Master's Thesis

Online Object Recognition using MSER Tracking

Hayko Riemenschneider

Graz University of Technology
Erzherzog-Johann-Universität

at the

Institute for
Computer Graphics and Vision

Accessor and supervisor:
Vert. Prof. Dipl.-Ing. Dr. techn. Horst Bischof

Graz, January 2008

Abstract

This thesis presents a robust online learning and recognition system. The basic idea is to exploit information from tracking an object during the recognition and/or learning stage to obtain increased robustness and better recognition results. Object tracking by means of an extended MSER tracker is utilized to extract local features and construct their trajectories. Compact object representations are formed by summarizing the trajectories to corresponding *frontal MSERs*. All steps are performed online including the MSER extraction, tracking, summarization, SIFT description as well as learning and recognition based on a vocabulary tree.

The online learning by tracking approach is evaluated on realistic video sequences which prove the increased performance for robust online recognition. The whole system runs at a frame rate of 9 fps on a standard PC.

Kurzfassung

Diese Arbeit präsentiert ein robustes online Lern- und Erkennungssystem. Die Grundidee basiert darauf Informationen durch Verfolgen eines Objekts zu nutzen, um während der Erkennungsphase bzw. Lernphase die Robustheit und Erkennungsergebnisse zu steigern. Objektverfolgung wird durch einen erweiterten MSER Tracker verwendet, um lokale Merkmale zu extrahieren und ihre Trajektorien zu konstruieren. Kompakte Objektrepräsentationen werden durch zusammenfassen der Trajektorien zu entsprechend *frontalen MSERs*. All Schritte werden online durchgeführt inklusive der MSER Detektion, Tracking, Zusammenfassung, SIFT Beschreibung sowie das Lernen und Erkennung anhand eines Vocabulary Trees.

Der Ansatz von online Lernen durch Tracking wird in realistischen Videosequenzen evaluiert, welche die Steigerung in Erkennungsrate für robuste online Erkennung belegen. Das gesamte System läuft auf einem Standard-PC mit einer Bildrate von 9 fps.

Acknowledgments

At this place I would like to thank all those who have contributed to this thesis by providing technical and social support.

In particular Vert. Prof. Dipl.-Ing. Dr. techn. Horst Bischof for making this work possible as well as for guiding me throughout this project with much advice and support together with Dipl.-Ing. Dr. techn. Michael Donoser.

I also would like to thank my girlfriend Kathi for all her support and patience as well as all other friends – national and international – for their friendship during last several years.

And finally, I would like to thank my parents for enabling me to pursue my studies.

Graz, in January 2008 Hayko Riemenschneider

Contents

Chapter 1

Introduction

1.1 Motivation

In the past robust learning and recognition required some form of offline processing to cope with the large amount of required training data for the complex learning algorithm [70]. In this thesis a robust system is introduced which handles learning and recognition online with convincing recognition rates. Most object recognition systems ignore the fact that usually a short sequence of the object is available. In the proposed system the training is handled by a tracking algorithm which pursues and learns the visible sides of an object sequence. Wallis and Bülthoff [87] show that a connection between continuous views improves the recognition capabilities in humans. This association between various appearances of the same object is realized by tracking its distinctive features. Tracking features on the object provides a better learning experience by carefully learning the best views and summarizing these to a robust and online retrievable object representation.

1.2 Overview of Approach

This thesis deals with the learning and recognition of 3D objects in an online process without further post-processing. The proposed approach utilizes tracking of an object to retrieve more representative feature information for a better object representation. The following gives an overview of the tracking system in general, the benefit due to tracking and finally, the choice of detection, description and matching methods is briefly discussed.

Tracking an object and its interest points provides comprehensive analysis of its motion and appearances. Typical single-image or multiple-image learning suffers from either only few good features or many unrelated less distinctive features. The tracking of recognized features allows for a greater understanding and learning experience than multiple independent features can provide.

The goal of tracking is to collect information about an object's appearance and to use it for constructing trajectories containing the evolution of the appearance and position of each of the object's individual distinctive features. The wealth of information is summarized to a robust object representation which is sufficiently compact for online learning and recognition.

Maximally Stable Extremal Regions (MSER) [51] are used as interest point detection. Matas et al. proposed this detector for wide-baseline stereo matching and defined extremal regions which posses properties such as affine transformation invariance, multi-scale detection, and a fast enumeration. Evaluations [8, 16, 17, 57, 59] show that MSERs are detecting stable features of arbitrary shape and scale, proving to be one of the most repeatable detectors. Despite the number of detected regions being low, the repeatability is better than with other detectors especially for viewpoint and illumination changes and other distortions.

The learning of various views involves tracking an object by means of MSER tracking [9] which delivers an efficient and accurate matching of MSERs between consecutive frames. The developed *compound MSER tracking* is suitable for tracking multiple MSERs simultaneously. This approach detects and matches multiple smaller MSERs directly. The key ideas are that the bounding boxes of the individually tracked MSERs are combined to a global bounding box which is then used as restriction for the tracking quality and detection of new features on the object. Second, the evaluation of matching stability is a vital part to ensure robust tracking. Once a feature on the object is no longer visible or cannot reliably be matched for several frames, its trajectory should be terminated quickly to maintain fast processing and high quality tracking. Third, in order to optimally learn 3D objects all sides must be visible during tracking and new features must be detected. An efficient method for restricting the MSER detection to features on the object is demonstrated. This guarantees that only previously untracked features are detected and efficiently merged with the previously tracked MSERs.

The tracking is used to construct trajectories to comprise all of the collected motion information and appearances of an object. Previous work [22, 23] used the relative change of a SIFT descriptor to detect a stable minimum where the trajectory contains the best representation. In this thesis we introduce the notion of *frontal MSERs* for optimal representation. The selection of a single interest point is the key idea for summarizing the wealth of information.

These object representations are described by a SIFT descriptor [49] and stored for later retrieval. For this purpose, a vocabulary tree data structure [63] is incorporated to efficiently insert new online learned objects and also to recognize objects during the tracking. A confidence measure is developed to evaluate the recognition score retrieved through the vocabulary tree.

The proposed system thus uses tracking in learning as well as recogni-

tion to compact the appearance of an object. The combination of state-of-the-art detection, tracking, robust summarization, description and retrieval techniques provides the necessary boost to perform all steps in an online process.

1.3 Goal

Tracking provides the basis for creating a better learning process in this thesis. It is used to extract more feature information and connections between motion and the feature appearance to create robust and compact representations of the tracked object within an online process.

