum anderen ergeben sich bislang ungeahnte Möglichkeiten für neue, innovative Anwendungen rund um *Multiview Video*. In beiden Fällen ergibt sich dabei der Bedarf nach effektiven Verfahren für die Datenanalyse.

In dieser Arbeit werden verschiedene Verfahren und Algorithmen vorgestellt, die die Analyse von *Multiview Video* ermöglichen und vereinfachen sollen. Dabei werden der zeitliche und örtliche Aspekt von Video getrennt voneinander untersucht. Insbesondere wird ein neuartiger, vollautomatischer Videosynchronisationsalgorithmus vorgestellt, der sehr robust gegenüber Störeinflüssen und äußerst flexibel einsetzbar ist. Des Weiteren führt die Arbeit ein universelles Evaluierungssystem ein, mit dessen Hilfe die Präzision skaleninvarianter Bildmerkmale bewertet werden kann. Schließlich wird einen neuer Ansatz zum Entwurf visueller Marker präsentiert und darauf basierend zwei neuartige, hocheffiziente Bildmerkmalsdetektoren.

Damit leistet diese Arbeit wissenschaftliche und zugleich praxisrelevante Beiträge, die mehrere Teilbereiche der zeitlich-örtlichen Analyse von *Multiview Video* umfassen.

# Contents

# List of Figures

# List of Tables

# Abbreviations and Acronyms

**CV** Computer Vision

**SfM** Structure-from-Motion

**QP** Quantization Parameter

**GOP** Group Of Pictures

**VBR** Variable BitRate

**MB** MacroBlock

**TEX** TEXture data

**MV** Motion Vector

**RDO** Rate-Distortion Optimization

**SATD** Sum of Absolute Transformed Differences

**PSNR** Peak Signal to Noise Ratio

**ZNCC** Zero-mean Normalized Cross-Correlation

**RANSAC** RANdom SAmple Consensus

**ConCor** Consensus-based cross-Correlation

**PCCF** Partial Cross-Correlation Function

**NPCCF** Normalized PCCF

**LoG** Laplacian of Gaussian

**DoG** Difference of Gaussians

**SIFT** Scale-Invariant Feature Transform

**ASIFT** Affine-invariant SIFT

**SURF** Speeded-Up Robust Features

**suSURF** speeded-up SURF

**AsuSURF** Affine-invariant suSURF

**MDR** Maximum Detector Response

**MDRM** Maximum Detector Response Marker

**VIS** Version-Independent Signature

**AWGN** Additive White Gaussian Noise

**CAVE** CAVE Automatic Virtual Environment

**GPU** Graphics Processing Unit

**GPGPU** General-Purpose GPU

# 1 Introduction

In the 1990s, the term *multimedia* was popularized, describing the convergence of different types of media in single applications in order to improve the tangibility of concepts presented to the user. At the same time, static content was more and more abandoned in favor of more dynamic forms of presentation, increasingly encouraging user interaction. An important catalyst for this development was digital video which found its way into a then still young and mostly text and image based Internet. Back in these days, video hence played an indispensable, if not the defining role in the emergence of multimedia.

Two decades later, with haptic communication still in its infancy, and olfactory or even gustatory data communication far out of reach, audio-visual data is still the most prevalent form of multimedia today. An interesting turn that video communication took in the early 2000s, and with it the entire Internet, is a blurring of the boundaries between consumption and production of content. With the advent of the so coined Web 2.0, more and more users started to actively participate and furnish their own content, sharing it with an interested audience through specific internet platforms. This phenomenon affected traditional media forms, such as text, with professional content providers seeing new competitors in the form of blogs, as well as newfangled forms of multimedia such as video. Users did no longer content themselves with their former role of mere consumers, they increasingly began to produce content themselves and have gradually become *prosumers* since.

Another aspect that fueled the progress of user-generated content is the availability of mobile devices capable of recording high quality video. Along with video sharing platforms encouraging their users to upload their recordings on the go, this has led to a vast and ever increasing amount of video data being available online. In urban areas nowadays, virtually every busy spot, and every noteworthy event is potentially covered by one or even more laymen cameramen.

Neglecting privacy issues, this enables a variety of exciting new applications that range from 3-D scene reconstruction to automatic video editing to free viewpoint video navigation. Alas, the sheer amount of video data also puts a burden on both the hosting service providers and consuming users who are confronted with a vast disarray of videos. Consequently, efficient solutions for structuring and organizing videos are necessary. The mentioned recovery of 3-D structure can serve a valuable purpose here beyond its primary applications in immersive multimedia. More precisely, the spatio-temporal relationship between videos, be it recoverable, can help put videos acquired by different users into a common context. Once it is known where and when different videos have been acquired, they can be easily grouped into sets which are then also very likely to be linked semantically [SSFK09].

In the following section, some concrete example applications will be discussed that exploit the spatio-temporal relations of user-generated videos to semantically group them into multiview sets and present them to users with additional value. Besides the se-

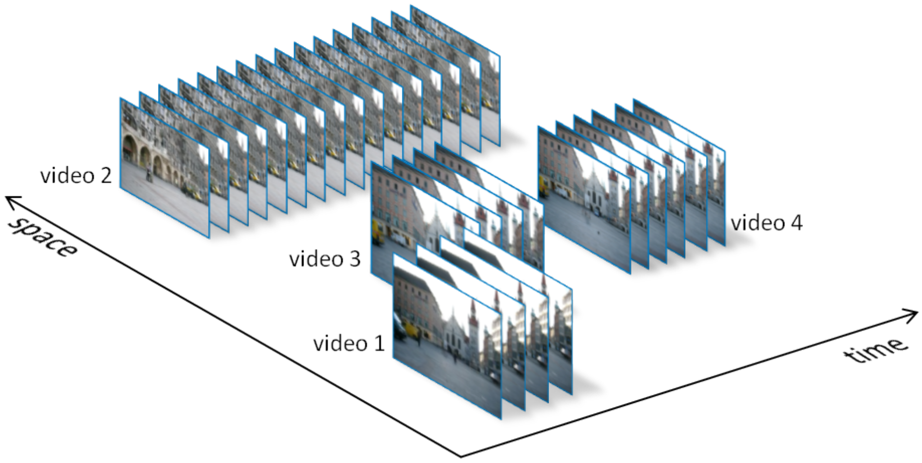


***Figure 1.1:*** *Schematic visualization of the spatio-temporal registration of videos. The objective is to determine where and when videos have been recorded with respect to each other.*

mantic organization of user-generated content and multimedia applications exploiting the spatio-temporal relationships between casually recorded videos there are of course other fields where multi-camera setups are involved. Examples are video surveillance, post-production in the movie industry, and robotics or computer vision in general.

## 1.1 Multiview Video Applications

Equipped with mobile devices that comprise a video camera, a GPS receivers and a connection to the Internet, users can record geo-tagged video clips and instantly upload them to a central server. For some applications it might be sufficient not to send the entire video data but only a reduced representation thereof. For other applications the upload process might consist of streaming the video on the fly.

At the server, videos that happen to have been recorded at the same spot can be registered with respect to each other and incorporated into a database. Initially, this database contains only a few reference views with precisely known position and viewing direction. In the course of operation, it is expanded with every uploaded video. The goal is to recover the exact 3D location of the involved cameras, their viewing directions, as well as the temporal relationship between the different recordings. How to perform the spatio-temporal registration is still subject of intense research. This thesis provides tools and algorithms designed to help solve some of the various open problems in this context. Once videos are registered, it is a straightforward task to find spatio-temporal clusters that most probably represent a common event.

A system that enables mobile users to determine both their location and viewing direction from images taken with their cameras has many potential applications. The most obvious one is a *visual compass* that allows a single user to orient herself in an unknown environment. In this particular case it is not necessary to transmit entire videos or still images but only a reduced representation based on image features that can be extracted on the device

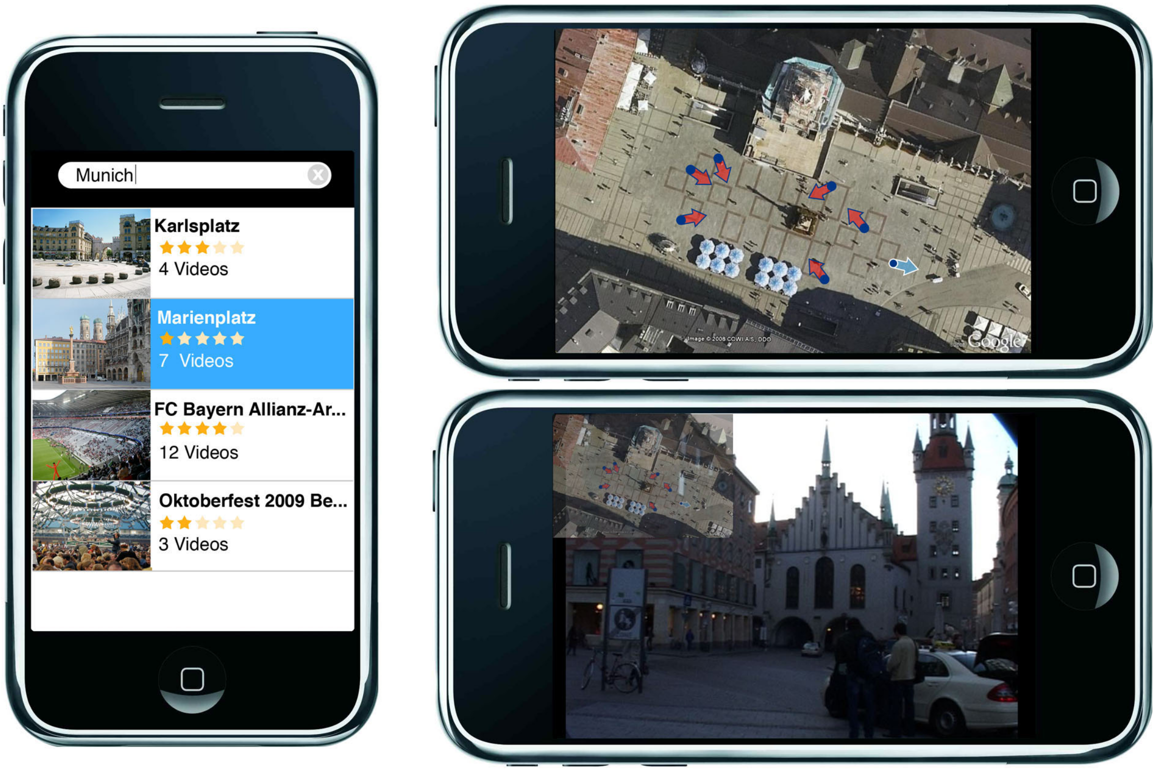


***Figure 1.2:*** *Multiview video sharing portal. Left: Search results for a particular query. Right: Interactive playback of the "Marienplatz" video set. Arrows indicate the location and viewing direction of each video, the currently selected one is highlighted in blue.*

locally. But based on this key functionality an even greater number of applications can be conceived that revolve around the concept of *user cooperation*. People can be offered a video sharing platform that allows them to upload their content which is then semantically linked to that of other users. So formed multiview video sets describe the depicted events much better than an unstructured collection of single clips. Other users interested in the such an event can intuitively choose between different viewpoints and get a more immersive feeling of what the event was like. At the same time, the size of a video set can be an indicator for an event's relevance or popularity. In specific scenarios, videos could also be streamed in real-time from the producing users to the consumers. Whenever an event is captivating enough to have more than one spectator filming it, a multi-angle live stream could so be made available. Example scenarios for live multiview video streaming are sport events or concerts.

Figure 1.2 shows the interface of the mobile video sharing platform proposed in [SSFK09] from a *consumer's* point of view. A query for the term "Munich" results in a number of video sets each containing multiple videos. On a conventional video platform such a request would give rise to a vast and confusing amount of results, comprising all different kinds of videos that have been tagged with "Munich" for some reason. The way in which this system structures the results makes it much more convenient to find videos of relevant content. Once the user selects a video set, the available viewpoints are displayed as arrows on a map of the site. Since the videos in the set are all synchronized, they all refer to a common time line. If a video clip does not span the entire duration of the event, its arrow symbol is simply faded in and out at the respective times. During playback, the arrow locations and orientations are updated in accordance with the motion of the cameras over time. Once the user selects a view the corresponding video clip is displayed. The map is minimized but still visible in the upper left corner so that orientation within the scene remains possible. At any time, another perspective may be selected and, due to the synchrony, the new video seamlessly sets in where the previous one has stopped.

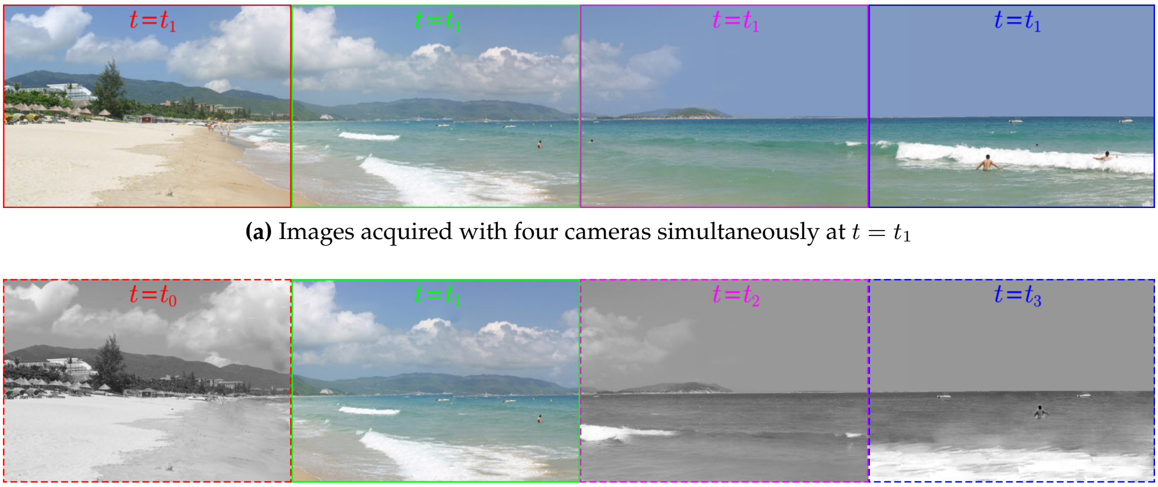
In the applications discussed so far, *user cooperation* referred to the passive agreement of users to have something done with their public videos. But this concept can be pushed even further when users decide to *deliberately* cooperate with each other in order to create a specific multiview video set. An example is the cooperative shot of a panorama image or even a video panorama, as illustrated in Figure 1.3. Using a single camera, it is obviously impossible to simultaneously capture a dynamic scene from different viewpoints. With the help of friends or passers-by and their cameras, however, it can be accomplished given an appropriate image stitching algorithm. Another occasion to arrange the cooperative shot of multiview video is for instance at family reunions, say a wedding. The produced video material then lends itself to fully or semi automatic editing where objects of interest can be kept in focus by transitioning between views, always selecting the most suitable one. Another application is the combination of the high quality audio signal captured by a nearby camera (say again, at the wedding ceremony) with the total view on the scene that only a second camera that is situated further back can provide. Or imagine the visual enhancement of parts of the video image through *super-resolution* techniques based on the input from various views on the scene. The last two examples are particularly useful in a lecture scenario where a lecturer and the blackboard are in the view of several cameras, maybe including a stationary system installed in the auditorium, but legible writing and comprehensible audio can only be recorded from front row seats.

## 1.2 Technical Challenges and Contributions

The focus of this thesis is on the visual analysis of video signals acquired in multi-camera environments. Each video signal[1] constitutes a three-dimensional object, with two-dimensional images, or frames, stacked along a third, temporal dimension. These complementary aspects of video, space and time, are studied separately. From the applications described previously, two technical challenges can immediately be identified that exactly correspond to these aspects. First, one must solve the problem of video synchronization, second, the problem of spatial registration. Both these problems have been intensively studied over the past decades and legion of solutions have been proposed for each one of them, specialized for specific requirements and with certain constraints.

Regarding video synchronization, preferable properties are high accuracy, minimum constraints on the depicted scene and the viewpoints, on the camera or video parameters,

[1] The term *video signal* refers to the visual component of video only. The audio component will be disregarded throughout this thesis.



**(a)** Images acquired with four cameras simultaneously at $t = t_1$

**(b)** Images acquired with a single camera at distinct time instants $t = t_0, t_1, t_2, t_3$

***Figure 1.3:*** *(a) One frame of a video panorama stitched from multiple simultaneous recordings. (b) A single camera cannot capture the dynamic contents of the scene.*

tolerance towards camera motion, varying lighting conditions, etc. Section 2.1 on related work in video synchronization gives an extensive overview of existing algorithms and lists their strengths and deficiencies in this regard. With the novel video synchronization algorithm presented throughout the rest of Section 2, an elaborate approach is proposed that comes very close to satisfying the requirements listed above. Unlike most other state-of-the-art methods it furthermore operates automatically, basically requiring no user intervention at all, which makes it unrivaled in terms of flexibility. A great deal of the robustness is owing to a novel cross-correlation approach coined ConCor, described in Section 2.3, which is capable of detecting and eliminating incoherent parts in the videos. But ConCor is not limited to video synchronization. It is rather a general concept that can be applied to a number of problems where robust cross-correlation is needed.

As to the problem of spatial registration, different disciplines have established tackling it in different flavors. Structure-from-Motion (SfM), for instance, addresses the problem of reconstructing 3-D structure from multiple 2-D images. The images are typically frames taken from a video captured with a moving camera, hence the name of the discipline. But conceptually there is no difference to using images acquired by different (video or still) cameras. As with all stereo vision approaches, the missing depth information is inferred from parallax effects due to the camera displacement which is typically small in SfM. In wide baseline reconstruction, the same problem is studied for the case where the involved cameras are further apart. Once the geometry of the scene can be reconstructed, the camera poses are known too, and video registration is achieved. Without going into further detail, most of the approaches available today rely on image features. There are different feature detectors out there, each with distinct properties and applications. An important family are multiscale or scale-invariant features which can naturally compensate for variations in camera distance, zoom level, and size changes in general. Scale-invariant image features are extensively studied in Section 3 of this thesis. More specifically, the location uncertainty of image features detected with a given algorithm is studied in Section 3.3. The proposed evaluation framework allows the quantification of this uncertainty, and makes it available to improve and facilitate subsequent processing, for example in SfM.

In Section 3.4, two novel feature detectors are presented that distinguish themselves by their low computational complexity. On the basis of an established feature detection algorithm, a radically simplified, linear detector, coined suSURF, is conceived. It achieves high detection performance, yet at a fraction of the processing time. The second novel detector is AsuSURF which, at the cost of a tolerable complexity increase, additionally incorporates affine invariance. This makes AsuSURF highly suitable for wide baseline matching tasks, *e.g.*, in 3-D reconstruction, but even more so in object detection and image retrieval.

Moreover, a new class of visual markers is introduced in Section 3.2. Unlike existing marker systems, the proposed Maximum Detector Response Markers (MDRMs) do not require a separate detection step, but are detected with standard feature detectors. The markers are designed to be optimally detectable and thus offer superb detection and low false negative rates.

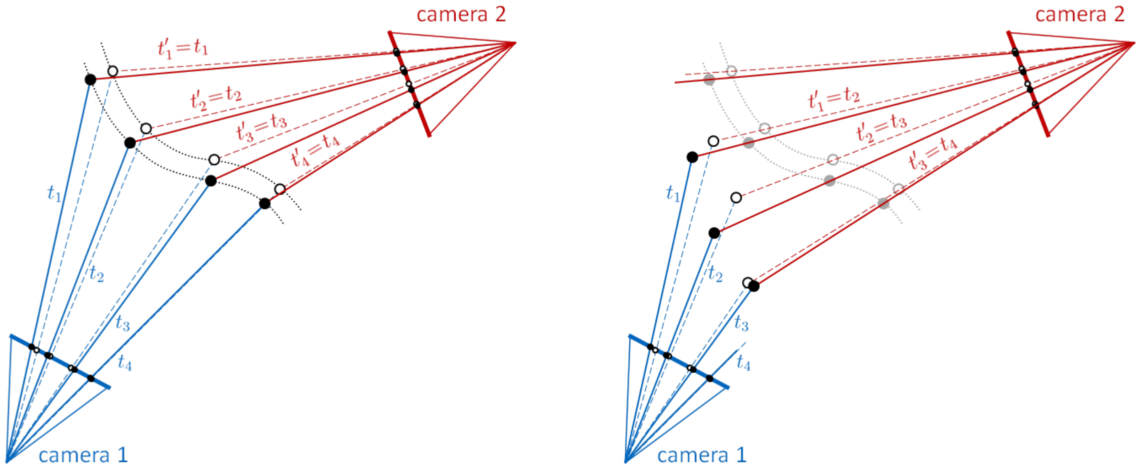
Parts of this thesis have been published in [ZGS+09, SSFK09, SZG+09, SSE+10, SSE+11, ANCS+12, SSE+13].

# 2 Temporal Video Analysis: An Information-theoretic View on Video Synchronization

Every multimedia application involving multiple videos of the same scene requires exact temporal synchronization in order to extract useful information from the given data. The inference of depth information from multiview videos can in general only be performed if the corresponding frames have been acquired at the same time instant. Otherwise, the projections of dynamic objects are inconsistent. The same holds for applications which aim at stitching the input videos together to form a panoramic view, or at editing videos in such a way as to provide seamless transitions between different perspectives on the portrayed action. Fields of application are very diverse and include basically every domain where multiple cameras are deployed. Be it any kind of camera network, used for instance in a surveillance application, in television or film production, or novel community based video sharing applications; whenever two or more videos of the same scene are available, there is an interest in aligning the image sequences in time. The required precision of this alignment may vary depending on the particular application, typically a synchronization with integer frame accuracy or below is desired. Another important requirement in many applications is that the synchronization process should be fully automated, without the need for user intervention.

While the above demands can be satisfied with hardware-based solutions, these approaches are not applicable in many of the targeted scenarios. In fact, cameras related by a central clock are only used in high-end applications, such as professional multiview sportscasts or automotive crash tests. An alternative are deliberately placed, external synchronization cues, *e.g.*, by use of a clapperboard in film productions. None of them are practical unless the camera setup is permanent to some extent, or the acquisition is planned well in advance. For less costly productions, or in scenarios where a camera network forms in an ad-hoc manner, possibly without control over the used cameras, only software-based synchronization approaches are viable. In the case of user-generated content, an event might even have been captured independently by total strangers. Their recordings can only be unified afterwards through some common video sharing service. Many mobile devices, such as camcorders and mobile phones, contain receivers for either or both the Global Positioning System (GPS) and the Global Navigation Satellite System (GLONASS), which both provide highly accurate clock signals. However, there is no reliable temporal association between a device's satellite navigation and camera modules. Timestamps attached to video recordings (typically just a file creation time) are always based on the device clock and are only exact to the second. For the purpose of video synchronization with frame or even subframe precision (*i.e.*, tolerances below 30–40 ms), this is insufficient.

The less influence one has on the acquisition process, the more robustness a video synchronization algorithm needs to offer. Assumptions about perfectly stationary cameras, identical frame rates, or similar viewing directions are more often than not invalid in prac-



**(a)** Two cameras in sync

**(b)** Cameras out of sync

***Figure 2.1:*** *The principle behind* feature-based *video synchronization: scene rigidity as a measure for synchrony. Two cameras record a moving pair of rigidly linked points (marked with • and ○) at time instants $t_1, t_2, \ldots$ and $t'_1, t'_2, \ldots$, respectively. (a) Rays back-projected from images acquired at the same time instant $t'_i = t_i$ intersect at the corresponding scene point location. (b) With a temporal offset of one frame ($t'_i = t_{i+1}$), the points of each pair are reconstructed at incorrect locations, their distance is not preserved. The pair's rigidity is violated in this case.*

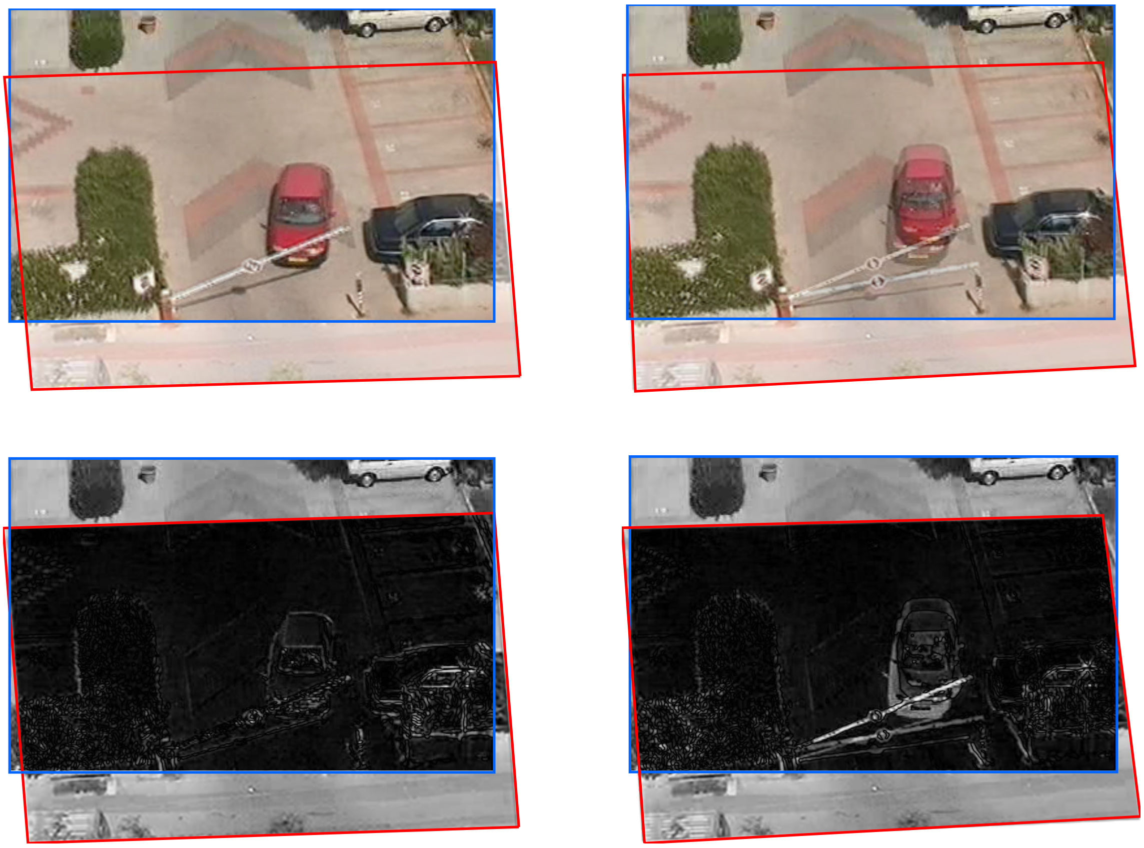
tice. Ideally, a video synchronization algorithm can deal with unknown input sequences and, as long as they show the same event, reliably determine their temporal alignment.

The remainder of this chapter is organized as follows: In Section 2.1 an overview of existing video synchronization approaches is given. Throughout Section 2.2, a fundamentally different, bitrate-based synchronization algorithm is presented and described in detail. Section 2.3 is devoted to Consensus-based cross-Correlation (ConCor), an extension proposed on top of the basic approach which increases robustness against erroneous data and allows for largely autonomous operation without user intervention. The ConCor enhanced synchronization algorithm is then applied to several challenging datasets in Section 2.4.

The ideas and contributions developed in this chapter have been published in parts in [SSE$^+$10, SSE$^+$11, ANCS$^+$12, SSE$^+$13].

## 2.1 State-of-the-art

In the late 1990s, when more and more applications involving multiple videos of one scene began to emerge, the first software-based video synchronization algorithms were proposed. One of the pioneer authors to take on this subject was Gideon Stein [Ste99] who presented a *feature-based* approach assuming coplanar object motion. Estimating a homography, points along trajectories in two views are brought to alignment, the estimation error being used as a measure of asynchrony. Most subsequent feature-based



**(a)** Videos in sync

**(b)** Videos out of sync

***Figure 2.2:*** *Example for* intensity-based *video synchronization: Two telephoto views of the same scene are warped and superimposed in the top row. Their gray level difference is shown below. (a) For simultaneously acquired frames, the alignment is rather accurate. (b) Between frames from different time instants, however, dynamic objects cause significant intensity differences.* (Source images from [CI]

methods have seized this fundamental principle of establishing geometric consensus between dynamic features. To deal with more general object motion, other approaches quantify misalignment by means of epipolar geometry, estimating a fundamental matrix [PBVDHC03, CSI06, WHK06, BPCP08, PCSK10] or a trifocal tensor [WLB05, LY06]. Rao et al. [RGSSM03] avoid the explicit computation of epipolar geometry and evaluate rank constraints instead. Some authors apply voting schemes to find the most consistent temporal alignment among feasible candidates. Pooley et al. employ the Hough Transform to establish an affine relationship between timelines [PBVDHC03], Pádua and Carceroni et al. use RANSAC instead [PCSK10]. Tuytelaars and Van Gool have detached their approach from epipolar constraints and evaluate the distance between back-projected rays of sight in affine space [TVG04]. In [YP04], Yan and Pollefeys extract spatio-temporal interest points from the videos and cross-correlate their occurrence over time. Raguse and Heipke present an approach aiming at accurate alignment of footage acquired with multiple high-speed cameras [RH06]. They regard the cameras' temporal deviations as additional intrinsic parameters, and so incorporate them into regular bundle adjustment. Wedge et al. and Brito et al. have proposed dedicated algorithms, respectively, to synchronize recordings of objects in free fall [WHK06], and of mobile sensors actively tracking their own position [BPCP08] (adopting the principles of [PCSK10]). Relying on external aids, both approaches can be considered on the verge to hardware-based methods. The general principle of feature-based video synchronization is illustrated in Figure 2.1.

A second major branch of synchronization approaches is formed by *intensity-based* methods. Instead of matching image features and their trajectories, constraints on the alignment are derived from the body of pixels in all video frames. In 2000, Caspi and Irani presented their work on sequence-to-sequence alignment [CI00] where the temporal alignment between frames and their spatial transformation is solved for simultaneously. In an iterative approach operating on scale pyramids generated from the input videos, the actual deviation in gray levels is minimized (*cf.* Figure 2.2). In later publications, a different similarity measure replacing mean squared error was introduced [UI06], as well as a feature-based variant of the initial algorithm [CSI06]. Another early work by the same authors deals with the synchronization of rigidly linked, moving cameras [CI02], exploiting similar changes over time in both views. Dai et al. have seized the principle behind [CI00], but solve for the spatio-temporal alignment through 3-D phase correlation [DZL06b]. Along completely different lines, Ushikazi et al. derive a frame-wise measure of appearance change that can be matched using cross-correlation [UOD06]. Recently, Shresta et al. published a multimodal approach exploiting audio fingerprints and the occurrence of camera flashes which they align across videos using dynamic programming.

All these methods have their specific strengths and limitations in terms of requirements to be imposed on the cameras, their setup, and the portrayed scene itself. In particular, there are differences in the number of supported cameras, their relative orientation, the nature of allowed motion, as well as image resolutions, frame rates, etc.. Scene objects sometimes must be sufficiently textured, so as to detect, track and match reliable features, their movements restricted both in nature and intensity. Generally speaking, a main issue of feature-based approaches is their restriction of relative camera viewing angles due to limited matchability [MTS$^+$05]. If features are derived from silhouettes, view points are typically confined to a plane, depending on the assumptions on object shapes and motion (*e.g.*, horizontal baselines in case of upright posture). Intensity-based methods on the other hand tend to be incompatible with independently moving cameras.

## 2.2 Bitrate-based Video Synchronization

In [SSE$^+$10], a fundamentally different approach to video synchronization has been proposed. Instead of imitating the human eye in detecting synchronous events in two videos, this approach depends on a more abstract concept: the information content of individual video frames. Based on the fundamental understanding that synchrony is inextricably linked with motion in the scene, the primary goal is to reliably quantify motion throughout a video. Obviously, static scenes do not carry any information from one frame to the next. From an information theoretic point of view, a frame that does not differ from its neighbors

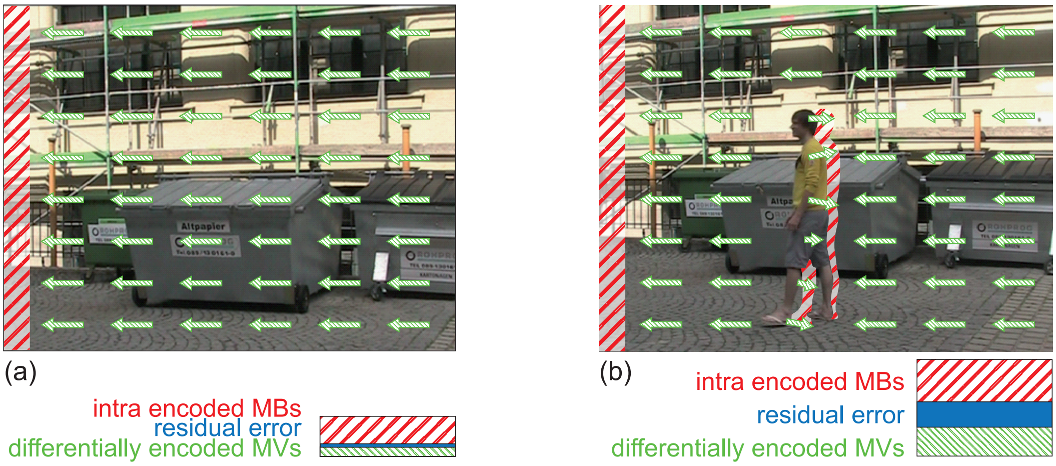
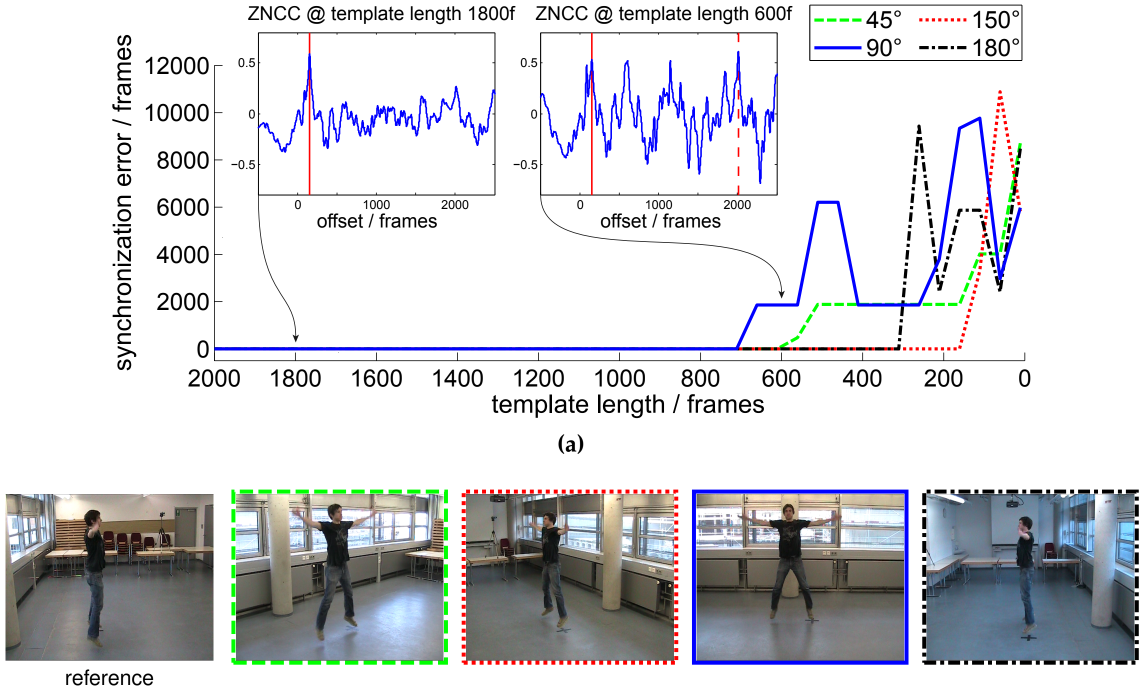


***Figure 2.3:*** *Simplified qualitative view on bitrate contributions in the cases of (a) sheer camera motion and (b) additional scene changes*

exhibits vanishing conditional entropy, its additional information content is zero. Only if there is deviation from the already observed statistical behavior, a frame brings about an information increase; which is exactly the case if objects in the scene are in motion.

There have been several advances towards quantifying scene changes specifically for the purpose of video synchronization, *e.g.*, in [UOD06]. The biggest challenge for all these approaches is to make the proposed measures as robust as possible to detrimental effects. In order to avoid restrictions to be imposed on the videos, the measures need to be designed to deal with all kinds of external influences, which amounts to an incessantly complex task. The most important issue is to reliably distinguish between camera motion and scene motion. While the latter carries precious information closely related to synchrony, camera motion is entirely independent, thus irrelevant for synchronization (unless the cameras are rigidly linked to each other, *e.g.*, on a stereo rig).

A field where this same problem – viewed from a different perspective – has already been solved to a great extent is video compression. Here, the goal is to represent video data with the least possible rate, hence to reduce it to its very essential information content. State-of-the-art video compression algorithms efficiently compensate for predictable motion, reducing the bit-rate demand of corresponding macroblocks to a minimum. Figure 2.3 schematically illustrates how a hybrid video codec handles motion in the video. For homogeneous motion patterns, which are characteristic for camera pans, prediction from previous frames is highly efficient. The merely translatory displacement of each macroblock is encoded in a motion vector, achieving vanishing residual error. A motion vector field as smooth as in Figure 2.3a can further be encoded differentially at very low rate. The major bitrate contribution in this example stems from image parts that are uncovered on the left border due to the camera motion. Ordinarily, these macroblocks need to be encoded independently, in so called INTRA mode. In the case of additional scene motion, the bitrate composition is different, as depicted in Figure 2.3b. Not only do moving objects uncover additional background areas, which results in an increased contribution of independently encoded blocks. The motion vector field is also less regular, and thus



(a)

(b)

*Figure 2.4: (a) Absolute synchronization error as a function of template length. Four videos with varying viewpoints, depicted in (b), are compared to the given reference. The two embedded graphs show the Zero-mean Normalized Cross-Correlation (ZNCC) functions for the view at 90° for exemplary template lengths, the solid red lines therein indicate the ground truth offset, the dashed one an erroneous synchronization outcome.*

more difficult to compress. Since object motion is in general more complex, prediction efficiency is lower, leading also to a higher residual error. To summarize, camera motion can be represented very efficiently while scene motion requires higher bitrates. Ideally, the effects of camera motion are reduced to "background noise" negligible in comparison with contributions caused by scene motion.

Even though camera motion cannot be fully eliminated in practice, its contribution is limited and, most notably, not correlated between views. It has been shown in [SSE$^+$10] that cross-correlating bitrate sequences is indeed a reliable way to determine the temporal offset between two videos of the same scene. It has further been demonstrated that the bitrate-based approach imposes only minimal restrictions on the videos to be synchronized. Owing to the sophisticated motion compensation qualities of H.264, it can cope with moderate camera motion, and it is independent of the cameras' viewing directions since the amount of motion in the scene is quantified rather than its precise appearance. The only prerequisites obviously are the presence of motion in the observed scene and that the depicted actions of interest remain in the focus of both cameras throughout the recording. It will be shown that this requirement can even be relaxed to a certain extent when ConCor is employed.

The key property of viewpoint independence is illustrated with an example in Figure 2.4.

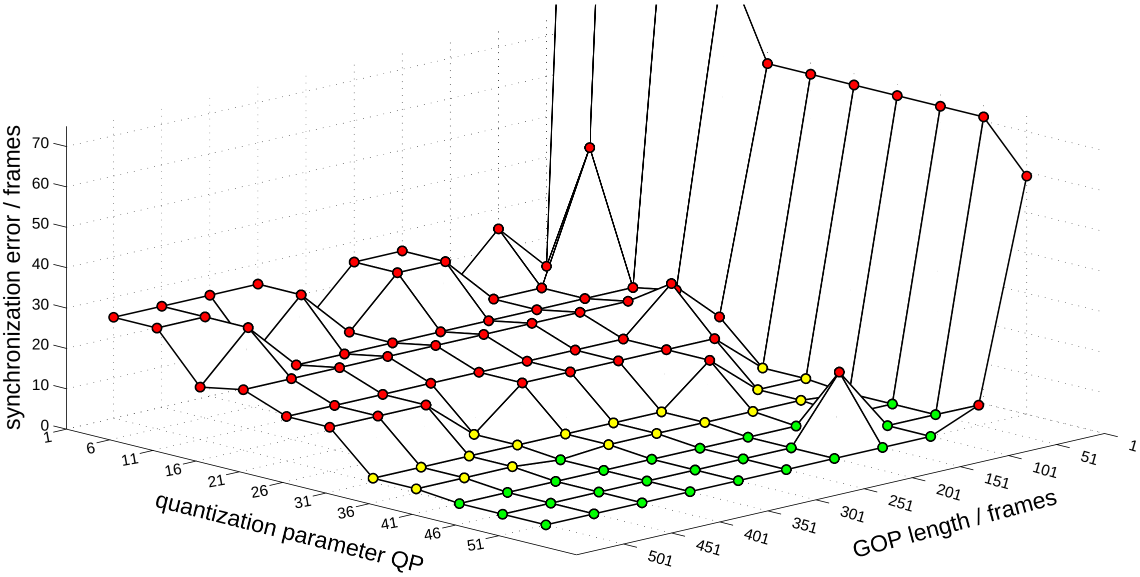


***Figure 2.5:*** *Synchronization error for different parameter settings. Perfect synchronization is indicated by green markers, yellow stands for an absolute error of exactly one frame, red for error values of two frames and higher.*

Here, an individual performing exercises is recorded from five different directions. Excerpts from the corresponding bitrate sequences are then aligned with the chosen reference using ZNCC. As long as the selected excerpts are long enough, the correct temporal offsets are retrieved in all cases. Only if the template length is insufficient the excerpts loose their distinctiveness and misalignment can occur. In this scenario, 800 frames or 4 seconds are enough to guarantee frame accurate synchronization. For one of the pairs, an excerpt as short as 200 frames is sufficient. Another observation is that although the true offset is no longer found at the highest ZNCC peak in some of the cases, it does remain a reasonable offset candidate with a local ZNCC maximum of comparable strength. This important fact will be exploited in the ConCor algorithm described in Section 2.3.

### 2.2.1 Influence of Encoding Parameters

In [SSE$^+$10], bitrate sequences were generated by re-encoding a given video using the x264 encoder implementation of the H.264 codec. A fixed quantizer needs to be used to produce Variable BitRate (VBR) output, and bi-directional prediction disabled in order not to interrupt the sequence's chronology. This implies a Group Of Pictures (GOP) structure of the form IPPP. . . , *i.e.*, single I-frames separated by a series of P-frames. The most crucial parameters in the re-encoding process are hence the Quantization Parameter (QP) which adjusts the fidelity of the re-encoded video, and the length of the Group Of Pictures (GOP), *i.e.*, the distance between I-frames separated by a series of P-frames. In Figure 2.5, the synchronization error for a representative video pair is displayed subject to different settings of these parameters. It can be observed that frame accurate synchronization can only be

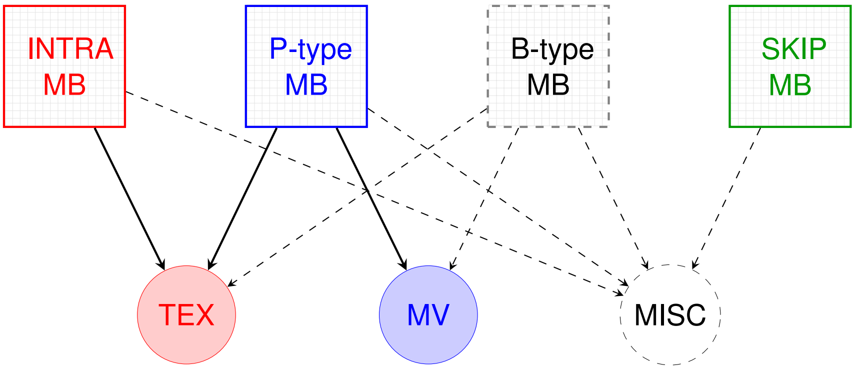


*Figure 2.6: Contributions to bitrate components. The three macroblock types allowed in our implementation are INTRA, P and SKIP. B-type prediction being disabled, the only contributors to the relevant texture and motion vector components are INTRA and P-type MBs. The signaling and syntax overhead, denoted MISC in the figure, is negligible.*

consistently achieved in a region of high QP and GOP length values. This and other experiments suggest that maximally long QPs, and maximally coarse quantization should be used. In the following, an attempt is made at explaining why coarse quantization and long GOPs facilitate bitrate-based synchronization.

### Quantization Parameter

To understand why coarse quantization is beneficial, different bitrate components need to be examined separately. There are basically two complementary types of information contributing to the bitrate output of a hybrid video encoder: data necessary to represent the Motion Vectors (MVs) of every predicted macroblock, and so called TEXture data (TEX) which comprises the associated residual prediction error and the contribution of individually encoded INTRA macroblocks, both after transform coding and quantization. Figure 2.6 illustrates the obvious linkage between different coding modes and these components.

Intuitively, one associates scene motion primarily with the MV component. Later in this section, it will be shown under which circumstances this assumption is valid. For the time being, let us go with intuition and identify MV data with dynamics in the video sequence.

From Figure 2.7 it can be seen that in the case of fine quantization, *i.e.*, for small QP values, the (very noise-like) texture component is predominant. Only for large QPs the bitrate for motion vectors can compete with and even surpass their TEX counterpart. It is further remarkable how the TEX component assumes the shape of the MV signal for increasingly coarse quantization, and thus also carries information about scene motion. This interesting fact will also be investigated later in this paragraph.

Figure 2.8 explicitly illustrates the behavior of both components for varying QP values. The rapid decrease of the TEX component is related to the obvious effect of QP in the quantization process, directly controlling the quantizer step sizes, which consequently leads to the observed descent in 2.8a. The characteristic behavior of the MV component in 2.8b, however, requires finer dissection.

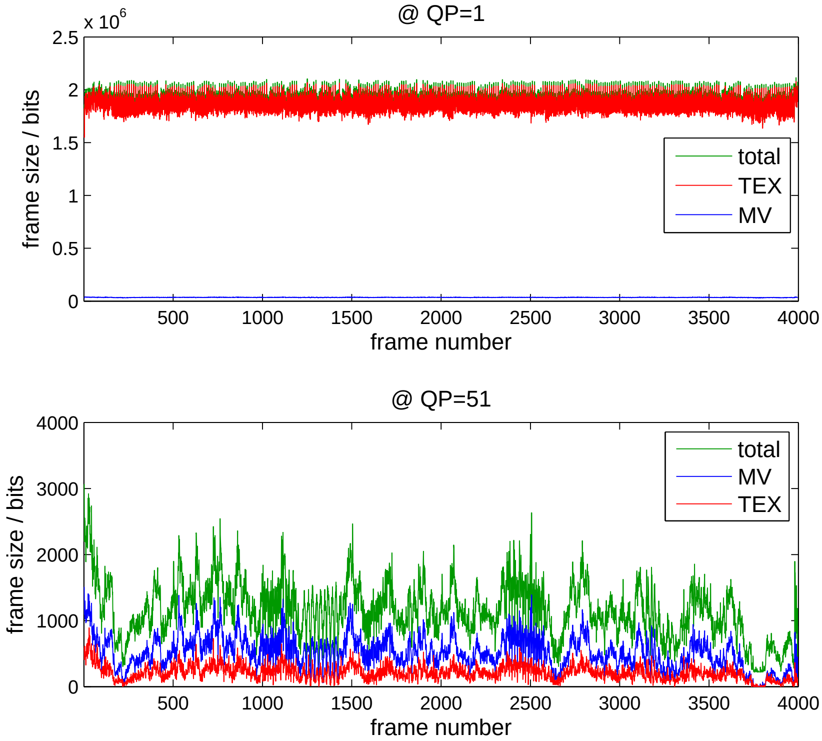


***Figure 2.7:*** *The bitrate profiles for one of the "Human Adam" sequences, at two different quantization parameter values QP and identical GOP length 499 (I-frames removed). The bitrates for the TEX and MV components are plotted separately, as well as the total bitrate (TEX+MV+overhead). For QP $= 1$, TEX dominates the total bitrate without following any characteristic evolution. At the other end of the scale, for QP $= 51$, MV comes out on top and imposes its temporal behavior which is closely related to the actual motion present in the video.*

Given the fact that in this setting, with B slices disabled, P-type macroblocks are the only contributors to the MV component (*cf.* Figure 2.6), the focus will be on the occurrence of P-frames and their properties. Figure 2.9a shows the allocation of a frame's macroblocks into the three types allowed by our settings as a function of the Quantization Parameter (QP). The exact numbers are of course subject to the video content and the encoder implementation, but the qualitative QP dependency is the one presented here, with diminishing INTRA and increasing SKIP rates, and a P-type percentage increasing up to a certain QP, labeled $q_0$ in the figure, receding thereafter. To understand this behavior, the choices the encoder makes during mode selection using Rate-Distortion Optimization (RDO) [SW98] must be considered. The established technique applied in RDO is the macroblock-wise minimization of a Lagrangian cost relating the necessary rate $R$ for a given coding mode and the associated distortion $D$.

$$J = D + \lambda R \tag{2.1}$$

The non-negative relative weight $\lambda$ is typically set as a function of QP, *e.g.*, in [SW98] an experimentally motivated $\lambda = 0.85\,(\mathrm{QP}/2)^2$ is proposed. Figure 2.10 schematically illus-

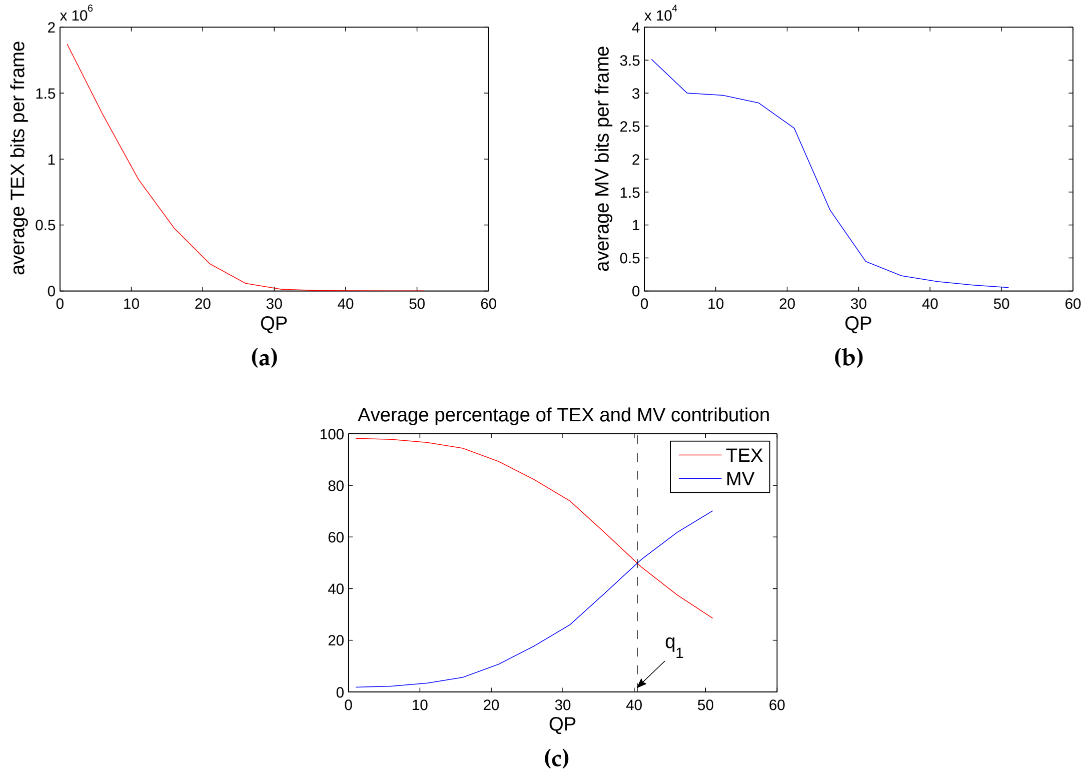


***Figure 2.8:*** *The average bitrates (averaged over all frames) for (a) the TEX and (b) the MV component as a function of the quantization parameter, and (c) their relative contribution to the total bitrate. At QP* $= q_1$*, the two components switch roles.*

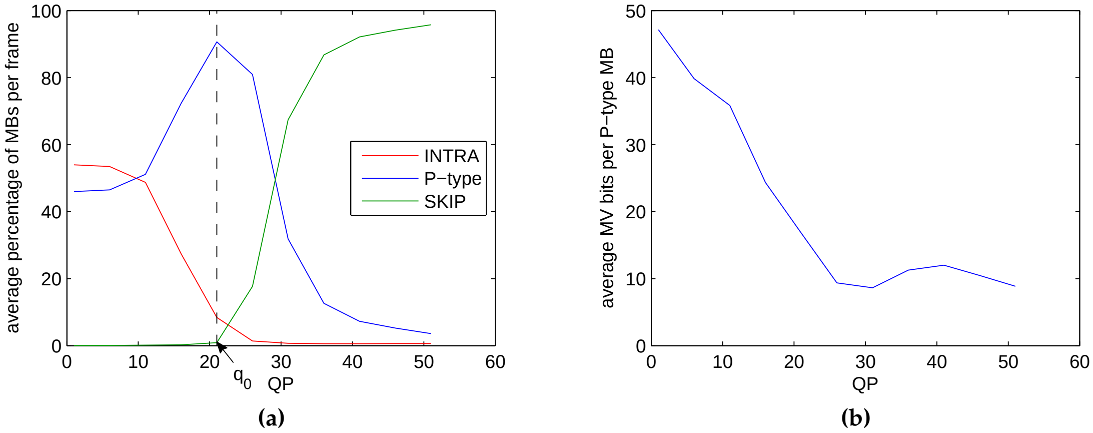


***Figure 2.9:*** *The average number of the different macroblock types in a frame for varying quantization parameter (a), and the size in bits of an average P-type MB (b). The latter, together with the P-type curve from (a), explains the behavior of the MV component over QP from Figure 2.8b.*

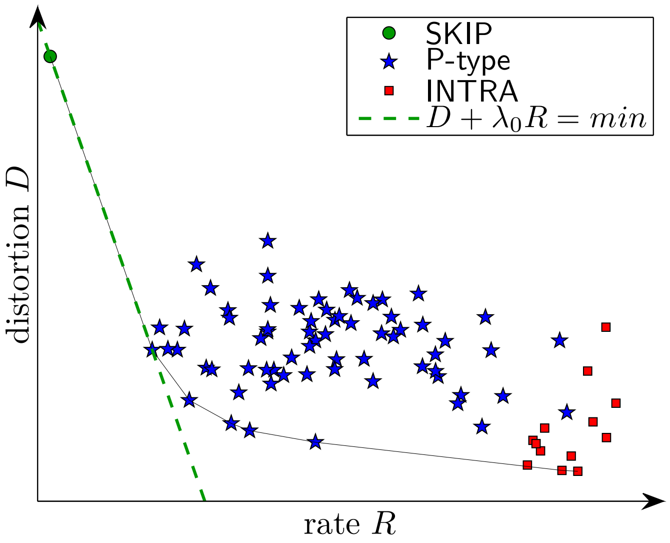


*Figure 2.10: The x264 encoder uses Rate-Distortion Optimization (RDO) to determine the encoding mode for a given macroblock. According to the H.264 standard, there are 13 INTRA and 67 P-type variants to choose from, in addition to SKIP mode which offers vanishing rate (neglecting overhead) at usually heightened distortion. The exact position of the different modes in the R-D plane depends on the particular macroblock. The dashed line with slope $-\lambda_0 = -0.85\,(q_0/2)^2$ corresponds to the $q_0$ value from Figure 2.9a for an "average" macroblock. See text for details.*

trates the RDO driven coding mode decision for a particular macroblock. In this diagram, operating points with constant cost $J$ form a family of parallel, straight lines with slope $-\lambda$. The optimal mode is determined as the contact point of the family member which is tangent (from below) to the convex hull of all possible operating points. With increasing QP the slope $-\lambda$ becomes steeper, and modes are selected towards lower rates and higher distortions. Above a certain QP value, indicated by the slope of the dashed line in Figure 2.10, SKIP mode is always the optimal choice, permanently providing the lowest possible rate.

Now consider the QP value labeled $q_0$ in Figure 2.9a. It corresponds to the dashed limit slope from Figure 2.10 and marks the turning point above which SKIP mode becomes optimal for more and more of the individual macroblocks. Consequently, the ratio of SKIP blocks experiences a rather abrupt increase beyond $q_0$, at the expense of INTRA and P-type macroblocks. For the QP region below $q_0$, where SKIP is not an option due to the higher emphasis of (2.1) on low distortion rather than on low rate, the only choice for the encoder is to balance resources between the INTRA and P-type modes. In this region, lower QPs (hence smaller $\lambda$, and heightened emphasis on low distortion) encourage the encoder to choose INTRA over P-type for an increasing number of macroblocks.

Next, let us have a look at the number of bits necessary to encode the motion vector information of an average P-type macroblock. H.264 allows for a multitude of block par-

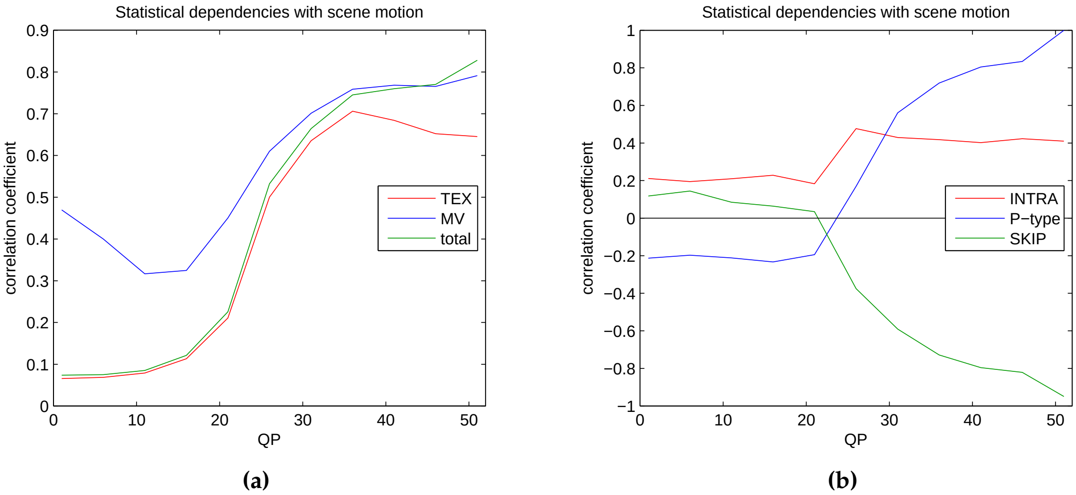


*Figure 2.11: At coarse quantization, scene motion (measured by the number of P-type MBs at coarsest quantization) has a strong correlation with the bitrate components TEX and MV, as well as with the total bitrate (b). The occurence of the different block types is also linked to motion, only moderately for INTRA MBs, but significantly for P-type and SKIP MBs, the latter being negatively correlated (a).*

titioning schemes with subblocks as small as $4 \times 4$ pixels. In total, there are 67 different partitionings with one to 16 motion vectors per macroblock. Again, an RDO argument can be used to explain the transition to less subtle partitionings when QP is increased. And yet again, there is a limit value for QP above which it is always the partitioning with one motion vector per macroblock that will be selected, yielding the lowest possible rate. Hence, for increasing QPs the number of motion vectors, and accordingly the number of bits per P-type macroblock, can be expected to decrease, finally going to saturation. This behavior is validated by the experimental results plotted in Figure 2.9b.

Altogether, the product of the average number of P-type macroblocks (*cf.* Figure 2.9a) and the average size of each P-type macroblock in bits (*cf.* Figure 2.9b) leads to the observed MV component behavior reported in Figure 2.8b. Unlike the TEX component, MV decays less rapidly when going towards coarser quantization. In fact, above a certain point, marked $q_1$ in Figure 2.8c, MV supersedes TEX in its relative contribution to the overall bitrate.

Following the initial intuition that MV is the key component for synchronization, QP values above $q_1$ should be most suitable for the purpose of video synchronization because in this region the bitrate is dominated by the motion vector information which is inherently related to the actual motion in the scene. However, the TEX component is not entirely irrelevant for synchronization either. Every MV is accompanied by a prediction residual contributing to the TEX component. In the case of complex motion, the residuals can be significant and are certainly useful for synchronization. Consequently, it would make sense to consider QP values below $q_1$ in order to include more of this P-type related TEX contribution. Nevertheless, TEX contributions that stem from from INTRA macroblocks need to be excluded.

According to Figure 2.9a, the influence of INTRA blocks is generally very limited for QPs above $q_0$. One might thus be tempted to reduce QP below $q_1$, eventually approaching $q_0$. Indeed, the number of P-type macroblocks would further increase towards mid-range values of QP, but, as is visible from Figure 2.12, this effect is not necessarily related to scene motion. A large amount of the additional P-type macroblocks are rather due to the increasingly fine subblock partitioning which was discussed earlier. Since smaller subblocks are less distinctive, the encoder sometimes decides – for the sake of minimizing Sum of Absolute Transformed Differencess (SATDs) – to predict individual macroblock pieces from arbitrary references, whereas unpartitioned macroblocks would have been skipped unless more consistently predictable. In general, the optical flow, representing the true motion, is better captured when motion compensated prediction works on entire macroblocks, thus for higher QP values.

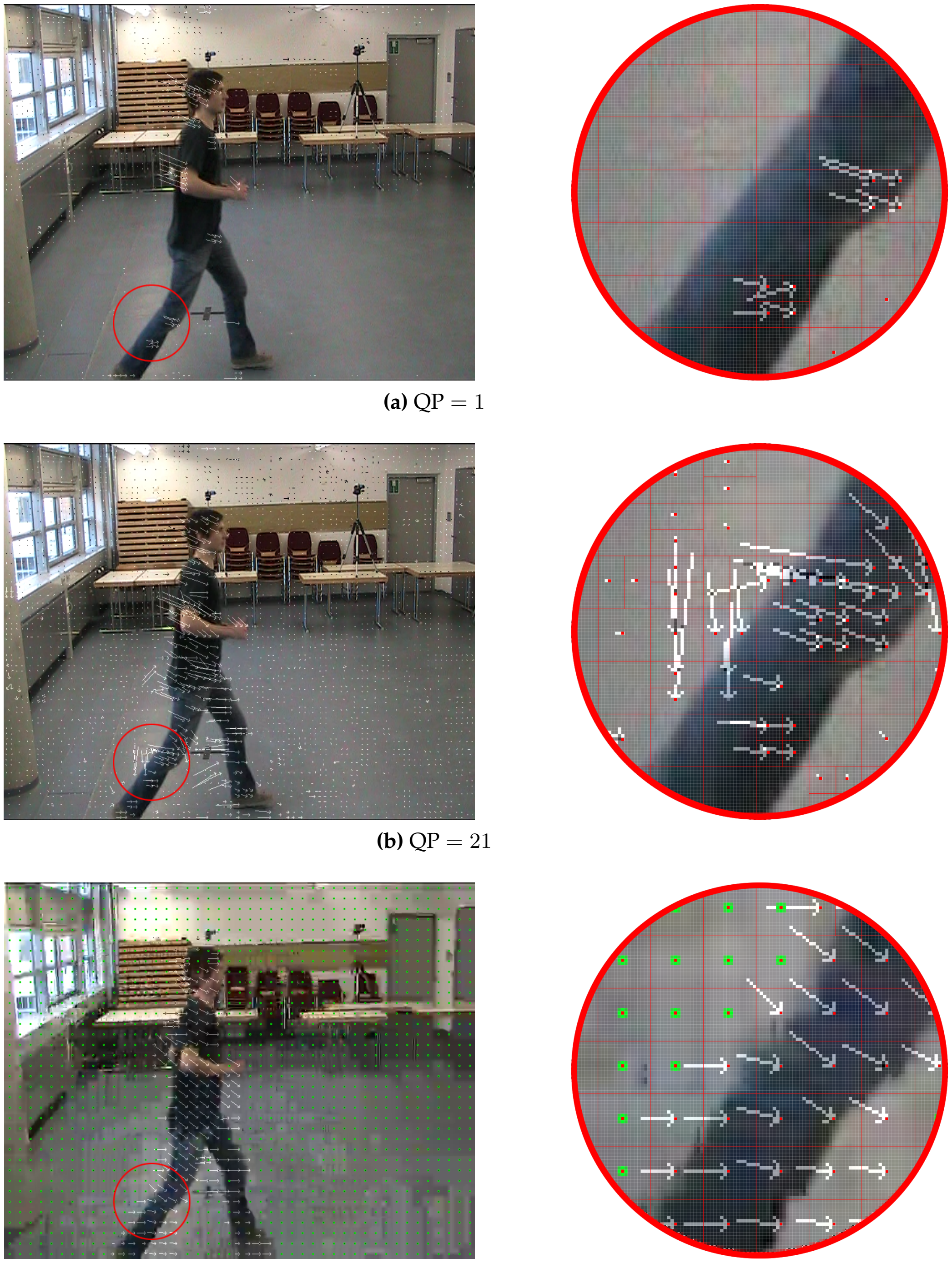
Finally, in order to assess the range of QP values that are suitable for our video synchronization algorithm, the statistical dependencies between scene motion and the bitrate components are analyzed. To this end, the number of P-type macroblocks at coarsest quantization is considered a legitimate indicator for scene motion, and its correlation coefficient with the TEX and MV time series obtained is evaluated for varying QP. The results in Figure 2.11a suggest that the MV component is in general more closely related to scene motion than TEX, and that both gradually lose their meaningfulness for decreasing QP. Both MV and TEX retain their usefulness on a constantly high level before it starts to drop at QP values around 35. This result is in accordance with the initial experiment presented in Figure 2.5.

In practice, depending on the control one has over the videos to be synchronized, it can be beneficial to make the effort and treat the two components separately. If for some reason the videos cannot be specifically re-encoded, it makes sense to combine TEX and MV in the maximum likelihood approach detailed in [SSE$^+$10], or to omit the TEX component altogether unless it is known that the existing videos have been produced using a coarse quantizer. If re-encoding is not an issue, a quantizer with $\mathrm{QP} = 40$ or above should be selected, in which case the total bitrate (MV+TEX, neglecting overhead) will perform just as well as the generally superior MV component alone. Note how the performance of the total rate clings to either of the two components in Figure 2.11a, the turning point being $\mathrm{QP} = q_1$ from Figure 2.8c.

For the sake of completeness, Figure 2.11b shows how scene motion correlates with the number of different macroblock types, pieces of information which, extracted by a more sophisticated bitstream parser, could as well serve for the purpose of synchronization. While the number of INTRA blocks is only remotely linked to scene motion, the number of P-type and SKIP blocks can very well prove useful, again at coarse quantization. It is worth noting that the number of SKIP MBs exhibits an (almost perfectly) negative correlation with scene motion, which stands to reason since it is highly unlikely that a macroblock on a moving object were to be efficiently encoded in SKIP mode.

### Group of Pictures Length

It lies in the nature of I-frames to disrupt the temporal dependences in the video stream. Exclusively composed of INTRA macroblocks, they do not carry information about scene



**(a)** QP $= 1$

**(b)** QP $= 21$

**(c)** QP $= 51$

***Figure 2.12:*** *Frame 733 from one of the "Human Adam" sequences, re-encoded with different quantization parameter values. For fine quantization, large areas of the frame are encoded in INTRA mode, as is evident from the lack of motion vectors in (a). At mid-value QPs, P-type mode is predominant, also in static parts of the frame; the true optical flow is not accurately captured. Only for coarse quantization in (c), most of the static image content is encoded in SKIP mode (indicated by green dots), while the motion vectors exclusively represent the person's movements. Note that in (b) there are multiple motion vectors per macroblock, whereas the majority of motion vectors in (c) represent single macroblocks.*

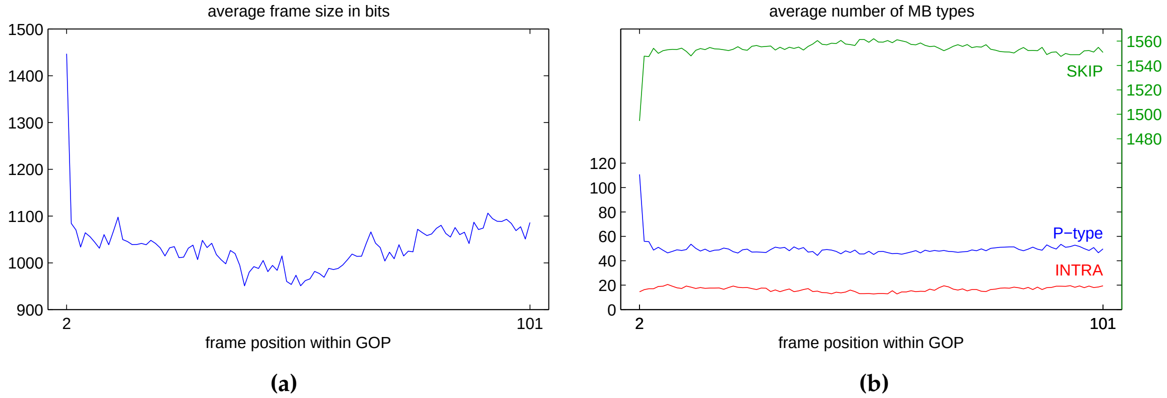


***Figure 2.13:*** *The size of P-frames (a) and the occurrence of different macroblock types (b) as a function of the frame's position inside the GOP. The values are averaged over all GOPs of the given video containing 10000 frames. The GOP length is 101 frames, the initial I-frame is not displayed in either graph.*

motion themselves, yet they are much bigger in size than the relevant P-frames. Consequently, it has been proposed in [SSE$^+$10] to interpolate the bitrate values at I-frame positions prior to cross-correlation.

However, as it turns out, immediately subsequent P-frames are also influenced by the (removed) I-frame. As can be seen from Figure 2.13a, which depicts an average GOP after I-frame removal, it takes several frames before the bitrate samples actually level off. The exact manifestation of this effect depends on the amount and nature of motion in the scene. In the given example, a steady camera was used to capture a person whose moderate movements cover approximately ten percent of the image. Picture quality experiences a "refresh" with every I-frame, then degrades again very fast within subsequent P-frames. Accordingly, prediction becomes less and less efficient, and SKIP mode is increasingly selected over P mode. Eventually, P-type macroblocks are only used in dynamic scene parts, while static background is entirely represented in SKIP mode. At the same time, the number of INTRA macroblocks remains on a constantly low level. Consequently, a quickly decaying bitrate peak is observable at the beginning of each GOP. The shorter the chosen GOP length, the more such peaks occur in the re-encoded sequence. In order to mitigate their repercussions on synchronization, sufficiently long GOPs need to be used. Following practical experiences (see also Figure 2.5), values beyond 300 frames per GOP are suitable.

## 2.2.2 Practical Considerations

### Re-encoding Artifacts

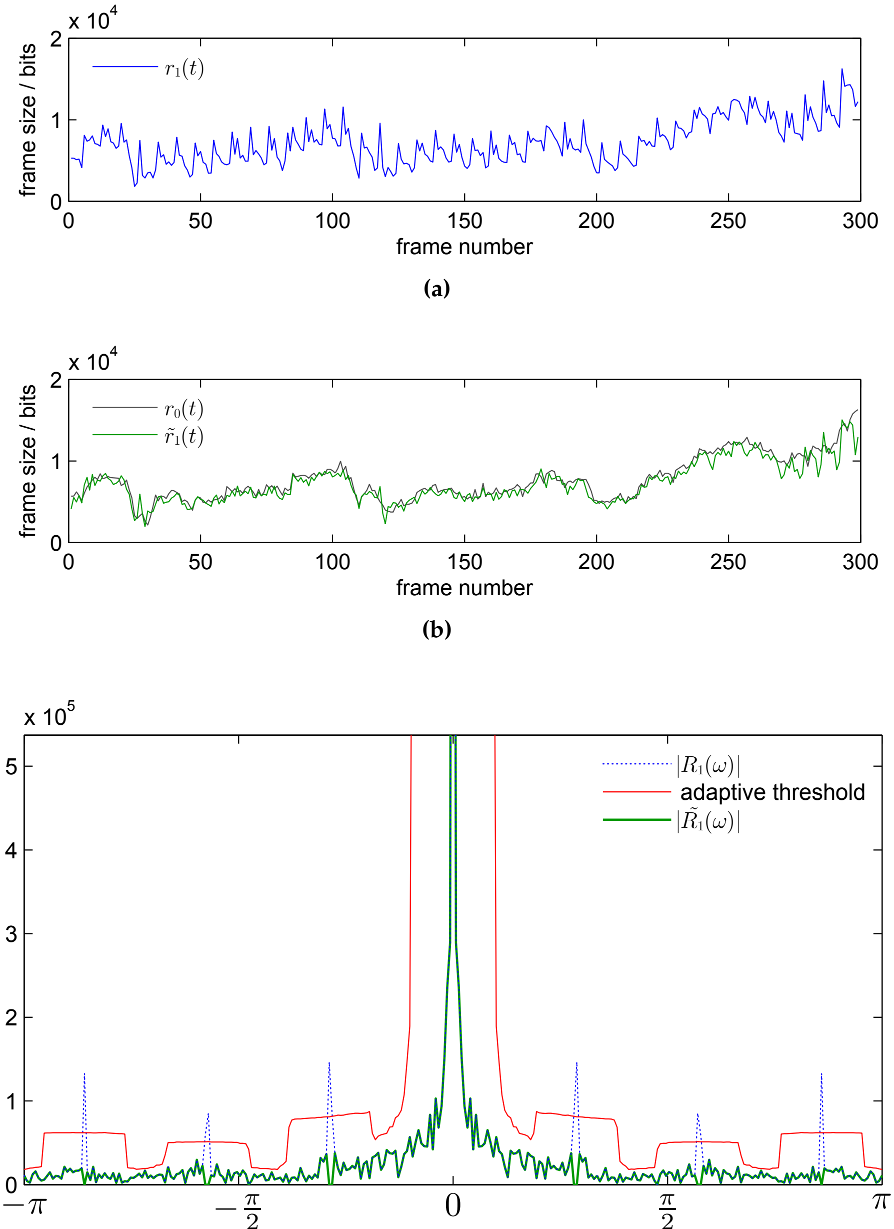
When bitrate sequences are obtained through re-encoding as described in [SSE$^+$10], the nature of the original video data needs to be taken into account. Ideally, the source video is available in hi-quality raw format, ready to be directly encoded into the desired IPPP... scheme, with parameters as suggested in Section 2.2.1. The bitrate samples then coherently reflect the motion complexity throughout the video – except at the I-frame positions which are discarded [SSE$^+$10]. However, more often than not the source videos are compressed in an arbitrary format and with unknown parameters, both evading direct control. In case of highly compressed source material, the persistence of the original GOP structure after re-encoding can be observed. This is due to the typically encountered Peak Signal to Noise Ratio (PSNR) differences between the different frame types. A former I-frame usually forces the encoder to spend more rate in order to avoid an otherwise disproportionate distortion. This is accomplished by increasing the number of INTRA macroblocks (in case of fine quantization), or by using P-type macroblocks instead of SKIP mode. In any case, this implies a bitrate spike at the position of the given frame, which thereby retains its "INTRA character". The same holds for former P-frames which similarly differ in PSNR from possibly present B-frames. The periodicities caused by this combination of persistent GOP structure and overshoots at the beginning of each new GOP (see Section 2.2.1) are of course detrimental during synchronization. Two remedies are proposed to counter this effect: an appropriate choice of the new GOP length, and an active removal of periodic components from the re-encoded bitrates.

First of all, the periodicities can be diluted by a proper choice of the new GOP length. The periodic length of the spike pattern is given by the least common multiple $\mathrm{lcm}(G_0, G_1)$ of the GOP lengths $G_0$ and $G_1$ before and after re-encoding. If the new GOP length $G_1$ is chosen to be a prime number, the period always assumes the maximum length of $G_0 \cdot G_1$ samples, irrespective of the given value of $G_0$. As a consequence of the enlarged period length, the spike disturbance takes on a more aperiodic character. Consider the re-encoding of GOPs with $G_0 = 6$ for different values of $G_1$. With the prime $G_1 = 499$, the spike pattern repeats itself exactly every $\mathrm{lcm}(6, 499) = 499 \cdot 6 = 2994$ samples. Every variation of $G_1$ drastically shortens this period, thus emphasizing the periodicity of the spike disturbance. With $G_1 = 500$ and $G_1 = 501$, for instance, the period length drops to $\mathrm{lcm}(6, 500) = 1500$ and $\mathrm{lcm}(6, 501) = 1002$ samples, respectively. For $G_1 = 498$, the period is as short as $\mathrm{lcm}(6, 498) = 498$ samples.

In addition to choosing a prime GOP length, an active removal of periodic components from the bitrate sequences after re-encoding is proposed. If the re-encoding history of the input videos is known[1], the corresponding frequencies and their harmonics can be precisely suppressed in the Fourier domain. For the case where this information is unavailable, the *adaptive spectral cleanup* illustrated in Figure 2.14 has proven very effective. In the Fourier domain, unnatural periodicities show as prominent peaks in the magnitude spectrum. Such peaks are identified by judging them against the local average and standard deviation computed within a sliding window of width $\frac{\pi}{5}$. If a value exceeds the local average by more than three times the local standard deviation, it is set to zero, and hence the corresponding spectral component eliminated from the signal. Figure 2.14 illustrates this for an example video that is available in two versions: as an uncompressed image sequence $v_0(t)$, and as low-quality H.264 encoded[2] video $v_1(t)$ with GOP structure IBBPBBP. Both versions are (re-)encoded with the x264 encoder at QP $= 40$ and GOP length 499, which leads to the bitrate sequences $r_0(t)$ and $r_1(t)$, respectively. Artifacts in $r_1(t)$ due to

[1] A given video might have been re-encoded multiple times, containing traces of several of the GOP structures used each time.

[2] produced with the H.264/AVC JM reference encoder avilable at http://iphome.hhi.de/suehring/tml/



**(a)**

**(b)**

**(c)**

***Figure 2.14:*** *Adaptive spectral cleanup: (a) Bitrate sequence $r_1(t)$ obtained by re-encoding a low quality video exhibiting an IBBPBBP GOP structure. Artifacts from the original GOP structure are clearly visible. (b) As a reference, the bitrate sequence $r_0(t)$ obtained by encoding the uncompressed image sequence. (c) With the adaptive spectral cleanup performed in the Fourier domain the GOP artifacts can largely be mitigated.*

the old GOP structure are clearly visible in Figure 2.14a. Corresponding spectral peaks occur in its Fourier transform $R_1(\omega)$ displayed in Figure 2.14c. Figure 2.14c also shows the adaptive threshold based on local average and standard deviation, and the clipped spectrum $\tilde{R}_1(\omega)$. The corresponding adjusted bitrate sequence $\tilde{r}_1(t)$ shown in Figure 2.14b comes very close to the artifact-free bitrate sequence $r_0(t)$. This adaptive spectral cleanup is versatile enough to mitigate other adverse periodicity effects as well. During frame rate conversion, for instance, repeated frames exhibit vanishing rate, and dropped ones disrupt predictability, leading to elevated rate in the following frame.

### Synchronization without Re-encoding

In some scenarios it might be impractical to specifically re-encode the source videos. Instead, in order to obtain useful bitrate sequences, the frame sizes are determined by parsing the existing representations. In that case, there is no control whatsoever over the exact nature of the encoding.

As long as both videos are encoded in H.264, there is a good chance that synchronization can, to a certain extent, be successful nonetheless. In the experiment reported in Figure 2.15, two H.264 videos produced with differing encoding parameters are synchronized with our approach. From the second video, a characteristic excerpt has been selected to ensure optimal synchronization in the case of identical parameters. For varying encoding parameters, the development of the synchronization error is then monitored. Within a range of reasonably similar QP and GOP length values, frame exact synchronization is achieved. Slight misalignments are encountered only if the encoding parameters, especially QP, deviate unduly between the videos.

In Figure 2.16, a more challenging scenario is investigated where two videos are encoded with different codecs. In this example, H.264 and MPEG-2 are used to produce the bitrate sequences for the synchronization process. With the given settings, the resemblant temporal behavior of both signals in Figure 2.16a is striking. As can be observed in Figure 2.16b, ZNCC retrieves the alignment almost perfectly. While the number of possible codec configurations is sheer limitless, this last experiment suggests the feasibility of bitrate-based synchronization for differently encoded input videos.

## 2.3 Consensus-based Cross-correlation (ConCor)

Apparently, cross-correlating bit-rate sequences is a simple, yet reliable approach to synchronize a pair of videos. So far, the template that was to be matched within the longer bitrate sequence had been manually selected from the shorter one. In order to be truly independent of user intervention, an automatic mechanism is necessary to select the most suitable parts from the sequences. This mechanism needs to discard parts of the second bitrate sequence $b(t)$ which do not overlap with the first one $a(t)$. Furthermore, it should identify those signal parts where non-stationary disturbances occur, either in $b(t)$ itself, or in corresponding parts of $a(t)$. Such disturbances can be of very different causes, including temporary occlusions present in one of the views, or sudden movements of one of the cameras that cannot be fully compensated by the video codec. Another effect that ren-

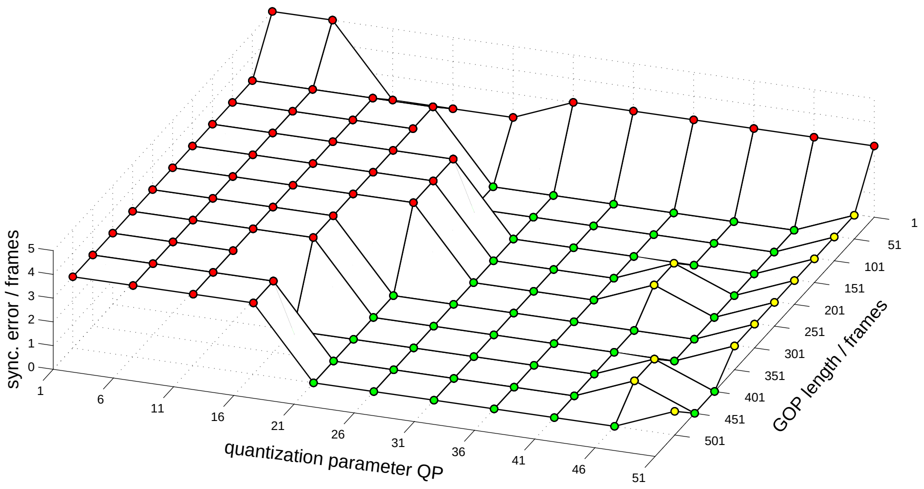
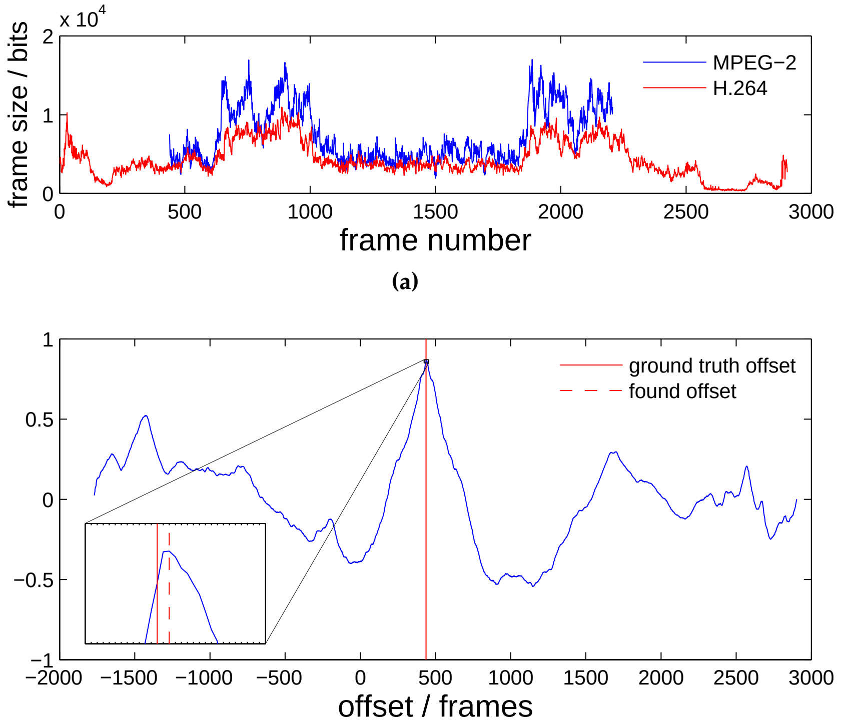


***Figure 2.15:*** *Bitrate-based synchronization with differing H.264 encoding parameters. A video encoded with QP = 36 and GOP length 451 is synchronized with a second view opposed by 180° and encoded with varying parameters, leading to the plotted synchronization errors.*



(a)

(b)

***Figure 2.16:*** *Bitrate-based synchronization with different codecs. Bitrate sequences obtained by H.264 (QP = 41, GOP length 499) and MPEG-2 (QP = 31, GOP length 351) are juxtaposed in (a). Despite the misalignment by 2 frames observed in (b), the principal temporal similarity is well apparent.*

ders specific signal parts useless for synchronization is, *e.g.*, the temporary lack of motion altogether.