First, a method to use efficient tracking to retrieve trajectory information is sought. Second, a method should be defined to summarize the robust feature information, which would enable a minimal effort during learning and recognition and hence, allow online processing. Finally, a measure to evaluate the recognition scores providing an online confidence for the recognition decision should be found.

1.4 Outline

Chapter 2 provides an overview of the existing state-of-the-art interest point detection and description methods. Further adequate methods for storage and retrieval of object representations are compared to fulfill the requirements of an online recognition system.

Chapter 3 a system is proposed which uses tracking to extract additional information for learning and recognition of objects. The various aspects such as tracking, object representation through robust trajectory selection and summarization and the recognition tasks are discussed in detail.

In Chapter 4 the proposed learning and recognition through tracking system is evaluated. Various comparisons against single image recognition as well as the recognition rate over time and with respect to the object's motion show the benefit of tracking for learning.

In Chapter 5 provides conclusions drawn from the experiences and results of this thesis. The benefits and shortcomings are summarized and improvements are proposed to further increase robustness and recognition capabilities along with future extensions.

Chapter 2

Related Work

Any learning and recognition system requires three main parts to fulfill its purpose. First, discriminating locations must be extracted to identify the most interesting and most distinctive regions of an object. Second, the extracted regions of interest must be described in a way that ensures their discriminative information is maintained. Such a representation is designed to allow the regions to be compared and identified against many other region representations. This learning part builds the object representations and stores them for later retrieval. Third, a new unknown object undergoes the first two steps of detection and description and additionally, the representation is matched against all previously learned objects. In this step, the design of the matching and retrieval system are essential to the performance and efficiency.

A further step related to the proposed approach is the involvement of tracking in the learning process. In the following sections each of these parts is discussed in detail concerning its goals and state-of-the-art techniques with the purpose of creating a robust online learning and recognition system.

2.1 Region Extraction

Local features have taken a dominant role when dealing with object recognition – specific or by class, wide baseline matching, scene classification, texture recognition and robot navigation.

The local appearance-based approach for region detection has many benefits over the global approach such as robustness to occlusion, noise, and large illumination changes. The challenge is to detect interest points – together with a support region or interest regions directly – which are invariant to viewpoint changes, numerous and various in the type of image element they localize. This demands a detection process to be invariant to scale changes, in-plane rotation, 3D viewpoint changes and insensitive to illumination changes in a robust and repeatable way. Localization accuracy

is an even more important aspect when geometric calculation is applied for post-processing verification or extraction of shape.

Over the last years, a variety of detection methods have been proposed and adapted to cope with the requirements mentioned above. Research has advanced the detection processes of local features to a state where repeatability and invariance are no longer the main concerns. Tuytelaars and Mikolajczyk [81] claim the real benefit of local features lies elsewhere. It is the advantage of being able to neglect the semantic segmentation step. The concept of local features scattered over foreground objects as well as background delivers an implicit scene representation robust to occlusion. This effect may be used in next level processing to distinguish the important information from background clutter.

The range of feature detectors is split into many categories with various properties and advantages. To select a method for our extraction purposes the following detectors are examined. The Harris [25] and Kanade-Lucas-Tomasi (KLT) [80] detectors find corner points. The Hessian and Difference-Of-Gaussian (DoG) [48, 49] methods detect blobs and ridges. The Maximally Stable Extremal Regions (MSER) [50, 51], intensity-based regions (IBR) [83, 84], edge-based regions (EBR) [82, 84], salient regions (SR) [32, 33, 34] and Principal Curvature-Based Regions (PCBR) [8] detectors provide regions. This list is a small subset of the available detectors and is based on the evaluations over the last years [8, 16, 17, 57, 59].

2.1.1 Detection Goals

The common ground for evaluation is either a comparison based on a manually defined ground truth or the performance within an application. The work by Schmid et al. [74] creates the basis for objective comparison of detection methods by creating evaluation measures for repeatability and information content. Mikolajczyk et al. [59] provide a common framework to allow measuring repeatability and matching score of any given detector whilst undergoing various transformations such as scale change, in-plane rotation, 3D viewpoint changes, image blur, JPEG compression as well as illumination changes. Their work builds the ground truth for future detection algorithms and extensions to affine invariance or non-planar scenes [16, 17, 85] to be evaluated on common aspects and determine their effectiveness.

The main properties of an ideal detector developed through various evaluations [57, 59, 74, 75] consist of the following:

Repeatability is a measure first formulated by Schmid et al. [74] to evaluated the reliability of a detector to find the same image parts in multiple images containing the same scene. The measure defined as a relation between the total number of detections and those common to both images.

According to Tuytelaars and Mikolajczyk [81] repeatability is achieved by either invariance – i.e. to model and adapt to distortions allowing for a detection independent of these transformations – or by robustness which is the opposite approach. A robust detection is less sensitive and allows for slight variations that cannot be modelled well – such as noise, compression artifacts, blur or any unknown photometric deviations. This deteriorates the accuracy but allows for a more repeatable detection.

Distinctiveness defines how the detected regions should be distinguishable due to their high variation in intensity patterns amongst one another. Distinctive regions carry more inherent information allowing better matching to take place.

Locality is part of the initial requirement for the features to have a limited local area they are covering. This is the essential argument to cope with occlusion since local features do not require the entire object of interest to be visible. Further, constraining features to be local allows the approximation of transformation under various viewing changes.

Quantity describes the number of detected regions and poses a desirable property. Without a sufficiently high number the success of the recognition process may be limited while a low number requires less processing time.

Accuracy is necessary in respect to geometric localization, scale selection and registration of a region's shape.

Efficiency provides a comparison for the runtime of a detection process which is a vital component in online applications.

Some of these properties contradict each other and an improvement in one results in a deterioration of the other. Locality and distinctiveness are two of those competitors. For a region to be highly distinctive it requires a large enough size of the image patch to describe. The more local or smaller the region becomes, the more limited is the intensity information and thus the descriptive power.

Further, distinctiveness is part of a balance with invariance and robustness. A high insensitivity during the detection process allows for invariance to viewpoint changes or photometric transformation while relinquishing part of its distinctiveness to achieve this invariance.

In most cases the application directly influences the choice of importance of each property and the desired minimal levels. The final selection is based on all these aspects with special consideration to repeatability, accuracy and efficiency. A high quantity is not a necessary property because efficiency is of more interest.

2.1.2 Detection Methods

In this section a subset of all available state-of-the-art detectors is analyzed based on the aforementioned requirements. Additionally, two further aspects of detection are mentioned dealing with changes in scale and perspective transformations.