With *consensus-based cross-correlation (ConCor)*, an algorithm has been devised that specifically addresses these requirements [SSE$^+$11]. ConCor can detect unapt signal parts, and exclude them from the computation of the cross-correlation measure. Not only does this increase the robustness against the described deranging effects, it also automatizes the process of template selection. In the following sections, the concept behind ConCor will be introduced, together with an extension based on the ZNCC. Finally, optimal values for ConCor's most significant parameters will be derived.

### 2.3.1 The Basic ConCor Algorithm

The basic idea behind consensus-based cross-correlation is to split one of the signals into shorter segments, and to cross-correlate each of them with the second signal:

$$\begin{aligned} b_i(t) &= \begin{cases} b(t) & : (i-1)M \le t < iM \\ 0 & : \text{else} \end{cases} \\ c_i(\Delta t) &= \sum_t a(t+\Delta t)\, b_i(t) \end{aligned} \tag{2.2}$$

In (2.2), the regular cross-correlation between $a(t)$ and $b_i(t)$ is computed. The resulting $c_i(\Delta t)$ are referred to as Partial Cross-Correlation Functions (PCCFs). The PCCFs are then combined using RANdom SAmple Consensus (RANSAC) [FB81] in order to separate the corrupted segments from the valid ones. In terms of RANSAC, the model to be fitted is the offset, hence a scalar quantity. The data points are the PCCFs which are combined, *i.e.*, summed up in order to determine potential offset candidates. Obviously, the sum of all PCCFs yields the cross-correlation function of the original signals:

$$\sum_i c_i(\Delta t) = \sum_i \sum_t a(t+\Delta t)\; b_i(t) = \sum_t a(t+\Delta t) \sum_i b_i(t) = \sum_t a(t+\Delta t)\; b(t) = c(\Delta t)$$

By omitting outlier PCCFs in the above summation, their influence on the resulting cross-correlation function can be specifically excluded.

The algorithm below summarizes the ConCor approach given two input sequences $a(t)$ and $b(t)$ of lengths $L_a$ and $L_b$, respectively, where $a(t)$ is assumed to be the longer of the two ($L_a \ge L_b$). Both sequences are assumed to be normalized with respect to their global means and standard deviations.

```
BASIC CONCOR ALGORITHM
  1. Chop the shorter signal into $m = \lfloor L_b/M \rfloor$ segments $b_i(t)$ of equal length $M$.
  2. Compute the PCCFs according to $c_i(\Delta t) = \sum a(t+\Delta t)\ b_i(t)$.
     Repeat ..................................................................
       3a) Make a random selection of $s$ PCCFs and compute their sum.
       3b) Extract candidate offsets from that sum.
```

3c) For every offset candidate, evaluate the number of consenting PCCFs (inliers).

........ *until confidence is reached that at least one outlier-free PCCF set has been selected.*

4) Select the offset with most consenting PCCFs .

5) Recompute the offset from the sum of consenting PCCFs .

■

As is usual for RANSAC, the stop criterion for the loop over steps 3abc is determined online, based on the current worst case estimate of the outlier ratio.

In step 3a of the algorithm, a single PCCF would essentially be sufficient to determine the offset between the input signals. The minimum number of data points required to fit the model is hence $s = 1$. However, PCCFs and their combinations are more trustworthy the more samples have been involved in their computation. Consequently, the total number $sM$ of effectively contributing samples should be raised to a maximum. In general, increasing $s$, hence combining several small segments rather than using few large ones, leads to a better chance of avoiding defective signal parts. In turn, including more samples always increases the risk of introducing corrupt samples. This trade-off in the choice of $s$ and $M$ depends on the burst error behavior of the considered signals and will be examined in detail in the next section.

During step 3b, a combination of PCCFs does not necessarily exhibit a single, conclusive peak. Instead, it will contain several local maxima of comparable strength, leading to more than one candidate offset. This effect is more pronounced when only a small number of PCCFs are combined or when each of them has been computed from very few samples. Multiple candidates are kept, each of which then undergoes the consensus check of step 3c with respect to all remaining PCCFs . Nevertheless, it makes sense to reduce the number of offset candidates, which can be achieved by morphological closure prior to the local maximum search. The closure operation leaves only the most dominant peaks intact, preserving their exact position, while less significant side lobes get filled in. A structuring element width of $50$ samples has been experimentally determined appropriate.

In step 3c, a PCCF votes for a given offset candidate if it has a local maximum in close proximity to it. To this end, the PCCF peaks within 10 frames from the candidate offset are counted. This threshold is tolerant enough to robustly allow for slight misalignments that can occur naturally. At the same time, it denies the support of PCCFs whose peaks are more than half a second away, and thus most likely unrelated.

Figures 2.17 to 2.21 illustrate the ConCor algorithm with an example. The used videos were captured with static cameras and show the same scene of a person acting in front of a static background. The input sequences $a(t)$ and $b(t)$ are exceptionally short, comprising only around $1200$ frames each, but have the instructive advantage to clearly reflect the actions of the recorded scene. For instance, the first notable peak in sequence $a(t)$ which attains its maximum around $t = 90$ corresponds to the main performer in the yellow sweatshirt entering the scene from the right, then pausing. During the second peak centered around $t = 200$, he approaches the paper bank and opens the hatch. He then climbs into the paper bank, which takes roughly until $t = 450$. The rather sharp peak just before $t = 500$ is caused by the hatch being abruptly closed. There is hardly any activity until $t = 600$. Then, both the container is reopened and the group of passers-by walk in

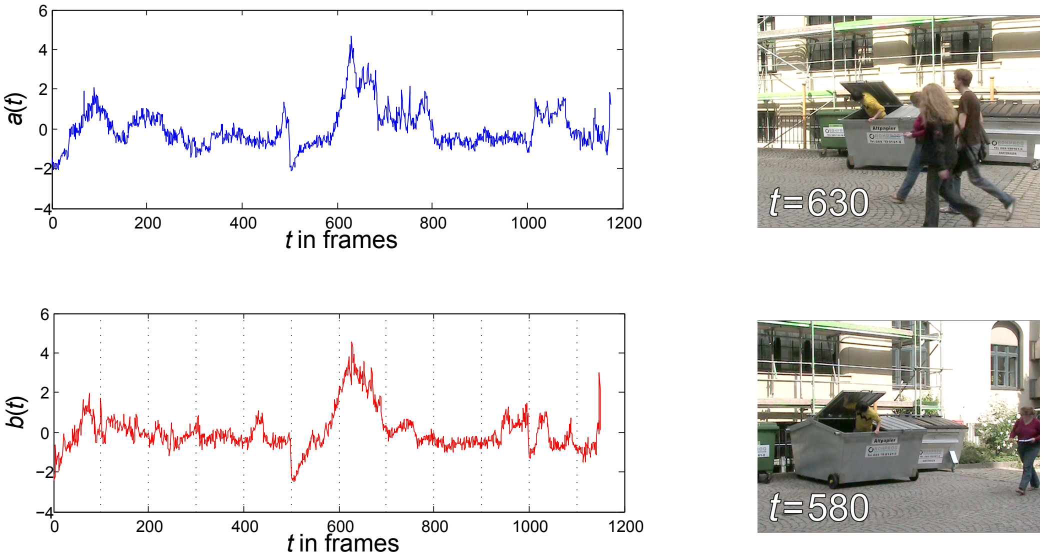


**Figure 2.17:** *The two input signals $a(t)$ and $b(t)$ and synchronous frames from the video sequences they have been derived from. The segmentation of $b(t)$ used in the consensus-based approach is indicated by the dotted vertical lines.*

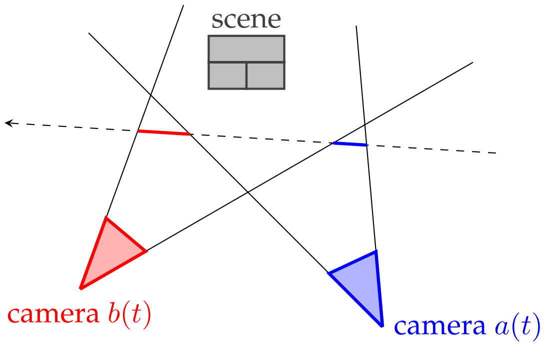


**Figure 2.18:** *Passer-by effect in schematic top view of the setup from Figure 2.17: Along the depicted traversal path, the blue and red parts lie in the field of view of only one of the cameras, having an impact either on $a(t)$ or $b(t)$, respectively.*

from the right. The group leave the scene approximately at $t = 700$, being responsible for the dominant peak between $t = 600$ and $t = 700$. After this event, the main performer jumps out of the container and leaves the scene as well, causing some minor activity in the bitrate signal $a(t)$ which lasts until $t = 800$. The scene remains without motion until $t = 1000$ where a second individual passes through the image from left to right.

The same events are observed by the second camera which is associated with the bitrate sequence $b(t)$. While the main line of action (hide-and-seek in the container) leaves bitrate traces equivalent to those in $a(t)$, the peripheral actions are slightly shifted in time. This is due to the differing viewpoints in the given camera setup: Events at the extreme right end of the scene are only visible in the image that corresponds to $a(t)$, while events occurring at

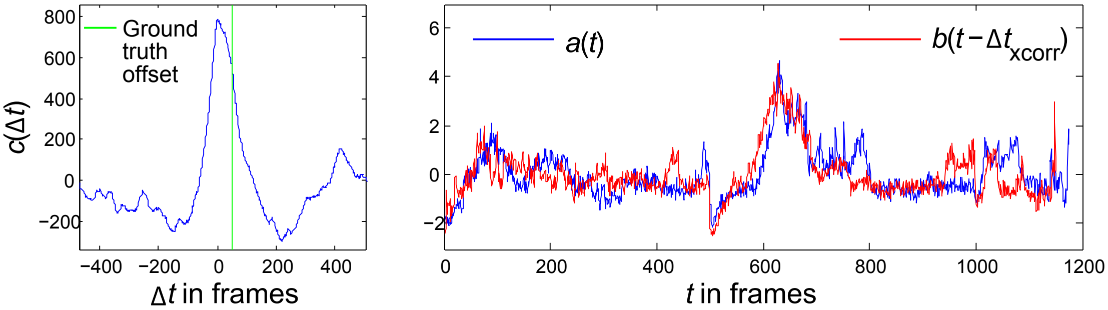


*Figure 2.19:* *Conventional normalized cross-correlation of sequences $a(t)$ and $b(t)$ yields the erroneous offset $\Delta t_{xcorr} = 1$ frame. This is due to the dominant, yet spurious, peak between frames 600 and 700 whose alignment is enforced.*

the very left exclusively show up in $b(t)$. Activities associated with passers-by traversing the scene from right to left thus appear slightly delayed in $b(t)$ as compared to $a(t)$, and vice versa. This incoherence due to peripheral motion is illustrated in Figure 2.18. Since the bitrate representation does not (and cannot) describe precise movements but rather the amount and complexity of motion present in each frame, these shifts pose a problem to regular cross-correlation. Seemingly shifted events would lead to an according misalignment of the bitrate sequences. In this example, the major peak caused by the group of three is subject to this problem, and because of its mere dominance, it determines the cross-correlation result, overruling more consistent signal parts. In Figure 2.19, this issue becomes apparent in the erroneous result of regular cross-correlation.

The aim of ConCor is to identify and exclude all incoherent signal parts. Figure 2.20 shows the eleven partial cross-correlation functions $c_i(\Delta t)$ resulting from a segmentation of $b(t)$ with $M = 100$. In this example, a combination of $s = 3$ PCCFs is taken in every iteration to generate offset hypotheses according to step 3b) of the algorithm. During the voting in step 3c), local maxima in the $c_i(\Delta t)$ assess the validity of each offset hypothesis. In the plots of Figure 2.20, this is exemplified for the ground truth offset marked by the vertical green line. With local maxima within the default distance threshold, $c_2(\Delta t)$, $c_3(\Delta t)$, $c_5(\Delta t)$, $c_8(\Delta t)$ and $c_{11}(\Delta t)$ are in support of the ground truth offset. These PCCFs correspond to the found inliers which eventually determine the ConCor result, presented in Figure 2.21.

### 2.3.2 Normalized ConCor

In [SSE$^+$11], as well as in the previous section, ConCor was proposed as an extension to regular cross-correlation. Neither the PCCFs $c_i(\Delta t)$ nor their combinations had been normalized with respect to the partial signals' means or standard deviations. Instead, the global means had been removed from $a(t)$ and $b(t)$, and both rescaled with their global standard deviations. The reason for this was to gain independence of the signal magnitudes, emphasizing their similarities in signal shape. If the bitrate sequences were wide-sense stationary, this global treatment would be equivalent to true normalization in the sense of ZNCC. For realistic bitrate sequences, however, this is only a first approximation.

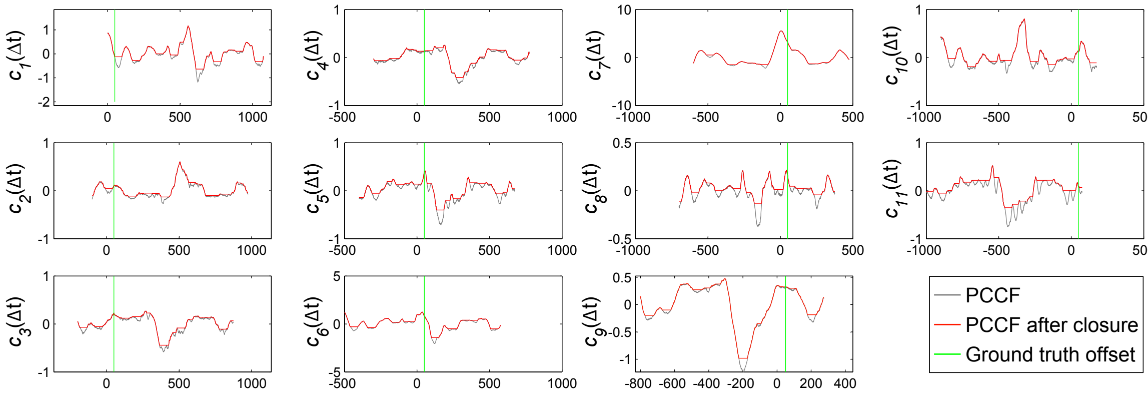


***Figure 2.20:*** *Partial cross-correlation functions ( PCCFs ) corresponding to the signals in Figure 2.17.*

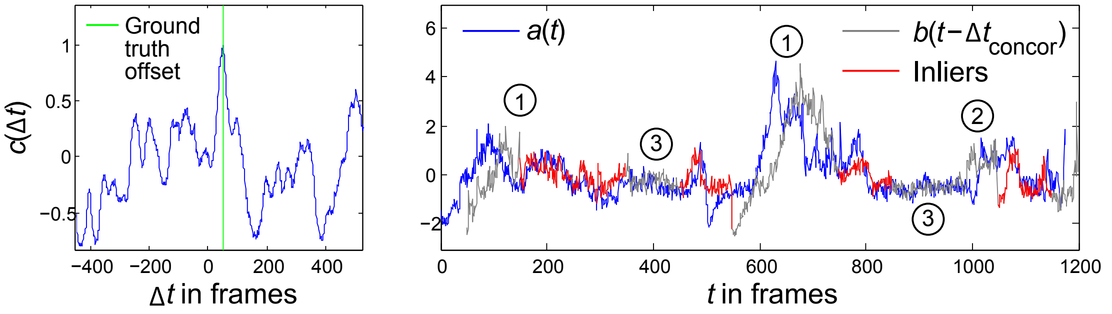


***Figure 2.21:*** *Consensus-based cross-correlation yields the correct offset $\Delta t_{concor} = 50$ frames. In particular, segments $b_i(t)$ are discarded where ① people walk into the scene from the right, ② people walk in from the left, and ③ where there is not enough scene motion to reasonably establish temporal relationships between both videos.*

In the following, modifications to the basic ConCor algorithm will be proposed in order to incorporate normalization.

To this end, several auxiliary quantities need to be defined. In particular, the following moving averages computed over $M$ consecutive samples of $a(t)$ are required:

$$\overline{a}(\Delta t) = \frac{1}{M} \sum_{t=\Delta t}^{\Delta t+M-1} a(t) \qquad \textit{(moving average)} \tag{2.3a}$$

$$\overline{a^2}(\Delta t) = \frac{1}{M} \sum_{t=\Delta t}^{\Delta t+M-1} a^2(t) \qquad \textit{(moving average energy)} \tag{2.3b}$$

$$\sigma_a(\Delta t) = \sqrt{\overline{a^2}(\Delta t) - (\overline{a}(\Delta t))^2} \qquad \textit{(moving standard deviation)} \tag{2.3c}$$

Corresponding measures are considered for the individual segments $b_i(t)$:

$$\overline{b}_i = \frac{1}{M} \sum_{t=(i-1)M}^{iM-1} b(t) \qquad \text{(segment average)} \tag{2.4a}$$

$$\overline{b_i^2} = \frac{1}{M} \sum_{t=(i-1)M}^{iM-1} b^2(t) \qquad \text{(average segment energy)} \tag{2.4b}$$

$$\sigma_{b_i} = \sqrt{\overline{b_i^2} - \overline{b}_i^2} \qquad \text{(segment standard deviation)} \tag{2.4c}$$

The *normalized partial cross-correlation functions* (NPCCFs ), defined as the ZNCC of $a(t)$ with each of the $b_i(t)$, can then be expressed as follows:

$$\widetilde{c_i}(\Delta t) = \frac{1}{M} \sum_{t=(i-1)M}^{iM-1} \frac{a(t+\Delta t) - \overline{a}_i(\Delta t)}{\sigma_{a_i}(\Delta t)} \cdot \frac{b_i(t) - \overline{b}_i}{\sigma_{b_i}}, \tag{2.5}$$

where $\overline{a}_i(\Delta t) := \overline{a}(\Delta t + (i-1)M)$ and $\sigma_{a_i}(\Delta t) := \sigma_a(\Delta t + (i-1)M)$ are shifted versions of the moving average and standard deviation from Equations (2.3a) and (2.3c).

It can be shown that the Normalized PCCF (NPCCF)s $\widetilde{c_i}(\Delta t)$ relate to the corresponding PCCFs $c_i(\Delta t)$ from Section 2.3.1 in the following way:

$$\widetilde{c_i}(\Delta t) = \frac{c_i(\Delta t) - M \cdot \overline{a}_i(\Delta t) \cdot \overline{b}_i}{M \cdot \sigma_{a_i}(\Delta t) \cdot \sigma_{b_i}}$$

It follows that, in order to combine several NPCCFs , they need to be denormalized, added up, and the sum renormalized according to the union of all participating segments. Let $I = \{i_1, i_2, \ldots, i_s\}$ be the index set of $s$ segments to be combined. The combination of NPCCFs , denoted by $\widetilde{c_I}(\Delta t)$, can then be calculated as follows:

$$\widetilde{c_I}(\Delta t) = \frac{\sum_{i \in I} \left[\sigma_{a_i}(\Delta t) \cdot \sigma_{b_i} \cdot \widetilde{c_i}(\Delta t) + \overline{a}_i(\Delta t) \cdot \overline{b}_i\right] - s \cdot \overline{a}_I(\Delta t) \cdot \overline{b}_I}{s \cdot \sigma_{a_I}(\Delta t) \cdot \sigma_{b_I}} \tag{2.6}$$

The quantities relating to the union of segments can be readily calculated from the corresponding measures initially computed subject to (2.3a), (2.3b), (2.4a) and (2.4b):

$$\overline{a}_I(\Delta t) = \frac{1}{s}\sum_{i\in I}\overline{a}_i(\Delta t) = \frac{1}{s}\sum_{i\in I}\overline{a}(\Delta t + (i-1)M) \tag{2.7a}$$

$$\overline{a^2_I}(\Delta t) = \frac{1}{s}\sum_{i\in I}\overline{a^2_i}(\Delta t) = \frac{1}{s}\sum_{i\in I}\overline{a^2}(\Delta t + (i-1)M) \tag{2.7b}$$

$$\sigma_{a_I} = \sqrt{\overline{a^2_I}(\Delta t) - \overline{a}^2_I(\Delta t)} \tag{2.7c}$$

$$\overline{b}_I = \frac{1}{s}\sum_{i\in I}\overline{b}_i \tag{2.8a}$$

$$\overline{b^2_I} = \frac{1}{s}\sum_{i\in I}\overline{b^2_i} \tag{2.8b}$$

$$\sigma_{b_I} = \sqrt{\overline{b^2_I} - \overline{b}_I^{\,2}} \tag{2.8c}$$

Ultimately, the basic ConCor algorithm only requires modification in that additional quantities be precomputed, and steps 3a) and 5) be adjusted:

```
NORMALIZED CONCOR ALGORITHM
1. Chop the shorter signal into $m = \lfloor L_b/M \rfloor$ segments $b_i(t)$ of equal length $M$.
2. Compute $\overline{a}(\Delta t)$, $\overline{a^2}(\Delta t)$, $\sigma_a(\Delta t)$, and all $\overline{b}_i$, $\overline{b^2_i}$ and $\sigma_{b_i}$ according to Equations (2.3a)
   through (2.4c), as well as the NPCCFs $\widetilde{c}_i(\Delta t)$ following Equation (2.5).
   Repeat ................................................................
     3a) Make a random selection of $s$ NPCCFs and combine them according to Equation (2.6).
     3b) Extract candidate offsets from the combination $\widetilde{c}_I(\Delta t)$.
     3c) For every offset candidate, evaluate the number of consenting NPCCFs (inliers).
   .......until confidence is reached that at least one outlier-free NPCCF set has been selected.
4) Select the offset with most consenting NPCCFs .
5) Recompute the offset from the combination of all consenting NPCCFs according to
   Equation (2.6).
```

The added computational burden is moderate, amounting to $O(1)$ for the precomputations in step 2). In steps 3a) and 5) of the algorithm, the NPCCF combination according to Equations (2.6) through (2.8c) comes at a complexity increase by $O(n)$, based on the input signal lengths.

The tremendous advantage, however, is the improved localizability of segments $b_i(t)$ within the longer sequence $a(t)$, with all the benefits that ZNCC has over unnormalized cross-correlation.

### 2.3.3 Optimal ConCor Parameters

There is the fundamental trade-off in choosing a segment length $M$ and the number $s$ of segments to be combined in each RANSAC step. On the one hand, it is desirable to have as many samples as possible contribute to the cross-correlation functions computed in every RANSAC iteration. On the other hand, increasing $M$ leads to a higher risk of involving corrupt samples in each segment, and raising $s$ obviously increases the chance to include one or more outlier segments. In the following, the optimal choice for $M$ and $s$ will be determined, maximizing the total number of involved samples while maintaining a high probability that RANSAC can successfully deliver a solution. The event of RANSAC success is defined as follows:

$\mathcal{R}_N$ : "An outlier-free set of segments is selected at least once in at most $N$ random draws."

The optimization problem to be solved is then given by

$$\max_{s,M} sM, \qquad \text{s.t.} \quad \Pr\{\mathcal{R}_N\} \geq 0.99, \quad M \geq 50, \tag{2.9}$$

where an arbitrarily high confidence level of $99\%$ has been chosen. The lower bound on $M$ stipulated in (2.9) is intended to ensure that individual (N) PCCFs are sufficiently meaningful to be used for consensus checks. The minimum length of $50$ samples is empirically motivated and corresponds to 2 seconds of video for 25 fps footage. It should be noted that this constraint on $M$ is hardly determining for the maximization in practical scenarios. The parameter $N$ bounds the complexity of ConCor, with typical values ranging from $1000$ to $10000$.

The following notation and assumptions will further be used: The sequences $a(t)$ and $b(t)$ are $L_a$ and $L_b$ samples long, where $L_a \geq L_b$ by definition. Consequently, $b(t)$ contains $m = \lfloor L_b/M \rfloor$ full segments of length $M$. We assume a worst-case overlap between $a(t)$ and $b(t)$ of at least $L_0$ samples.

As discussed earlier, there are two main effects responsible for a sample $b(t)$ to be corrupt. Either it lies outside the overlap region of both signals, or it is part of a burst error in the video content itself (due to occlusions, lack of motion, etc.). Both phenomena will be treated jointly in the following.

#### Burst Error Model

To realistically model the occurrence of burst errors, a homogeneous, two-state Markov chain is devised with transition matrix $\mathbf{P} = \left(\begin{smallmatrix} p_0 & 1-p_1 \\ 1-p_0 & p_1 \end{smallmatrix}\right)$, as depicted in Figure 2.22a. Let $x(t)$ denote the state of sample $b(t)$ which can either be error-free ($x(t) = 1$) or corrupt ($x(t) = 0$). The transition probabilities

$$p_0 = \Pr\{x(t)\!=\!0 \,|\, x(t\!-\!1)\!=\!0\} \quad \text{and} \quad p_1 = \Pr\{x(t)\!=\!1 \,|\, x(t\!-\!1)\!=\!1\} \tag{2.10}$$

determine how likely it is for the current sample to remain in the same state as the previous one. The complementary probabilities $(1–p_0)$ and $(1–p_1)$ quantify the rate of state changes,

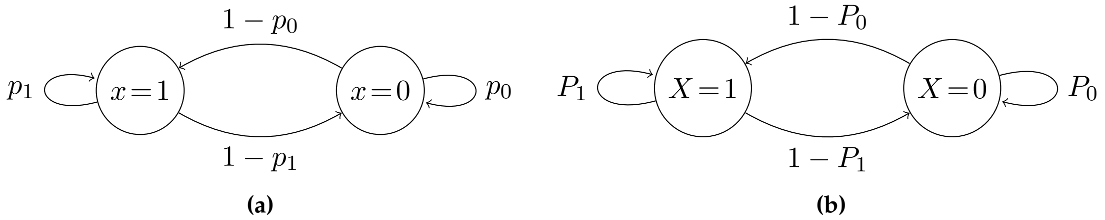


***Figure 2.22:*** *The Markov chains governing the occurrence of corrupt samples (a), and outlier segments (b). The states* $0$ *and* $1$ *represent erroneous and valid data, respectively.*

accordingly.

In order to describe the Markov chain in a more intuitive way, it is parametrized by its *mean error burst length* $B$, and its *steady-state sample error rate* $p$. For an error burst to last $l$ samples, the Markov chain needs to remain in the corrupt state exactly $(l-1)$ times before leaving it. The probability for this to happen is $p_0^{l-1}(1-p_0)$, and consequently:

$$B = \mathrm{E}\{l\} = \sum_{l=1}^{\infty} l\, p_0^{l-1}(1-p_0) = \frac{1}{1-p_0} \tag{2.11}$$

The stationary distribution of $x(t)$ can be computed as the eigenvector of $\mathbf{P}$ corresponding to its unity eigenvalue. It is thus the solution to $\begin{pmatrix} p_0-1 & 1-p_1 \\ 1-p_0 & p_1-1 \end{pmatrix} \begin{pmatrix} p \\ 1-p \end{pmatrix} = \mathbf{0}$, resulting in

$$p = \frac{1-p_1}{2-p_0-p_1}. \tag{2.12}$$

Accordingly, the recurrence probabilities can be expressed as a function of $B$ and $p$ in the following way:

$$p_0 = \frac{B-1}{B}, \qquad p_1 = 1 - \frac{p}{B(1-p)} \tag{2.13}$$

Both $B$ and $p$ can be set to default values or adapted to a particular class of scenarios, allowing for specific values of the expected duration and frequency of disturbances. For instance, videos acquired during a sports event will contain many frequent but possibly brief occlusions, while surveillance videos are generally more steady.

Given this model for the sample outlier occurrence parametrized by $p$ and $B$, let's now turn to the error distribution for segments of length $M$ samples. There are $m = \lfloor L_b/M \rfloor$ such segments, each of which is considered an outlier if it contains one or more corrupt samples. The random process determining the occurrence of outlier segments then obeys a Markov chain too, very similar to the one governing the sample state (see Figure 2.22b). Capital letters are used for the segment-based quantities, namely $X_i \in \{0,1\}$ for the state of the $i$-th segment, and $P_0$, $P_1$ and $P$ for the recurrence probabilities and the steady-state error rate, respectively.

If the previous segment was an inlier, it takes $M$ recurring error-free samples for the current segment to be entirely error-free as well, thus

$$P_1 = \Pr\{X_i\!=\!1|X_{i-1}\!=\!1\} = p_1^M. \tag{2.14}$$

An arbitrary segment is an inlier if its first sample is error-free, and the $(M-1)$ subsequent samples are too. Accordingly, the complementary event that the segment is an outlier occurs with probability

$$P = 1 - (1-p)\, p_1^{M-1}. \tag{2.15}$$

With a relationship analog to the one in Equation (2.12), the remaining transition probability can then be expressed as

$$P_0 = 2 - P_1 - \frac{1-P_I}{P}. \tag{2.16}$$

To evaluate the overall RANSAC success probability $\Pr\{\mathcal{R}_N\}$ for a given pair of $s$ and $M$, first the probability $\Pr\{\mathcal{D}\}$ is computed indicating how likely it is that, in one of the RANSAC iterations, a random draw of $s$ segments is successful.

$\mathcal{D}$: "$s$ randomly selected segments are all inliers."

Ignoring the overlap effect for the moment, let's define the following random variables:

$$G = \sum_{i=1}^{m} X_i, \ \text{ the number of inlier segments,} \tag{2.17}$$

$$H = X_1 + X_m, \ \text{ and} \tag{2.18}$$

$$K = \sum_{i=2}^{m} X_{i-1}X_i, \ \text{ the correlation between adjacent segments.} \tag{2.19}$$

It can be shown [Klo72, Klo73] that the joint probability of $G$, $H$, and $K$ is given by

$$\Pr\{G\!=\!g, H\!=\!h, K\!=\!k\} = \binom{2}{h}\binom{g-1}{k}\binom{m-g-1}{g-k-h}\,\alpha\,\beta^g\,\gamma^h\,\delta^k \tag{2.20a}$$

$$\text{where} \quad \alpha = (1-2P+P_1P)^{m-1} \,/\, (1-P)^{m-2}, \tag{2.20b}$$

$$\beta = (1-P_1)^2\,P\,(1-P) \,/\, (1-2P+P_1P)^2, \tag{2.20c}$$

$$\gamma = (1-2P+P_1P) \,/\, ((1-P)(1-P_1)). \tag{2.20d}$$

The probability that, among the $m$ segments of $b(t)$, exactly $g$ are inliers is then given by

$$\Pr\{G=g\} = \sum_{h=0}^{2}\sum_{k=0}^{m-1} \Pr\{G\!=\!g, H\!=\!h, K\!=\!k\}. \tag{2.21}$$

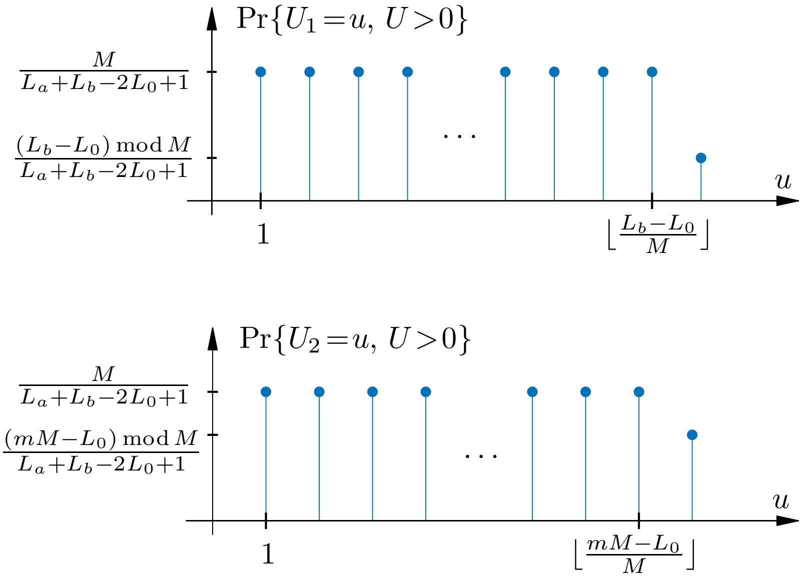


***Figure 2.23:*** *Probability distribution of the numbers $U_1$ and $U_2$ of unusable segments at the head and tail of $b(t)$, respectively.*

Together with

$$\Pr\{\mathcal{D}|G = g\} = \prod_{i=0}^{s-1} \frac{g-i}{m-i} = \binom{g}{s} \Big/ \binom{m}{s} \tag{2.22}$$

and

$$\Pr\{\mathcal{R}_N|G = g\} = 1 - (1 - \Pr\{\mathcal{D}|G = g\})^N \,, \tag{2.23}$$

this eventually leads to

$$\Pr\{\mathcal{R}_N\} = \sum_{g=0}^{m} \Pr\{\mathcal{R}_N|G = g\} \Pr\{G = g\}. \tag{2.24}$$

### Overlap Effect

In order to incorporate the so far neglected overlap effect, the random variable $U$ is introduced counting the segments of $b(t)$ that are not fully contained in the overlap region, and thus unusable. Obviously, $U$ depends on the unknown offset between both sequences. Assuming that the offset is uniformly distributed between $(L_0{-}L_b)$ and $(L_a{-}L_0)$, such that the assumed minimum overlap of $L_0$ samples is guaranteed, there are $(La{+}L_b{-}2L_0{+}1)$ equiprobable shifts in total. For $(L_a{-}mM{+}1)$ of them, all segments of $b(t)$ fully overlap with sequence $a(t)$, hence $U = 0$. Let's furthermore define $U_1$ and $U_2$ as the number of unusable segments at the beginning and at the end of $b(t)$, separately. By definition, $b(t)$ is the shorter sequence, so either its head *or* its tail protrudes beyond $a(t)$. Hence, the events

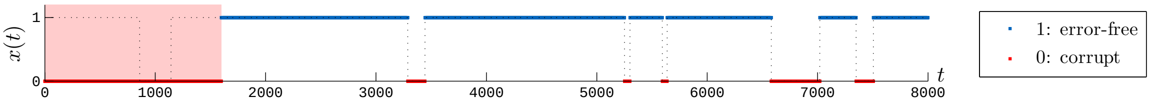


*Figure 2.24: Possible realization of the sample state $x(t)$ of sequence $b(t)$ according to our joint error model, with $L_a = 10000$, $L_b = 8000$, $L_0 = 4000$, $p = 10\%$ and $B = 150$. The dotted line indicates the Markov process generating error bursts, the region shaded in red marks the samples outside the mutual overlap of $a(t)$ and $b(t)$.*

$U_1 = u$ and $U_2 = u$ are mutually exclusive as long as $u > 0$. Accordingly, the distribution of $U$ can be expressed as given in Equation (2.25) below.

$$\Pr\{U = u\} = \begin{cases} \frac{L_a - mM + 1}{L_a + L_b - 2L_0 + 1} & : u = 0 \\ \Pr\{U_1 = u\} + \Pr\{U_2 = u\} & : u > 0 \end{cases} \tag{2.25}$$

Shifting the sequence $b(t)$ to the left, more and more segments leave the overlap region, $U_1$ being incremented every $M$ samples. In the worst case, $\lfloor \frac{L_b - L_0}{M} \rfloor$ segments lie completely outside the overlap region, and another one protrudes by $((L_b - L_0) \bmod M)$ samples. Similarly, if $b(t)$ is shifted to the right, $U_2$ will increase by one every $M$ samples, eventually leading to $\lfloor \frac{mM - L_0}{M} \rfloor$ non-overlapping segments and one partially protruding by $((mM - L_0) \bmod M)$ samples. The resulting distributions of $U_1$ and $U_2$ are depicted in Figure 2.23.

With $U$ unusable segments due to the overlap effect, the number of available segments in $b(t)$ is de facto reduced from $m$ to $(m - U)$. This requires modifications to the equations involved in the computation of $\Pr\{G = g\}$, namely (2.20a), (2.20b) and (2.21), which become

$$\Pr\{G = g, H = h, K = k \,|\, U = u\} = \binom{2}{h} \binom{g-1}{k} \binom{m-u-g-1}{g-k-h} \tilde{\alpha}\, \beta^g\, \gamma^h\, \delta^k, \tag{2.26a}$$

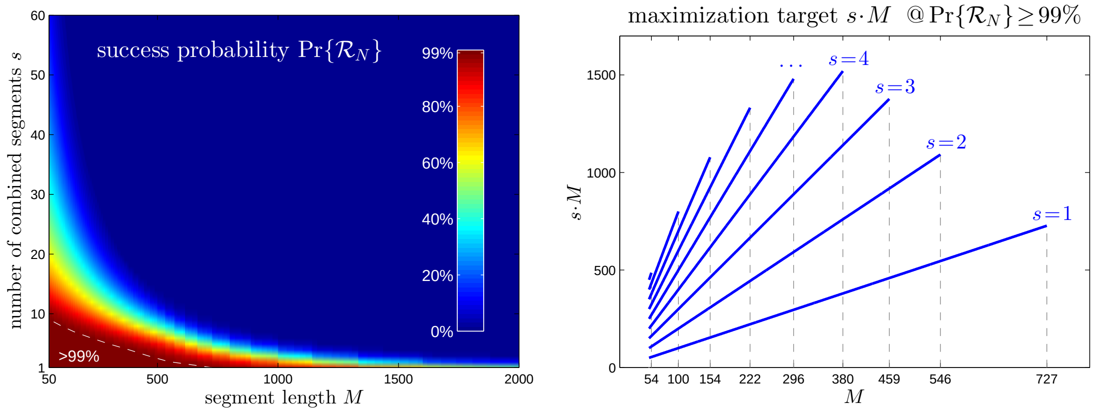
$$\text{where} \quad \tilde{\alpha} = (1 - 2P + P_1 P)^{m-u-1} \,/\, (1 - P)^{m-u-2}, \tag{2.26b}$$

$$\Pr\{G = g\} = \sum_h \sum_k \sum_u \Pr\{G = g, H = h, K = k \,|\, U = u\} \; \Pr\{U = u\}. \tag{2.27}$$

The rest of the equations, especially (2.22) through (2.24), remain valid.

Figure 2.24 shows an example realization generated by this error model which is determined by the burst-error characteristics of the sequences, described by $B$ and $p$, the sequence lengths $L_a$ and $L_b$, and their minimum overlap $L_0$.

Fig. 2.25a shows the behavior of the ConCor success probability $\Pr\{\mathcal{R}_N\}$ for this case. As to be expected, there is a high probability of avoiding corrupt samples when $s$ and $M$ are small, *i.e.*, when only a small number of short segments are used. For increasing values of $s$ and $M$ the success probability drops rapidly. There is only a relatively small region in the $(s, M)$ plane for which the success probability exceeds the stipulated $99\%$. For all combinations of $s$ and $M$ within this region, Fig. 2.25b shows the total number



**(a)** **(b)**

***Figure 2.25:*** *(a) The ConCor success probability for the specific error model from Fig. 2.24 with* $L_a = 10000$*,* $L_b = 8000$*,* $L_0 = 4000$*,* $p = 10\%$*,* $B = 150$*,* $N = 10000$*. For the values of* $s$ *and* $M$ *where* $\Pr\{\mathcal{R}_N\} \geq 99\%$*, the number of effectively used samples* $s{\cdot}M$ *is plotted in (b).*

| | $p$=1% | $p$=2% | $p$=3% | $p$=4% | $p$=5% | $p$=**10%** | $p$=20% | $p$=50% |
|---|---|---|---|---|---|---|---|---|
| $B = 10$ | $s$=5<br>$M$=346 | $s$=4<br>$M$=296 | $s$=4<br>$M$=232 | $s$=4<br>$M$=190 | $s$=4<br>$M$=161 | $s$=3<br>$M$=119 | $s$=3<br>$M$=59 | n/a |
| $B = 25$ | $s$=5<br>$M$=470 | $s$=5<br>$M$=363 | $s$=4<br>$M$=375 | $s$=4<br>$M$=320 | $s$=4<br>$M$=280 | $s$=4<br>$M$=170 | $s$=3<br>$M$=119 | $s$=1<br>$M$=58 |
| $B = 50$ | $s$=5<br>$M$=571 | $s$=5<br>$M$=467 | $s$=5<br>$M$=400 | $s$=5<br>$M$=347 | $s$=5<br>$M$=307 | $s$=4<br>$M$=252 | $s$=3<br>$M$=186 | $s$=2<br>$M$=78 |
| $B = 100$ | $s$=5<br>$M$=615 | $s$=5<br>$M$=533 | $s$=5<br>$M$=470 | $s$=5<br>$M$=421 | $s$=5<br>$M$=380 | $s$=4<br>$M$=339 | $s$=4<br>$M$=199 | $s$=3<br>$M$=78 |
| $B =$ **150** | $s$=6<br>$M$=533 | $s$=5<br>$M$=571 | $s$=5<br>$M$=500 | $s$=5<br>$M$=470 | $s$=5<br>$M$=421 | $s$=**4**<br>$M$=**380** | $s$=4<br>$M$=234 | $s$=3<br>$M$=96 |

***Table 2.1:*** *Optimal values of* $s$ *and* $M$ *for two sequences with* $L_a = 10000$*,* $L_b = 8000$*,* $L_0 = 4000$*, given a maximum of* $N = 10000$ *RANSAC iterations. The bold numbers correspond to the signal class represented in Figures 2.24 and 2.25.*

of samples $s \cdot M$ that are effectively used in every ConCor iteration. In this example, the global maximum is attained with $s = 4$ and $M = 380$ which corresponds to a total of 1520 contributing samples. An exhaustive search or any suitable discrete maximization strategy can be used to determine these optimal values.

In Table 2.1, optimal values for $s$ and $M$ are listed for example sequences of varying error characteristics $p$ and $B$. In the extreme case where $p = 50\%$ and $B = 10$, the confidence requirement of 99% cannot be met.