Harris Detector

Harris Detector [25] is based on auto-correlation matrix (or second-moment matrix, or structure tensor) and measures the similarity of the image patch when shifted in each direction. The corner measure avoids eigenvalue decomposition and speeds up the detection process. A study by Schmid et al. [74] comparing various interest point detectors determined that the Harris detector is the most repeatable and distinctive against noise and illumination conditions. This is partially supported by later evaluations comparing more detectors [54, 59]. Disadvantages of this detector are the sole dependence on spatial location ignoring the scale and any viewpoint distortions except the obvious rotational invariance.

Hessian Detector

Hessian detector uses the Hessian matrix which is similarly using second-order derivatives of the intensity function. The response measure is calculated via the Hessian determinant combined with non-maxima suppression. The derivatives are also the reason this detector locates blobs and ridges similarly to the Laplacian operator [59]. However, the determinant of the Hessian matrix used for the response is less sensitive to lines, i.e. image structures for which one of the second derivatives only has a small value. Even though the efficiency is only slightly worse compared to the Harris detector [59], the main disadvantage is the limitation to scale since only blobs at the same scale as the Hessian matrix are found. This detection process also only shows rotational invariance.

Scale Adaptation and Laplacian

The inability to deal with feature detections of different scales is addressed in the scale-space theory by Lindeberg [46]. In his work, he shows that the automatic selection of a blob is possible by determining its characteristic scale [47]. This approach is known as multi-scale detection [12, 53] since detection is performed simultaneously at multiple scales of the image or iteratively narrowed down to a final scale [59].

Lowe [48] based his work also on Lindeberg's scale-space to develop a close approximation of the Laplacian scale selection. Lowe shows that the difference of two Gaussian (DoG) smoothing functions with a constant ratio

of their standard deviations provides a good approximation and speeds up the typical Laplacian at similar results [53]. The DoG detector provides a substantial speedup over the Laplacian, however still requires more computation time than the previous version without the scale adaptation. Recent work by Grabner et al. [24] and Bay et al. [3] remedy the efficiency loss and maintain the performance.

Affine Adaptation

The second main challenge of feature detection is the invariance toward perspective distortions. The key idea is to approximate them by an affine transformation. This is possible provided the image patches are locally planar. A transformation by the inverse of the covariance matrix is first used by Lindeberg when describing an affine invariant neighborhood [46]. Mikolajczyk and Schmid [54] show that such an approximation to the perspective transformation group presents nearly the same results yet allows faster calculation. The anisotropic ellipse around the shape of an interest point is normalized to create a circle. The region still changes but in a covariant fashion with transformation [57]. After this adaptation only a residual arbitrary rotation is left which may be addressed by other means such a orientation histograms or localization of extremal points. This affine normalization results, of course, in a higher computational effort. Yet the tradeoff is outweighed by the significant improved recognition performance [58, 59].

Edge-based Detector (EBR)

Edge-based detector (EBR) by Tuytelaars and Van Gool [82, 84] extracts edge geometry by combination of a Harris and Canny detector [5]. The idea is to reduce the 6D search space by constraining it to local invariants. Tuytelaars and Van Gool use edge information around an interest point which is robust to changes in illumination or viewpoint. The EBR detector works well in structured scenes and with edge contours. However, a lack of distinct edges [83] as well as a mediocre computational time [59] are the major drawbacks.

Intensity-based Regions (IBR)

Intensity-based regions (IBR) extend the EBR approach to explore a circular region around intensity extrema [83, 84]. This intensity profile along rays emitted from the center is evaluated according to an invariance function giving a profile of the rate of intensity change. Extrema (usually maxima due to a sudden change in intensity) are selected and connected to an arbitrary shaped directly fit into an ellipse. The detected blob-like structures accurately represent regions – especially of printed material [81]. This process

does not require distinct edges to be present. However, the computational time is not practical for online detection [59].

Salient regions detector (SR)

Salient regions detector (SR) developed by Kadir and Brady [32, 33, 34] is based on entropy distribution to emulate the pre-attentive stage of the human visual system. The central concept is to find rare patches inside an image which have a high information content and thus provide the necessary distinctiveness. The process evaluates candidates based on an image point along with an ellipse described by scale, orientation and ratio of major to minor axes. Candidates are then ranked by their entropy achieving invariance to transformation in geometry and photometric changes [35]. SR provide only a limited number of regions and extract these at at very slow rate due to exhaustive evaluation of each pixel in the image [81]. This complexity is reduced when adopting a local search strategy such as mentioned in [2, 54, 72]. However, this approach is still impractical for online use.

Intensity-based Approximations

Lepetit et al. [44] propose a new interest point detector for their tracking application which is closely trimmed to their classification approach by decision trees. Lepetit et al. detect interest points by evaluating the quantized pixels of a circle at a certain radius. The center of the circle is selected as a stable interest point if diagonally opposing pixels and their neighbors show a minimum difference in intensities. This – along with a simple substraction as approximation of a Laplacian response measure – provides the detection of interest points [42].

This method achieves robust and stable detection to be used for 3D pose estimation. The approximations allow efficient frame-rate processing and additionally, recognition using randomized tree, see Section 2.3.2 for details, creates a promising approach. This detector however still lacks invariance in scale and affine transformation. Rotational invariance is provided through choosing a canonical direction in an orientation histogram. The affine invariance in the system by Lepetit et al. is managed by artificially distorting and learning detected interest regions [44].

Maximally Stable Extremal Regions (MSER)

MSERs were proposed by Matas et al. [50, 51] as a novel interest region detector for wide-baseline stereo matching. Matas et al. define extremal regions as distinguished regions with properties such as affine transformation invariance of the intensity function, multi-scale detection, a measure of stability, covariance to adjacency preserving continuous transformation and an enumeration in O(n log log n) with n the number of pixels in the image.

The detection of these regions is achieved by thresholding the image at every intensity. The resulting binary slices are analyzed for connected neighbors (components) from dark to brighter intensities. The components are termed extremal regions because their boundary pixels have a darker intensity relative to their bright interior. The reverse direction detects dark regions with brighter boundaries (known as `MSER-`). MSERs are an extension to the definition of extremal regions by additionally representing the maximally stable regions. They are selected based on their stability in size along the levels of the intensity function.

One advantage of the MSER detector lies in the inherent scale-invariance since no smoothing or additional steps are required to simultaneously detect coarse and fine structures. Although recent work by Forssén and Lowe [15] show a scale-space approach leads to further scale-invariance and higher repeatability. Second, the detection is invariant to photometric changes. Third, it is also covariant to adjacency preserving transformations due to the inherent hierarchies within the intensity levels. A higher intensity level includes the lower level thresholds. This provides a benefit over other detectors which explicitly need to develop robust countermeasures. Affine adaptation to perspective distortions is however required.

The nature of MSER detection is to find regions which are stable in their size, i.e. do not change when other intensities levels are explored. This delivers most homogenous regions representing structured scenes [19, 59]. MSER thus ignore text if the letters themselves are too small to exist as independent stable homogenous regions.

The evaluations by Mikolajczyk et al. [59] show that MSERs are detecting stable features of arbitrary shape and scale. Despite the number of detected regions being low, the repeatability is better than with other detectors especially for viewpoint changes, slightly less for image blur [57] and their detection efficiency is unbeaten [59].

Principal Curvature-Based Regions (PCBR)

Principal Curvature-Based Regions (PCBR) are based on the principle curvature information extracted from an image. Deng et al. developed this detector as an extension to the typical corner measures based on eigenvalues of the Hessian matrix [8]. They created a method to extract shape and pattern information while achieving robustness against illumination variations and other noise. The implementation by Deng et al. combines a scale-space approach, a morphological closing to remove noise, a stability measure to select regions stable across multiple scales and affine invariance. The principle curvature itself is created from the smallest and largest eigenvalue of the Hessian matrix.

PCBR thus provide a well-designed approach, however the repeatability is less than for other detectors and usually half as repeatable as the MSER

detector [8]. EBR, PCBR along with the MSER are the only detectors which lose information when fitted by an ellipse for affine invariant representation since their arbitrary shapes are dismissed in this process.

2.2 Region Description

Once interest points or regions are detected the next important step is to find an appropriate description best suited for matching and recognition. The description is used to identify the underlying image data in a way to create a repeatable, compact and distinctive representation.

2.2.1 Description Goals

Similarly to region detection, the desired characteristics have been developed through various evaluations [55, 58] to compare the wide range of descriptors available. The following is an overview of the most important aspects of feature description:

Repeatability ensures the invariance or robustness when the underlying image information is affected by noise, affine geometric or photometric transformations. The repeatability is again the most valuable aspect along with the distinctiveness. The two sides of repeatability however are invariance towards any undesirable changes and complete description of information of the interest region. The more distinctive the descriptor is the more information it carries and equally the less invariant it becomes.

Distinctiveness is a measure how much detail a descriptor is able to encode in its representation while not failing its invariance requirements.

Compactness or dimensionality describe the size of the representation and are a crucial part in matching and indexing.

Efficiency evaluations allow to compare detectors based on the computational complexity and execution time.

Additionally, there are other factors involved when creating a representation for a detected interest point including the size of the support region and the matching strategy.

Depending on the level of invariance relating to affine transformation in geometry, intensity function, scale and orientation of the detector providing the support region, the process for description has to be equally able to robustly handle such distortions and other noise. The comprehensive evaluations by Mikolajczyk and Schmid [55, 58] provide comparisons for effects of scale changes, image rotation, blur, JPEG compression, illumination changes as well various region and scene types for feature description.

2.2.2 Description Methods

There is a large variety of descriptors of which a subset is discussed in the following sections. Each of them performs differently depending on the types of image information like individual pixel intensity, color, gradient, texture, edges, etc. The application itself determined the balance of the desired properties such as repeatability, distinctiveness, compactness or efficiency. However, all are equally important to achieve a fast robust online learning and recognition system.

There exists a vast range of description methods in various types such as distribution or region-based descriptors, non-parametric transformations using statistical relations, spatial-frequency descriptors and differential based on local derivatives (jets) and many more [56]. Since efficiency is a vital part we are considering less complex descriptions method which create distinctive representation in minimal time. Thus cross-correlation, steerable filters [20], moment invariants [82], spin images [38], RIFT [39], SIFT [48, 49], and LAFs [65, 66] are discussed and evaluated in detail.

Normalized Cross-Correlation

The simplest and easiest description is no description at all. For normalized cross-correlation the image region is taken as it is and compared to the query region. Cross-correlation measures the similarity between two such pixel intensities and performs well when trying to find correspondences between these two. However, this requires the complete or sub-sampled image regions to be stored and analyzed resulting in impractically high memory and computation needs [81]. Additional problems arise when the image patches are distorted by affine transformations or misaligned.

Scale Invariant Feature Transform (SIFT)

The Scale Invariant Feature Transform (SIFT) by Lowe [48, 49] is a carefully designed combination of a detection and description process. Lowe approximates the blob detection ability of a Laplacian of Gaussian (LoG) through a difference of two Gaussian kernels (DoG). The next steps reuse the image gradient and selected characteristic scale for efficient description of the detected features. The four steps encompassing the complete process from image to feature description are:

1. Scale-space extrema detection.
2. Accurate interest point localization.
3. Orientation assignment.
4. Interest point descriptor.

The interesting steps for feature description are the orientation assignment based on peak selection of gradient orientations. The gradient information is available through the previous steps of extrema detection. For each feature the support region is analyzed based on the direction of its derivatives. Peaks in this 360 degree histogram are canonical orientations which provide invariance to rotation.

At this point the underlying image data for a feature is already transformed relative to its assigned scale, orientation and image location. Thus the last step of feature description is also invariant to these transformations. The descriptor is built from sub-patches each with individual orientation histograms filled by the image gradients. Lowe uses 16 sub-patches in a 4x4 grid with a resolution of 45 degree for the orientation histogram. This creates a 128-dimensional descriptor.

In the evaluation framework by Mikolajczyk and Schmid [55] the original SIFT by Lowe outperforms other descriptors in all experiments except illumination changes where it comes second. In a more recent comparison [58] the SIFT descriptor maintains its good performance, however comes overall second. The best descriptor in this evaluation is the gradient location and orientation histogram (GLOH) which is an extension of the original SIFT with a modified location grid. The changed grid results in 17 bins which creates a more distinctive and repeatable 272-dimensional descriptor at a resolution of 16 orientation bins [58]. The high dimensionality is reduced at more computation cost by a PCA post-processing step.

Spin Images

Spin images were introduced by Johnson and Hebert [28] as data level shapes descriptor. Johnson and Hebert designed them to describe 3D surfaces by a vertex and its surface normal. Two cylindrical coordinates derived from the orientation build a 2D accumulator space. Lazebnik et al. [38, 39, 40, 41] adapt this concept to 2D intensity representation. Their intensity-domain spin images also use two parameters – distance from the center and pixel intensity – to describe the distribution of intensities in a 2D histogram. Each bin of the spin image descriptor is filled according to the frequency distribution of the corresponding distance and intensity. The advantage of this descriptor is obviously inherent rotation-invariance as well as ease of computation. The distinctiveness varies depending on the resolution of the histogram. Lazebnik et al. use a 100-dimensional descriptor with ten bins for each parameter. They additionally compensate some of the noise and resampling artifacts by applying a Gaussian blurring.