To summarize, a probabilistic model has been established that realistically describes the disturbances which occur in bitrate sequences extracted from videos. It covers both the existence of unapt samples in either of the sequences, as well as the loss of samples outside the mutual overlap. It links the characteristics of the videos ($L_a$, $L_b$), of the expected disturbances ($p$, $B$, $L_0$) as well as the ConCor parameters ($s$, $M$, $N$) to the success probability of our approach.

## 2.4 Experimental Results

Extensive tests have been performed to validate the algorithm. For Section 2.4.1, specific video pairs have been selected that exhibit certain characteristics typically not handled by the state-of-the-art video synchronization algorithms. In Section 2.4.2, ConCor is used to synchronize a number of publicly available multi-perspective recordings, and specifically those provided by other groups who have developed video synchronization algorithms. Specifically, unless mentioned otherwise, normalized ConCor is applied to bitrate sequences generated with the x.264 encoder at QP$=40$, with GOP length $499$ frames. Furthermore, the adaptive spectral cleanup described in Section 2.2.2 is used. The videos stem from diverse sources, and vary in codec, resolution, original encoding parameters, etc. (see Table 2.2). With that said, the used ConCor parameters have been derived from a very generic error assumption which allows for a mean error burst length of $B=50$ frames and a sample error rate up to $p = 5\%$. A maximum number of $N = 10000$ RANSAC iterations are allowed and a minimum overlap of $50\%$ of the shorter sequence's length is assumed. Unless otherwise available, ground truth offsets have been determined by visual inspection.

### 2.4.1 ConCor Tested for Different Effects

Here a number of effects are described that ConCor is able to handle successfully. The focus is on four particularly interesting effects that pose severe challenges to the state-of-the-art:

1. **Wide baselines**: Approaches that rely on matching texture features suffer from the limited invariance towards view point changes. Hence feature-based approaches not only exclude scenes where the cameras stand opposite to one another but also scenes with wide angle overlapping views.
2. **Camera motion**: A "shaking" camera renders many synchronization algorithms unusable. However, the application of video synchronization of casually captured multi-perspective events requires an algorithm that can handle also this kind of scenario.
3. **Dynamic backgrounds**: A changing background can confuse any synchronization algorithm in identifying the real object of interest.
4. **Occlusions**: Another problematic effect, especially for approaches that involve the tracking of features, is the temporary disappearance of objects of interest, as well as self-occlusions among several moving objects.

In the following four synchronization scenarios are presented which address the issues pointed out above.

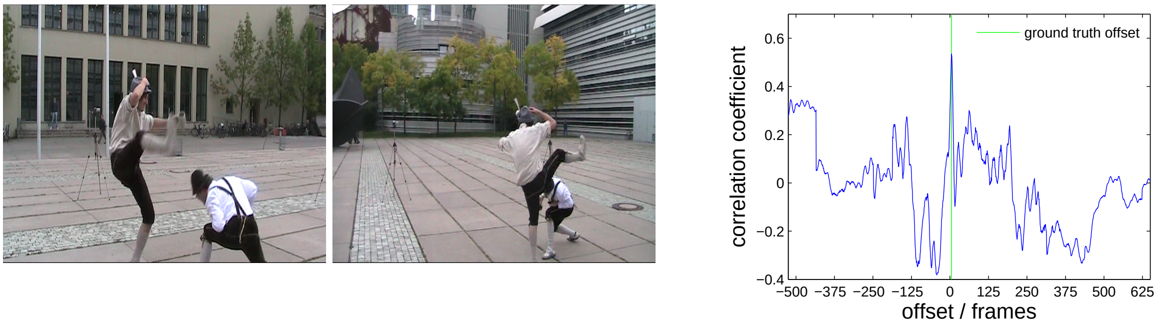


***Figure 2.26:*** *Synchronous frames from the* CapoEHA *sequences and synchronization outcome.*

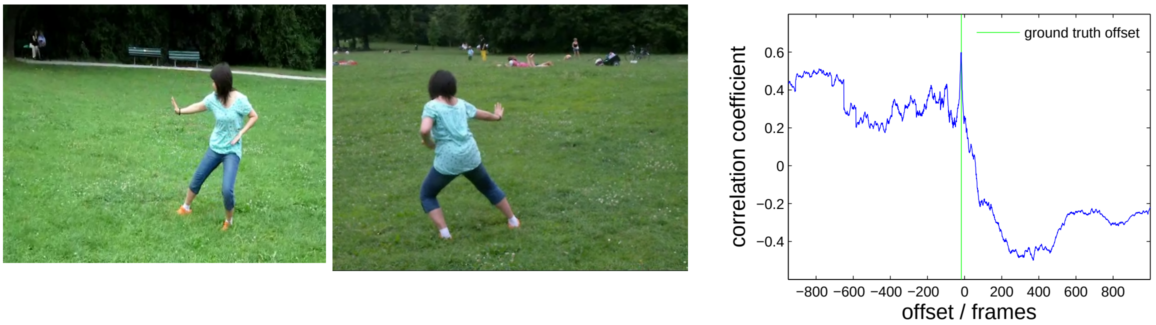


***Figure 2.27:*** *Synchronous frames from the* Taiji *sequences and synchronization outcome.*

The screen shots shown in Figure 2.26 show two views of the *CapoEHA* video set. As can be seen, the perspectives differ in viewing direction by approximately 90°. Furthermore, since the two subjects wildly dance around each other, frequent self-occlusions occur, and motion blur becomes an issue. The background appears completely different in the two views but is mostly static, with the exception of single pedestrians. One of the cameras is hand-held which introduces slight shaking movements. It can be seen from the result on the right of Figure 2.26 that frame accurate synchronization is achieved by our approach nonetheless.

The second experiment, presented in Figure 2.27, showcases ConCor's ability to deal with videos recorded with different cameras. The videos, showing *Taiji* exercises, most notably differ in spatial resolution. This example contrasts with the previous one in that scene motion is way more subtle, whereas background motion is more prominent. The cameras are more than 90° apart, both are hand-held. In one view, several persons walk behind the main actress, in the other one non-involved persons are visible in the background playing ball. Despite these unrelated distractions ConCor successfully syncs the video pair and finds the correct offset.

An exceptional effect is covered in a third demonstration. The screen shots shown in Figure 2.28 are taken from a scene where three individuals engage in rope skipping. The inherently repetitive motion pattern can be expected to be highly challenging for a method that seeks to find the offset by correlating a motion measure. Nevertheless, ConCor successfully recovers the temporal delay. A remarkable detail about these videos is the wall-filling mirror in the background which obviously aids our approach by multiplying the

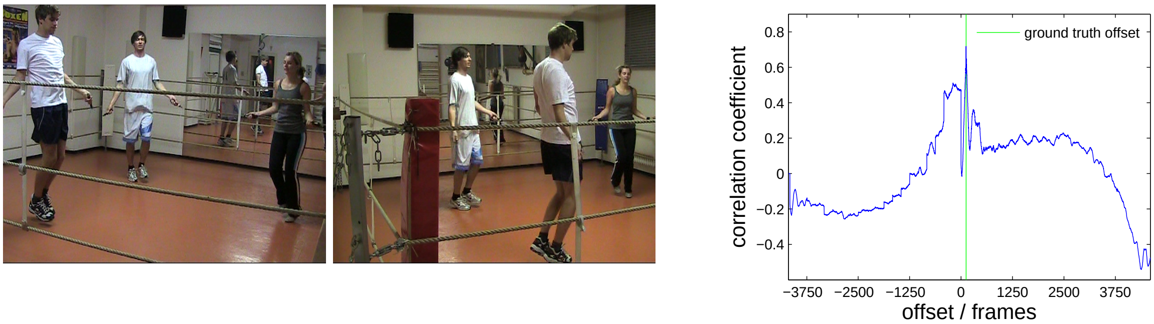


***Figure 2.28:*** *Synchronous frames from the* Rope Skipping *sequences and synchronization outcome.*

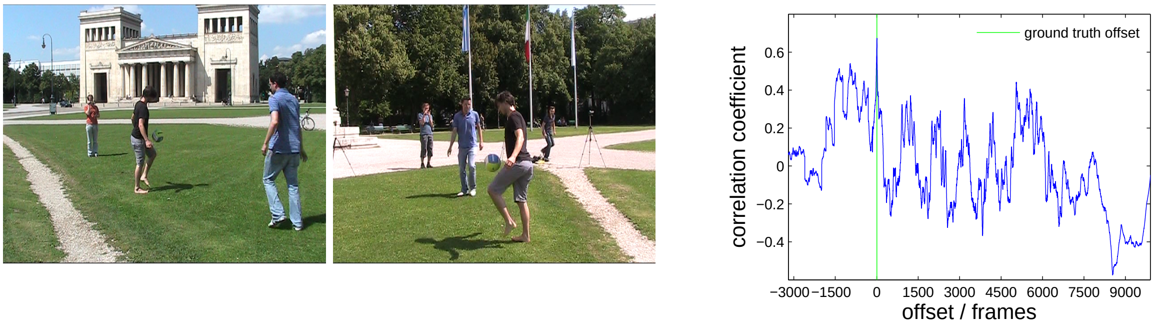


***Figure 2.29:*** *Synchronous frames from the* Soccer *sequences and synchronization outcome.*

effective amount of scene motion.

Finally, a scenario is presented with a very realistic composition of all the discussed effects. In the videos shown in Figure 2.29, a ball is kicked back and forth by two individuals who are surrounded by spectators filming the event. The selected views are diametrically opposed to each other and show significant background motion; one of the cameras is hand-held. The particular challenge is that both players regularly leave the cameras' fields of view, which leads to inconsistencies to be detected and eliminated by ConCor. According to the synchronization result, ConCor is able to achieve this, yielding accurate alignment.

### 2.4.2 Evaluation of ConCor on External Data Sets

Here, ConCor's performance is demonstrated by reference to a number of scenario relevant video sets available for download from external sources. Focus is particularly on video sets that are provided by authors of other video synchronization algorithms. In the lower half of Table 2.2, the tested videos are summarized, Figure 2.30 shows the obtained synchronization results. Unfortunately, some of the available clips are markedly short, comprising few hundred frames only. An asterisk in the third column of the table indicates those videos too short to be divided into enough segments of acceptable length. An optimal choice of the $s$ and $M$ parameters according to (2.9) being impossible in these cases, we select the parameters such that the prospect of success $\Pr\{\mathcal{R}_N\}$ is maximized for the least permissible segment length, *i.e.*, $M = 50$ frames and $s = 1$. In the exceptional case of the extremely short *Train* sequences, it is necessary to further lower the segment length to $M = 45$, in order to obtain at least three segments for ConCor to work with.

The *Dog*, *Martial Arts* and *HumanEva* datasets have all been produced in a meticulously controlled studio environment, and show one or two active performers in front of orderly, static backgrounds. From these sets, video pairs with increasing viewpoint difference, ranging from around 20° to 180°, have been chosen. All of them are successfully synchronized by the presented approach. A particularity about the selected *Martial Arts* pair is that one of the videos shows a top view of the scene while the second one portrays the action from an unusually low worm's eye perspective.

The *Basketball* and *Hall* sequences have been recorded in natural environments, both with stationary cameras facing each other. They show the outdoor practice of two basketball players and an anteroom scenario with multiple persons taking seats and moving on, respectively. The main difficulty lies in the fact that the performers repeatedly enter and leave their scenes, an effect ConCor proves able to cope with. It should be noted that the *Basketball* sequences lack distinctive cues that would allow to conclusively determine the (actually mid-frame) ground truth offset from the interlaced footage. Consequently, the observed synchronization error could liberally be interpreted as amounting to half a frame.

The very challenging *Rothman* and *Magician* datasets show an outdoor street performance and an indoor magic show, respectively. The selected videos are affected by heavy rocking motion and camera pans. In the *Magician* set, the cameras are even displaced by several meters during the recording. In the *Rothman* set, crowds of spectators constitute a rather dynamic background, with motion intensities comparable to those of the street performer himself. A particularly interesting effect in one of the *Magician* videos is a blackout during which the screen remains black for several hundred frames. It is also noteworthy that there is a difference in image format (portrait vs. landscape), a factor to which our synchronization approach is inherently invariant. ConCor sucessfully excludes the blackout and, altogether for both the *Rothman* and *Magician* datasets, retrieves the sequences' offset (up to one frame).

Finally, the *Train* dataset points out the limits of the presented bitrate based approach. One problem, as mentioned earlier, is the very limited number of available frames ($L_b = 146$). The main challenge, however, is the indistinctive, linear motion described by the depicted toy train, which is also superimposed by relatively strong camera shaking. The only cues useful for synchronization (both for our algorithm and a human observer) are the train's alternately flashing head lights and two hardly discernible collisions with other toys in the scene towards the end of the recordings. Without violating the lower bound on $M$, stipulated in (2.9), no more than two segments would be available. But as it turns out, only the last third of the shorter signal is actually suitable for our bitrate based approach. With $M = 45$, three segments are obtained, but each of the corresponding NPCCFs favors a distinct set of potential offsets. Given the dissent among the three NPPCFs, it would be preposterous to speak of a "consensus"-based decision in this case. Which segment eventually becomes classified as *the* inlier is basically random: in one out of three cases this choice is made in favor of the actually faithful third segment. This needs to be borne in mind regarding the rather accurate synchronization result reported here.

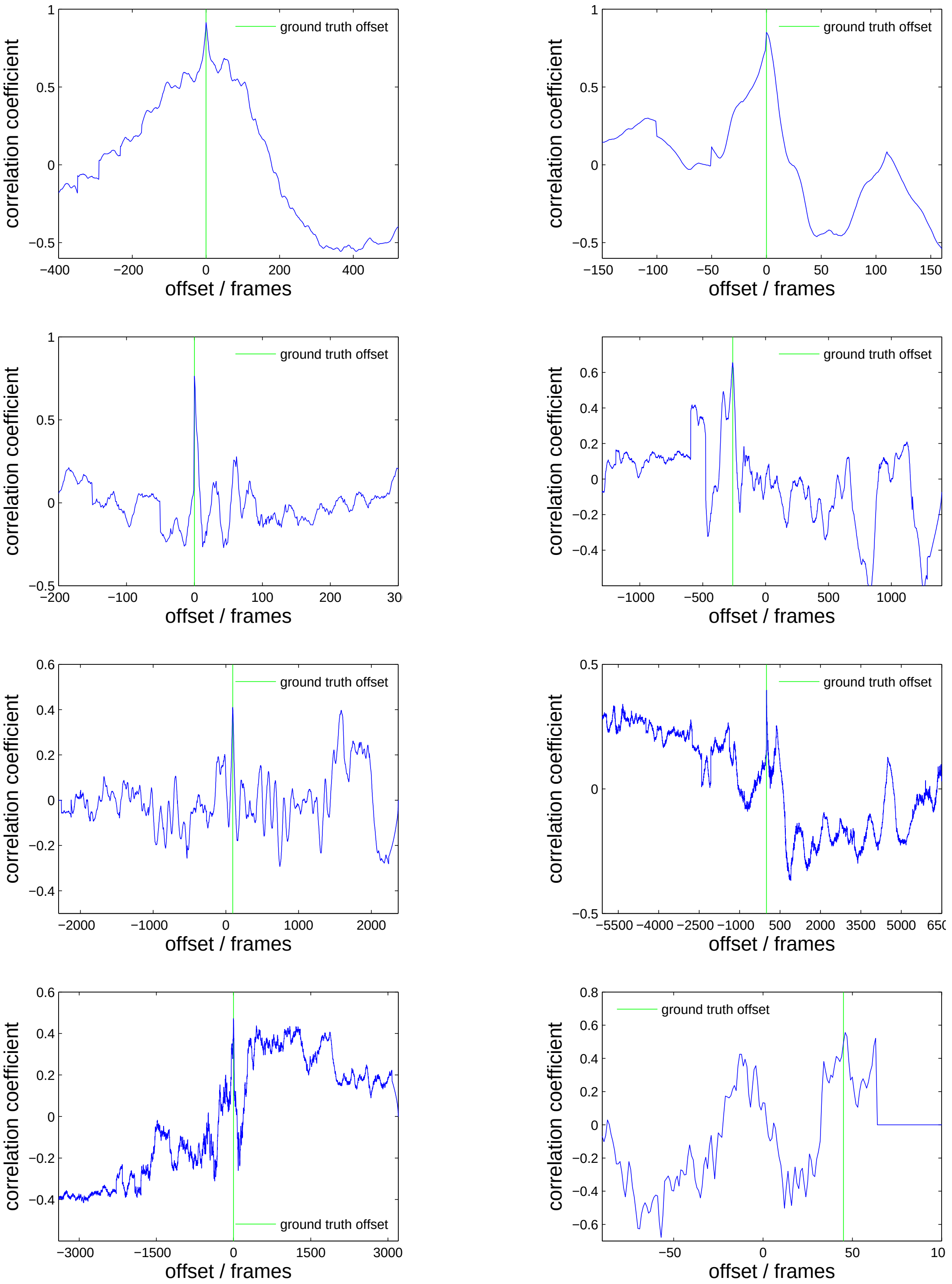


***Figure 2.30:*** *Synchronization outcome for the external datasets (from left to right, and top to bottom):* Dog, Martial Arts, HumanEva, Basketball, Hall, Rothman, Magician *and* Train.

| Video Set | Video Properties | ConCor Parameters | Ground Truth | Sync. Result |
|---|---|---|---|---|
| *CapoEHA*<br>– scene 4<br>– views 1 & 4 | • viewpoint difference: $\sim 90°$<br>• stationary / hand-held cameras<br>• $L_a = L_b = 934$<br>• highly dynamic scene, frequent self-occlusions<br>• MPEG-2, 25 fps, 720 × 576 pixels | $M = 62$<br>$s = 4$ | 5 | 5 |
| *Taiji*<br>– views 2 & 3 | • viewpoint difference: $> 90°$<br>• hand-held cameras<br>• $L_a = 1184$, $L_b = 1177$<br>• smooth, subtle motion<br>• WMV-9, 25 fps, 720 × 576 / 640 × 480 pixels | $M = 65$<br>$s = 5$ | $-18$ | $-18$ |
| *Rope Skipping* | • viewpoint difference: $< 90°$<br>• stationary cameras<br>• $L_a = 4836$, $L_b = 4584$<br>• periodic motion, mirror in scene<br>• MPEG-2, 25 fps, 720 × 576 pixels | $M = 208$<br>$s = 5$ | 122 | 122 |
| *Soccer*<br>– views 1 & 4 | • viewpoint difference: $\sim 180°$<br>• stationary / hand-held cameras<br>• $L_a = 10068$, $L_b = 4401$<br>• dynamic background, subjects leave visual field<br>• MPEG-2, 25 fps, 720 × 576 pixels | $M = 153$<br>$s = 4$ | 4 | 4 |
| *Dog*<br>– scene *Walking*<br>– views 0 & 1<br>[Jan09] | • viewpoint difference: $\sim 20°$<br>• stationary cameras<br>• $L_a = L_b = 581$<br>• PNG image sequence, 25 fps, 1624 × 1080 pixels | $M = 58$<br>$s = 2$ | 0 | 0 |
| *Martial Arts*<br>– scene *Kick One*<br>– views 0 & 6<br>[Jan09] | • viewpoint difference: $\sim 90°$ (worm's eye/top view)<br>• stationary cameras<br>• $L_a = L_b = 211$<br>• PNG image sequence, 25 fps, 1624 × 1080 pixels | $M = 50^*$<br>$s = 1$ | 0 | 0 |
| *HumanEva*<br>– scene *S1-Box1*<br>– views C2 & C3<br>[SBB10] | • viewpoint difference: $\sim 180°$<br>• stationary cameras<br>• $L_a = L_b = 360$<br>• MPEG-4, 60 fps, 640 × 480 pixels | $M = 50^*$<br>$s = 1$ | 0 | 0 |
| *Basketball*<br>[CSI06] | • viewpoint difference: $\sim 180°$<br>• stationary cameras<br>• $L_a = 1881$, $L_b = 1798$<br>• Indeo v5, 25 fps, 720 × 576 pixels | $M = 119$<br>$s = 4$ | $-262$ | $-263$ |
| *Hall*<br>[CSI06] | • viewpoint difference: $< 180°$<br>• stationary cameras<br>• $L_a = 2638$, $L_b = 2533$<br>• Indeo v5, 25 fps, 720 × 576 pixels | $M = 133$<br>$s = 5$ | 94 | 94 |
| *Rothman*<br>– views 1 & 2<br>[BBPP10] | • viewpoint difference: $\sim 90°$<br>• hand-held cameras<br>• $L_a = L_b = 6899$<br>• Lagarith, 25 fps, 960 × 544 / 544 × 960 pixels | $M = 344$<br>$s = 4$ | 0 | 1 |
| *Magician*<br>– views 2 & 3<br>[BBPP10] | • viewpoint difference: $\sim 45°$ (varying)<br>• hand-held cameras<br>• $L_a = L_b = 3800$<br>• Lagarith, 25 fps, 960 × 544 / 544 × 960 pixels | $M = 120$<br>$s = 4$ | 0 | $-1$ |
| *Train*<br>[TVG04] | • viewpoint difference: $\sim 45°$<br>• hand-held cameras<br>• $L_a = 200$, $L_b = 146$<br>• MPEG-4, 25 fps, 720 × 576 pixels | $M = 45^*$<br>$s = 1$ | 45 | 46 |

***Table 2.2:*** *Summary of ConCor results*

## 2.5 Concluding Remarks

With the bitrate-based description of scene changes, in combination with consensus-based cross-correlation, a reliable and versatile video synchronization algorithm has been presented. In comparison with the state-of-the-art, the approach is highly independent of camera and image, as well as scene properties. Owing to the very abstract quantification of relevant scene changes, attributes such as source codec, image resolution, orientation, brightness, etc. have no or very limited influence on the synchronization process. Since the amount of scene changes is quantified, rather than their exact appearance, true viewpoint independence is achieved. Furthermore, the proposed approach mostly compensates for global camera motion, a capability directly inherited from the underlying qualities of H.264/AVC. The only two requirements of the presented synchronization approach are the presence of meaningful scene changes, usually in the form of object motion, and the continuous focus of both cameras on the same objects of interest.

The prototypical application scenario where these requirements are met is multiview video sharing. Uploaded video clips usually undergo re-encoding into a common format anyway. During this process the bitrate data necessary for synchronization can be tapped at no extra cost. It should be noted that this implies a certain trade-off between the desired video quality (obtained with fine quantizers) and synchronization performance (achieved with coarse quantization). In the common scenario where the multiview video sharing platform maintains multiple versions of the same video (destined for different devices, available in several levels of quality, etc.), this is more of a formal issue. Unless a parameter setting suitable for synchronization is already available, a supplementary re-encoding pass carries only marginal additional weight.

The general video synchronization approach is complemented with normalized ConCor which deals with distracting, unrelated object motion, monotonous signal parts, occlusions, sudden camera motion too strong to be fully compensated by H.264/AVC, and other temporary, disturbing effects.

In contrast to other video synchronization methods, the approach as presented in this chapter does not yield sub-frame accuracy. With little extra effort, this can be achieved by interpolation of the bitrate sequences at the desired resolution. It remains to be investigated how the performance of this straightforward approach compares to the more sophisticated, and by far more complex frame rate upconversion of the initial video data.

# 3 Spatial Analysis: Scale-invariant Feature Extraction

Complementary to the investigation of temporal coherences presented in the previous chapter, the analysis of spatial relationships between videos is an equally important subject of study. In applications, such as cooperative video, the spatial properties of interest are the 3-dimensional poses of the cameras with respect to each other and with regard to the scene. *Spatial analysis* can also refer to the alignment of structures within the 2-dimensional frames of one or multiple cameras, for instance for the purpose of object recognition or tracking. In either case, a very common approach to obtain these kinds of information is to identify characteristic image features and relate them between different frames or views.

In this chapter, the focus is on *scale-invariant image features*, a particular class of features highly relevant in state-of-the-art applications. The contributions of this thesis are a novel system of visual markers adapted to scale-invariant feature detection algorithms, further a framework to assess the location accuracy of detected features, and last but not least novel, highly efficient feature detectors, extending the scale-invariant paradigm to affine-invariance.

## 3.1 Introduction to Scale-Invariant Feature Detection

This section provides a very general introduction to scale-invariant image features, so as to cover basically all available algorithms of this category. In particular, the *Scale-Invariant Feature Transform* (SIFT) [Low04] and *Speeded-Up Robust Features* (SURF) [BTVG06], which will be subject of the subsequent sections, obey the principles outlined here.

The motivation for scale-invariance is the fact that image features representing the same physical structure in the real world are unlikely to appear at the exact same size in different images. An example for this is given in Figure 3.1. Depending on the distance of the object to the camera and the particular camera properties, such as focal length or level of zoom, the same structures appear at different size in an image.

In order to compensate for these scale differences an image is expanded into a *scale space* representation prior to feature extraction. In this way, all possible acquisition scales are simulated, which eventually leads to the desired scale-invariance. This effectively introduces a third dimension, namely the scale $\sigma$. In the example of Figure 3.1, in the left image, delicate twigs would be detectable on a fine-scale pyramid layer with low $\sigma$. In the close-up on the right, the same twigs, that now appear about the width of stronger branches in the original resolution, would be dominant higher up in the image pyramid, *i.e.*, at scales with higher values of $\sigma$.

Figure 3.2 shows a typical image pyramid used in scale space representations, organized in octaves. The original image is located at the bottom, towards the top increasingly low-

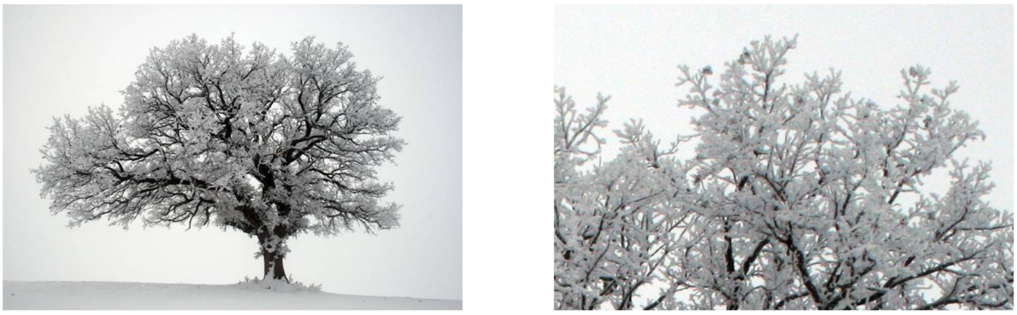


***Figure 3.1:*** *Branches vs. twigs: Two photographs of the same tree. Bold structures in one image appear at a comparable size as the finer details in the other.*

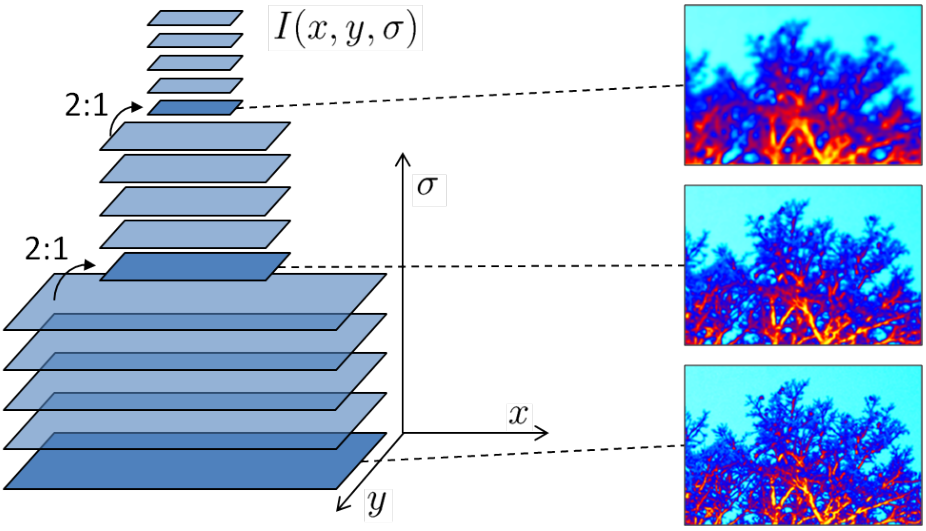


***Figure 3.2:*** *An octave-divided image scale space representation $I(x, y, \sigma)$, with three octaves and five scale intervals per octave. Example pyramid layers are shown (in pseudocolor and resampled for comparison), based on the right image from Fig. 3.1.*

pass filtered versions, referred to as *scales*, are stored. In general, a new octave is started once the image resolution has halved, by subsampling the previous scale level by a factor two. The choice of the scale range to examine, the exact numbers of octaves and scales per octave, the scale sampling strategy (uniform versus non-uniform), as well as the involved filters are parameters of specific algorithms.

Again from a very general point of view, an algorithm-specific operator is then applied to identify image features in three-dimensional scale space. In a final step, a characteristic *descriptor* is computed for every feature point which can be used to reliably compare features with each other. To this end, a unique orientation is usually assigned to the feature in order to achieve additional rotation-invariance.

It's the goal to make the feature detection process as stable and repeatable as possible, especially under external influences such as noise or general perspective transformations. Both Scale-Invariant Feature Transform (SIFT) and Speeded-Up Robust Features (SURF) have proven to perform very well in this respect. The following sections will delve more deeply into the specifics of both the SIFT and SURF algorithms and point out their differences. The focus will especially be on their two main functional components, *detection operator* and *descriptor construction* which will be of importance for the rest of this chapter.

### 3.1.1 Scale-invariant Feature Transform (SIFT)

The Scale-Invariant Feature Transform, or SIFT, has been proposed by David Lowe in 2004 and is still one of the most commonly used scale-invariant image features. Especially the highly distinctive and rotation and scale invariant SIFT descriptor is still considered as a benchmark in feature description. In the following, an overview of the SIFT algorithm is given, with particular focus on the specifics exploited in later sections of this thesis.

#### Feature Detection

It has been shown that gradually filtering with a Gaussian kernel is particularly suited to build the scale space representation of natural images [Lin94]. Furthermore, the scale-normalized Laplacian operator has proven very performant in detecting interest points within scale space [Lin98, Mik02]. Due to the linearity of derivation, applying the Laplacian operator in Gaussian scale space is equivalent to filtering the original image with a *Laplacian of Gaussian* (LoG) filter:

$$\Delta(G^\sigma * I)(x, y) = (\underbrace{\Delta G^\sigma}_{\text{LoG}} * I)(x, y), \tag{3.1}$$

$$\text{where} \quad G^\sigma(x, y) = \frac{1}{2\pi\sigma^2}\, e^{\frac{-(x^2+y^2)}{2\sigma^2}} \tag{3.2}$$

Here, $I$ is the original image, and $G^\sigma$ a bivariate Gaussian with standard deviation $\sigma$ representing scale. SIFT uses logarithmically sampled scale levels, *i.e.*, neighboring scales differ by a constant factor: $\sigma_{n+1} = k\,\sigma_n = k^{n+1}\,\sigma_0$. With $N$ intervals per octave, it follows that $k = 2^{1/N}$, and the $n$-th scale then explicitly takes the form

$$\sigma_n = 2^{n/N}\sigma_0 = 2^{o+i/N}\sigma_0, \tag{3.3}$$

where $o$ is the octave in which the scale is located as the $i$-th layer. This logarithmic scale sampling scheme is shown in Figure 3.6. For computational reasons, image sizes are halved from one octave to the next, implementing the general image pyramid structure presented previously in Figure 3.2.

Lowe shows in [Low04] that the scale-normalized Laplacian of Gaussian (LoG) ($\sigma^2\Delta G^\sigma$) can be approximated by a *Difference of Gaussians* (DoG) function which lends itself to a computationally convenient implementation.

$$\sigma^2\Delta G^\sigma = \sigma \frac{\partial G^\sigma}{\partial \sigma}\bigg|_{\sigma=\sigma_n} \approx \sigma_n\, \frac{G^{\sigma_{n+1}} - G^{\sigma_n}}{\sigma_{n+1} - \sigma_n} = \frac{1}{k-1}\underbrace{(G^{\sigma_{n+1}} - G^{\sigma_n})}_{\text{DoG}} \tag{3.4}$$

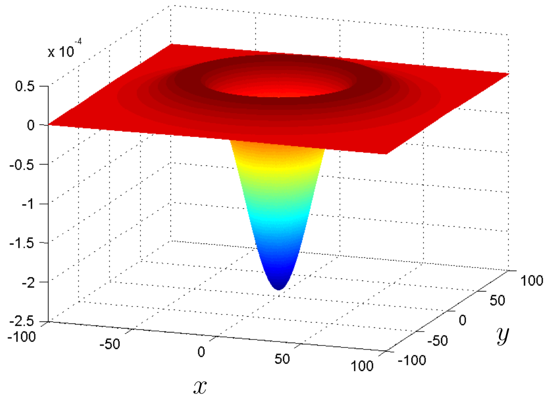


***Figure 3.3:*** *Difference of Gaussians (DoG) filter $H^n_{\text{DoG}}$ used by SIFT at scale $\sigma_n = 20$.*

The first equality in (3.4) is due to the fact that the Gaussian function obeys the heat diffusion equation $\frac{\partial G^\sigma}{\partial \sigma} = \sigma \Delta G^\sigma$.

With Difference of Gaussians (DoG)s as feature detection filters, the SIFT detector response $D_{\text{SIFT}}$ at pixel position $(x, y)$ obtained on scale level $\sigma_n$ hence reads:

$$D_{\text{SIFT}}(x, y, \sigma_n) = \left(H^n_{\text{DoG}} * I\right)(x, y), \tag{3.5}$$

$$\text{where} \quad H^n_{\text{DoG}} = G^{\sigma_{n+1}} - G^{\sigma_n} \tag{3.6}$$

Figure 3.3 shows the typical "flipped mexican hat" shape of $H^n_{\text{DoG}}$. Features are found as local extrema of the three-dimensional filter output (3.5). Every feature $(x, y, \sigma)$ is hence the combination of location and scale maximizing $D_{\text{SIFT}}(x, y, \sigma)$, where, quadratic interpolation is used to refine the exact feature position in scale space.

### Feature Description

After detection, every feature is assigned a "fingerprint" computed from the gradient distribution around its pixel location. The more distinct these *descriptors* are, the better the results obtained in a subsequent feature matching step will be. In the framework presented in Section 3.2, descriptors will also be used to identify markers in an image.

The SIFT descriptor is constructed from a square neighborhood of side length $12\sigma$ pixels, where $\sigma$ is the scale at which the feature was detected. This neighborhood is aligned with the dominant local gradient direction of the feature and divided into 16 subblocks. Figure 3.4 shows this $4 \times 4$ subdivision. For each of the 16 subregions a 8-bin histogram of weighted gradients is compiled, and finally the descriptor is constructed by sorting the bins' contents from all the subblocks into a vector of length $8 \times 4 \times 4 = 128$. Finally, the descriptor vector is normalized to unit length which allows for affine pixel intensity changes. Due to the scale and rotation adaptive creation process, SIFT descriptors are mostly invariant to moderate geometric transformations.

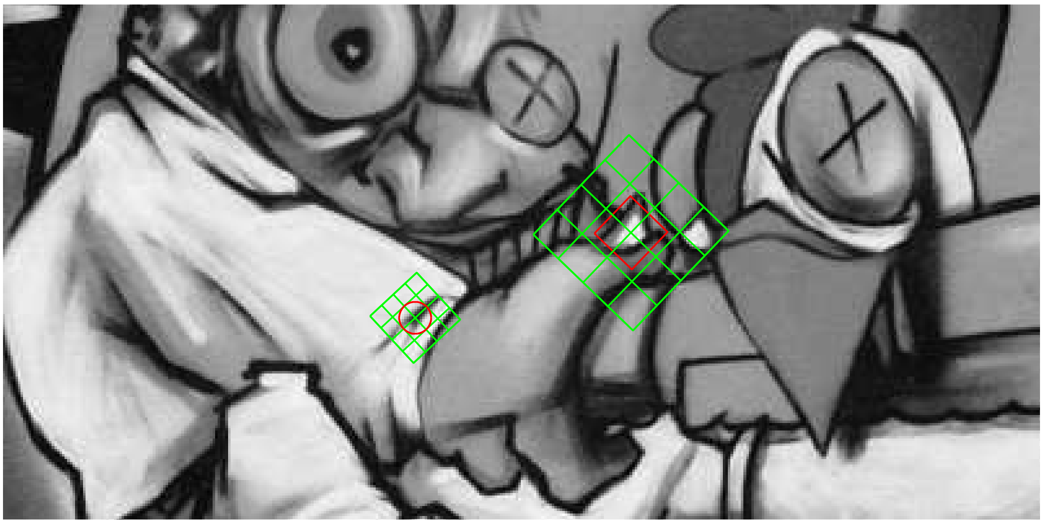


*Figure 3.4:* *A SIFT feature (left) and a SURF feature (right) detected in an image. The red circle (plotted with radius $3\sigma$) and square represent the feature extent, respectively. The green squares oriented along the dominant local image gradient mark the descriptor support regions,* i.e., *the pixel regions the descriptors are computed from.*

### 3.1.2 Speeded-up Robust Features (SURF)

SURF builds on the concepts of SIFT but introduces more radical approximations in order to speed up the detection process. With the use of integral images, the complexity of SURF is greatly reduced in comparison to SIFT. Still, SURF is reported to achieve detection performance even superior to its predecessor [BTVG06].

#### Feature Detection

Instead of the Laplacian operator, SURF uses the determinant of the Hessian matrix (DoH) to detect features in scale space. Both operators are closely related – the Laplacian is in fact the trace of the Hessian matrix – but the DoH has the advantage of being more selective with respect to weak features along edges in the image, especially when the approximations proposed by SURF are used [Low04, BTVG06].

Figure 3.5 shows the box filters $\tilde{G}^{\sigma}_{xx}$, $\tilde{G}^{\sigma}_{yy}$ and $\tilde{G}^{\sigma}_{xy}$ approximating the second order derivatives of the Gaussian $G^{\sigma}$. Scales are again discretized and depend on the size of the used box filters. By definition, a filter kernel with side length $s$ pixels corresponds to the scale $\sigma = \frac{1.2}{9} s$.

Filter sizes are chosen appropriately such as to cover a reasonable range of scales [BTVG06]. In particular, SURF introduces a simplified, slightly coarser scale sampling strategy, as illustrated in Figure 3.6. Octaves are still distributed logarithmically, inside octaves, however, scales are sampled uniformly.

Applying the approximative box filters from Figure 3.5 to an image $I$ yields (approximations for) the entries of the Hessian matrix. The detector output at pixel position $(x, y)$

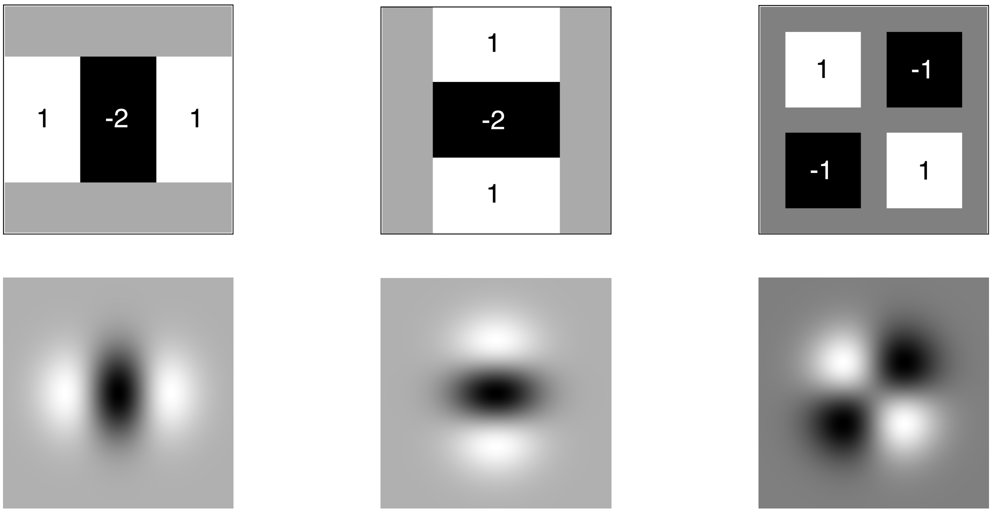


***Figure 3.5:*** *The box filters $\tilde{G}^{\sigma}_{xx}$, $\tilde{G}^{\sigma}_{yy}$ and $\tilde{G}^{\sigma}_{xy}$ used by SURF (top row) approximating the second order Gaussian derivatives $G^{\sigma}_{xx}$, $G^{\sigma}_{yy}$ and $G^{\sigma}_{xy}$ (bottom row).*

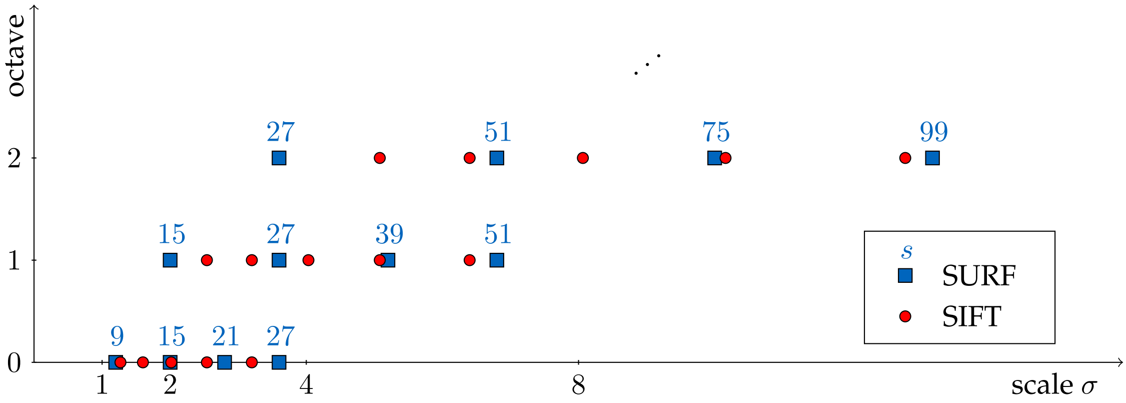


***Figure 3.6:*** *Comparison of scale sampling schemes: SIFT consistently samples the scale $\sigma$ logarithmically. SURF increases its filter size $s$ linearly within each octave, only doubling the filter size increments from one octave to the next.*

on scale $\sigma$ is defined as the determinant of this approximate Hessian matrix, given by the following equations:

$$D_{\text{SURF}}(x, y, \sigma) = \det \begin{bmatrix} H_{11}(x, y) & H_{12}(x, y) \\ H_{21}(x, y) & H_{22}(x, y) \end{bmatrix} \tag{3.7a}$$

$$\text{where} \quad H_{11} = \tilde{G}^{\sigma}_{xx} * I, \quad H_{22} = \tilde{G}^{\sigma}_{yy} * I, \quad H_{12} = H_{21} = \tilde{G}^{\sigma}_{xy} * I \tag{3.7b}$$

Since SURF's computational complexity is independent of the box filter size due to the use of integral images, subsampling the original image is not required. Nevertheless, the stride length of the filters is doubled from one octave to the next so that, ultimately, the SURF detector output $D_{\text{SURF}}$ has a pyramid layout congruent to that of SIFT (*cf.* Figure 3.2).