Rotation-Invariant SIFT (RIFT)

Rotation-invariant SIFT (RIFT) is a generalization of Lowe's SIFT. Lazebnik et al. designed this descriptor to relieve spin images of their drawbacks – mainly noise and resampling artifacts [39]. RIFTs similarly use gradient information for rotation assignment. However, they inherently encode the orientation relative to the direction of the current position to the center of the analyzed patch. The representation is built from circular bands of the patch. Again, the resolution of the parameters – distance from center or number of concentric rings and orientation angle – determine the dimensionality. Lazebnik et al. use a 32-dimensional representation built from four rings and eight orientation bins [39].

Steerable Filters

Steerable filters by Freeman and Adelson [20] are oriented filters. They allow the calculation of a filter response at different orientations. The design of steerable filters specifically enables the steering and combination of various filters. This means multiple filters can be combined and filter responses can be interpolated for between orientations. According to recent evaluation [58] steerable filters provide the best low-dimensional descriptors.

Moment Invariants

Tuytelaars and Van Gool [82, 83] use a range of different combinations of moment invariants to describe region patches. The regions have undergone transformations for scale and affine-invariance as well as illumination insensitivity. Thus any further description also shows invariance towards these distortions. The moment invariants are calculated over the first-order coordinates or second-order color information. Tuytelaars and Van Gool state that their adaption of moment invariants is robust since information such as shape, color, and intensity is directly characterized. This method is successfully applied in wide-base stereo matching [84] as well as simultaneous segmentation and recognition of objects [14]. Due to their straightforward calculation and robustness, they allow a compact way of describing detected features.

Local Affine Frames (LAF)

Obdržálek and Matas [65, 66] create local affine frames (LAF) to achieve invariance towards scale, illumination and perspective distortions. Their approach approximates an inverse transformation based on the covariance of the shape of detected regions as well as illumination normalization to create frames which are locally invariant and thus can be used to describe local features.

The rotational invariance is achieved through localization of various extreme points in distance, concavities or bitangent lines. Bi-tangents provide a valuable addition since they do not need a complete object to be present to provide accurate directional information [64]. Furthermore, Obdržálek and Matas apply a contour smoothing through a polygon fitting of the shape. This increases the computation complexity of an already high dimensional descriptor.

Although LAFs do not constitute descriptions themselves, the correspondence matching between two LAFs does not require additional description since the frames are aligned and photometrically normalized. The dimensionality is limited by also normalizing to a resolution of 21 by 21 pixels reducing the effects of noise [66]. Obdržálek and Matas use an intensity-normalized distance on these reduced frames to evaluate the matching score which they successfully used in wide-baseline stereo matching [52] and image retrieval [65]. For larger frame sizes, Obdržálek [64] also proposes a discrete cosine transformation with few coefficients which obtains similar discriminative recognition as Lowe's SIFT.

2.3 Region Matching

The previous sections described the two processes of detecting interest points and describing these. This section deals with the indexing and matching of feature descriptions. Special attention is required once the total number of features rises above what simple exhaustive search can handle realistically. A nearest-neighbor search suffices in finding the best matches only for a low number of features due to its quadratic runtime. For larger amounts more complex approaches are required which manage the indexing of higher dimensional feature descriptors efficiently.

2.3.1 Matching Goals

The indexing and matching is a vital part when dealing with large amounts of feature vectors. In the last years this count has steadily risen from a few images, 100.000 keypoints by Lowe [49], the one million image database by Nistér and Stewénius [63] to the most recent work by Philbin et al. [68] up to 1.2 million. The detected features are described and then used to recognize the objects by querying a database. In this section, various matching and indexing mechanism are discussed and evaluated.

Unlike the requirements for region detection and description, the use of indexing and matching methods has a smaller set of commonly desired characteristics. For the goal of creating a robust online learning and recognition system the following requirements are needed:

- Matching the query image features against all stored features.

- Efficient matching of query objects for online processing.
- Fast online insertion and retrieval of new or unknown objects.
- Extensibility to handling large amounts of objects.

The core challenge is to find the minute number of inliers among a vast majority of outliers. Exhaustive search only leads to impractical solutions. Thus the common approach is to rule out outliers quickly and narrow down the search space to the ones most likely to be inliers. Voting schemes are applied to increase the score for each of the potential matches.

2.3.2 Matching Methods

In the following sections, various approaches to dealing with large amounts of feature representation are discussed including the kd-tree [21], k-means clustering for video indexing [79], randomized trees [44], the LAF-tree [67], and hierarchical clustering such as the vocabulary tree [63].

(Best-Bin-First) kd-tree

Beis and Lowe [4] introduced the 'best-bin-first' strategy as an approximate improvement to the well-known kd-tree by Friedman et al. [21]. Kd-trees split the data into two parts based on the median of the data. At each level the k-dimensional data is sorted and divided into roughly same sized subparts. The kd-tree has its limit at ten dimensions after which an exhaustive search is equally efficient [49]. Beis and Lowe achieved a ten-fold speedup for a 20-dimensional search space over exhaustive search and similarly for the plain kd-tree.

The process behind the 'best-bin-first' algorithm is an adapted search ordering enabling the closest neighbors to be searched first. After checking only the first 200 nearest-neighbor candidates [4] the search is stopped and evaluated. With this technique they achieve the remarkable speedup while on average still correctly matching 95%.

Lowe uses this 'best-bin-first' modified kd-tree in his work to find the closest neighbors in his 100.000 descriptor database with high probability. This set of descriptors represents only about 50 images given the typical number of 2000 interest points per image [49]. Philbin et al. make use of the concept in their recent work [68] where a randomized kd-tree is combined with the approximate nearest neighbor method.

Flat K-Means

The work by Sivic and Zisserman [79] (frequently referred to as 'Video Google') successfully adapts the notion of specific recognition as an analogy of text retrieval. They utilize standard methods from text retrieval systems

such as search engines to visual words. The k-means clustered descriptors are equally incorporated into a term frequency - inverse document frequency (TD-IDF) scheme.

To avoid clustering all of their data – two full length feature films – they obtain a subset of a few frames for 48 shots at various locations in the movies. The resulting 200.000 descriptors cover about 10% of the total frames. After well-known refinements such as affine transformation, stop list for common and rare feature vectors and the Mahalanobis distance for matching, Sivic and Zisserman also perform spatial constraints in further analogy of test retrieval. The consistency is verified by computational efficient measures such as approximate locations within a surrounding area or similar layout for neighboring matches in both the query and retrieved frames [79].

Vocabulary Tree

The vocabulary tree by Nistér and Stewénius [63] is a hierarchical clustering of the features vectors. The vocabulary tree similarly implements a voting scheme modelled after the textual search using inverted file lists with unique identifiers. It provides two main benefits due to this structure. First, the hierarchical k-means clustering of visual words is adaptive to the actual data. The tree has a compact and efficient form even for a large number of images [63]. Second, the hierarchies are used simultaneously during clustering for the quantification and during querying for discarding unrelated descriptors. This combination provides efficient access to the stored images.

Nistér and Stewénius state their system enables on-the-fly insertion of new objects after the unsupervised offline training phase is completed. Due to the adaptive nature the clustered tree is easily reused without repeating expensive training steps. Their work shows applications using 35.000 training images and up to 16 million leaf nodes. The final tree holds one million images and has a query time of under one second [63].

Randomized Trees

Introduced by Amit and Geman [1], randomized trees are applied to solve a classification problem. The concept is to build a tree where each node represents a decision that narrows down the choices to a final outcome. The randomization plays a part in the selection of training and test cases included to build the tree. This is necessary to reduce the overall amount of information stored in the tree. The small random subsets allow smaller, faster trees, each one representing part of the data and a weak classifier itself.

This system was used for classification of handwritten digits. Lepetit et al. [44] create a fast and robust object recognition system after adapting the randomized trees approach instead of the previous nearest neighbor

matching [45]. Lepetit et al. extend the binary decision with a middle way and also a refinement through combination of multiple trees. Their approach achieves frame-rate performance due to approximations which do not create major drawbacks on their application of pose estimation and recognition. First, the detection method does not require a separate feature description. Second, for training multiple affine distorted images, patches are generated for each detected interest point. This provides the learning with the necessary invariance to affine transformation or illumination changes. Third, the gray-level image information is directly used to make the decisions [42, 43].

While the system is fast during the recognition phase it requires up to 15 minutes of processing and training beforehand. Recently, they have reduced this to less than one minute when reusing prior trees with updated node information [44].

Obdržálek and Matas [67] are inspired by this approach and propose a *LAF-tree*. The number of randomized trees used are dependent on the number of objects in the recognition system and provide an indexing time of log(N). Their description method uses LAFs and thus is inherently invariant to affine deformations. A further addition is the concept to move away from fixed image patch sizes. Obdržálek and Matas argue that too much background is learned and a fixed patch is not representative for many irregular shaped object features. Their experiments show a recall time of about two milliseconds for 400 object images. However, the tree requires a long time for training objects, and more so, retraining is required for new objects [67].

2.4 Summary

Each of the discussed feature detectors has different properties such as type, amount, structure or location of the detected features which make the methods complementary [17, 59]. The selection of a single or more detectors crucially depends on the application purpose. For a robust and online learning and recognition system the main requirements are efficiency, repeatability and distinctiveness.

Fraundorfer and Bischof [16] and Mikolajczyk et al. [59] conclude that the MSER detector performs best under a variety of transformations. An extension by Fraundorfer and Bischof [17] confirms this even for non-planar scenes. The overwhelming performance of the MSER detector by Matas et al. in terms of repeatability and efficiency is unrivalled.

Equally, each of the discussed description methods provides a different approach to an affine invariant and robust representation of an interest region. Their properties as well as their performance vary substantially. The evaluations by Mikolajczyk and Schmid [55, 58] show the superior descriptor to be GLOH closely followed by SIFT. This ranking is consistent for the precision-recall evaluations under various affine transformations and distor-

tions as well as for distinctiveness based on information content of PCA [29] components.

However, in terms of efficiency the SIFT descriptor is unbeaten. Unless low-dimensionality and thus less distinctiveness is sought, here steerable filters provide the best balance. The computational benefits of SIFT are partly due to the combination of detection and description into one process and its careful recycling of calculated information. The SIFT description is a straightforward process providing a careful balance between repeatability, distinctiveness and efficiency.

There are many approaches available to deal with fast indexing and matching of feature descriptions. Online recognition as desired is thus possible as well as handling large numbers of objects. However, for the online learning of objects we need to be able to insert new object representations into the data structures as well.

Kd-trees provide the potential for fast online addition, yet only seem to cope with smaller datasets unless considering the most recent adaptation by Philbin et al. [68]. Still, randomized trees require about one minute of training even when reusing previously calculated trees [44]. Flat k-means clustering provides a valuable aspect of adapting to the data distribution. The vocabulary tree by Nistér and Stewénius uses this benefit in addition to hierarchical fast recognition and compact representation of millions of images without the need for retraining.

Together, these methods – MSER detection, SIFT description and storage by means of the vocabulary tree – provide the fast, robust, distinctive and adaptive processing needed for an online learning and recognition system. These methods thus form the basis for the proposed system along with one more aspect discussed in the next section.

2.5 Learning through Tracking

The state-of-the-art technologies for detection, description and storage provide a solid basis for robust and fast learning and recognition. The final key aspect is the use of tracking to improve the learning experience. Typical single-image or multiple-image learning suffers from either few good features or too many unrelated and less distinctive features. Tracking an object and its interest points provides comprehensive analysis of the motion and the appearances. Research by Wallis and Bülthoff [87] suggests that this is the same process as within humans. The tracking of recognized features allows for a greater understanding and learning experience than multiple independent features can provide.

The straightforward approach for matching and thus combining interest points is to compare each of them in an exhaustive search between consecutive frames. This process of finding nearest neighbors in terms of their

description vectors quickly becomes very time-consuming if the number of interest points is too large and impractical if a high accuracy of matching is not maintained. Tracking has the advantage of directly matching interest points in each frame and has three major benefits. First, the tracking provides the inherent connections between features in each frame. These connections are used to build trajectories [76] representing the motion of an object and its interest points. Second, the matching accuracy is increased due to the ability to use previous tracking information in consecutive frames to reduce the search regions. Third, constraining the search space during tracking also speeds up the processing significantly.

The goal of tracking is to collect information about an object's appearance and use it to build a more distinctive object representation which is sufficiently robust and compact to employ in online learning and recognition. In the following section recent work which makes use of tracking for learning is discussed and evaluated.

2.5.1 Tracking Methods

Roth et al. [70] created a tracking system based on the notion of MSER tracking [9] and applied a global appearance learning. The tracking collects object appearances in form of image patches containing the object. These are used to train an object representation by means of incremental PCA [29]. The MSER tracking concept by Donoser and Bischof is an efficient and accurate method for tracking MSERs. It achieves a significant boost in processing and accuracy by three improvements. First, only one type of detection – either `MSER-` or `MSER+` – is performed. Second, only an adaptive part of the analysis is performed through limiting the gray value range of the image. Each MSER which represents an extremal region is attributed two specific gray values defined by the brightest and darkest pixel included in its region. And finally, the search area within the image for matching possible extremal regions is reduced significantly by looking in a location near its predecessor.