It is also important to note that, as opposed to SIFT's DoG filters, the SURF detection operator is nonlinear in the input image $I$.

The SURF algorithm searches the detector response (3.7a) for local extrema (maxima as well as minima). In keeping with SIFT, quadratic interpolation is eventually used to refine a feature to inter-scale and sub-pixel accuracy.

### Feature Description

SURF's descriptor layout is very similar to that of SIFT, also based on a square region around the feature point which is aligned with the dominant gradient. It is also divided into 16 subblocks (see Fig. 3.4), but spans a slightly wider neighborhood of $20\sigma \times 20\sigma$ pixels.

The oriented descriptor support region is resampled at resolution $20 \times 20$ from which the discrete gradient $(d_x, d_y)$ is computed. Instead of histogram values, SURF uses the sums and absolute sums of each subblock's gradient components. Every subblock hence contributes exactly four descriptor entries, namely $\sum d_x$, $\sum |d_x|$, $\sum d_y$, $\sum |d_y|$. In total, the SURF descriptor hence comprises $64$ entries. The SURF descriptor is normalized to unit length ensuring invariance to affine intensity variations.

## 3.2 Visual Markers Adapted to Scale-invariant Feature Detectors

Visual markers play an important role in augmented reality applications where reliable correspondences between the real world and 2D projections thereof are required. Subsequent tasks such as pose estimation or object recognition are hence vastly simplified. The simplest designs consist of passive markers containing distinctive planar patterns. Due to their versatility and low installation costs, these so called *fiducial markers* are often chosen over more complex tracking systems employing infra-red or active markers. Example applications from other fields include visual servoing in robotic surgery and monitoring in production logistics.

In many applications, fiducial markers are supplemented with image features originating in salient points in the scene itself. Maybe the most widely used algorithms used for feature detection today are the previously discussed SIFT and SURF algorithms. At the other end of the scale, in merely feature-based applications, it might even so be desirable to place reliably detectable reference points on unstructured surfaces, which would usually find themselves featureless.

In this section, a light-weight marker framework will be proposed that conflates the 2-stage strategy consisting of marker detection and feature point extraction into a more efficient 1-step approach. The aim is on the vast number of computer vision applications which are based on feature points, enhancing them with markers that fully integrate into the existing detection process. The developed markers are provably optimal in that they trigger maximum response in the SIFT and SURF detectors.

This part of the thesis is based on results presented in [SZG$^+$09].

### 3.2.1 Related Work on Fiducial Markers

Fiducial markers are widely used in a variety of different fields. Depending on the particular application, different designs have been proposed. An overview and comparison of different marker systems can be found in [ZFN02]. Fiducial marker systems typically consist of a set of distinguishable labels that are placed in the scene and can be detected and decoded by an associated algorithm. According to the design of the markers, the detection consists of several nontrivial steps such as edge detection, linking and line fitting. *Decoding* refers to the identification of a detected marker. Early systems used correlation-based approaches to match the appearance of a marker against a database of templates [KB99, KBP$^+$00]. State-of-the-art marker systems employ binary error correcting codes to allow the unique and robust identification of thousands of different markers [Fia05]. Recently, there have been efforts towards lowering the computational complexity for mobile real-time applications [WS07, WLS08]. Compared to these highly specialized systems, our proposed marker scheme offers very elementary, yet highly valuable functionality: making feature points reliably detectable, at no additional expenses.

Desirable properties of a conventional marker system are low *false positive* and *false negative* rates, as well as low *inter-marker confusion* rates if the system comprises more than one distinguishable marker. That is, the system should neither report a detected marker when there is none, nor miss or mistake an actually present marker.

As mentioned before, state-of-the-art marker systems like, *e.g.*, *ARTag* [Fia05] use error correction mechanisms which allow them to reach unrivaled performance in terms of false positive and inter-marker confusion rates. As the light-weight marker system proposed in this chapter only provides one marker, respectively two markers in the case of SURF, inter-marker confusion is not an issue. The main goal is to have a highly detectable low-cost marker, not necessarily a uniquely detectable one.

The key virtue of the system presented here is its remarkably low false negative rate. As demonstrated in Section 3.2.4, it can be virtually guaranteed that the SIFT and SURF markers get detected, even under dramatically varying imaging conditions. This makes them highly useful in applications where SIFT or SURF points need to be found at a desired location.

### 3.2.2 Detector Response Maximization

In this section, optimal input images for the SIFT and SURF detectors will be derived, which then lend themselves as ideal markers for the respective detector. In this context, *optimal* refers to giving rise to the highest possible detector output.

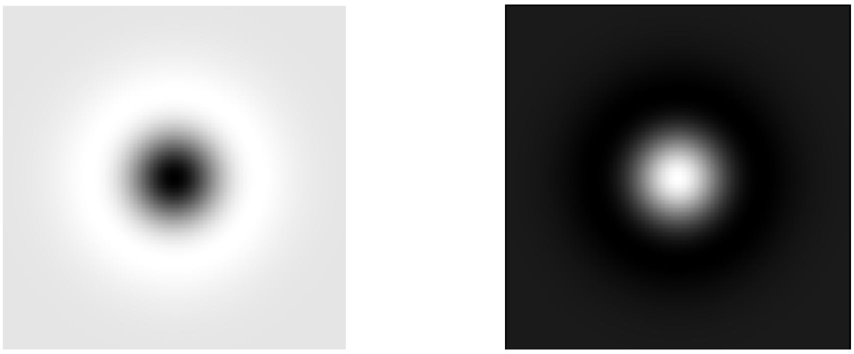


***Figure 3.7:*** *Maximum detector response markers for SIFT.*

### The SIFT Marker

In case of SIFT, the task of determining the optimal marker is that of maximizing the output of a linear filter. As discussed in Section 3.1.1, this filter has a Difference of Gaussians function as impulse response, as given in Equation (3.6) and Figure 3.3.

In order to find the pixel pattern which maximizes the detector response (3.5) the concept of *matched filters* is borrowed from signal processing. A matched filter is commonly used to recover a signal of known shape that has been corrupted on its way over a noisy channel [Tur60]. Example applications include the detection of reflected radar impulses or the decoding of base band signals. The determining property of the matched filter is that it maximizes the signal-to-noise ratio at the receiver. It can be shown that this is achieved when filter impulse response and signal are mirrored versions of each other.

In the present case, the inverse problem is to be solved. For the given DoG filter, an energy-limited signal is to be determined which, superimposed with image noise, will yield maximum response at the filter output. Consequently, following the matched filter paradigm, the SIFT marker is chosen to be a (mirrored[1]) DoG. The shape and appearance of the resulting marker is shown in Figures 3.3 and 3.7. Note that a sign-reversed DoG is a valid "matched signal", thus marker, as well.

Figure 3.8 shows the associated descriptor which can subsequently be used to identify the marker. Its characteristic shape due to the fact that the main lobes of the DoG fall inside the inner four subblocks is beneficial for descriptor matching. These descriptor parts will be referred to as the *SIFT marker signature*.

### The SURF marker

As discussed in Section 3.1.2, SURF uses a nonlinear detection operator. Hence, the matched filter approach that was taken for SIFT is no longer applicable in this case. Nevertheless, the input image that will maximize the detector response can be obtained by solving a properly defined optimization problem.

[1] Mirroring obviously has no effect on $H^n_{\mathrm{DoG}}$ due to its symmetry.

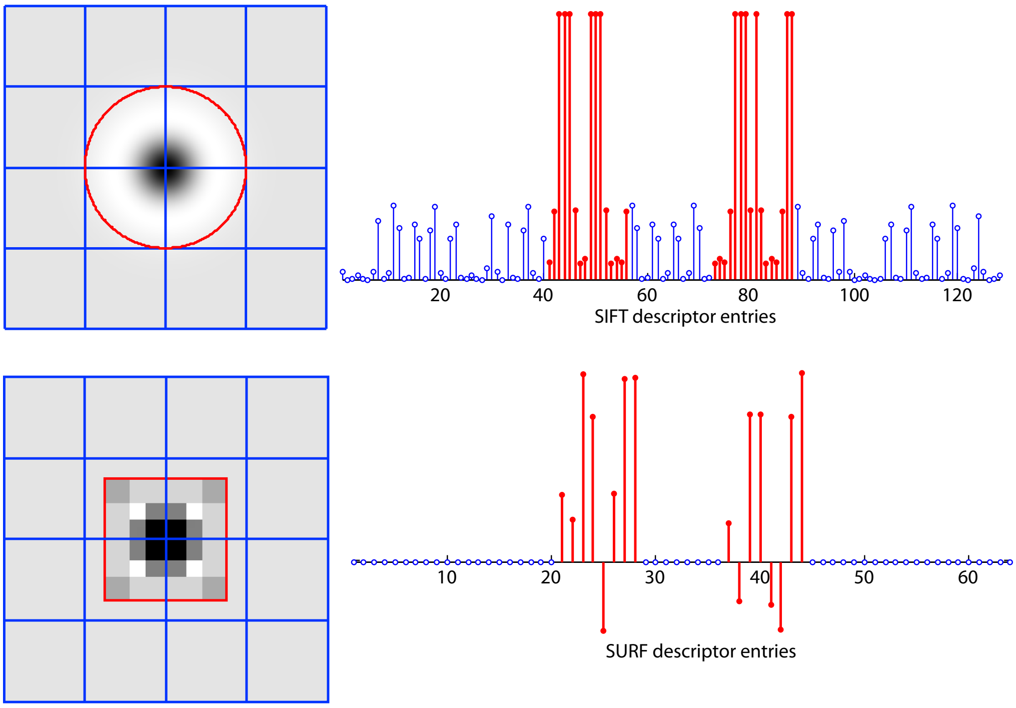


**Figure 3.8:** *Left: The support region used to compute the descriptors in size comparison with the SIFT and SURF markers* (cf. *Fig. 3.4). Right: The descriptor contributions of the respective markers. The highlighted entries correspond to the inner four subblocks and constitute the markers'* signatures.

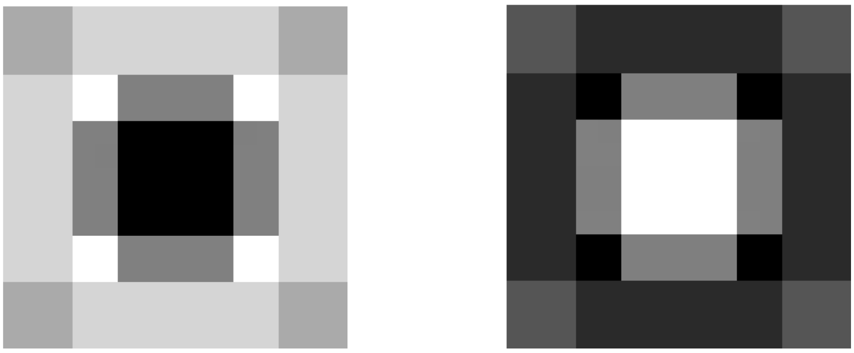


**Figure 3.9:** *Maximum detector response markers for SURF.*

Without loss of generality, the SURF marker of size 9×9 pixels will be derived here, which involves 81 variables in our optimization. The problem can of course also be solved for the other kernel sizes used by SURF, *i.e.*, 15, 21, 27, and so on. However, the resulting markers will just be upscaled versions of the $9 \times 9$ marker. Hence, only the elementary case $s = 9$, corresponding to $\sigma = 1.2$, will be treated in the following. To enforce a maximum of (3.7a) at position $(x_0, y_0)$, the 81 pixels in the square region centered on $(x_0, y_0)$ need to be taken into consideration. Their values must be adjusted such as to maximize $D_{\text{SURF}}(x_0, y_0, 1.2)$.

First, the 81 variables are rearranged into a vector $\mathbf{x}$, and accordingly the entries of the box filters into vectors $\mathbf{g}_{xx}$, $\mathbf{g}_{xy}$ and $\mathbf{g}_{yy}$. So, the filter output (3.7a) can be rewritten as follows.

$$\begin{aligned} D_{\text{SURF}}(x_0, y_0, 1.2) &= \begin{vmatrix} \mathbf{g}_{xx}^\top \mathbf{x} & \mathbf{g}_{xy}^\top \mathbf{x} \\ \mathbf{g}_{xy}^\top \mathbf{x} & \mathbf{g}_{yy}^\top \mathbf{x} \end{vmatrix} \\ &= \mathbf{x}^\top \underbrace{\left(\mathbf{g}_{xx}\mathbf{g}_{yy}^\top - \mathbf{g}_{xy}\mathbf{g}_{xy}^\top\right)}_{\mathcal{G}} \mathbf{x} \\ &= \tfrac{1}{2}\mathbf{x}^\top(\mathcal{G}^\top + \mathcal{G})\,\mathbf{x} \end{aligned}$$

With $\mathcal{A} = \mathcal{G}^\top + \mathcal{G}$, this leads to the following quadratic optimization problem, with the natural additional requirement that the pixel values be limited.

$$\max_{\mathbf{x}} \mathbf{x}^\top \mathcal{A}\,\mathbf{x}, \quad \text{s.t. } \|\mathbf{x}\| \leq 1 \tag{3.8}$$

It can be shown that rank($\mathcal{A}$) = 3, so the eigenvalue decomposition $\mathcal{A} = \mathcal{U}\Lambda\mathcal{U}^\top$ together with the substitution $\mathbf{y} = \mathcal{U}^\top\mathbf{x}$ yields the equivalent problem:

$$\begin{aligned} \max_{\mathbf{y}} \mathbf{y}^\top \Lambda\, \mathbf{y} = \max_{\mathbf{y}} &(\lambda_1 y_1^2 + \lambda_2 y_2^2 + \lambda_3 y_3^2), \\ &\text{s.t. } \|\mathbf{y}\| \leq 1 \end{aligned}$$

There are two solutions $\mathbf{y}_{\text{opt}} = [\pm 1, 0, \ldots, 0]^\top$, and back-substitution reveals that $\mathbf{x}_{\text{opt}} = \mathcal{U}\mathbf{y}_{opt}$ is the eigenvector corresponding to the largest eigenvalue of $\mathcal{A}$ (and its inverse respectively). Rearranging $\mathbf{x}_{\text{opt}}$ into a 9×9 image and adjusting its values to span the whole range of gray values eventually leads to the desired SURF markers, as depicted in Figures 3.10 and 3.9.

Due to their discrete nature, the SURF markers leave an even more characteristic *descriptor signature* than their SIFT counterparts (see Figure 3.8). It is particularly noteworthy that the signatures of the dark and light versions of the SURF marker are distinguishable. However, the entries corresponding to the sum of absolute gradients values are identical for both.

### 3.2.3 Distinguishable Markers

The proposed Maximum Detector Response (MDR) markers are available in two variants only, a light and a dark version. Conventional marker systems have the advantage to

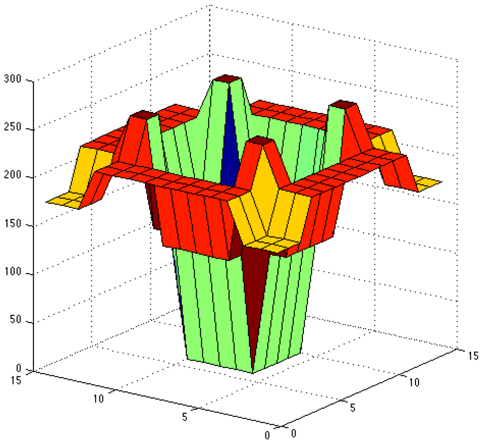


***Figure 3.10:*** *Maximum response SURF marker optimized for a kernel size of* 15 *pixels. Note the qualitative similarity to the reversed Mexican hat of Figure 3.3.*

provide a myriad of markers equipped with distinct IDs. In this section, the possibilities to enhance the SURF MDR marker with unique IDs are briefly discussed. An important requirement is to keep the detection and identification process compatible with the established SURF framework.

Regarding the SURF marker signature in Figure 3.8, it is obvious that many descriptor entries are unused. In particular, the descriptor components attributable to the twelve marginal subblocks of the descriptor support region vanish, amounting to 48 unused descriptor entries. Selectively modifying these subregions allows the marker's descriptor to be *modulated*.

From a theoretic point of view, optimally distinguishable markers with maximum inter-descriptor distance are desired. Given a fixed number $N$ of distinct markers, the goal is thus to uniformly[2] distribute their 64-dimensional descriptor vectors on the hypersphere $\mathbb{S}^{64}$. This task is related to the so called *Thomson problem* known in physics, where $N$ electrons are to be distributed on a sphere in 3-space such that their potential energy is minimized [Tho04]. In 64-dimensional space, and for $N < 64$, the desired solutions are *regular simplexes*, *i.e.*, configurations where all points have equal distance to each other (equilateral triangle, tetrahedron, and so on). In case where $N$ is a power of 2, the descriptors can be constructed from the Hadamard matrix of the same order [AZL03]. For arbitrary values $N < 64$, the recursive approach illustrated in Figure 3.11 and inspired by [Cox73] is proposed to construct the descriptor vectors. Note that to distribute $N$ descriptors, only $N-1$ of the 64 dimensions need to be considered. Should the desired number of distinguishable marker IDs exceed 63 , an iterative optimization can be used to determine the $N$ uniformly distributed descriptors on $\mathbb{S}^{64}$.

In practice, however, setting the optimal descriptors is not a simple task. The mapping

[2] Here, *uniformly* refers to the configuration where the minimum pairwise distance between descriptors is maximized. This configuration is only defined up to rotations about the origin in 64-space.

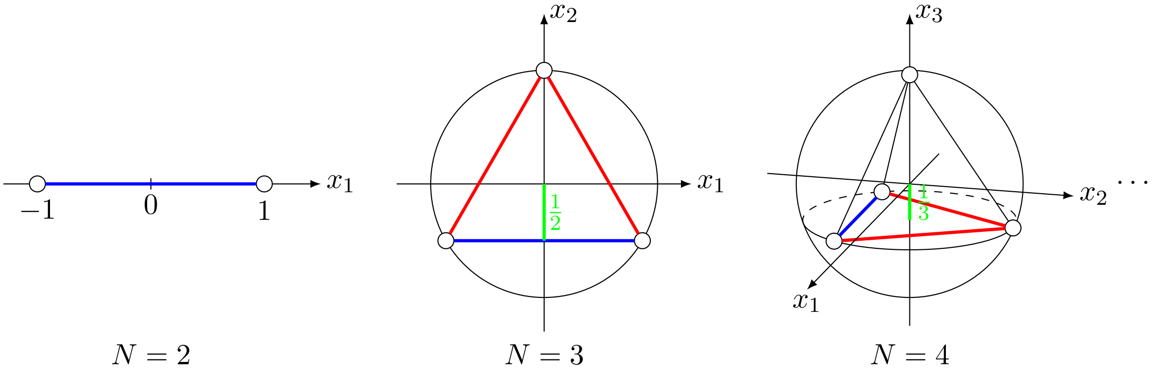


***Figure 3.11:*** *Recursive construction of $N$ optimally distributed descriptors on the unit sphere $\mathbb{S}^{N-1}$: In every step, the previous configuration is shifted along the negative $x_{N-1}$ axis by $d_N = \frac{1}{N-1}$ and shrunk accordingly with factor $\sqrt{1 - d_N^2}$. The $N$-th descriptor is always placed at $x_{N-1} = 1$.*

from the pixel domain to descriptors is highly non-linear and cannot be easily inverted. Furthermore, the descriptor components are interdependent which renders significant parts of $\mathbb{S}^{64}$ inaccessible. Another important hurdle is the orientation assignment which is heavily influenced by any modification of the marker's surrounding.

An investigation of descriptor-based marker IDs compatible with Upright-SURF (a variant without orientation assignment, lacking rotation-invariance) is conducted in [Wie12].

### 3.2.4 Experiments on Detectability

The markers derived in Section 3.2.2 are provably optimal only if viewed under perfect conditions, *i.e.*, frontal to the camera, upright, and at the exact same size as the operator they were optimized for. Experiments on synthetic and real data suggest, however, that the markers are still detectable if imaged at sizes halfway between scales, and under perspective distortions. In this section, the results of these detectability experiments will be presented. For all our experiments, the SIFT implementation by Andrea Vedaldi [Ved] and the OpenCV [BK08] implementaion of SURF were used.

In general, there are two properties related to the performance of the proposed markers that have to be distinguished.

**Detectability:** A marker is *detectable* in an image if it generates a local maximum in scale space, *i.e.*, the detector will report a feature point at the marker position. This is the minimum requirement towards the markers.

**Unique detectability:** If the imaged marker even triggers the global maximum in scale space, or if the combination of high detector response and signature similarity identifies it as a marker, it is *uniquely detectable*. The experiments described in Section 3.2.5 suggest that the markers also have this property.

The goal of the following experiments is to back up the theoretic optimality of the presented marker design. More specifically, it remains to be shown that even under unfavor-

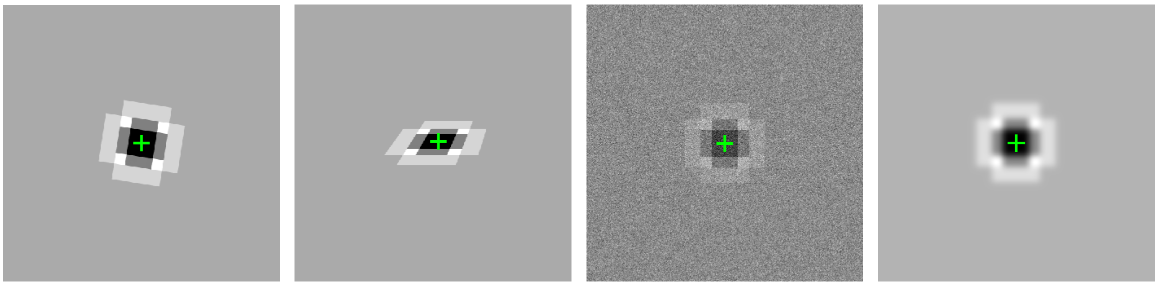


***Figure 3.12:*** *Examples of the distortions applied to the SURF marker in the synthetic experiments: In plane rotation by $10^\circ$, out-of-plane rotation by $40^\circ$, additive white Gaussian noise with standard deviation equal to 50 gray values, Gaussian blur with standard deviation 6 px. The green cross indicates the position at which the marker has been detected.*

able viewing conditions, *i.e.*, other than those assumed during the derivations, the markers still yield over-average detector response. Ideally, the detector response should be invariant to all these effects.

### First Experiments on Synthetic Data

In order to assess the general viability of the proposed SIFT and SURF marker systems, a first set of experiments is run in a fully controllable, synthetic environment. A synthetically generated marker, either SIFT or SURF, is placed centered in front of a uniform background in an image of size $513 \times 513$ pixels. The initial size of the SURF marker is $s = 147$ pixels, and $\sigma = 15$ pixels in the SIFT case. After applying different distortions to the image, the respective feature detector is applied, and its response measured. At the same time, the localization error with respect to ground truth is monitored. If the marker is detected more than three pixels off, it is declared undetected. This is a rather strict rule in comparison to the used marker size. At first, the effects of the following distortions are investigated:

(a) *scaling*,

(b) *in-plane rotation*, and

(c) *out-of plane rotation*.

The visual impact of these distortions in the case of the SURF marker is illustrated in Figure 3.12. Figures 3.13 to 3.15 show their effect on the detector response values. The overall observation is that the detector response stays on a fairly high level in general, *i.e.*, close to the maximum value reachable in the absence of distortion. Within the plotted ranges the 3 pixel threshold was never violated (except for major angles in the out-of-plane rotation scenario). However, there are some unexpected effects which require further investigations.

A first interesting observation is the behavior of the response over the actual marker scale (Fig. 3.13). Especially SURF exhibits two severe response drops at mid-scales, roughly at marker sizes 50 and 100 pixels. Apparently, the spacing between neighboring scale values is adversely coarse in these regions. This effect will be examined more closely in Section 3.2.4. Apart from that, the two curves show the expected behavior. At too small sizes,

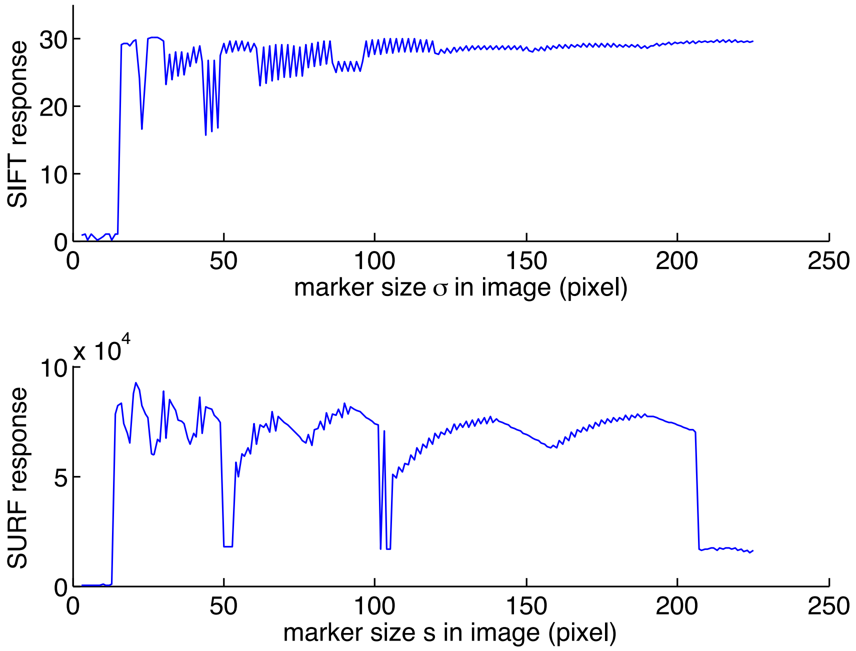


***Figure 3.13:*** *Behavior of detector response under scaling of the marker.*

inferior to the smallest detector operator, the markers do not get detected at all. In the case of SURF, there is also an upper limit to the detectable marker size as, unlike SIFT, SURF uses a predefined number of scale steps. This experiment suggests a minimum size for the SURF marker of 12 pixels, and a maximum size of 210 pixels. In practice, typical sizes are likely not to exceed 50 pixels as markers are preferred to be as unobtrusive as possible.

Regarding the behavior with respect to in-plane rotation, the results are fully convincing (see Figure 3.14). Even though there are minor variations in the detector output, the overall response level remains on a constantly high level. This was to be expected for the rotationally symmetric SIFT marker, because the used resolution is sufficiently high to avoid severe aliasing artifacts. It is remarkable in case of the SURF marker however, for which reduced detectability for rotation angles around 45° would seem inevitable.

For out-of-plane rotations (Fig. 3.15), the markers hit the limits given by the respective feature detectors themselves. It is known that both detectors do not cope well with angles beyond some 40 degrees. While the SIFT marker does well in terms of high detector output, it exceeds the 3 px localization accuracy test for angles greater than 35 degrees. The SURF marker exhibits a significant response decrease and also deviates from ground truth more than 3 pixels for angles superior to 25 degrees.

### In-depth Evaluation of the SURF Marker

In this section, a more detailed experimental evaluation of the SURF marker system is conducted. Again, all the experiments make use of the SURF algorithm provided by the OpenCV library [BK08].

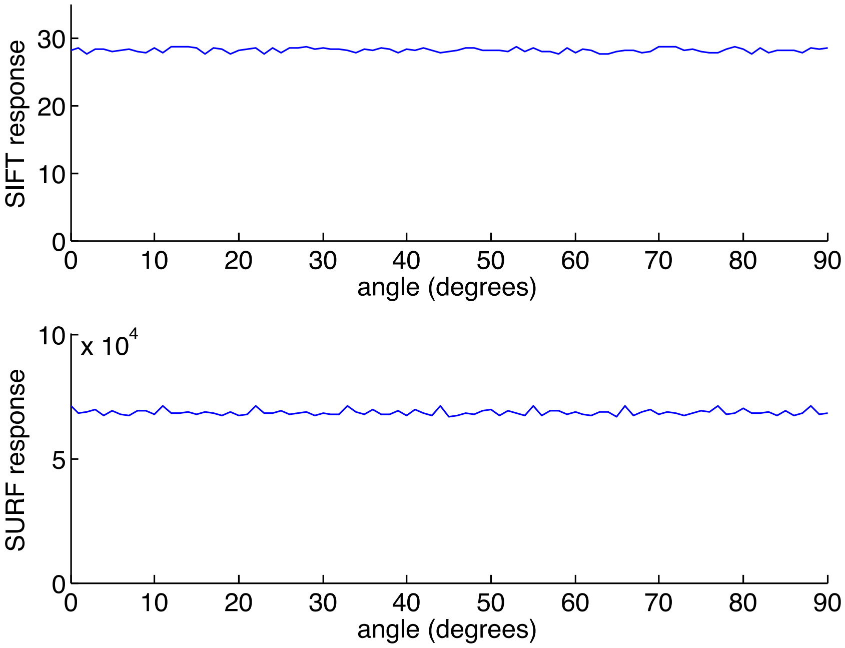


***Figure 3.14:*** *Behavior of detector response under in-plane rotation.*

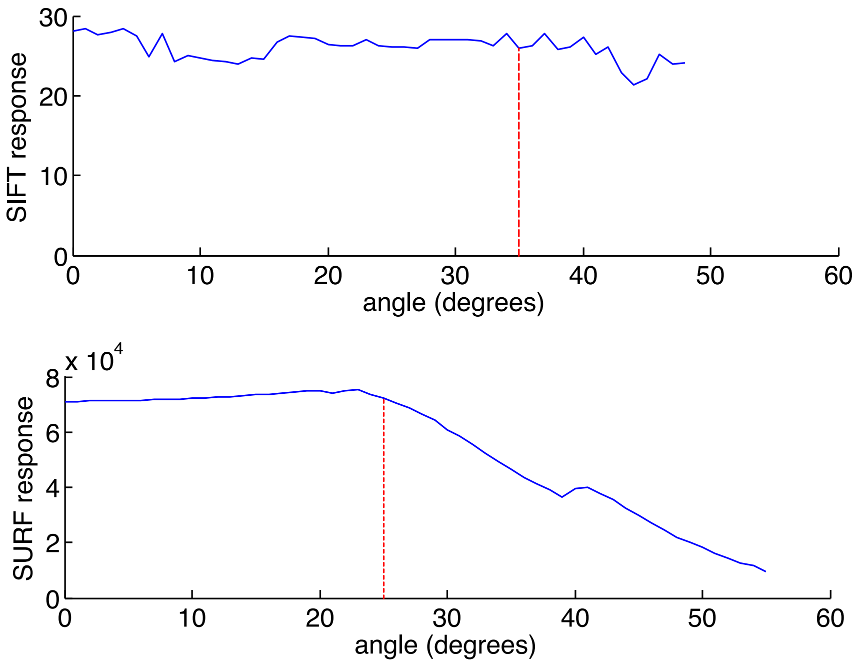


***Figure 3.15:*** *Behavior of detector response under perspective distortion (out-of-plane rotation). The dotted red lines indicate the limit above which the localization error exceeds 3 px.*

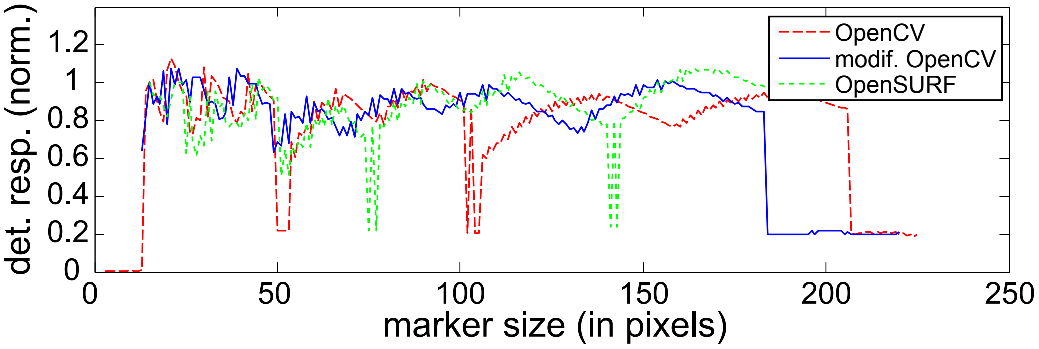


***Figure 3.16:*** *Influence of different scale sampling approaches on the SURF detector response. The values are normalized with the maximum response for the initial marker size.*

The performance analysis from Section 3.2.4 is extended by explicitly considering the evolution of the localization error and the SURF marker signature under distortion. Furthermore, the apparent scale sampling issue observed in Figure 3.13 is addressed, and a more elaborate examination of perspective distortion (out-of-plane rotation) is provided. Additionally, the effects of image noise and blur are investigated.

A quantitative assessment of the marker localization error in *real images* will show that the results obtained on synthetic images also transfer to real-world data.

**Experiments on Synthetic Images** To fathom the scale sampling problem discovered in Section 3.2.4, a marker of initial size 15 pixels is placed in front of a gray background and gradually enlarged by 2 pixels at a time. Figure 3.16 shows the SURF detector response as generated by the implementation provided by OpenCV in dashed red. The same response drops as in Figure 3.13 are observable. The explanation for this effect is the slightly coarser scale sampling scheme used in the OpenCV's v2.0 implementation of SURF which differs from the originally proposed scale sampling described in [BETVG08]. Adjusting the scale sampling accordingly leads to a curve without drops, denoted by *modified OpenCV* in Figure 3.16. For the sake of completeness, the same experiment is conducted using another widely used SURF implementation [Eva], observing similar problems as for the unmodified OpenCV version, yet at different marker sizes. Given that the marker is the prototypical SURF feature, this experiment suggests to give preference to the originally proposed scale sampling scheme as it seems to cover the examined parts of scale space more consistently.

Figure 3.17 gives the direct comparison between the two variants of the OpenCV implementation in terms of the resulting localization errors. The modified version is generally more accurate for the relevant range of marker sizes below 60 pixels and, in particular, does not contain the tremendous error peak at 160 pixels.

In the following, a more comprehensive study into SURF marker detectability under *perspective transformations* is given. In Section 3.2.4, only a one-dimensional out-of-plane rotation was considered. Here, the full range of perspective transformations parametrized by the two-dimensional viewing direction (in azimuth and elevation angles measured from the marker's normal) is tested. For this experiment a medium sized marker (115 px from

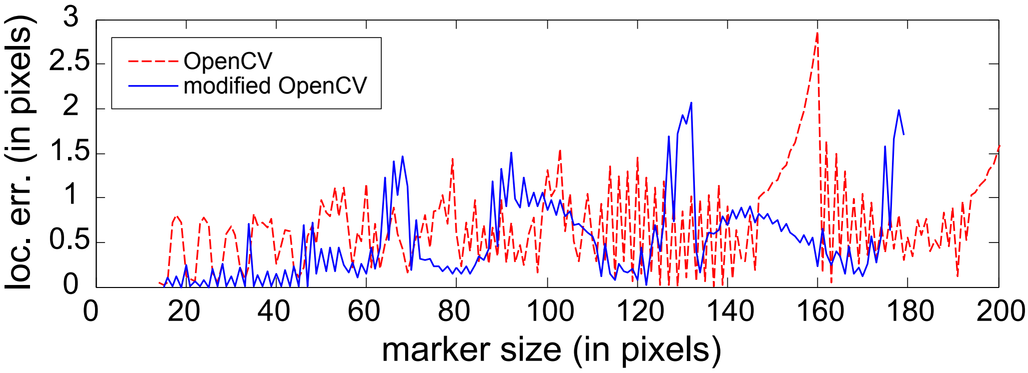


***Figure 3.17:*** *Localization error as a function of marker size for the two rival SURF scale sampling schemes.*

a frontal vantage point) is placed in front of a gray background, and a perspective transformation applied according to the varying relative viewing direction. Figure 3.18 shows the *detector response* relative to its theoretical maximum, the marker's *signature distance* to the undistorted reference marker, and the *localization error* of the marker in the image. Obviously, the marker gets accurately detected (*i.e.*, with low localization error) for a wide range of orientations, namely azimuth and elevation of up to 60°. In order to uniquely identify a marker, outstanding detector response and low signature distance are required (see [SZG$^+$09] for more details). Depending on the surrounding natural features, this can be expected for azimuth and elevation angles up to 40° where the detector response is still above 80% of its maximum. The discontinuity in the detector response beyond this area is caused by the fact that the marker appears clinched to such an extent that it suddenly gets detected on the next lower scale.

In the next experiment, the performance of the SURF marker is studied under parasitic influences, namely *Additive White Gaussian Noise* (AWGN) and *isotropic Gaussian blur*. In both cases, the detector response degrades gracefully with increasing disturbance, as shown in Figures 3.19 and 3.20. For practically relevant noise and blur levels, unique marker detection remains unimpaired, especially since the strength of natural features typically degrades likewise under these effects. The localization error grows approximately linearly with the noise intensity and never exceeds half a pixel for ranges to be expected in real applications. Interestingly, excessively increased blurring causes the localization error to recede. However, these effects take place in the range of insignificantly small errors and can most likely be neglected.

**Experiments on Real Images** In addition to the synthetic experiments presented in the previous paragraphs, the marker performance is now evaluated based on real images. Figure 3.21 shows the experimental setup with the SURF marker mounted on a tripod at approximately 1 m from a consumer camera with focal length 34 mm (expressed as 35 mm-equivalent) and resolution 1200 × 960. Figure 3.21 also shows selected detail views of the marker being incrementally turned in yaw and pitch direction. It is ensured that the marker remains in the image center so as to minimize the effects of lens distortion.

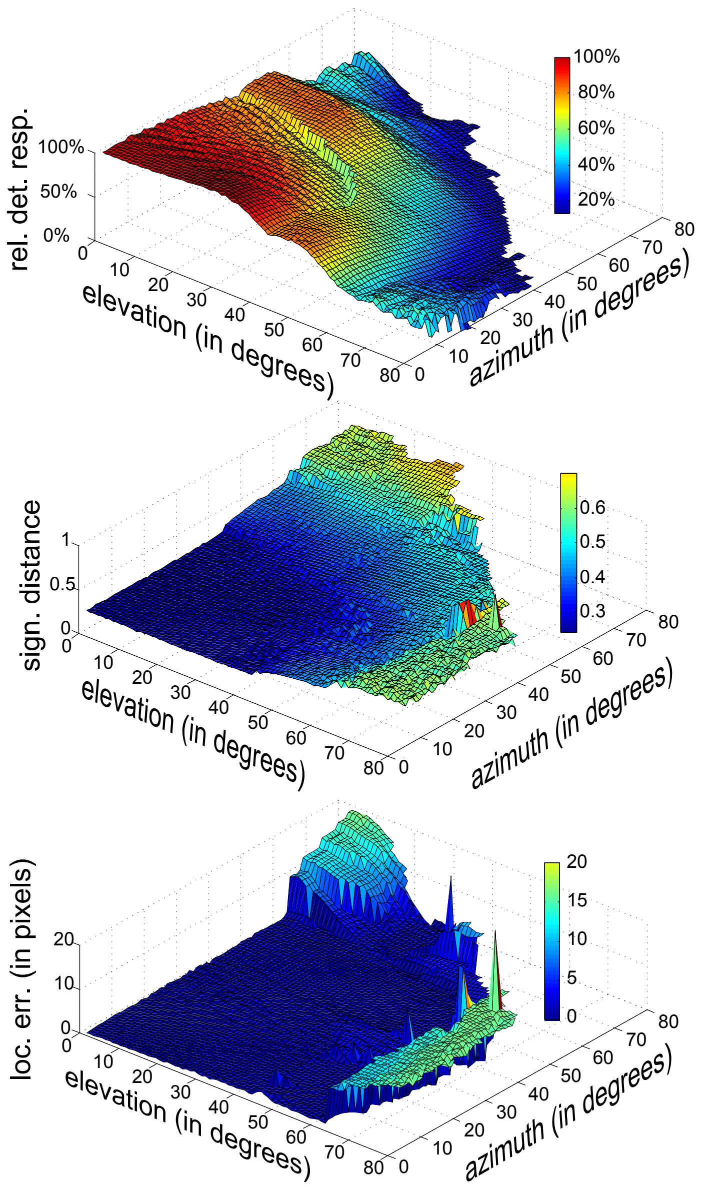


*Figure 3.18:* *Relative detector response, signature distance and localization error for varying marker orientation. Angles are measured from the marker's normal vector. The response is relative to its theoretical maximum, the signature distance is based on the undistorted marker's SURF descriptor, and the localization error is defined as the deviation of the detected marker center from its ground truth position.*

Moreover, the marker is printed in front of a checkerboard pattern which allows the exact extrinsic calibration of the camera with respect to the marker. This is achieved using the DLR camera calibration toolbox [JYB] and Zhang's calibration method [Zha99]. The imaged size of the frontally viewed marker itself is approximately 120 pixels.

In Figure 3.22, the obtained values for the SURF detector response and signature distance are plotted. Note the similarity to the results on synthetic data from Figure 3.18. While the

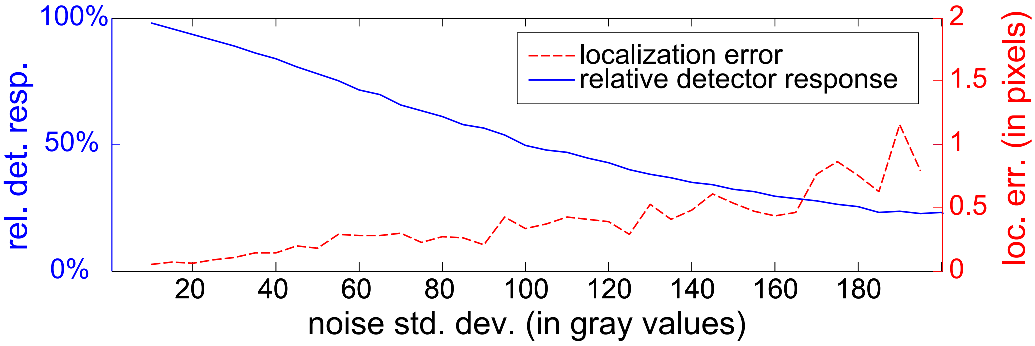


*Figure 3.19:* *The influence of AWGN on the marker localization error and the detector response normalized by its maximum value.*

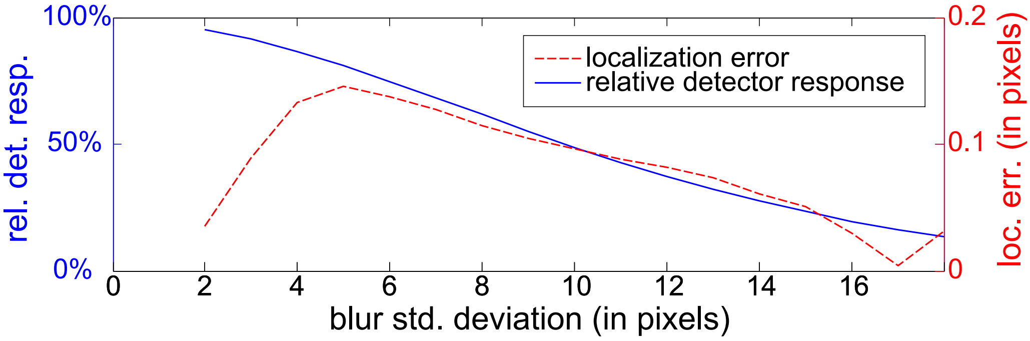


*Figure 3.20:* *The influence of Gaussian blur on the marker localization error and the detector response normalized by its maximum value.*

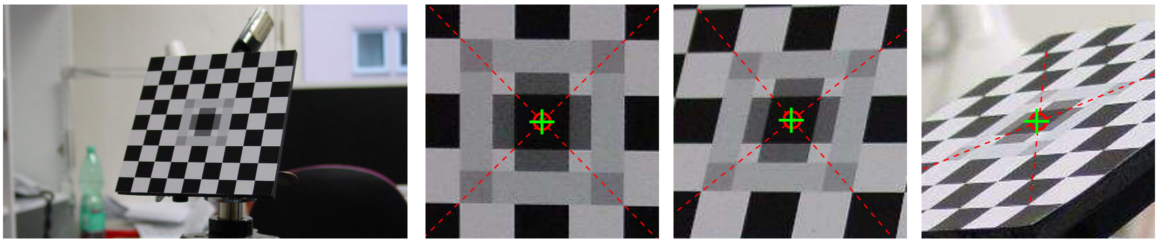


*Figure 3.21:* *A SURF marker placed on a checkerboard pattern for the experiments on real images (far left), and the detected marker location (+) compared to ground truth (○) for three selected orientations (cf. Table 3.1).*

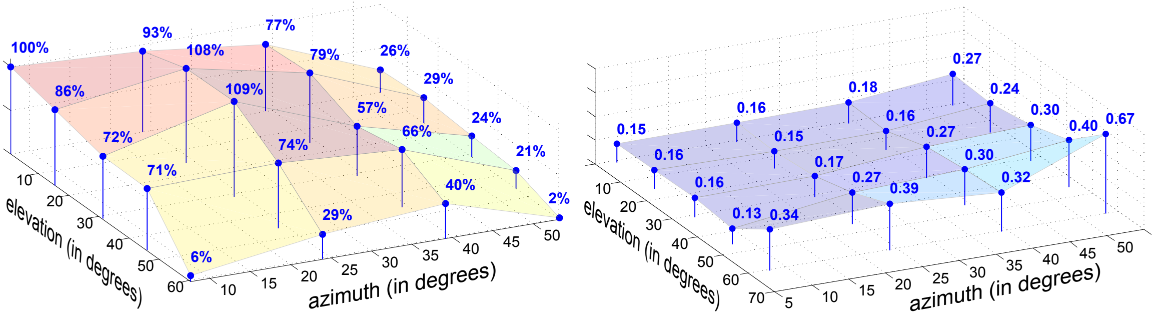


***Figure 3.22:*** *Detector response (left) and signature distance (right) for different marker orientations. The former has been normalized with respect to the view closest to frontal.*

| | | azimuth angle | | | |
|---|---|---|---|---|---|
| | | 7.7° | 23.9° | 39.1° | 53.3° |
| elevation | 0.8° | **0.48 px** | 0.65 px | 0.82 px | 0.18 px |
| | 15.4° | 0.60 px | 1.09 px | 1.43 px | 0.72 px |
| | 31.1° | 0.73 px | **1.11 px** | 0.94 px | 0.79 px |
| | 45.9° | 1.19 px | 1.42 px | 1.46 px | 1.27 px |
| | 60.4° | 9.18 px | 3.62 px | 0.29 px | **1.46 px** |

***Table 3.1:*** *Localization error in pixels for different marker orientations. Angles are measured from the normal vector of the marker. The images corresponding to the bold values are depicted in Figure 3.21.*

detector response remains relatively high for angles up to roughly 40°, the signature distance remains low in the same range. This gives reason to expect good distinctiveness of the marker in comparison with natural image features. The measured localization errors for the different azimuth-elevation pairs are given in Table 3.1. The angles were obtained from the camera calibration, and ground truth for the marker position was computed from the intersection of the diagonals through the extremal corners of the checkerboard pattern. The values for the localization error are similar to those expected from established, conventional marker systems. For instance, in [ZFN02], some quadrilateral designs are tested in a comparable setup, yielding average localization errors ranging from 0.17 to 1.57 pixels

### 3.2.5 Example Applications

The proposed markers are particularly useful whenever homogeneous, unstructured surfaces need to be "spiced up" with detectable features. Here, two example applications are presented.

#### Robotic Navigation

Imagine an experimental setup where a solely vision-based robot is to navigate through the lab in order to fulfill a certain task. Without a map or detectable markers in the scene,

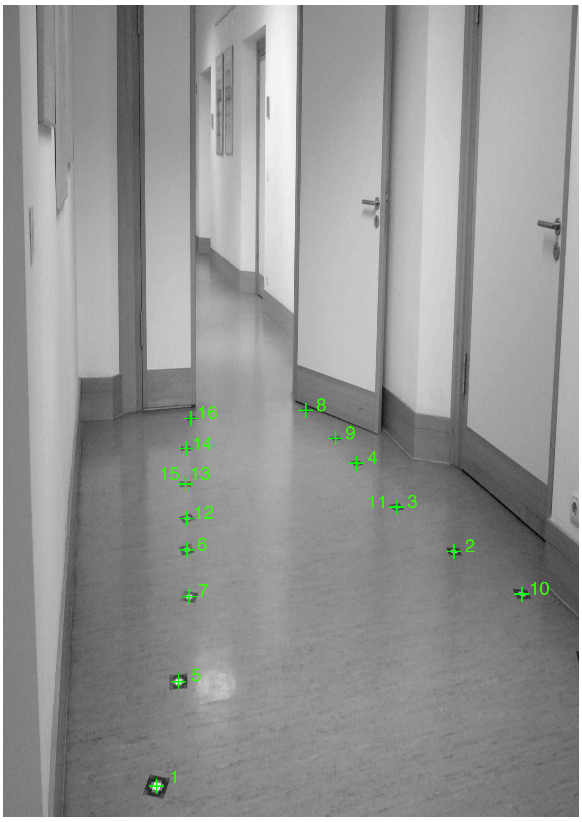


***Figure 3.23:*** *Image captured with a Panasonic DMC-FX1 at original resolution* 1536×2048 *pixels. The detected SURF markers are labelled according to the ranking in Figure 3.24.*

the robot would certainly lose track. Assume the task involves the recognition of certain objects which is based on the use of SURF features already. Such a prototypical scenario is the ideal application for the maximum detector response markers: The path to be followed by can simply be lined with markers reliably detectable by the robot with its already built in functionalities. To facilitate the navigation, the light and dark versions of the marker could even be used to identify and tell apart the left and right lane boundaries.

Given the sensitivity to perspective distortions due to out-of-plane rotations discovered in the experiments from Section 3.2.4, a simple preprocessing step is proposed for this particular scenario: As the height of the robot mounted camera remains constant, we can assume that the markers are always perceived perspectively distorted in the same way, *i.e.*, they appear vertically compressed. Stretching the image vertically, *e.g.*, by a factor 2 (using bilinear interpolation) partly undoes the distortion and improves marker detection significantly.

As illustrated in Figure 3.23, 14 $8 \times 8$ cm markers are placed on the floor outlining the desired path. When SURF is applied to the depicted image with low response threshold, in total some 30000 features get detected. Nevertheless, the MDR markers are reliably found as those features with the highest detector response values. Two of the markers get detected twice, each time with different orientation, so there are 16 highest response features for 14 present markers.

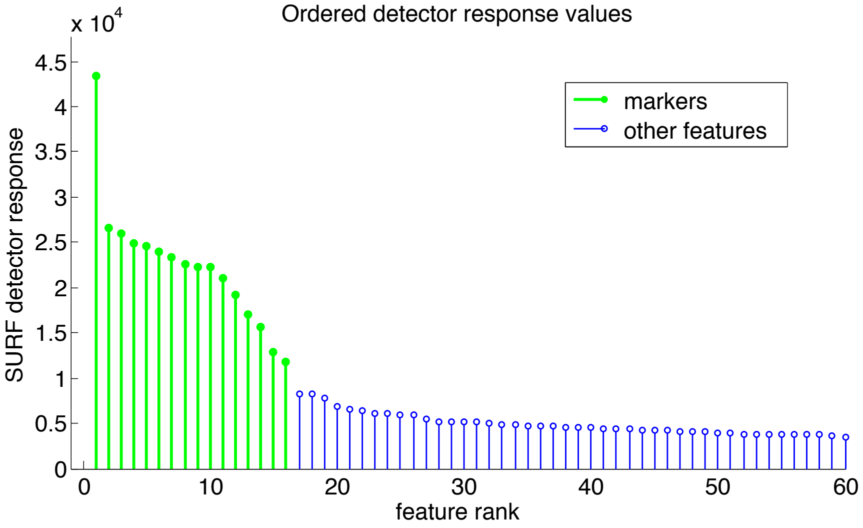


***Figure 3.24:*** *The SURF detector responses from the example in Figure 3.23 in descending order. The first 16 values belong to the imaged markers.*

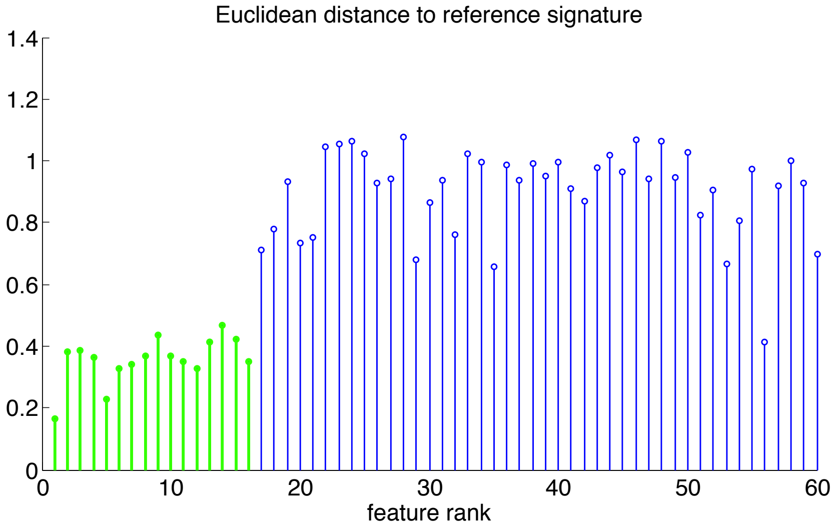


***Figure 3.25:*** *Signature similarity for the SURF features from the example in Figure 3.23. For every feature, the Euclidean distance between its signature and the signature of the template marker is plotted.*

Figure 3.24 shows the response values in descending order. The labels in Figure 3.23 correspond to this ranking. A drop from feature 16 to feature 17 is clearly visible., *i.e.*, the markers are indeed *uniquely detectable* in the image. Figure 3.25 shows the similarity between the descriptor signature of detected points and the signature of the dark marker template. Again, the markers clearly stand out. Note that there are other features, *e.g.*, number 56, with similar signatures, but their rank, and equivalently the detector response they trigger are much lower.

A property of this example setup that is worthwhile mentioning is that, since every detected marker comes with a scale assigned by the SURF algorithm, the robot can already make first assumptions about the marker distances without the need for stereo vision and triangulation.

### SIFT/SURF CAVE

Another example application that takes a similar line is to have a wallpaper or poster textured with maximum detector response markers. By simply putting up such posters, a computer vision lab can easily be turned into a "SIFT/SURF CAVE" with dense feature points all over the walls.

Mixing light and dark versions of the marker, such posters can carry one bit per marker allowing, *e.g.*, to encode information about different parts of the room. In principle, there are two ways to distinguish both marker variants from each other, either based on marker signature or by means of the Laplacian of the respective marker, *i.e.*, the sum of second derivatives. For SIFT, it is just the detector output (3.6) that approximates the Laplacian. In the case of SURF, it is given by the trace of the Hessian matrix introduced in (3.7a). The Laplacian is strictly negative for dark markers and strictly positive for light ones, hence a simple thresholding is sufficient for separation. Experiments showed that this approach is more reliable than signature comparison, it will be used in the following.

Figure 3.26 illustrates the combination of markers using arrays composed of three SURF markers each. The upper left corner of the two-by-two arrays is left blank to have an unambiguous array orientation. This design allows eight different marker arrays which could be used to tag strategic areas in the room.

The following simple algorithm proved sufficient to identify and decode these marker arrays. First, the SURF markers are uniquely detected in the image by means of their outstanding detector response and signature similarity. The feasibility of this step will be demonstrated shortly. Second, those markers are identified whose closest two neighbors satisfy certain geometric constraints, namely (a) their respective distances are similar and (b) they lie in roughly perpendicular directions. This yields 3-marker clusters in linear time with regard to the number of markers in place. Assuming that all the marker arrays are imaged close to upright, the topmost marker of each triple is then defined to carry the first bit, the leftmost one the second and the remaining marker the third bit. Finally, the bit value of each marker is determined using the sign of its Laplacian (see Figure 3.28). This approach is supposed to illustrate that the combination of markers is a feasible approach in general, it can of course be extended in a more sophisticated way.

Figure 3.27 shows the results of the SURF detection step. Similarly to the robot navigation

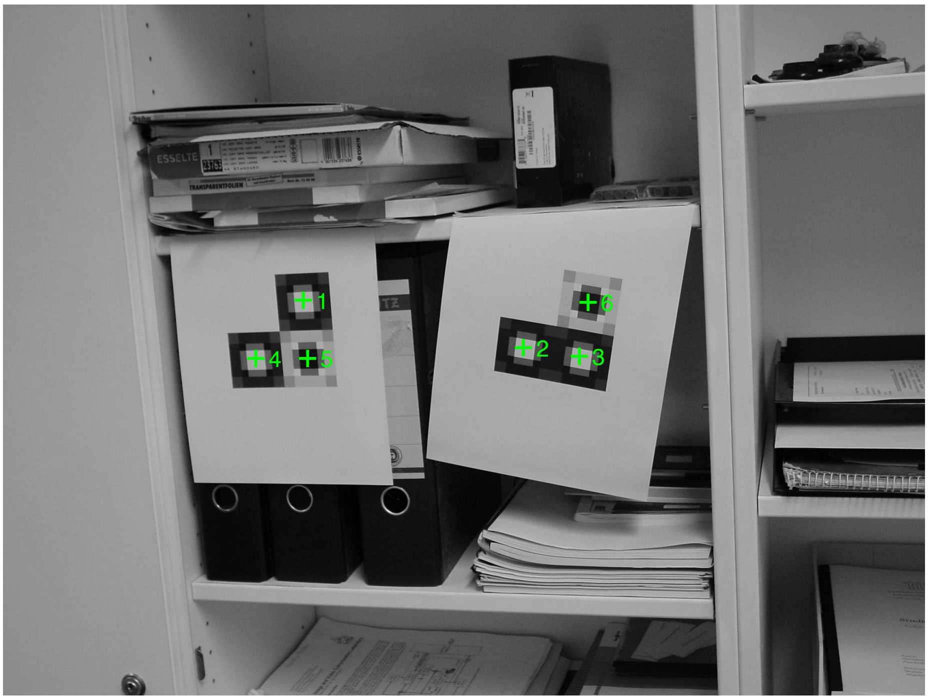


***Figure 3.26:*** *Two out of eight possible 3-bit marker arrays photographed with a Sony DSC-S85 at resolution 2272×1704 pixels. The location of the six SURF features which yielded highest detector response are displayed as green crosses.*

example, the detected features are examined in terms of detector response and distance to the SURF reference signature from Figure 3.8. Every point in the figure corresponds to a detected feature, and obviously, the six features that belong to the markers form a cluster in an area with high detector response and low signature distance. As opposed to the previous experiment where only dark markers were deployed, in this example, both versions of the SURF marker need to be identified simultaneously. Therefore, instead of the full marker signatures, only those entries which are common to both marker variants are considered. In the OpenCV-based implementation of SURF the corresponding descriptor dimensions are 23, 24, 27, 28, 39, 40, 43 and 44. This reduced representation is referred to as *Version-Independent Signature* (VIS).

## 3.3 Accuracy Assessment for Scale-invariant Image Features

Scale-invariant image features are a widely used tool to identify similar or identical parts across images. In multi-camera setups, in particular, geometric relations between images can be inferred exploiting feature correspondences. For trustworthy results it is necessary to work with robust features that can be reliably matched. Relevant measures in this context are the repeatability of feature detection under varying conditions, as well as the matching accuracy of corresponding feature pairs. Another important aspect which is of-

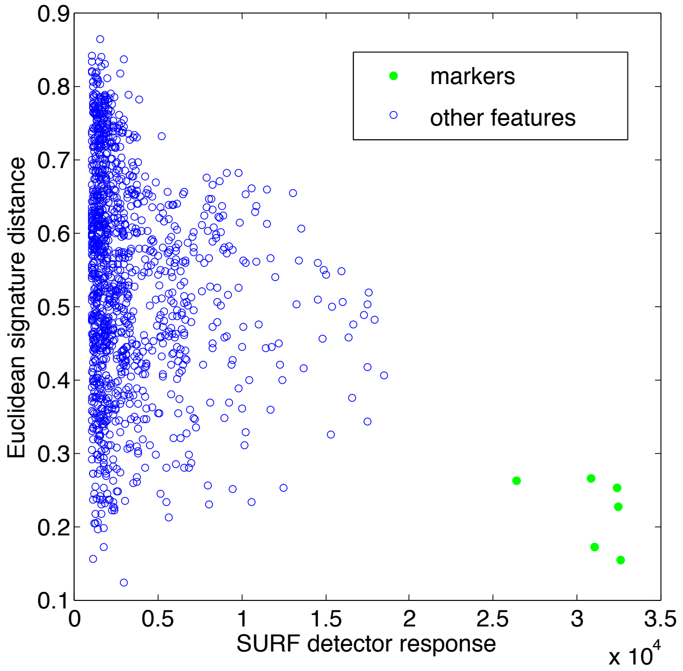


***Figure 3.27:*** *Detector response and VIS distance values for all the SURF features extracted from the image in Figure 3.26. The markers form an apparent cluster.*

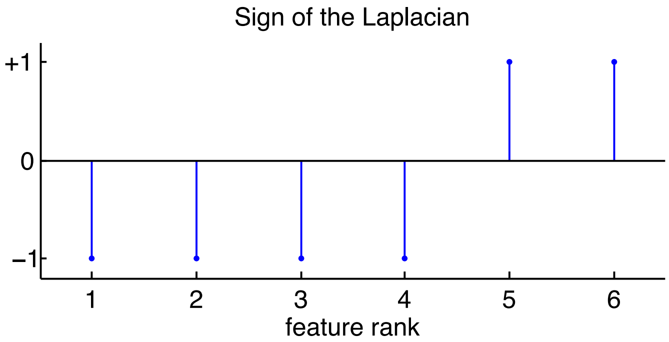


***Figure 3.28:*** *The sign of the Laplacian allows to reliably distinguish the light (+1) and dark (-1) versions of the SURF marker (compare with Figure 3.26).*

ten overlooked is the accuracy of the detection process itself. In current feature detection frameworks, it is assumed that the strongest features (according to an appropriate detection criterion) are automatically well localized. But that is not necessarily true, especially for scale-invariant features that can be detected on arbitrary scales. As a result of this thesis, it was effectively shown that the particular image content has a significant impact on the localization of the features. An obvious example are features located along an edge in the image which lack location precision in the direction of the edge. This effect has been dealt with, *e.g.* in SIFT, by identifying overly unreliable features in a second pass and simply excluding them from further processing. Another, also quite plausible realization confirmed by the results presented in this section is that large scale features are less accurately locatable than fine details.

This work proposes a way to precisely quantify the localization uncertainty of a given feature based on the underlying image intensities. This allows applications to incorporate this information and propagate the uncertainty estimates all the way to the computed results. For instance, [Sur10] makes use of the uncertainty measure proposed in this thesis to improve feature matching between images.

This part of the thesis is based on results published in [Zei09, ZGS$^+$09].

### 3.3.1 Related Work on Feature Uncertainty

There are several publications dealing with the localization accuracy of image features. Many of them, especially the earlier works [KK01, BCGVDH01, Kan04, SJ05], focus on classical corner detectors such as the Harris [HS88] and Förstner [FG87] algorithms. Only more recent publications specifically consider scale-invariant features [HJA08]. In all works, the localization error of an image feature is modeled as zero-mean Gaussian noise, defined by its $2 \times 2$ covariance matrix.

Kanatani and Kanazawa [KK01, BCGVDH01, Kan04] consider the so called *self-matching residual*, *i.e.*, the absolute sum of pixel value differences in a window around an interest point, as a measure of localization definiteness. They derive, irrespective of the actually used feature detector, that from a second order approximation of the self-matching residual an estimate for the covariance matrix of a feature's localization error can be obtained. Specifically for Harris corners, however, they conclude that the so acquired covariances cannot be exploited to significantly improve typical computer vision tasks such as structure-from-motion. They argue that Harris corners are markedly well selected features in that their covariances already are very similar, and especially isotropic. Reweighting Harris corners individually during an optimization, *e.g.*, employing the Mahalanobis distance [Mah36] (see also Section 3.3.5), would thus be futile.

In a series of experiments, Brooks et al. succeeded to demonstrate that under certain conditions Harris corners equipped with covariances actually can have a positive influence on parameter estimation [BCGVDH01]. Their research also casts light on the accuracy of the covariance estimate itself and its influence on the overall results.

Still considering traditional single scale features, Steele et al. derive covariance estimates for a given detector, specifically Förstner corners in their case, regarding image noise as the main source for misdetection [SJ05]. Investigating different noise models, they propagate the respective stochastic properties through the detection process, which leads to the desired covariance estimates. According to their results complex noise models are necessary to capture the underlying effects, and to obtain realistic covariance estimates.

Haja et al. go beyond single-scale features and study scale-invariant algorithms [HJA08]. The compared feature detection algorithms are found to differ in terms of location accuracy. Furthermore, the location error varies significantly depending on where in the image a particular feature was detected. Coarser features on higher scales tend to be less precisely localized. However, no attempt is made in [HJA08] to quantify the observed localization errors, *i.e.*, to estimate the corresponding covariances.

The approach presented in this section particularly addresses the covariance estimation of the location error for scale-invariant features. The proposed framework has been successfully applied in several applications, ranging from robust feature matching with *a contrario* models [Sur10] to photogrammetric surface reconstruction [BDBL+11].

### 3.3.2 Location Uncertainty Estimation

Based on the generic scale-invariant feature extraction scheme presented in Section 3.1, a framework is established to estimate the location uncertainty of image features [ZGS+09]. Consider the scale space representation discussed in Section 3.1, with a stack $I(\mathbf{x}, \sigma_i)$ of $N$ discrete scale layers. Here, $\mathbf{x} = (x, y)$ specifies the spatial dimensions in the image, and $\{\sigma_i,\ i = 1, \ldots, N\}$ is the set of discrete scale levels.

Following the work by Lindeberg [Lin90, Lin93, Lin94, Lin98], a derivative-based detection operator $f_{det}$ is applied to each layer of the scale-space stack, and local extrema are identified as features in the three-dimensional detector response. The extremum search can be regarded as a two-stage process, first identifying local extrema in $D(\,\cdot\,, \sigma_i) = f_{det}[I(\,\cdot\,, \sigma_i)]$ as feature candidates, individually in each scale layer. Second, scale selection is performed for each feature candidate by finding local extrema along the scale dimension. Formally, here for the case where features are detected as response maxima, the two steps can be described by Equations (3.9) and (3.10). When response minima are also considered features, as is common practice in, *e.g.*, some implementations of SIFT, analogous equations apply with maximization replaced by minimization.

$$\mathcal{F}_1 = \bigcup_{i=1}^{N} \left\{ (\mathbf{p}, \sigma_i) \,\middle|\, \mathbf{p} = \arg \max_{\mathbf{x} \in \mathcal{N}_\mathbf{p}} D(\mathbf{x}, \sigma_i) \right\} \tag{3.9}$$

$$\mathcal{F}_2 = \left\{ (\mathbf{p}, \sigma) \in \mathcal{F}_1 \,\middle|\, \sigma = \arg \max_{s \in \mathcal{N}_\sigma} D(\mathbf{p}, s),\ D(\mathbf{p}, \sigma) > \tau \right\} \tag{3.10}$$

Here, $\mathcal{N}_\mathbf{p}$ and $\mathcal{N}_\sigma$ are local neighborhoods around $\mathbf{p}$ and $\sigma$ in space and scale, respectively. The threshold $\tau$ further eliminates all members from the initial candidate set $\mathcal{F}_1$ with insufficient response. The final set $\mathcal{F}_2$ contains the eventually detected images features [3]. During both maximizations in (3.9) and (3.10), interpolation can be used to obtain subpixel and inter-scale accuracy.

It is important to note that the scale selection process given in (3.10) does not alter the location at which a feature is ultimately detected. Only the spatial extremum search in $D(\,\cdot\,, \sigma_i)$, given by (3.9), is determining.

Obviously, the shape of the function $D(\,\cdot\,, \sigma_i)$ determines how well a local extremum is localized. Peaky maxima lend themselves to accurate localization, whereas shallow extrema are less reliable. For the following argument, let's introduce the residual error $R(\Delta\mathbf{p}, \sigma_i)$

[3] Actually, the set of true image features is a subset of $\mathcal{F}_2$. A pairwise comparison of neighboring features in $\mathcal{F}_2$ is still required to discard false local maxima. This additional step, however, has no consequence on the following considerations and is hence neglected.

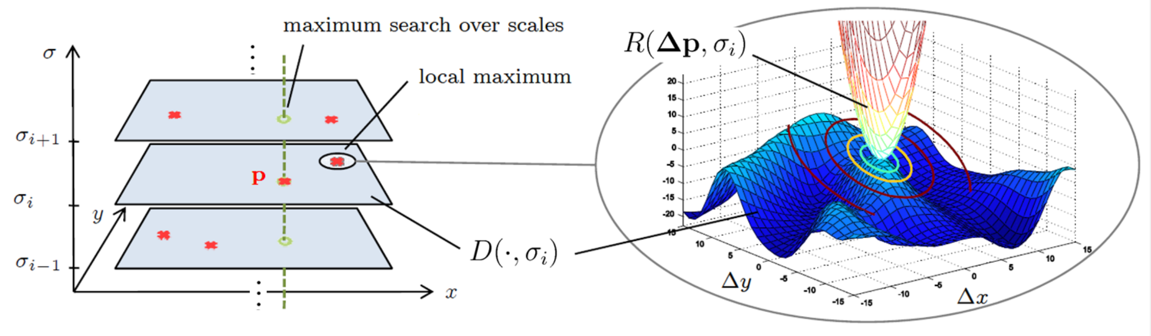


***Figure 3.29:*** *A detailed view on the extrema search in the scale space stack. The residual error $R(\mathbf{\Delta p}, \sigma_i)$ takes on a minimum where $D(\,\cdot\,, \sigma_i)$ has a local maximum.*

(Image reproduced from [Zei09])

encountered when a detected feature is displaced from its true location by $\Delta\mathbf{p}$:

$$R(\Delta\mathbf{p}, \sigma_i) = |D(\mathbf{p}, \sigma_i) - D(\mathbf{p} + \Delta\mathbf{p}, \sigma_i)| \tag{3.11}$$

Note that $D(\,\cdot\,, \sigma_i)$ has a local extremum when $R(\,\cdot\,, \sigma_i)$ is minimized. See Figure 3.29 for an illustration.

For small displacements, (3.11) can be adequately approximated by a Taylor series up to second order:

$$R(\Delta\mathbf{p}, \sigma) \approx \underbrace{R(\mathbf{0}) + \left.\frac{\partial R(\Delta\mathbf{p}, \sigma)}{\partial \Delta\mathbf{p}}\right|_{\Delta\mathbf{p}=\mathbf{0}}}_{=0} + \frac{1}{2}\Delta\mathbf{p}^\top \underbrace{\left.\frac{\partial^2 R(\Delta\mathbf{p}, \sigma)}{\partial \Delta\mathbf{p}^2}\right|_{\Delta\mathbf{p}=\mathbf{0}}}_{=:\mathbf{H}} \Delta\mathbf{p} \quad = \frac{1}{2}\Delta\mathbf{p}^\top\mathbf{H}\Delta\mathbf{p} \tag{3.12}$$

The zeroth and first order terms vanish because the residual $R(\Delta\mathbf{p}, \sigma)$, by definition, reaches its global minimum at $\Delta\mathbf{p} = 0$.

The Hessian matrix $\mathbf{H}$ describes the curvature of the residual, and its inverse is a good estimate for the shape of the localization error's covariance [KK01], hence the feature uncertainty. The covariance of a feature detected at position $(\mathbf{p}, \sigma)$ in scale-space is consequently given by

$$\mathbf{\Sigma} = \mathbf{H}^{-1} = \begin{bmatrix} R_{xx}(\Delta\mathbf{p}) & R_{xy}(\Delta\mathbf{p}) \\ R_{yx}(\Delta\mathbf{p}) & R_{yy}(\Delta\mathbf{p}) \end{bmatrix}^{-1}_{\Delta p=0} = \mp \begin{bmatrix} D_{xx}(\mathbf{p}, \sigma) & D_{xy}(\mathbf{p}, \sigma) \\ D_{xy}(\mathbf{p}, \sigma) & D_{yy}(\mathbf{p}, \sigma) \end{bmatrix}^{-1} . \tag{3.13}$$

The sign on the right hand side of (3.13) depends on whether the feature $(\mathbf{p}, \sigma)$ has been detected as a local maximum $(-)$ or as a local minimum $(+)$. Note that $\mathbf{\Sigma}$ is only determined up to scale, *i.e.*, only the shape and relative extent of the underlying localization error can be appraised.

Depending on the particular structure of the scale-space representation and the derived detector stack $D$, the so computed covariance matrix $\mathbf{\Sigma}$ needs to be normalized with respect to a common reference scale $\sigma_0$. This ensures that covariances retain their relative proportions. In particular, normalization is necessary if the spatial resolution differs between scale layers, *e.g.*, when the scale space is divided into octaves and scales of higher octaves are subsampled. Introducing the notation $\mathrm{res}(I)$ for the spatial resolution of image $I$ (measured in *pixels per image width*), normalization of a covariance matrix $\mathbf{\Sigma}$ computed at scale $\sigma$ is achieved by Equation (3.14).

$$\mathbf{\Sigma}_0 = \left(\frac{\mathrm{res}(D(\,\cdot\,,\sigma_0))}{\mathrm{res}(D(\,\cdot\,,\sigma))}\right)^2 \mathbf{\Sigma} \tag{3.14}$$

The normalized covariance matrix $\mathbf{\Sigma}_0$ describes the localization uncertainty of the feature $(\mathbf{p},\sigma)$ expressed in terms of the original image size.

### 3.3.3 Application to SIFT and SURF

SIFT and SURF use Difference of Gaussians filters and the determinant of the Hessian matrix as detection operators, respectively (see Sections 3.1.1 and 3.1.2). It is straightforward to apply the present location uncertainty estimation framework by evaluating the covariance matrix in (3.13) based on the respective detector response values, *i.e.*, either with $D = D_{\text{SIFT}}$ or $D = D_{\text{SURF}}$.

In practice, the required second order derivatives of the detector output are computed employing finite difference filters, evaluated at the feature location:

$$D_{xx}(\mathbf{x},\sigma) = h_{xx} * D(\mathbf{x},\sigma), \tag{3.15a}$$

$$D_{xy}(\mathbf{x},\sigma) = h_{xy} * D(\mathbf{x},\sigma), \tag{3.15b}$$

$$D_{yy}(\mathbf{x},\sigma) = h_{yy} * D(\mathbf{x},\sigma), \tag{3.15c}$$

$$\text{where} \quad h_{xx} = h_{yy}^{\top} = \begin{bmatrix} 1 & -2 & 1 \end{bmatrix}, \quad h_{xy} = \frac{1}{4}\begin{bmatrix} 1 & 0 & -1 \\ 0 & 0 & 0 \\ -1 & 0 & 1 \end{bmatrix} \tag{3.15d}$$

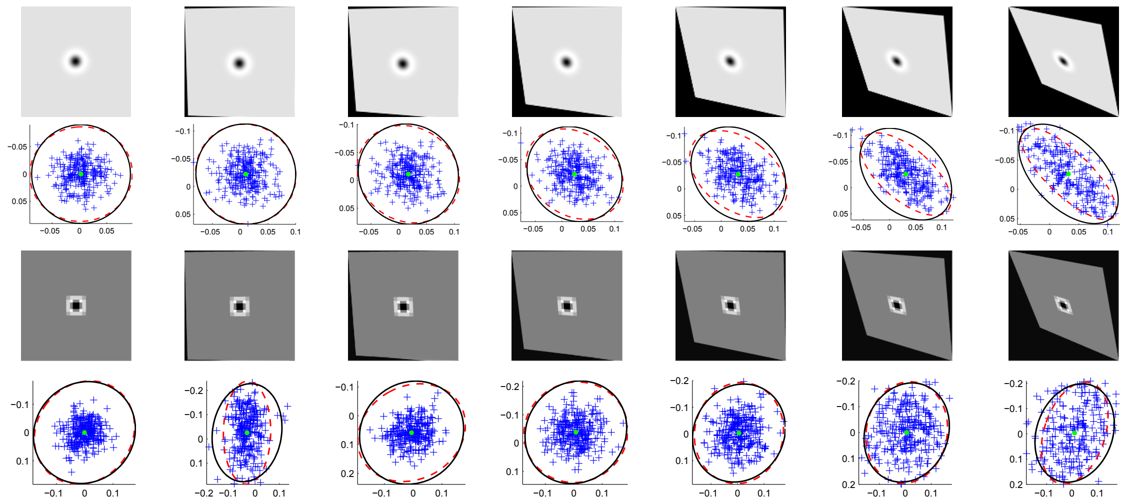
Since both SIFT and SURF use interpolation to refine the feature location in scale space, a feature $(\mathbf{p},\sigma)$ usually lies between scale layers. Ultimately, this would require to interpolate a non-existent scale from neighboring scale layers to eventually compute (3.15) and (3.13). As it turns out, the computations can also be performed on the scale layer closest to $\sigma$, without significant degradation of the results [ZGS+09].

Due to the subsampling of higher octaves used by both SIFT and SURF, scale normalization according to (3.14) is necessary. For a feature detected in octave $o$, normalization hence takes the form given in Equation (3.16):

$$\mathbf{\Sigma}_0 = 4^o\,\mathbf{\Sigma} \tag{3.16}$$

### 3.3.4 Experimental Validation

In order to assess the methodological correctness and the performance of the developed uncertainty estimation framework, several tests have been conducted. In accordance with



(a) 0° (b) 10° (c) 20° (d) 30° (e) 40° (f) 50° (g) 60°

***Figure 3.30:*** *The detection error for SIFT (top row) and SURF (bottom row) under varying viewing directions. The ground truth feature position is indicated by a green dot, measured positions by blue plus signs. The measured and estimated covariances are marked in dashed red and solid black, respectively.* (Image reproduced from [Zei09])

the previous sections, and especially Section 3.3.3, the SIFT and SURF algorithms are considered in the experiments. SIFT is used in the OpenCV-based implementation by Rob Hess [Hes10]. For SURF, the OpenCV implementation has been altered such that the original scale sampling [BETVG08] (*cf.* Figure 3.6) is in place, as proposed in Section 3.2.4 (see also Figure 3.13). Both synthetic data and real images are used to substantiate the approach.

### Measured versus Estimated Covariance

The first experiment simulates the detection of a feature at a given ground truth position in a synthetically generated image. To this end, the *maximum detector response markers* described in Section 3.2 are used to enforce the detection of a single, reliable feature at a known location. The markers, being the prototypical feature for the respective algorithm, are viewed from different perspectives, with Gaussian noise added to disturb the detection process in a realistic manner.

In hundreds of realizations, the deviation from the ground truth marker location is recorded, as depicted in Figure 3.30. From these samples, a least squares fit of the localization error's covariance matrix is computed which is marked with a dashed red line in Figure 3.30. This measured covariance is then compared to the covariance estimate calculated from (3.13) according to the presented approach (solid black line in Figure 3.30). Measurement and estimate are normalized to equal Frobenius norm before comparison.

As can be seen from Figure 3.30, the estimate follows the measured covariance quite accurately. In the case of SIFT, where the distribution of the localization error clearly fol-

| View | 0° | 10° | 20° | 30° | 40° | 50° | 60° |
|---|---|---|---|---|---|---|---|
| SIFT | 0.181 | 0.850 | 0.955 | 2.72 | 7.94 | 32.9 | 50.2 |
| SURF | 0.402 | 36.5 | 1.90 | 0.582 | 3.57 | 0.870 | 12.5 |

***Table 3.2:*** *The Bhattacharyya distance between the measured and estimated covariances from Figure 3.30. The overall order of magnitude is* $10^3$.

lows the distortion of the test image, the covariance estimate changes accordingly. But also for SURF, where the error evolves counter-intuitively with increasing distortion, the covariance estimate reliably captures the true distribution. In order to quantify the deviation of the estimated covariance from the measured one, the so called Bhattacharyya distance [Bha43], a common measure for the divergence of probability distributions, is used. Table 3.2 contains the resulting distances.

### Scale-dependence

In a second experiment, the relationship between the scale a feature gets detected on and its covariance estimated according to the presented framework is investigated. To this end, SIFT and SURF features are extracted from the image shown in Figure 3.32. The detection thresholds are deliberately set as low as to produce a sufficiently high number of features. For every feature, the location error covariance is estimated according to (3.16).

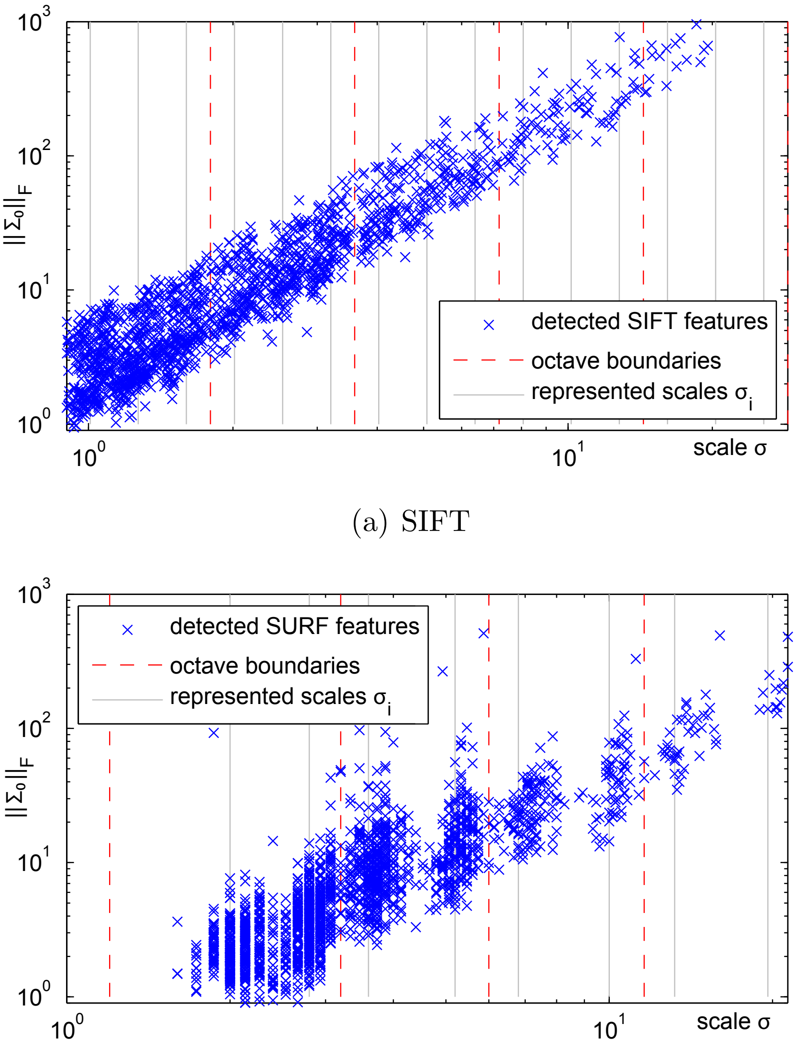
Figure 3.31 shows the results in the form of a scatter plot. Apparently, the uncertainty of a feature, measured by the Frobenius norm of its covariance matrix, increases with the scale of the feature. This was to be expected for features on different octaves for which, by definition (3.16), the covariances have been rescaled during the estimation process. Within an octave, where all covariances are normalized by the same factor, the same behavior can be examined as well. This is in accordance with the intuitive understanding that coarse features, detected on higher (possibly subsampled) scale layers are less well localized than delicate features.

In Figure 3.32, those SIFT features with markedly high and low covariance are plotted on top of the used image. This also illustrates the practical relevance of the location error covariance as a valid indicator for the uncertainty of a detected feature. Features with small covariances, marked with crosses in the figure, are well localized on fine structures, small blobs or corners. High covariance features, however, represent bigger, rather vague, blob-like structures that could just as well be located several pixels off under small distortions.

### Covariance under Perspective Distortion

Complementary to the initial experiments on synthetic data, the behavior of the covariance estimates under distortion has also been repeated with real-world images. The purpose of this is to demonstrate qualitatively the consistency of the proposed measure. Figure 3.33 shows how the covariance computed according to (3.16) changes when features undergo perspective transformation.