Their use of tracking allows efficient collection of image patches for a global appearance modelling. A drawback is that the bounding boxes have to be accurate to achieve a good recognition rate. Thus, only objects with MSERs defining the exact boundary can be used. This restriction is not robust for inhomogeneous object surfaces, however does not affect local feature approaches. Further, the general low quantity in MSER detections does not have an adverse effect in performance and is even an advantage when only a lower number of features have to be processed. The notion of MSER tracking along with an adaptation for tracking multiple local features is described detail in Section 3.1.1.

Wallraven and Bülthoff [88] present a tracking system which uses trajectories to identify key frames. These key frames have the characteristic that

enough of their content changed and they show new image information and different view points. Key frames are detected when less than 25% of the trajectories between consecutive frames are matched. Wallraven and Bülthoff use the set of key frames in global appearance-based learning.

Everingham et al. [13] show a tracking system for faces and pedestrians. The tracking of faces is achieved by a Kanade-Lucas-Tomasi tracker [80] and correspondences are matched between each frame globally. They apply a simple tracking procedure of counting the number of matches associated with faces to obtain robust tracking. It is capable of handling occlusions and variations in pose and facial expression while at the same time reducing the total number of tracked faces to a robust subset.

Kim et al. [36] make use of stereo vision to directly segment interest points on the object from background detections. Their elaborate robotic system is capable of learning and recognizing hundreds objects without any manual interaction.

Chetverikov and Verestoy [6] introduced a tracking system which focuses on building trajectories of interest points, however also globally determine correspondences. They identify the states which an interest point undergoes during tracking such as appearance, temporary occlusion and permanent disappearance. Chetverikov and Verestoy then include these events into repairing and combining broken trajectories.

Sivic et al. [78] employ the collected information of a tracking system for creating object level grouping. This implicitly creates simple 3D models of the tracked objects by connecting interest points of several frames during recognition. Their approach also includes short range repairs to handle small distortions to the tracking and long range repair for reappearance of objects. MSERs are used in their detection step, however matching occurs globally and not through the concepts of MSER tracking.

Grabner [22, 23] shows how the problem of the enormous information collected during tracking is used as a further advantage. A tracking system provides a range of appearances and thus descriptions for each interest point in each frame. Grabner uses the relative change of a SIFT descriptor to detect a stable minimum where the trajectory contains the best representation. The notion of a frontal view provides the learning process with a single description per feature. These however carry the most distinctive and representative information about an object.

To sum up, previous work has introduced successful learning and recognition systems. However, most of them are not suited for online processing and require additional interaction. In this thesis, the proposed tracking system is targeted at online learning and recognition. This is achieved by the aforementioned state-of-the-art detection, description and storage methods in combination with concepts from previous work such as trajectory construction, short range repair, handling of appearance and disappearance of tracked features and compact summarization by frontal views.

Chapter 3

Tracking System

This chapter describes the proposed robust online learning and recognition framework. The underlying concept uses temporal memory of an object to increase the learning effect. The work by Wallis and Bülthoff suggests that a connection between continuous views improves the recognition capabilities in humans. This association between various appearances of the same object provides a better learning experience [87]. This idea is realized by tracking the motion of an object and forming a compact representation. The tracking provides the learning experience which is defined as acquiring information about an object's appearance. A process of building an object representation through tracking is discussed which is common to both learning and recognition.

The combination of state-of-the-art detection, description and retrieval techniques provides the necessary boost to perform all steps within an online process. Figure 3.1 illustrates the proposed system with each of its steps.

In the first step, an object is followed in its motion by means of tracking interest points for which the Maximally Stable Extremal Regions (MSER) by Matas et al. [51] are used. The notion of MSER tracking introduced by Donoser and Bischof [9] provides a significant speedup and repeatability improvement useful for online processing. An advanced tracking method offers both the detection of interest points and the robust matching between consecutive frames. It is applied to ensure stable tracking results and is described in Section 3.1. During the course of tracking, all information associated with the tracked interest points is recorded and trajectories are constructed for each of these MSERs. This provides the basis for the learning and recognition steps. But please note that one-shot recognition may also be used.

The second step is an analysis of the trajectories to obtain a compact representation of the redundant information. The quality of the tracking is used to evaluate and select robust trajectories. Such a subset is more reliable and provides a better repeatability in other scenes. The appearances of all

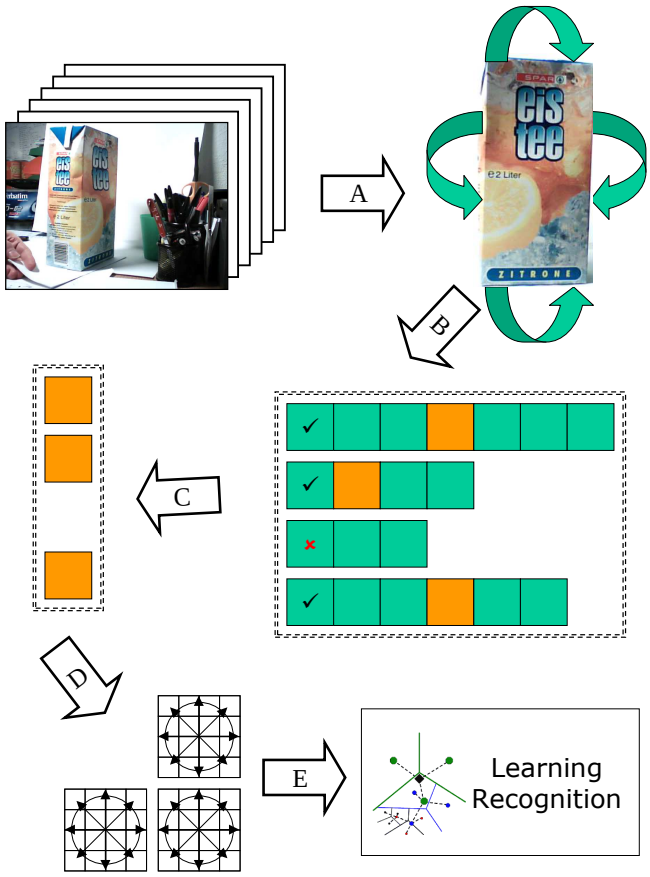


Figure 3.1: The system consists of five steps: (A) Tracking, (B) Extraction of trajectories, (C) *Frontal MSER* selection, (D) SIFT description and (E) either learning or recognition by means of a vocabulary tree.