The figure shows frames from three separate videos where SIFT features have been tracked



(a) SIFT

(b) SURF

***Figure 3.31:*** *Frobenius norm of the covariance for SIFT (a) and SURF (b) features detected in the image shown in Figure 3.32.* (Image reproduced from [Zei09])

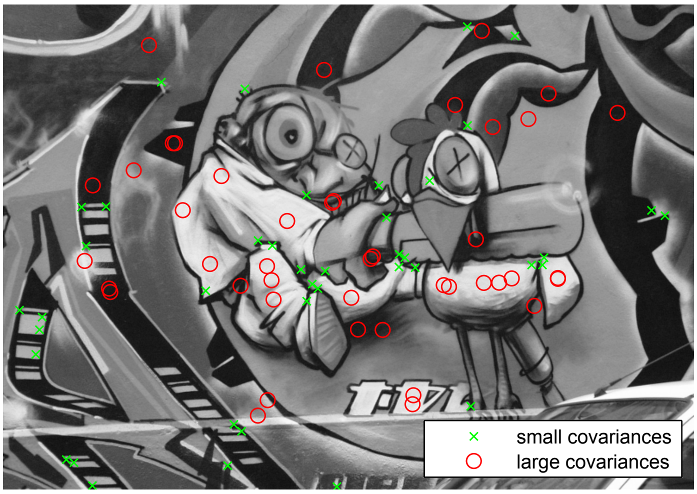


***Figure 3.32:*** *Selected SIFT features in a real-world image from the* Graffiti *dataset [MTS$^+$05]. Those with large covariances ($\sigma > 8$) are poorly localized in comparison to small covariance features ($\sigma < 2$).* (Image reproduced from [Zei09])

throughout the image sequences. In the first case, displayed in Figure 3.33(a), the pattern containing the features is zoomed into, and the covariance estimates follow the associated scale change accordingly. When a viewpoint change occurs where the object appears compressed along one direction, as shown in Figure 3.33(b), the computed covariances reflect this in a reduced uncertainty in exactly that direction. Finally, also rotating the object has the covariances follow in a consistent manner, as shown in Figure 3.33(c).

### 3.3.5 Application in Structure-from-motion

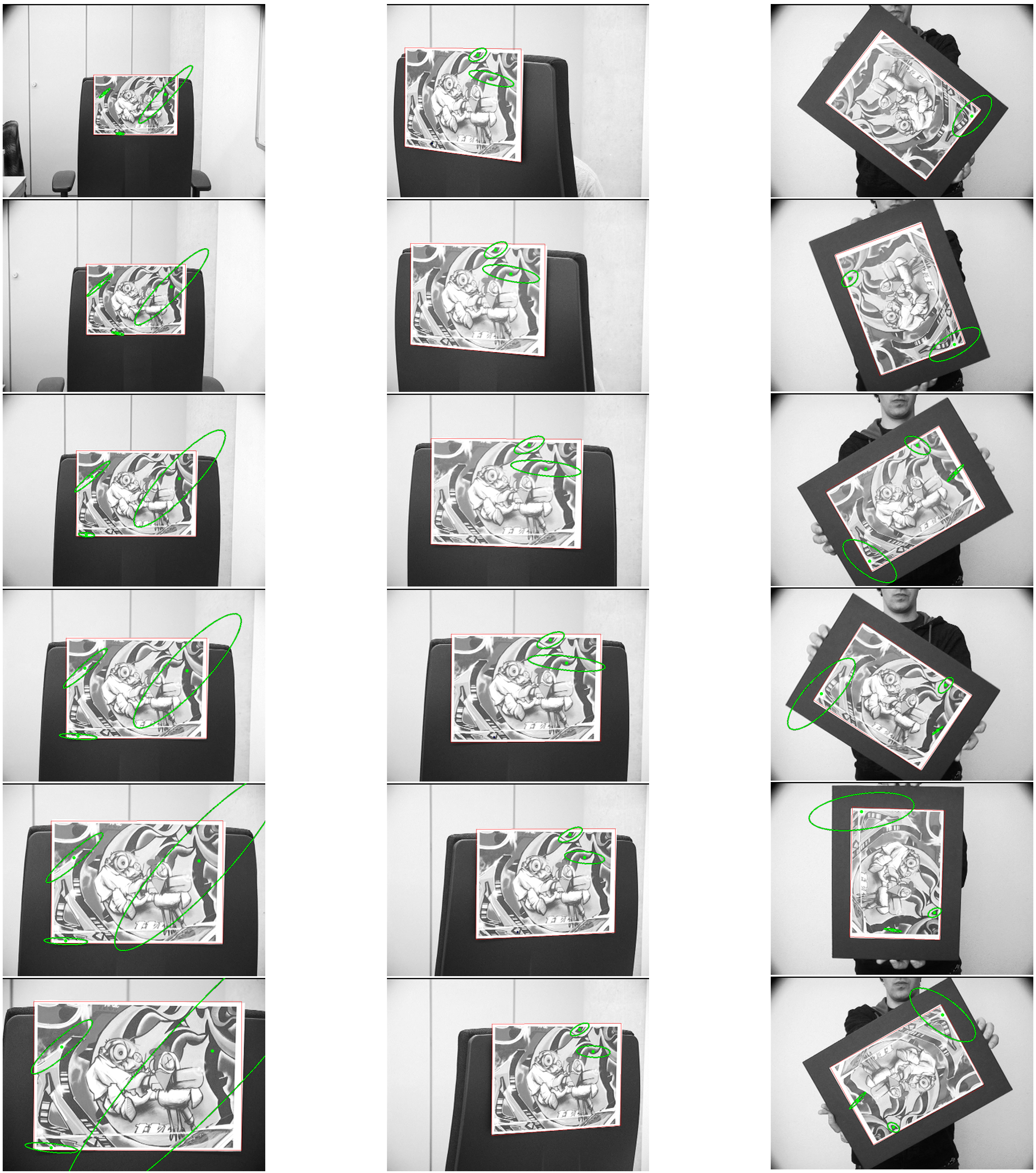
Having demonstrated the conceptual validity of the feature accuracy assessment method, it will be shown in this section how an existing computer vision algorithm can actually benefit from the obtained uncertainty measures. In particular, a structure-from-motion scenario is considered, where a 3D scene is reconstructed from 2D projections acquired with a stereo camera.

#### Setup and Problem Formulation

Formally, the task is to recover the position of $n$ points $\mathbf{X}_i \in \mathbb{R}^3$, and the pose of $m = 2$ cameras from the points' projections $\mathbf{x}_{ij} \in \mathbb{R}^2$. Introducing projection operators $P_j$ for each camera, the 3D-2D relationship can be expressed as follows:

$$\mathbf{x}_{ij} = P_j \mathbf{X}_i \tag{3.17}$$

In order to have full control over the setup, especially to have ground truth data available for quantitative evaluations, virtual scenes are used. Figure 3.34 shows an example setup. Each test scene is constructed from four parallel, planar patches of different sizes and located at varying distances from the cameras. Natural images are mapped onto the planes



(a) zoom (b) tilt (c) rotation

***Figure 3.33:*** *Three image sequences with different perspective distortions, and the evolution of the covariance for selected SIFT features. A small green dot marks the feature locations, the ellipses indicate their estimated covariance.* (Image reproduced from [Zei09])

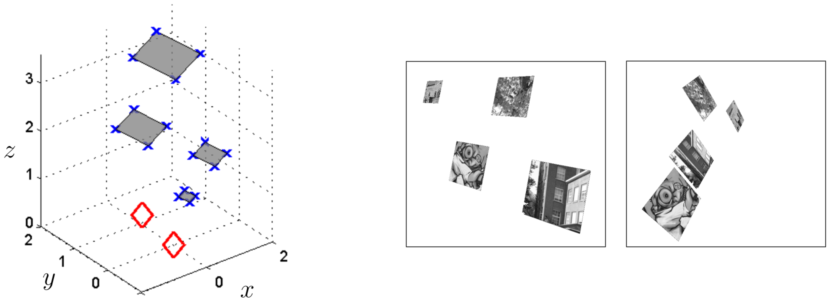


***Figure 3.34:*** *The virtual setup used in the structure-from-motion experiment composed of two cameras located in the $x$-$y$ plane (marked with red diamonds), looking along the $z$ axis, and four randomly placed planar patches (left). Two exemplary camera views on the scene are shown on the right.* (Image reproduced from [Zei09])

such as to provide enough texture for the SIFT and SURF algorithms. The position of the virtual stereo camera is fixed, the exact pose of the patches is chosen at random each time.

In the stereo camera images, either SIFT or SURF features are detected and matched between the image pair. To eliminate the influence of outliers, wrong matches are immediately identified through the known geometry and discarded. State-of-the-art reconstruction algorithms are then used to infer the 3D setup of planes and cameras. More specifically, an initial reconstruction is obtained by estimating and decomposing the *essential matrix* [Hor90, HZ00]. Subsequently, *bundle adjustment*, which will be briefly summarized shortly, is employed to iteratively refine the results. The popular SBA implementation by [LA09] is used for the experiments reported here. In case the reconstruction fails at any point, *e.g.*, because a random setup didn't spawn enough feature correspondences, the current run is aborted and a new random setup is synthesized.

After each successful run, the ground truth location of the 16 patch corners (marked with blue crosses in Figure 3.34) is compared with the reconstructed patch corners.

### Improvements in Bundle Adjustment

Bundle adjustment aims at amending the camera-scene configuration such that the average error between measured image points $\mathbf{x}_{ij}$ and their expected positions is globally minimized:

$$\min_{\mathbf{X}_i, P_j} \sum_{i=1}^{n} \sum_{j=1}^{m} \|\mathbf{x}_{ij} - P_j \mathbf{X}_i\|^2 = \min_{\mathbf{X}_i, P_j} \sum_{i=1}^{n} \sum_{j=1}^{m} (\mathbf{x}_{ij} - P_j \mathbf{X}_i)^\top (\mathbf{x}_{ij} - P_j \mathbf{X}_i) \tag{3.18}$$

Instead of minimizing the Euclidean distance as in (3.18), the measurements $\mathbf{x}_{ij}$ and the associated errors can be weighted according to their individual uncertainties. This leads to the established Mahalanobis distance [Mah36, HZ00] which directly incorporates the

| | average reprojection error | | |
|---|---|---|---|
| | without cov. | with cov. | improvement |
| SIFT | 2.031 px | 1.759 px | 13.4% |
| SURF | 2.554 px | 2.363 px | 7.5% |

***Table 3.3:*** *Average reprojection error* (3.20) *after 100 simulated bundle adjustment runs.*

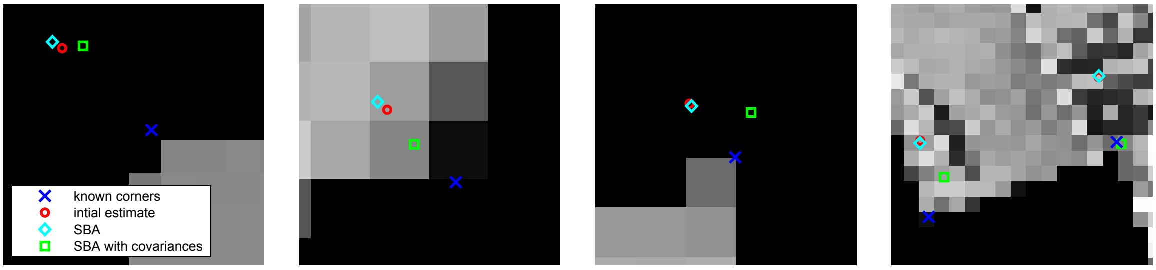


***Figure 3.35:*** *Four detail views on the patch corners from Figure 3.34 as seen by the first camera. The refined estimates tend more closely towards the ground truth position when covariances have been used.* (Image reproduced from [Zei09])

covariance matrix of each measurement, *i.e.*, each detected image feature:

$$\min_{\mathbf{X}_i,P_j} \sum_{i=1}^{n} \sum_{j=1}^{m} (\mathbf{x}_{ij} - P_j\mathbf{X}_i)^\top \boldsymbol{\Sigma}_{ij}^{-\frac{1}{2}} (\mathbf{x}_{ij} - P_j\mathbf{X}_i) \tag{3.19}$$

The covariance matrices $\boldsymbol{\Sigma}_{ij}$ in (3.19) are computed according to (3.16) for every feature $\mathbf{x}_{ij}$. Both with SIFT and SURF, 100 simulation runs are carried out, once without exploiting covariances minimizing (3.18), and a second time *with* covariances minimizing (3.19).

The average *reprojection error* over the 16 patch corners in one of the cameras is used a quality measure for the reconstruction:

$$e = \frac{1}{16} \sum_{i=1}^{16} \|\mathbf{c}_{i1} - \hat{P}_1 \mathbf{C}_i\|, \tag{3.20}$$

where the $\mathbf{C}_i$ are the 16 ground truth corner points in 3D, and $\mathbf{c}_{i1}$ are their projections in the first camera whose estimated projection matrix $\hat{P}_1$ is a result of the respective minimization.

The results are shown in Table 3.3 and in Figure 3.35. It can be seen that using the proposed covariance measure in combination with the Mahalanobis distance reduces the mean reprojection error significantly, by up to 13%. Figure 3.35 illustrates this accuracy improvement qualitatively, with detail views on selected patch corners.

## 3.4 Efficient and Affine-invariant Feature Detection

So far, the focus of this chapter was on the detection properties of existing feature detection algorithms, and specifically on SIFT and SURF. In this section, two novel, full-fledged feature detectors, coined speeded-up SURF (suSURF) and Affine-invariant suSURF (AsuSURF), will be presented, based on the findings from Section 3.2. The motivation behind these detectors is, for one thing, a reduction of computational complexity, and for another, additional invariance against affine geometric transformations.

The reason for these two requirements is obvious. Faster and especially leaner implementations enable real-time applications and the deployment on less powerful devices, *e.g.*, on smart phones. In the context of mobile media search for instance, as touched upon in Section 1, where a database of known features is maintained somewhere in the backend for mobile users to query, image features are the atoms for image and video retrieval. From a service provider's perspective, it is desirable to have the mobile devices perform the feature extraction locally, so that only features instead of whole images or videos need to be transmitted. However, mobile devices are significantly limited both in processing power and memory capacities. Consequently, faster and more efficient feature extraction algorithms are necessary, especially if real-time performance is required.

Affine-invariance on the other hand increases the applicability of a feature detector, allowing more drastic view point changes to be compensated. While (the only rotation- and scale-invariant) SIFT and SURF typically cope with viewing directions differing by up to 30° (see also Section 3.2.4), general perspective distortions prevent corresponding features to be detected and successfully matched between two views. When the dimensions of a feature are negligible in comparison with the viewing distance, which is typically the case, perspective transformations can be approximated with an affine model. A detector that is invariant to affine transformations is thus also largely invariant to general projective effects.

### 3.4.1 State-of-the-art

#### Fast Feature Detection

In the literature, several attempts at fast and efficient feature extraction have been made. One direction follows the development of simplified detectors, such as FAST [RD06] or the related ORB [RRKB11], that specifically target mobile, real-time tracking applications. Another line of thought aims at more efficient implementations of existing, well established algorithms. In the family of scale-invariant approaches, General-Purpose GPU (GPGPU) based implementations for SIFT and SURF have been proposed [SFPG06, HMS$^+$07, Wu, CVG08, FYZ$^+$11], taking advantage of their high parallelizability. The full potential of parallelization can of course only be exploited on architectures where GPGPU techniques are available. In a different approach, similar to SURF's box filter approximations, Grabner et al. have proposed to radically simplify the Difference-of Gaussian kernel used by SIFT, computing a Difference of Mean on integral images instead [GGB06]. The approach behind the suSURF algorithm presented in Section 3.4.2 is comparable to [GGB06] in that it simplifies the SURF detector, yet following a more substantiated line of argument.

### Affine-invariance

As for affine-invariance, there are several feature detectors that have been proposed in the literature. They can be divided into two main categories based on whether they compensate for affine transformations by *normalization* or by *simulation*. In this context, *normalization* refers to estimating the affine transformation parameters from the pixel neighborhood of a feature, and "undoing" the transformation, thus bringing the image patch around the feature to a normalized form. Both SIFT and SURF use this concept, for instance, to normalize the orientation of a feature. Among affine-invariant features, the most prominent representative of this category are Maximally Stable Extremal Regions (MSER) [MCUP04]. As the name implies, the MSER algorithm detects characteristic pixel regions in an image rather than single points in scale-space. Ellipses fitted to these regions contain the affine transformation parameters, and normalization is achieved by mapping them to the unit circle.

In the *simulation* based category, the space of affine transformations is sampled, and each of the resulting transformations is applied to the image. In one of the so distorted images, a given feature will appear as if seen straight from the front. An affine transformation is hence "undone" by simulating the frontal view of the feature. This concept is also employed for example during scale selection in SIFT and SURF. To be able to handle the vast number of image versions the simulative approach produces, the simulated affine transformations must be selected wisely, and sophisticated feature matching strategies are required. The foundation for the simulative concept has been laid by Yu et al. whose Affine-invariant SIFT (ASIFT) algorithm successfully applies it to SIFT [YM09,MY09,YM11]. With AsuSURF, presented in Section 3.4.3, the same concept is carried over to suSURF.

## 3.4.2 Speeded-up SURF (suSURF)

The detector presented here, as its name suggests, is based on the already extensively discussed SURF algorithm. The goal is to reduce the complexity in the detection process and hence make for an acceleration, while maintaining the superb detection quality of SURF. To achieve this, the insights gained during the development of the maximum-detector response (MDR) markers in Section 3.2 are exploited. Let us recall that SURF uses a non-linear detection operator. The approach followed here is to linearize the SURF detector and to reduce the number of operations needed to compute its response. All the other processing steps, especially extremum search via non-maximum suppression, scale-space interpolation, orientation assignment and descriptor computation, remain untouched.

### Detector Design

The MDR markers developed in Section 3.2 and depicted in Figure 3.9 on page 56 were constructed such as to trigger the strongest possible SURF response. They can thus be considered the prototypical features that the algorithm is supposed to detect. Conversely, to reliably locate features of this kind in an image, the SURF detector is the means of choice. Yet, it is not the only one. The optimal *linear* detection operator predestined for this task,

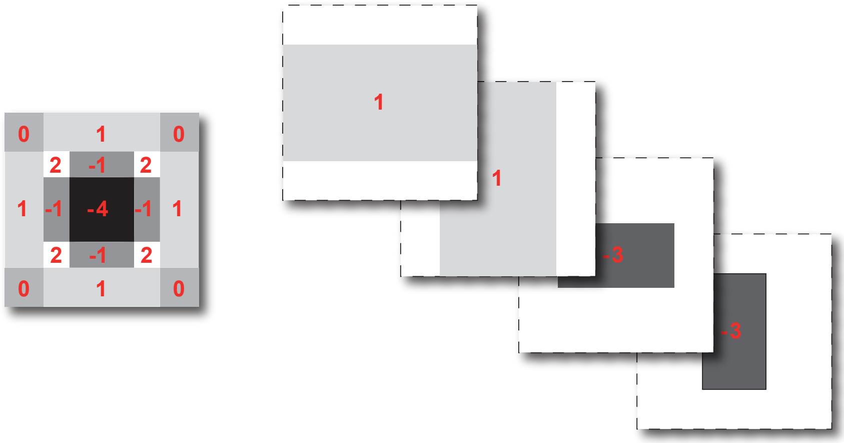


***Figure 3.36:*** *The suSURF kernel (left) can be efficiently decomposed as the superposition of four separate box filters (right).*

in that it achieves maximum signal-to-noise ratio, is the *matched filter* fitted to the wanted signal (*cf.* SIFT marker construction in Section 3.2.2).

Accordingly, the *speeded-up SURF* (suSURF) detector is defined as the matched filter adapted to the SURF MDR markers. That makes it the linear detector that shares its prototypical features with SURF. Due to the apparent symmetry, the suSURF kernel is identical to the underlying MDR marker[4]. Again, it is interesting to observe how the suSURF kernel qualitatively resembles the Difference of Gaussians used in SIFT (*cf.* Figure 3.3 on page 50). Thus, suSURF can simultaneously be considered a linearization of SURF and a non-uniformly quantized version of SIFT.

Figure 3.36 illustrates how the speed-up in suSURF is achieved. Given the block structure of the kernel, the concept of integral images can be exploited to efficiently perform the convolution at arbitrary scales. The suSURF kernel is essentially composed of four subfilters which require one box integration each. In comparison, the filters involved in the derivative computations carried out by SURF comprise a total of eight such integrations. In addition, SURF performs a non-linear combination of the filter outputs.

Specifically, the suSURF response $D_{\text{suSURF}}$ can be calculated as follows:

$$\begin{aligned} D_{\text{suSURF}} = \quad & I_\Sigma(\mathbf{a}_1) - I_\Sigma(\mathbf{a}_2) + I_\Sigma(\mathbf{a}_3) - I_\Sigma(\mathbf{a}_4) \\ & + I_\Sigma(\mathbf{b}_1) - I_\Sigma(\mathbf{b}_2) + I_\Sigma(\mathbf{b}_3) - I_\Sigma(\mathbf{b}_4) \\ & - 3 \cdot \big[ \; I_\Sigma(\mathbf{c}_1) - I_\Sigma(\mathbf{c}_2) + I_\Sigma(\mathbf{c}_3) - I_\Sigma(\mathbf{c}_4) \\ & \qquad + I_\Sigma(\mathbf{d}_1) - I_\Sigma(\mathbf{d}_2) + I_\Sigma(\mathbf{d}_3) - I_\Sigma(\mathbf{d}_4) \; \big], \end{aligned} \tag{3.21}$$

[4] Either version of the MDR marker can be used since their responses to a desired feature are simply the inverse of each other, and the extremum search considers both maxima and minima alike.

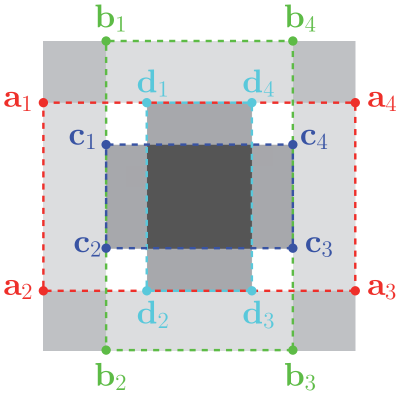


***Figure 3.37:*** *The coordinates relevant for the computation of the suSURF filter response from an integral image, exploiting the box decomposition from Fig. 3.36*

where $I_\Sigma$ is the integral image initially computed from a given image $I$ through

$$I_\Sigma(x,y) = \sum_{i=1}^{x} \sum_{j=1}^{y} I(i,j), \tag{3.22}$$

and the $\mathbf{a}_1$ through $\mathbf{d}_4$ are the scale-specific coordinates marked in Figure 3.37.

One can see that in (3.21) only 15 additions/subtractions and one multiplication per pixel and scale are necessary to compute the suSURF detector response. In comparison, SURF requires a total of 30 additions/subtractions and 5 multiplications. An according speed-up over SURF can thus be expected in the suSURF detector response computation. Moreover, any state-of-the-art implementation for fast convolution, either in software or hardware, can be used to further accelerate the suSURF detection process. Another favorable property of suSURF is that it seamlessly integrates with every DoH-based feature extraction algorithm. Other proposed extensions and advancements acting, *e.g.*, on the descriptor design can hence be combined with the suSURF detector.

### Performance Analysis

With the linearization of the detection operator presented so far, a speed advantage is achievable in theory. Quantitatively, this is verified by the results given in Table 3.4 where the actual detection times of suSURF and SURF are compared for OpenCV implementations on a 3 GHz desktop PC. Different test image sets from [MTS+05] are used in the comparison. Each set comprises six images of size $800 \times 640$ pixels, and the table displays average results over each set. The detection thresholds of SURF and suSURF are set such that the number of detected features is just about equal for both algorithms. It can be seen that the suSURF approach is faster, both for feature rich images (*e.g.*, from the *Graffiti* set) and less textured ones. In terms of detection time per feature, an average reduction by $31.6\%$ is achieved.

It remains to be shown that suSURF not only detects faster but also meaningful features. Figure 3.38 compares the results of both detectors qualitatively when applied to a natural

| image set | algo. | avg. number of features | average detection time | avg. detection time per feat. |
|---|---|---|---|---|
| *Graffiti* | SURF | 2129.5 | 1.6716 s | 784.97 µs |
| | suSURF | 1951.2 | 1.0389 s | 532.43 µs |
| *Boat* | SURF | 1338.0 | 1.1570 s | 864.70 µs |
| | suSURF | 1373.7 | 0.8577 s | 624.40 µs |
| *Wall* | SURF | 1546.3 | 1.3833 s | 894.55 µs |
| | suSURF | 1401.5 | 0. 8170 s | 582.95 µs |
| | | | **overall average:** | **848.07 µs**<br>**579.93 µs** |

***Table 3.4:*** *Detection times by SURF and suSURF on different test images from [MTS$^+$05].*

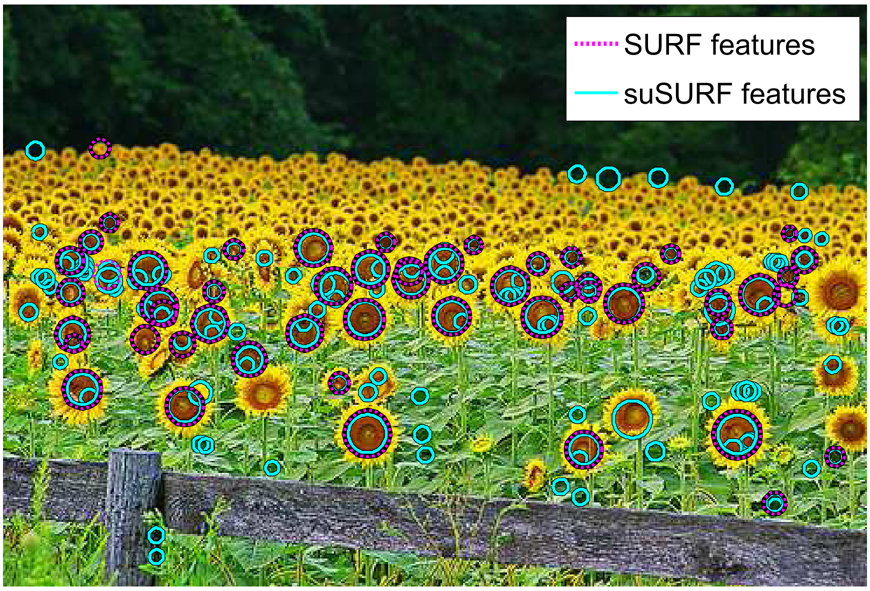


***Figure 3.38:*** *The suSURF detector responds to the same type of image structures as SURF.*
(Test image from flickr.com/janinerussell)

image. Apparently, suSURF and standard SURF react to the same class of features, many of which coincide. The fact that the extracted suSURF features are in parts identical to those detected by SURF is advantageous in that it allows the combination of suSURF and SURF features in one application, *e.g.*, in existing databases.

A quantitative evaluation is presented in Figure 3.39, again using test images, and the evaluation methodology, from [MTS+05]. Here, the so called *repeatability score* is evaluated. This measure gives the ratio of detected features which are successfully recovered in the image after applying a known distortion. In the *Graffiti* and *Wall* datasets, perspective distortion is parametrized by the viewing angle relative to the reference image. Other tested (yet not absolutely parametrized) effects are image blur, zoom and rotation, as well as lighting and JPEG compression. For all except the last two effects two datasets are available, differing in homogeneity/texturedness of the depicted scene.

Obviously, suSURF can compete with standard SURF, as well as with SIFT. In some categories, suSURF even outperforms SIFT and SURF, *e.g.*, in the presence of image blur (*Bike* and *Tree*) or JPEG compression artifacts (*UBC*). It is interesting to note that suSURF generally performs better on scenes that contain homogeneous regions with distinctive boundaries (the left column in Figure 3.39), as opposed to repetitively textured scenes (right column).

Experiments on other datasets have shown that suSURF tends to detect features that are located along straight edges and thus poorly localized (see also Section 3.3). This is a problem that suSURF shares with SIFT. Hence, similar counter measures can be taken, *e.g.*, applying the edge response elimination proposed in [Low04] or evaluating the Harris corner measure [HS88], only for already detected features.

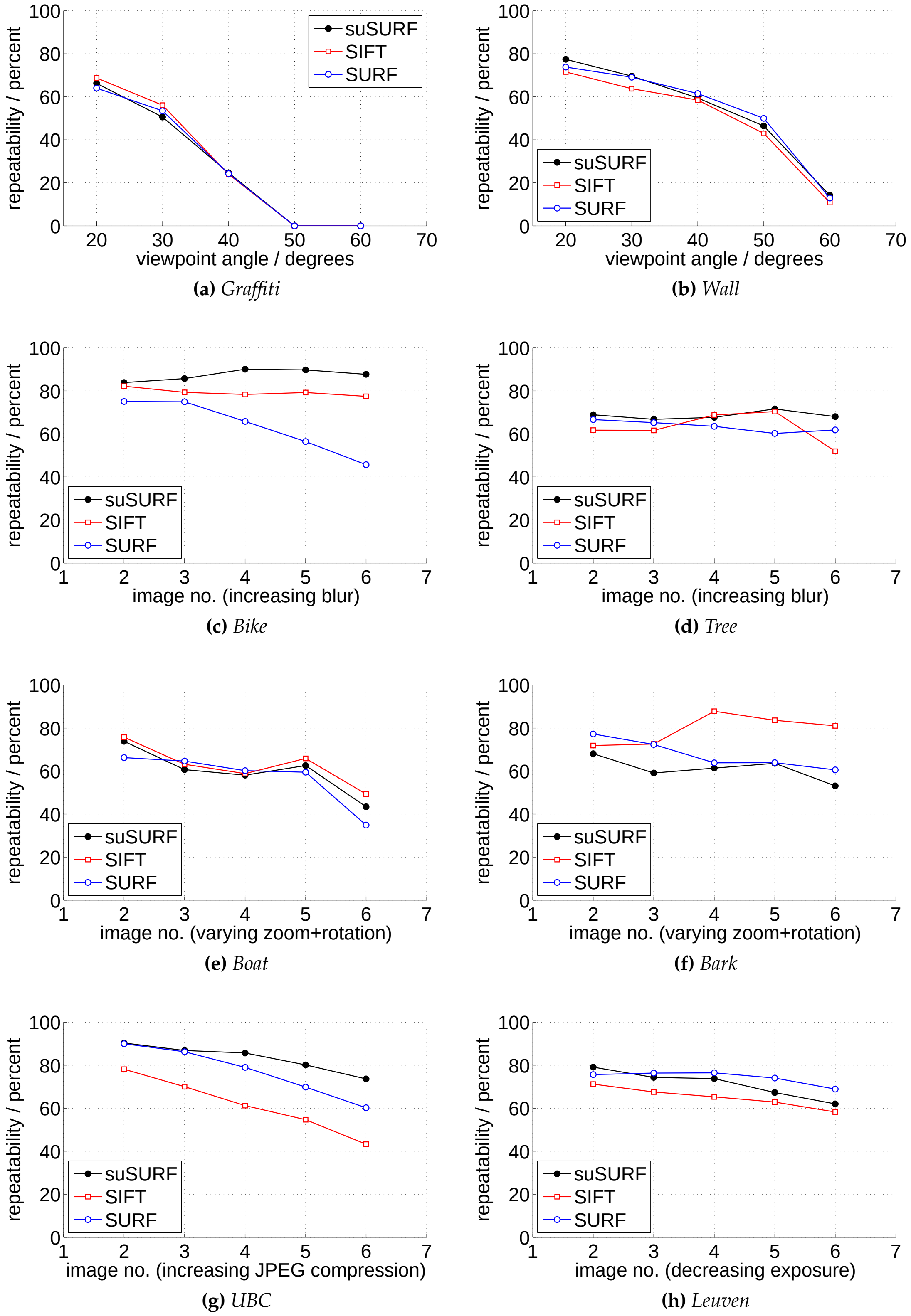
### 3.4.3 Affine-invariant Speeded-Up SURF (AsuSURF)

The suSURF detector presented in the previous section distinguishes itself from SURF by its reduced complexity. At the same time, its detection performance can keep up with SURF. With AsuSURF, presented next, the opposite direction is explored, trading off, to some degree, suSURF's low complexity against a gain in robustness. More specifically, affine invariance will be achieved by incorporating the suSURF detector into the ASIFT framework [YM09, MY09].

#### Review of ASIFT

As pointed out in Section 3.4.1, ASIFT simulates a variety of affine transformations in order to find the closest agreement between parts of two images. An affine transformation maps a 2D image $I(\mathbf{x})$ to $I(A(\mathbf{x}-\mathbf{x}_0))$. The matrix $A$ describing the non-translational parts of the transform can be partitioned using singular value decomposition:

$$A = U\Sigma V^\top = \lambda \underbrace{\begin{bmatrix}\cos\psi & -\sin\psi \\ \sin\psi & \cos\psi\end{bmatrix}}_{U} \underbrace{\begin{bmatrix} t & 0 \\ 0 & 1\end{bmatrix}}_{\frac{1}{\lambda}\Sigma} \underbrace{\begin{bmatrix}\cos\phi & -\sin\phi \\ \sin\phi & \cos\phi\end{bmatrix}}_{V^\top} \tag{3.23}$$



**(a)** *Graffiti*

**(b)** *Wall*

**(c)** *Bike*

**(d)** *Tree*

**(e)** *Boat*

**(f)** *Bark*

**(g)** *UBC*

**(h)** *Leuven*

***Figure 3.39:*** *Repeatability scores for suSURF based on image datasets from [MTS⁺05] in comparison to SIFT and SURF (at a similar number of detected features).*

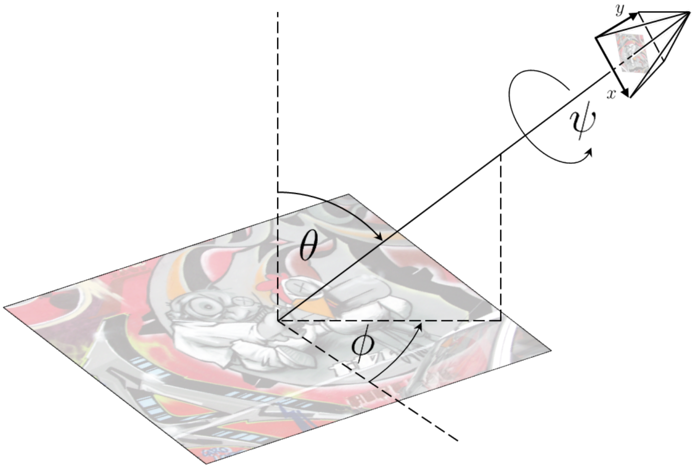


***Figure 3.40:*** *Camera model used in the ASIFT framework, specified by azimuth, elevation and roll angles $\phi$, $\theta$ and $\psi$ (in this order) of a camera performing parallel projection. The induced mapping from the viewed plane to the image plane is affine (see Figure 3.41).*

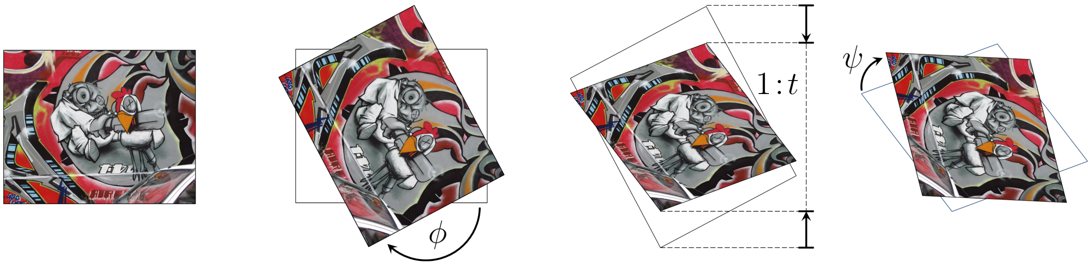


***Figure 3.41:*** *Image formation under the affine camera model depicted in Figure 3.40.*

Interpreting the individual transformation steps from right to left, $V^\top$ represents the rotation of the camera by an azimuth angle $\phi$, the diagonal matrix $\frac{1}{\lambda}\Sigma$ the vertical compression by factor $t$ due to the camera elevation $\theta$, and $U$ a camera roll by $\psi$. Finally, the image is uniformly scaled with factor $\lambda$ proportional to the camera distance. The geometric relationships and their effects on an image are illustrated in Figures 3.40 and 3.41, respectively. In keeping with [YM09], the compression factor $t$ is referred to a *tilt*. It is related to the elevation angle through $t = \frac{1}{\cos\theta}$.

SIFT, being scale and rotation invariant, can cope with two of the partial transformations in (3.23): the rescaling by $\lambda$ and the in-plane rotation induced by $U$. SIFT can also compensate for any translation $\mathbf{x}_0$. The two remaining affine parameters are the tilt $t$, being a function of $\theta$, and the azimuth angle $\phi$. The approach of ASIFT is to simulate the effects of these two parameters over a sufficiently wide range, so as to allow the successful matching between affinely warped images. Figure 3.42 illustrates how azimuth $\phi$ and elevation $\theta$ are sampled. ASIFT uses elevations $\theta_i$ such that the corresponding tilts $t_i$ follow a geometric

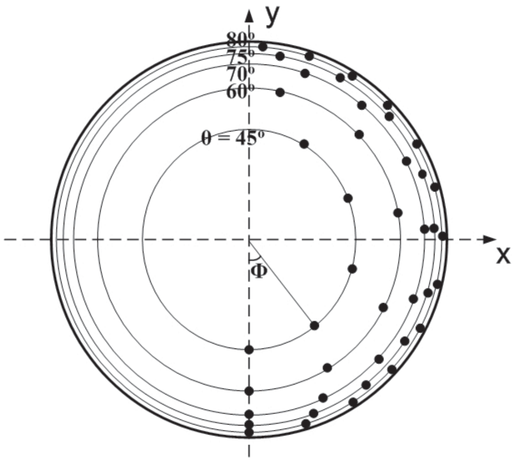


***Figure 3.42:*** *Azimuth and elevation angles sampled by ASIFT according to Eq.* (3.24) *and* (3.25). (Image reproduced from [MY09])

series, and equally spaced azimuths $\phi_{ij}$:

$$\theta_i = \arccos \frac{1}{t_i} \qquad \text{with} \quad t_i = (\sqrt{2})^i, \quad i \in \{0, \ldots, n\} \tag{3.24}$$

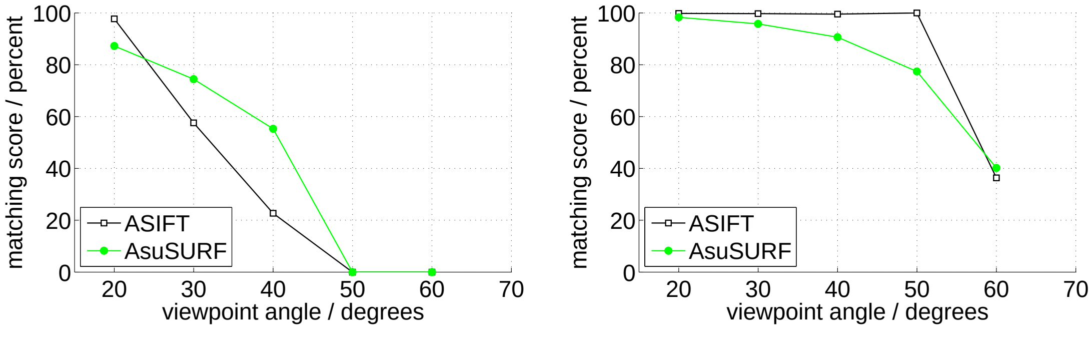
$$\phi_{ij} = \frac{\pi}{10} \cdot \frac{j}{t_i} \qquad \text{with} \quad j \in \{0, \ldots, \lfloor 10\, t_i \rfloor\}, \tag{3.25}$$

where $n = 5$ is suggested as a typical value in [YM09].

Since an image is virtually multiplied in the process (32 variations for $n = 5$, see Figure 3.42), the number of detected features increases accordingly. Given an image pair, the combinatorial possibilities for feature matches hence explodes, roughly surging by a factor 1000 when $n = 5$ is chosen[5]. To counter this, Yu and Morel propose a two-resolution scheme where ASIFT is first applied on decimated versions of the images followed by feature matching. In full resolution, ASIFT then no longer simulates all affine transformations, but only those which led to a significant number of matches in the first pass. More specifically, the $M$ best matching variations are used. The parameter $M$ can be regarded as a presumption about the geometry of the depicted scene. Consider the case of an urban environment with buildings and other man-made structures. It can be expected to find many planar surfaces in this scene each of which maps from one image to another through a certain (approximately affine) transformation. Features on such a plane will be properly matched when its particular inter-image transformation is simulated. So, there is one optimal transformation for every plane in the scene[6]. Yu and Morel suggest a default value of $M = 6$. Strictly speaking, this "pre-matching" step makes ASIFT more than just a feature detection algorithm. ASIFT inextricably combines feature detection and matching and is thus always applied to image pairs.

[5] With $N$ the number of features detected in an average image, there are $N^2$ possible matching pairs in the non-ASIFT case and $(32N)^2 = 1024N^2$ for ASIFT.

[6] More precisely, there is one ideal transformation for every vanishing line, *i.e.*, parallel planes share the same optimal transformation.



**(a)** *Graffiti*

**(b)** *Wall*

***Figure 3.43:*** *Matching scores for AsuSURF in comparison with ASIFT for those datasets from [MTS+05] with viewpoint changes.*

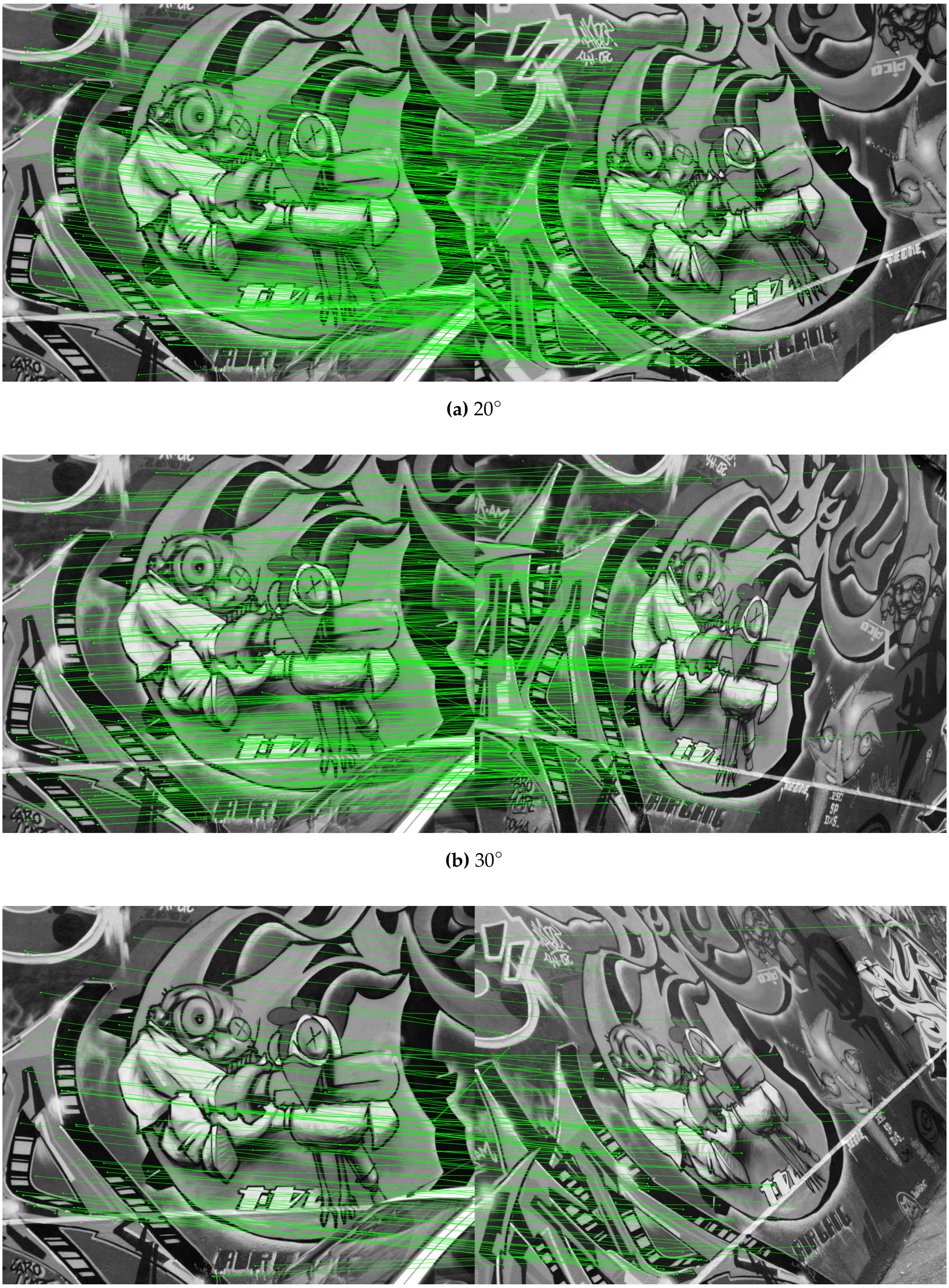
### Adaptation and Performance Analysis

For AsuSURF, the simulative framework for affine transformations proposed in [YM09, MY09] and reviewed in Section 3.4.3 is fully adopted. Instead of the SIFT dectector, however, the suSURF detector from Section 3.4.2 is used. This will reduce the computational burden without severe impacts on performance, as is confirmed by the following experimental results.

AsuSURF performs a combination of feature detection and matching and can thus not be directly compared to pure detection algorithms such as SIFT and SURF. In the following, the comparison is made between AsuSURF and ASIFT. The performance also depends on the used descriptor and the matching strategy. For the presented experiments, AsuSURF is equipped with the SURF descriptor and ASIFT uses the SIFT descriptor. In both cases, the number of simulated tilts is set to $n = 5$, and the $M = 14$ top ranking simulations are promoted from the low resolution stage (downsampling factor 1:3) to full resolution. Figure 3.43 shows the achieved *matching scores*, *i.e.*, the percentage of correctly determined feature pairs with respect to given ground truth transformations. Since the purpose of affine invariance is to defy perspective distortions between views, only the specific datasets from [MTS+05] have been selected which comprise varying viewpoints.

AsuSURF is clearly inferior in case of the *Wall* dataset which contains fine and highly repetitive structures. In Figure 3.45, it can be observed how the number of correctly matched features comes down with increasing viewpoint difference and, concurrently, the ratio of spurious matches goes up. For the *Graffiti* dataset, it is interesting to see that AsuSURF can outperform ASIFT at intermediate viewpoint angles. The feature matches are displayed in Figure 3.44. This is in accordance with the experimental finding from Section 3.4.2 that suSURF generally performs best on scenes with homogeneous regions and distinctive boundaries.

On a 2.6 GHz desktop PC, the average runtime per image pair was 19.954 s for AsuSURF and 33.784 s for ASIFT, given an image resolution of $800 \times 640$ pixels.

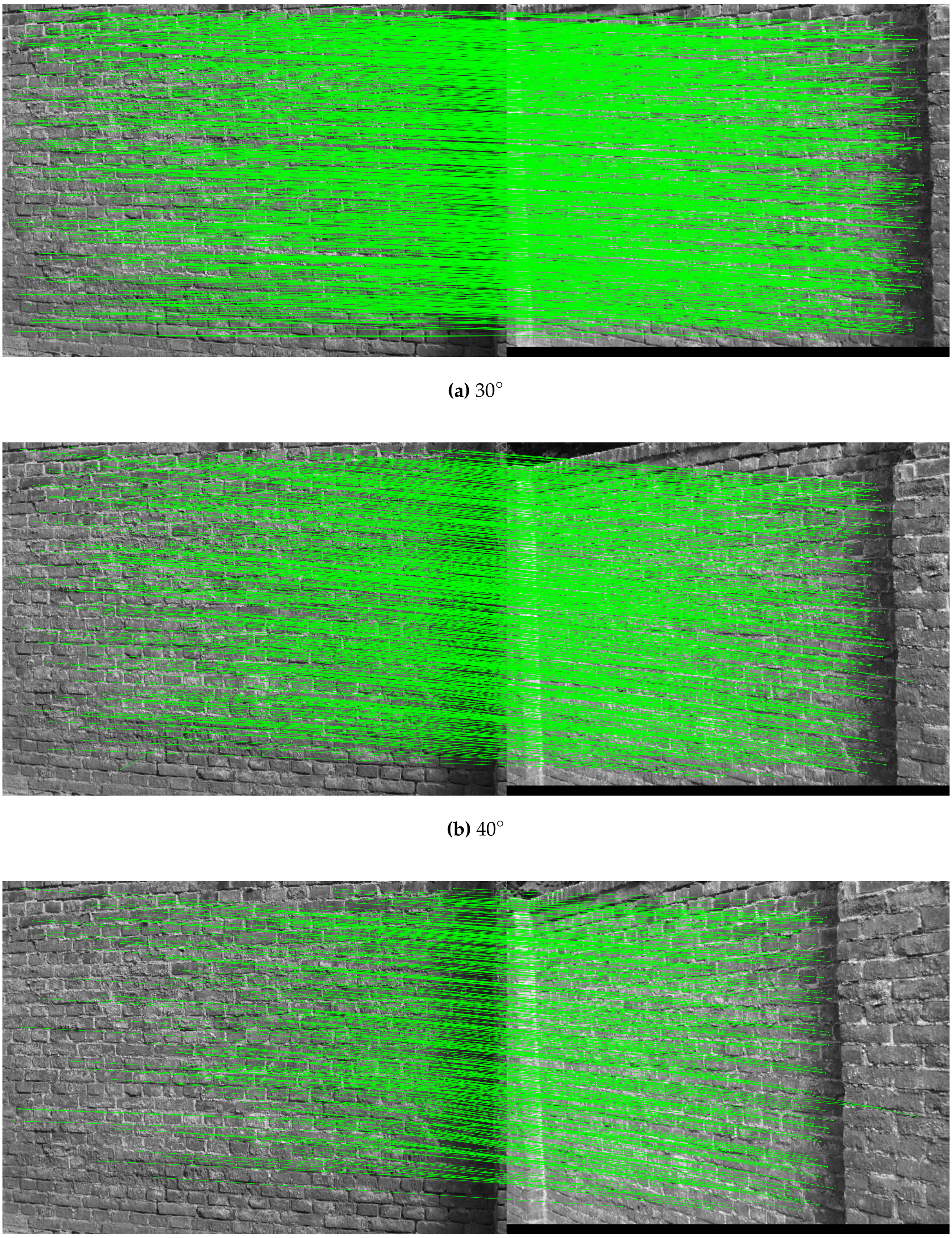


**(a)** 20°

**(b)** 30°

**(c)** 40°

***Figure 3.44:*** *Matching AsuSURF features in the* Graffiti *set for selected viewing angles*



**(a)** 30°

**(b)** 40°

**(c)** 50°

***Figure 3.45:*** *Matching AsuSURF features in the* Wall *set for selected viewing angles*

## 3.5 Concluding Remarks

In this chapter, a wide range of aspects related to spatial image and video analysis have been treated. The common theme was multiscale feature detection due to the prevalent use of this concept in a variety of applications. The specific contributions are a new paradigm for the design of visual markers, a novel accuracy assessment framework generally applicable to scale-space based features, and two accelerated feature detectors.

The visual marker design presented in this chapter distinguishes itself from previous approaches in that the markers are adapted to existing state-of-the-art feature detectors. The Maximum Detector Response (MDR) property guarantees optimal detectability by the specific algorithm the marker has been designed for. At the same time, the need for a separate marker detection algorithm is obviated.

With the feature accuracy framework, also proposed in this chapter, image features detected with scale-space based algorithms can be judged according to their localization fidelity. This is valuable information for subsequent processing tasks, such as triangulation or the computation of epipolar geometries and homographies.

The two feature detectors introduced in this chapter, namely suSURF and AsuSURF, are derived from the MDR marker principle. While the first simplifies the established SURF by linearizing its detection operator, the latter adds affine-invariance to enable successful feature matching even for extreme viewpoint differences. Both detectors are significantly faster than the algorithms from the literature that they are based on, but provide comparable performance nonetheless.

# 4 Conclusion and Outlook

Video data is omnipresent today, and the trend is towards more powerful, cheaper, and more available capture devices. The impact of video can thus be expected to increase even more. This brings about many challenges but also the opportunity for novel applications that improve our everyday lives. This is especially true for multiview video data where more than one shot of a particular scene is available. In order to make use of the video material out there methods to analyze it are necessary. In this thesis the two main components of video data have been looked into. Specifically, both the temporal and the spatial dimension of multiview video data has been scrutinized.

Synchronization is necessary to make use of multiple videos of a scene in a reasonable way. Only when video sequences are temporally aligned, sensible information can be extracted. Depth measurements for example can only be taken where corresponding image information is available. Unless the scene is entirely static, which is uncommon in the world of "moving pictures", slight misalignments in time can lead to severe biases in the retrieved depth values. For most applications frame-accurate synchronization is therefore a minimal requirement.

Spatial video analysis which translates to still image analysis once the temporal dependencies have been resolved, aims at extracting information about the geometric setup of the scene and the cameras. This involves the inversion of the projection performed by the cameras during acquisition. At this point, the *multi-view* aspect of the data comes into effect which enables triangulation in the first place. Especially for wide-baseline and sparse reconstruction[1] scenarios, local image features are an indispensable tool. An important property of local features is their distinctiveness which enables their reliable detection and identification. One specific requirement is the invariance against distortions of any kind, photometric, geometric, etc. A key property indispensable in almost any application is the scale-invariance of local image features. Multi-scale detectors meet this basic requirement. More sophisticated algorithms can also deal with more complex geometric image distortions such as affine transformations.

In the context of low-performance devices, another important property of feature detectors is their computational complexity. For mobile devices in particular, processing time is an essential factor, especially in real-time video processing, but also the induced battery drain. In the light of these requirements, given the increasing impact of smartphones and other mobile devices, efficient feature detectors are thus necessary.

The aspiration of this thesis is to make contributions to the areas mentioned above. Despite the broad variety of topics related to spatio-temporal multiview video analysis, several considerable improvements and proper innovations have been acquired.

Regarding the temporal dimension, a novel, full-fledged video synchronization algorithm has been proposed in this thesis. Based on the bitrate demand of the encoded video,

[1] as opposed to dense reconstruction which determines the depth of every pixel

a characteristic activity profile is derived which allows for accurate temporal alignment. With the novel and powerful correlation method ConCor, also proposed in this thesis, the synchronization process becomes very robust, withstanding various parasitic effects. Occlusions, camera motion and other inconsistencies can thus be compensated for to a large degree. Moreover, the bitrate-based synchronization approach is unaffected by the videos' resolutions, photometric properties and especially viewing directions. This makes it an extremely versatile, fully automatic algorithm that operates without the need for user intervention. In this respect, the presented video synchronization algorithm is unique.

As to the spatial analysis, contributions in different areas have been made. For one thing, a complete framework to assess the location accuracy of multiscale image features has been developed. This framework is generic and has been successfully applied to the established SIFT and SURF detectors. The obtained accuracy estimates can be exploited in a variety of computer vision algorithms to improve the results. This has been demonstrated in this thesis and in other authors' work.

Another contribution in the area of multiscale image features are the visual markers optimized for given feature detectors. The proposed markers are constructed such as to trigger the maximally possible response and are hence provably optimal to detect. The concept of response maximization is again very general; in this thesis, it has been applied to the prototypical SIFT and SURF.

Furthermore, novel, more efficient feature detection algorithms have been studied. Borrowing concepts from the marker optimization, a low-complexity detector has been developed which yields outstanding performance at significantly reduced complexity. This makes it particularly useful in mobile applications where computational resources are eminently precious. On the basis of this efficient detector, an existing affine-invariant feature detector has been modified, again reducing complexity at steady performance.

In summary, the thesis covers a wide range of topics revolving around the spatio-temporal analysis of multiview video. All presented concepts have been motivated and derived theoretically and then validated experimentally.

Future work drawing on the contributions of this thesis is conceivable in different directions. Regarding the practicability of the proposed bitrate-based video synchronization paradigm, extensive tests in a realistic server/client setup involving several hundreds of videos are necessary. Only then, a meaningful performance assessment becomes possible. The proposed synchronization algorithm can also be adopted to align other kinds of (multimedia) signals. In [ANCS$^+$12], video synchronization is boosted by applying consensus-based cross-correlation to downsampled Pulse-Code Modulated (PCM) audio sequences. Other fields could also benefit from the robustness provided by the proposed approach. Examples are template matching as suggested in [SSE$^+$11] or correlation-based beat detection in audio processing. Beyond the world of multimedia, potential application include the measurement of astronomical Doppler shifts in physics or the study of time series in economics. For many applications, an enhancement of the approach towards subframe or, more generally, subsample accuracy needs to be studied.

The feature uncertainty framework proposed in this thesis can be applied to general scale-space based image features. Furthermore, it can assist the computation of a variety of geometric and epipolar geometric quantities. Further formalizations are necessary to incorporate the covariance estimates into the respective calculations. Frédéric Sur has demonstrated this for homography based 2D image matching and 3D data fusion in [Sur10].

Regarding the visual markers developed in this thesis, a more thorough examination of potential marker discrimination methods is advisable. In the current, descriptor-based approach, it might be necessary to trade off (optimal) detectability against distinctiveness by abandoning the maximum response requirement and use a modified marker pattern. As a last resort, the proposed markers can be combined with existing marker ID systems. This requires of course additional processing steps such as corner localization and ID decoding, which would violate the minimalistic design objectives followed in this thesis.

The feature detectors described in this thesis require further testing under practical conditions and in real applications. First attempts with suSURF features in an existing visual location recognition framework [HSH$^+$12a] have already yielded promising results. An issue that has become evident from these experiments is suSURF's sensitivity to edges mentioned in Section 3.4.2. Suitable counter measures indicated in the same section should be implemented before suSURF and AsuSURF can be successfully deployed in practical environments.

# Bibliography

## Publications by the author

[ANCS$^+$12] A. Al-Nuaimi, B. Cizmeci, F. Schweiger, R. Katz, S. Taifour, E. Steinbach, and M. Fahrmair. ConCor+: Robust and confident video synchronization using consensus-based cross-correlation. In *Proceedings of the IEEE International Workshop on Multimedia Signal Processing (MMSP)*, pages 83–88, Banff, AB, Canada, September 2012. IEEE.

[HSH$^+$12a] R. Huitl, G. Schroth, S. Hilsenbeck, F. Schweiger, and E. Steinbach. TUMindoor: an extensive image and point cloud dataset for visual indoor localization and mapping. In *Proceedings of the IEEE International Conference on Image Processing (ICIP)*, pages 1773–1776, Orlando, FL, USA, September 2012. IEEE.

[HSH$^+$12b] R. Huitl, G. Schroth, S. Hilsenbeck, F. Schweiger, and E. Steinbach. Virtual reference view generation for CBIR-based visual pose estimation. In *Proceedings of the 20th ACM international conference on Multimedia*, pages 993–996, Nara, Japan, October 2012. ACM.

[KBSS12] J. Kammerl, F. Brandi, F. Schweiger, and E. Steinbach. Error-resilient perceptual haptic data communication based on probabilistic receiver state estimation. *Haptics: Perception, Devices, Mobility, and Communication*, pages 227–238, 2012.

[SANH$^+$11] G. Schroth, A. Al-Nuaimi, R. Huitl, F. Schweiger, and E. Steinbach. Rapid image retrieval for mobile location recognition. In *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 2320–2323, Prague, Czech Republic, May 2011. IEEE.

[SBS08] F. Schweiger, I. Bauermann, and E. Steinbach. Joint calibration of a camera triplet and a laser rangefinder. In *Proceedings of the IEEE International Conference on Multimedia and Expo (ICME)*, pages 1201–1204, Hanover, Germany, June 2008. IEEE.

[SES$^+$10] Florian Schweiger, Michael Eichhorn, Georg Schroth, Eckehard Steinbach, Michael Fahrmair, and Wolfgang Kellerer. Method and apparatus for synchronizing video data, November 12 2010. US Patent App. 12/926,383.

[SES$^+$11a] Florian Schweiger, Michael Eichhorn, Georg Schroth, Eckehard Steinbach, and Michael Fahrmair. Method and an apparatus for performing a cross-calculation, November 29 2011. US Patent App. 13/373,763.

[SES$^+$11b] Florian Schweiger, Michael Eichhorn, Georg Schroth, Eckehard Steinbach, Michael Fahrmair, and Wolfgang Kellerer. Method and apparatus for synchronizing video data, May 25 2011. EP Patent 2,326,091.

[SES$^+$12] Florian Schweiger, Michael Eichhorn, Georg Schroth, Eckehard Steinbach, and Michael Fahrmair. Method and apparatus for performing a cross-correlation, May 30 2012. EP Patent 2,458,510.

[SHAA+12] G. Schroth, R. Huitl, M. Abu-Alqumsan, F. Schweiger, and E. Steinbach. Exploiting prior knowledge in mobile visual location recognition. In *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 2357–2360, Nara, Japan, March 2012. IEEE.

[SHH+11] G. Schroth, S. Hilsenbeck, R. Huitl, F. Schweiger, and E. Steinbach. Exploiting text-related features for content-based image retrieval. In *Proceedings of the IEEE International Symposium onMultimedia (ISM)*, pages 77–84, Dana Point, CA, USA, December 2011. IEEE.

[SSAA+12] Eckehard Steinbach, Georg Schroth, Mohammad Abu-Alqumsan, Robert Huitl, Anas Al-Nuaimi, and Florian Schweiger. Visual localization method, February 16 2012. WO Patent WO/2012/019,794.

[SSAN+12] Eckehard Steinbach, Georg Schroth, Anas Al-Nuaimi, Robert Huitl, and Florian Schweiger. Visual localization method, February 15 2012. EP Patent 2,418,588.

[SSE+10] G. Schroth, F. Schweiger, M. Eichhorn, E. Steinbach, M. Fahrmair, and W. Kellerer. Video synchronization using bit rate profiles. In *Proceedings of the IEEE International Conference on Image Processing (ICIP)*, pages 1549–1552, Hong Kong, September 2010. IEEE.

[SSE+11] F. Schweiger, G. Schroth, M. Eichhorn, E. Steinbach, and M. Fahrmair. Consensus-based cross-correlation. In *Proceedings of the 19th ACM international conference on Multimedia*, pages 1289–1292, Scottsdale, AZ, USA, November 2011. ACM.

[SSE+13] F. Schweiger, G. Schroth, M. Eichhorn, A. Al-Nuaimi, B. Cizmeci, M. Fahrmair, and E. Steinbach. Fully automatic and frame-accurate video synchronization using bitrate sequences. *IEEE Transactions on Multimedia*, 15(1):1–14, 2013.

[SSFK09] F. Schweiger, E. Steinbach, M. Fahrmair, and W. Kellerer. CAMP: a framework for cooperation among mobile prosumers. In *Proceedings of the IEEE International Conference on Multimedia and Expo (ICME)*, pages 1776–1779, New York, NY, USA, June 2009. IEEE.

[SZG+09] F. Schweiger, B. Zeisl, P. Georgel, G. Schroth, E. Steinbach, and N. Navab. Maximum detector response markers for SIFT and SURF. In *Proceedings of the International Workshop on Vision, Modeling and Visualization (VMV)*, volume 6, pages 145–154, Brunswick, Germany, November 2009.

[ZGS+09] B. Zeisl, P. Georgel, F. Schweiger, E. Steinbach, and N. Navab. Estimation of location uncertainty for scale invariant feature points. In *Proceedings of the British Machine Vision Conference (BMVC)*, pages 57.1–57.12, London, UK, September 2009. BMVA.

## General publications

[AZL03] Joshua Adams, Peter Zvengrowski, and Philip Laird. Vertex embeddings of regular polytopes. *Expositiones Mathematicae*, 21(4):339 – 353, 2003.

[BBPP10] L. Ballan, G.J. Brostow, J. Puwein, and M. Pollefeys. Unstructured video-based rendering: Interactive exploration of casually captured videos. *ACM Transactions on Graphics*, 29(4):87, 2010.

[BCGVDH01] M.J. Brooks, W. Chojnacki, D. Gawley, and A. Van Den Hengel. What value covariance information in estimating vision parameters? In *Proceedings of the 8th IEEE International Conference on Computer Vision (ICCV)*, volume 1, pages 302–308, Vancouver, BC, Canada, July 2001. IEEE.

[BDBL$^+$11] L. Baglivo, A. Del Bue, M. Lunardelli, F. Setti, V. Murino, and M. De Cecco. A method for asteroids 3d surface reconstruction from close approach distances. *Computer Vision Systems*, pages 21–30, 2011.

[BETVG08] H. Bay, A. Ess, T. Tuytelaars, and L. Van Gool. Speeded-up robust features (surf). *Computer vision and image understanding*, 110(3):346–359, 2008.

[Bha43] A. Bhattacharyya. On a measure of divergence between two statistical populations defined by their probability distributions. *Bull. Calcutta Math. Soc.*, 35:99–109, 1943.

[BK08] G. Bradski and A. Kaehler. *Learning OpenCV: Computer Vision with the OpenCV Library*. O'Reilly Media, Inc., 2008.

[BPCP08] D.N. Brito, F.L.C. Pádua, R.L. Carceroni, and G. Pereira. Synchronizing video cameras with non-overlapping fields of view. In *Proceedings of the XXI Brazilian Symposium on Computer Graphics and Image Processing (SIBGRAPI)*, pages 37–44, Campo Grande, Brazil, October 2008.

[BTVG06] H. Bay, T. Tuytelaars, and L. Van Gool. SURF: Speeded up robust features. pages 404–417, Graz, Austria, May 2006. Springer.

[CI] Yaron Caspi and Michal Irani. Sequence-to-sequence alignment. `www.wisdom.weizmann.ac.il/~vision/VideoAnalysis/Demos/Seq2Seq/`. [Online; accessed: April 2013].

[CI00] Y. Caspi and M. Irani. A step towards sequence-to-sequence alignment. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 682–689, Hilton Head, SC, USA, June 2000.

[CI02] Y. Caspi and M. Irani. Aligning non-overlapping sequences. *International Journal of Computer Vision*, 48(1):39–51, 2002.

[Cox73] Harold Scott Macdonald Coxeter. *Regular polytopes*. New York: Dover, 1973.

[CPSK04] R.L. Carceroni, F.L.C. Pádua, G.A.M.R. Santos, and K.N. Kutulakos. Linear sequence-to-sequence alignment. 2004.

[CSI06] Y. Caspi, D. Simakov, and M. Irani. Feature-based sequence-to-sequence matching. *International Journal of Computer Vision*, 68(1):53–64, 2006.

[CVG08] N. Cornelis and L. Van Gool. Fast scale invariant feature detection and matching on programmable graphics hardware. In *Proceedings of the IEEE*

*Conference on Computer Vision and Pattern Recognition Workshops (CVPRW).*, pages 1–8, Anchorage, AK, USA, June 2008.

[DZL06a] C. Dai, Y. Zheng, and X. Li. Accurate video alignment using phase correlation. *Signal Processing Letters,*, 13(12):737–740, 2006.

[DZL06b] C. Dai, Y. Zheng, and X. Li. Subframe video synchronization via 3d phase correlation. In *Proceedings of the IEEE International Conference on Image Processing (ICIP)*, pages 501–504, Atlanta, GA, USA, October 2006.

[Eva] Christopher Evans. opensurf1 - OpenSURF - Open Source SURF feature extraction library - Google Project Hosting . `http://code.google.com/p/opensurf1/`. [Online; accessed: October 2009].

[FB81] Martin A. Fischler and Robert C. Bolles. Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. *Communications of the ACM*, 24(6):381–395, 1981.

[FG87] W. Förstner and E. Gülch. A fast operator for detection and precise location of distinct points, corners and centres of circular features. In *Proceedings of the ISPRS Intercommission Conference on Fast Processing of Photogrammetric Data*, pages 281–305, Interlaken, Switzerland, June 1987.

[Fia05] M. Fiala. ARTag, a fiducial marker system using digital techniques. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, volume 2, pages 590–596, San Diego, CA, USA, June 2005. IEEE Computer Society.

[FYZ+11] Z. Fang, D. Yang, W. Zhang, H. Chen, and B. Zang. A comprehensive analysis and parallelization of an image retrieval algorithm. In *Proceedings of the IEEE International Symposium on Performance Analysis of Systems and Software (ISPASS)*, pages 154–164, Austin, TX, USA, April 2011. IEEE.

[GGB06] Michael Grabner, Helmut Grabner, and Horst Bischof. Fast approximated SIFT. In *Proceedings of the 7th Asian conference on Computer Vision (ACCV)*, pages 918–927, Hyderabad, India, January 2006. Springer.

[Hes10] R. Hess. An open-source siftlibrary. In *Proceedings of the 17th ACM International conference on Multimedia*, pages 1493–1496, Ningbo, China, October 2010. ACM.

[HJA08] A. Haja, B. Jahne, and S. Abraham. Localization accuracy of region detectors. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1–8, Anchorage, AK, USA, June 2008. IEEE.

[HMS+07] S. Heymann, K. Muller, A. Smolic, B. Frohlich, and T. Wiegand. SIFT implementation and optimization for general-purpose GPU. In *Proceedings of the 15th International Conference in Central Europe on Computer Graphics, Visualization and Computer Vision*, page 144, Plzen-Bory, Czech Republic, January 2007.

[Hor90] B.K.P. Horn. Recovering baseline and orientation from essential matrix. *Journal of the Optical Society of America*, pages 1–10, 1990.

[HS88] C. Harris and M. Stephens. A combined corner and edge detector. In

*Alvey Vision Conference*, volume 15, pages 23.1–23.6, Manchester, UK, August 1988. BMVA.

[HZ00] R. Hartley and A. Zisserman. *Multiple view geometry in computer vision*, volume 2. Cambridge Univ Press, 2000.

[Ira99] M. Irani. Multi-frame optical flow estimation using subspace constraints. In *Proceedings of the International Conference on Computer Vision (ICCV)*, volume 1, pages 626–633, Kerkyra, Greece, September 1999.

[Jan09] Gaetan Janssens. 4d-repository :: Public. `http://4drepository.inrialpes.fr/public/datasets`, December 2009.

[JYB] Jean-Yves Bouguet. Camera calibration toolbox for matlab. `http://www.vision.caltech.edu/bouguetj/calib_doc/`. [Online; accessed: May 2010].

[Kan04] K. Kanatani. Uncertainty modeling and model selection for geometric inference. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 26(10):1307 –1319, oct. 2004.

[Kan05] K. Kanatani. Uncertainty modeling and geometric inference. *Handbook of Geometric Computing*, pages 461–491, 2005.

[KB99] H. Kato and M. Billinghurst. Marker tracking and HMD calibration for a video-based augmented reality conferencing system. In *Proceedings of the 2nd IEEE/ACM International Workshop on Augmented Reality (IWAR)*, pages 85–94, San Francisco, CA, USA, October 1999. IEEE/ACM.

[KBP$^+$00] H. Kato, M. Billinghurst, I. Poupyrev, K. Imamoto, and K. Tachibana. Virtual object manipulation on a table-top AR environment. In *Proceedings of the IEEE/ACM International Symposium on Augmented Reality (ISAR)*, pages 111–119, Munich, Germany, October 2000. IEEE/ACM.

[KK01] Y. Kanazawa and K. Kanatani. Do we really have to consider covariance matrices for image features? In *Proceedings of the 8th IEEE International Conference on Computer Vision (ICCV)*, volume 2, pages 301–306, Vancouver, BC, Canada, July 2001. IEEE.

[Klo72] J. Klotz. Markov chain clustering of births by sex. *Berkeley Symp. on Mathematical Statistics and Probability*, 4:173–185, 1972.

[Klo73] J. Klotz. Statistical inference in bernoulli trials with dependence. *The Annals of Statistics*, 1(2):373–379, 1973.

[LA09] M.I.A. Lourakis and A.A. Argyros. Sba: A software package for generic sparse bundle adjustment. *ACM Transactions on Mathematical Software (TOMS)*, 36(1):2, 2009.

[LF05] V. Lepetit and P. Fua. *Monocular model-based 3D tracking of rigid objects*. Now Publishers Inc, 2005.

[Lin90] T. Lindeberg. Scale-space for discrete signals. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 12(3):234–254, 1990.

[Lin93] T. Lindeberg. *Scale-space theory in computer vision*. Springer, 1993.

[Lin94] T. Lindeberg. Scale-space theory: A basic tool for analyzing structures at different scales. *Journal of applied statistics*, 21(1-2):225–270, 1994.

[Lin98] T. Lindeberg. Feature detection with automatic scale selection. *International journal of computer vision*, 30(2):79–116, 1998.

[Low04] D.G. Lowe. Distinctive image features from scale-invariant keypoints. *International journal of computer vision*, 60(2):91–110, 2004.

[LS08] Y. Liu and Y. Sato. Recovering audio-to-video synchronization by audiovisual correlation analysis. In *Proceedings of the 19th International Conference on Pattern Recognition (ICPR)*, pages 1–4, Tampa, FL, USA, December 2008.

[LY06] C. Lei and Y. H. Yang. Tri-focal tensor-based multiple video synchronization with subframe optimization. *Transactions on Image Processing*, 15(9):2473–2480, September 2006.

[Mah36] Prasanta Chandra Mahalanobis. On the generalised distance in statistics. *Proceedings of the National Institute of Sciences of India*, 2(1):49–55, 1936.

[MC05] É. Marchand and F. Chaumette. Feature tracking for visual servoing purposes. *Robotics and Autonomous Systems*, 52(1):53–70, 2005.

[MCUP04] J. Matas, O. Chum, M. Urban, and T. Pajdla. Robust wide-baseline stereo from maximally stable extremal regions. *Image and Vision Computing*, 22(10):761–767, 2004.

[Mik02] K. Mikolajczyk. *Detection of local features invariant to affines transformations*. PhD thesis, Institut National Polytechnique de Grenoble-INPG, 2002.

[MS04] K. Mikolajczyk and C. Schmid. Scale & affine invariant interest point detectors. *International journal of computer vision*, 60(1):63–86, 2004.

[MTS$^+$05] K. Mikolajczyk, T. Tuytelaars, C. Schmid, A. Zisserman, J. Matas, F. Schaffalitzky, T. Kadir, and L.V. Gool. A comparison of affine region detectors. *International Journal of Computer Vision*, 65(1):43–72, 2005.

[MY09] Jean-Michel Morel and Guoshen Yu. ASIFT: A new framework for fully affine invariant image comparison. *SIAM Journal on Imaging Sciences*, 2(2):438–469, 2009.

[PBVDHC03] D.W. Pooley, M.J. Brooks, A.J. Van Den Hengel, and W. Chojnacki. A voting scheme for estimating the synchrony of moving-camera videos. In *Proceedings of the IEEE International Conference on Image Processing (ICIP)*, Barcelona, Spain, September 2003.

[PCSK10] F.L.C. Pádua, R.L. Carceroni, G.A.M.R. Santos, and K.N. Kutulakos. Linear sequence-to-sequence alignment. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pages 304–320, 2010.

[RD06] Edward Rosten and Tom Drummond. Machine learning for high-speed corner detection. In *Proceedings of the 9th European conference on Computer Vision (ECCV)*, pages 430–443, Graz, Austria, May 2006.

[RGSSM03] C. Rao, A. Gritai, M. Shah, and T. Syeda-Mahmood. View-invariant alignment and matching of video sequences. In *Proceedings of the International*

*Conference on Computer Vision (ICCV)*, pages 939–945, Nice, France, October 2003.

[RH06] K. Raguse and C. Heipke. Photogrammetric synchronization of image sequences. In *Proceedings of the ISPRS Commission V Symposium on Image Engineering and Vision Metrology*, pages 254–259, Dresden, Germany, September 2006.

[RRKB11] E. Rublee, V. Rabaud, K. Konolige, and G. Bradski. ORB: an efficient alternative to SIFT or SURF. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, pages 2564–2571, Barcelona, Spain, November 2011. IEEE.

[RZ96] Ian D. Reid and Andrew Zisserman. Goal-directed video metrology. In *Proceedings of the 4th European Conference on Computer Vision (ECCV)*, pages 647–658, Cambridge, UK, April 1996. Springer.

[SB06] L. Sigal and M.J. Black. Humaneva: Synchronized video and motion capture dataset for evaluation of articulated human motion. *Brown Univertsity TR*, 120, 2006.

[SBB10] L. Sigal, A.O. Balan, and M.J. Black. Humaneva: Synchronized video and motion capture dataset and baseline algorithm for evaluation of articulated human motion. *International Journal of Computer Vision*, 87(1):4–27, 2010.

[SBWS10] P. Shrestha, M. Barbieri, H. Weda, and D. Sekulovski. Synchronization of multiple camera videos using audio-visual features. *on Multimedia*, 12(1):79–92, January 2010.

[SCI02] E. Shechtman, Y. Caspi, and M. Irani. Increasing space-time resolution in video. *Lecture Notes in Computer Science*, 2350:753–768, 2002.

[SFPG06] S.N. Sinha, J.M. Frahm, M. Pollefeys, and Y. Genc. GPU-based video feature tracking and matching. In *Proceedings of the Workshop on Edge Computing Using New Commodity Architectures (EDGE)*, volume 278, page 4321, Chapel Hill, NC, USA, May 2006.

[SJ05] R.M. Steele and C. Jaynes. Feature uncertainty arising from covariant image noise. In *Proceeding of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, volume 1, pages 1063–1070, San Diego, CA, USA, June 2005. IEEE.

[SSF$^+$] K. H. Strobl, W. Sepp, S. Fuchs, C. Paredes, M. Smisek, and K. Arbter. DLR CalDe and DLR CalLab. `http://www.robotic.dlr.de/callab/`. [Online; accessed: October 2012].

[Ste99] G.P. Stein. Tracking from multiple view points: Self-calibration of space and time. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Ft. Collins, CO, USA, June 1999.

[Sur10] F. Sur. Robust matching in an uncertain world. In *Proceedings of the 20th International Conference on Pattern Recognition (ICPR)*, pages 2350–2353, Istanbul, Turkey, August 2010. IEEE.

[SW98] Gary J Sullivan and Thomas Wiegand. Rate-distortion optimization for video compression. *Signal Processing Magazine*, 15(6):74–90, 1998.

[Tho04] Joseph John Thomson. Xxiv. on the structure of the atom: an investigation of the stability and periods of oscillation of a number of corpuscles arranged at equal intervals around the circumference of a circle; with application of the results to the theory of atomic structure. *The London, Edinburgh, and Dublin Philosophical Magazine and Journal of Science*, 7(39):237–265, 1904.

[TR03] P. Tresadern and I. Reid. Synchronizing image sequences of non-rigid objects. In *Proceedings of the British Machine Vision Conference*, volume 2, pages 629–638, Norwich, UK, September 2003.

[Tur60] G.L. Turin. An introduction to matched filters. *IRE Transactions on Information Theory*, 6(3):311–329, 1960.

[TVG04] T. Tuytelaars and L. Van Gool. Synchronizing video sequences. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, volume 1, pages 762–768, Washington, DC, USA, June 2004.

[UI06] Y. Ukrainitz and M. Irani. Aligning sequences and actions by maximizing space-time correlations. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 538–550, Graz, Austria, May 2006.

[UOD06] M. Ushizaki, T. Okatani, and K. Deguchi. Video synchronization based on co-occurrence of appearance changes in video sequences. In *Proceedings of the International Conference on Pattern Recognition*, volume 3, pages 71–74, Hong Kong, August 2006.

[Ved] A. Vedaldi. An open implementation of the SIFT detector and descriptor. Technical report, 070012, UCLA CSD, 2007.2.

[Vid] VideoLAN. x264, the best H.264/AVC encoder. `www.videolan.org/developers/x264.html`. [Online; accessed: September 2011].

[VW05] S. Velipasalar and W. Wolf. Frame-level temporal calibration of video sequences from unsynchronized cameras by using projective invariants. In *Proceedings of the IEEE International Conference on Advanced Video and Signal Based Surveillance (AVSS)*, pages 462–467, Como, Italy, September 2005. IEEE.

[WCL$^+$08] C. Wu, B. Clipp, X. Li, J.M. Frahm, and M. Pollefeys. 3D model matching with viewpoint-invariant patches (VIP). In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1–8, Anchorage, AK, USA, June 2008. IEEE.

[WHK06] Daniel Wedge, Du Huynh, and Peter Kovesi. Motion guided video sequence synchronization. In *Proceedings of the Asian Conference on Computer Vision (ACCV)*, pages 832–841, Hyderabad, India, January 2006.

[Wie12] M. Wiedemann. Unterscheidbare Maximum Detector Response Marker für SURF. Master's thesis, Technische Universität München, 2012.

[WKH05] D. Wedge, P. Kovesi, and D. Huynh. Trajectory based video sequence synchronization. In *Proceedings of the Digital Image Computing on Techniques and Applications (DICA)*, page 13, Cairns, QLD, Australia, December 2005. IEEE Computer Society.

[WLB05] A. Whitehead, R. Laganiere, and P. Bose. Temporal synchronization of video sequences in theory and in practice. In *Proceedings of the 7th IEEE Workshop on Applications of Computer Vision / IEEE Workshop on Motion and Video Computing (WACV/MOTION)*, volume 2, pages 132–137, Breckenridge, CO, USA, January 2005.

[WLS08] D. Wagner, T. Langlotz, and D. Schmalstieg. Robust and unobtrusive marker tracking on mobile phones. In *Proceedings of the 7th IEEE/ACM International Symposium on Mixed and Augmented Reality (ISMAR)*, pages 121–124, Cambridge, UK, September 2008. IEEE/ACM.

[WRM+08] D. Wagner, G. Reitmayr, A. Mulloni, T. Drummond, and D. Schmalstieg. Pose tracking from natural features on mobile phones. In *Proceedings of the 7th IEEE/ACM International Symposium on Mixed and Augmented Reality (ISMAR)*, pages 125–134, Cambridge, UK, September 2008. IEEE/ACM.

[WS07] D. Wagner and D. Schmalstieg. ARToolKitPlus for pose tracking on mobile devices. In *Proceedings of the 12th Computer Vision Winter Workshop (CVWW)*, pages 139–146, St. Lambrecht, Austria, February 2007.

[WSBL03] T. Wiegand, G.J. Sullivan, G. Bjontegaard, and A. Luthra. Overview of the H. 264/AVC video coding standard. *Transactions on circuits and systems for video technology*, 13(7):560–576, 2003.

[Wu] Changchang Wu. SIFT on GPU (siftgpu). `http://cs.unc.edu/~ccwu/siftgpu`. [Online; accessed: April 2013].

[WYT09] C. Wang, G. Yang, and Y.P. Tan. Reconstructing videos from multiple compressed copies. *Transactions on Circuits and Systems for Video Technology*, 19(9):1342–1351, 2009.

[WZ02] L. Wolf and A. Zomet. Correspondence-free synchronization and reconstruction in a non-rigid scene. In *Proceedings of the Workshop on Vision and Modelling of Dynamic Scenes*, Copenhagen, Denmark, 2002.

[YM09] G. Yu and J.M. Morel. A fully affine invariant image comparison method. In *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 1597–1600, Taipei, Taiwan, April 2009.

[YM11] Guoshen Yu and Jean-Michel Morel. ASIFT: An algorithm for fully affine invariant comparison. *Image Processing On Line*, 2011. `http://dx.doi.org/10.5201/ipol.2011.my-asift`.

[YP04] J. Yan and M. Pollefeys. Video synchronization via space-time interest point distribution. In *Advanced Concepts for Intelligent Vision Systems*, pages 501–504, Brussels, Belgium, September 2004.

[Zei09] B. Zeisl. Estimation and exploitation of localization uncertainty for scale invariant feature points. Master's thesis, Technische Universität München, 2009.

[ZFN02] X. Zhang, S. Fronz, and N. Navab. Visual marker detection and decoding in AR systems: A comparative study. In *Proceedings of the 1st IEEE International Symposium on Mixed and Augmented Reality (ISMAR)*, pages 97–106, Darmstadt, Germany, September 2002. IEEE Computer Society.

[Zha99] Z. Zhang. Flexible camera calibration by viewing a plane from unknown orientations. In *Proceedings of the 7th IEEE International Conference on Computer Vision (ICCV)*, volume 1, pages 666–673, Kerkyra, Greece, September 1999. IEEE.

[Zha00] Z. Zhang. A flexible new technique for camera calibration. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(11):1330–1334, 2000.

[Zuc03] S. Zucker. Cross-correlation and maximum likelihood analysis: a new approach to combine cross-correlation functions. *Monthly Notices of the Royal Astronomical Society*, 342(4):1291–1298, July 2003.