MSERs are analyzed and combined to a compact representation showing the optimal viewpoint of the tracked MSER as substitute for the entire trajectory, as introduced by Grabner [22]. In this thesis, the notion of *frontal MSERs* is a compact summary while providing the most information for feature description. For this third step, Lowe's SIFT approach is used in combination with an affine normalization to arrive at a robust, invariant and distinctive description of the summarized trajectory. Details of this step are described in Section 3.2.

And finally, all of the tracked trajectories are combined to a compact object representation. This set of descriptors is used to learn and recognize an object. For storing and matching these descriptions the vocabulary tree by Nistér and Stewénius fulfills the necessary requirements for online performance. The hierarchical data structure enables efficient insertion and recognition of objects without the need for any offline training of the feature descriptions. These processes are described in Section 3.3 along with introducing a confidence measure which is used for online evaluation of the recognition certainty.

Please note that even though for clarification each step is discussed separately, all of the steps are integrated into the tracking system where all processing is entirely performed online. During tracking object representations are updated in each frame including the SIFT descriptors. This allows learning of unknown objects and their recognition by a rotation and translation in front of a camera.

The key aspect is the building of object representations by tracking. This is used in training to learn the appearance of objects. During the recognition phase, the same benefits of finding *frontal MSERs* may be used to obtain better object representations. Alternatively, one-shot image recognition may be used to identify objects in the visible scene. In this approach, tracking is not used but instead a one time MSER detection of a single view and its SIFT description provides the representation to be matched against previously learned objects. The storage and matching allows any mixture of input methods. That is, the SIFT description for learning or recognition may be retrieved through tracking or single image analysis.

3.1 Trajectories through Object Tracking

In contrast to one-shot learning or recognition where a single image is used for each task, we use tracking to increase the gained information. This enables the system to perform two valuable operations. First, the object is more easily separated from the background. Tracking an object provides a segmentation of the scene into the object of interest and the unimportant background. The tracked regions determine this segmentation by limiting the regions to those on the surface of the object using the information available during tracking. This segmentation is not required for learning but it helps the tracking to focus more accurately on the object. Thus the object is not restricted to a fixed position and is able to move while maintaining a good separation of the scene's information and consequently regions of interest on the object.

Second, the motion itself delivers the most interesting benefit. In single image analysis the learning step only sees one view of an object. This restricts feature detection to the visible surfaces of the object. Remaining

features are hidden and cannot be used for the learning process. However, another important difference is that only one appearance is available for each feature. If the object is not in a frontal position and viewed at an angle, more features are visible but the angle introduces perspective distortions which deteriorates the quality of the feature.

The idea behind tracking is to follow the motion of an object, learn all sides and select the best representations for each feature independent of the view at which it is originally detected. Tracking an interest region over a period of time undergoing various motions is an indicator for a valuable robust feature of an object. The longer such a tracking is done successfully, the more stability is associated with it. This is not the case in a single shot method where no information about the repeatability is available. Here the choice comes down to either a single frontal view with clearly visible but limited amount of features or a single non-frontal view which provides a higher number of distorted features. Affine normalization helps to reduce some of the perspective distortions by modelling them with an affine transformation. Similarly, multiple views could be used to detect features on all surfaces of the object. This would, however, require a complicated matching step to ensure accuracy and thus, a higher computational effort.

The benefit of tracking is the accurate matching between consecutive frames which again is used to construct trajectories. A trajectory t_i is a set of features tracked over time and defined by

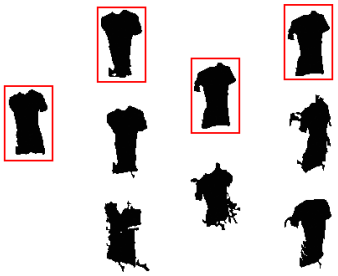
$$t_i = \{f_1, f_2, f_3, \ldots\} \tag{3.1}$$

where f_i are the tracked features associated with it. The process of constructing trajectories is described in the following sections by the notion of MSER tracking combined with an advanced handling of multiple MSERs, short range repair for unstable matching and new feature detections efficiently.

3.1.1 Maximally Stable Extremal Regions (MSER) Tracking

Maximally Stable Extremal Regions (MSER) were proposed by Matas et al. [50, 51] as a novel interest region detector for wide-baseline stereo matching. Matas et al. define extremal regions as distinguished regions with properties such as affine transformation invariance of the intensity function, multi-scale detection, a measure of stability, covariance to adjacency preserving continuous transformation and an enumeration in O(n log log n) with n being the number of pixels in the image.

The tracking process incorporates the notion of MSER tracking as proposed by Donoser and Bischof [9]. Their contribution is a sophisticated tracking approach where each initial MSER is individually matched in each consecutive frame. This allows fast and accurate tracking. The key concept is the efficient detection of MSERs combined with tracking information from previous frames to further reduce computation.



(a) t+0 (b) t+1 (c) t+2 (d) t+3

Figure 3.2: Illustration of the tracking concept: An MSER detected in a previous image is matched to its best fit in the next image by size, center of gravity, intensity and stability.

The tracking step is the process of finding the extremal region within the image area which best fits the previously tracked MSER, see Figure 3.2 for an example of the matching. The tracked (best fit) MSER is located by means of its size, center of mass, stability and intensity which are already available due to the incremental calculation during the building of the extremal regions data structure known as a component tree.

The component tree, originally introduced in statistics [26, 89] for classification and clustering, was redefined by Jones [30] as a representation for the connections between the thresholding the resulting regions. A component tree is an acyclic directed graph with nodes corresponding to pixels, or later detected regions, and edges defining the relationships between their intensity levels. At any chosen intensity level of the hierarchy the component tree contains the connected components detected at this level. A component tree allows to quickly access extremal regions and store the necessary meta-information in its nodes.

There exist several approaches [7, 31, 60, 61, 62, 71] to build component trees. The approach by Najman and Couprie [61] is used in MSER tracking and delivers the best runtime complexity to maintain even in worst-case a very efficient algorithm. Najman and Couprie describe an algorithm to build the component tree with a worst-case complexity is $O(N \times \alpha(N))$ with $\alpha(N)$ being the inverse Ackermann function and N the sum of the number of pixels and arcs in an image, i.e. linear for practical purposes. The component tree is the bases for detecting MSERs and matching them efficiently during MSER tracking. Table 3.1 summarizes the properties of an MSER that are used for tracking. Due to its arbitrary shape a combination of only a few of its characteristics suffices to identify similar extremal regions between frames.

A vector consisting of its size, center of mass, stability and intensity range is used to determine the best match which is then used as next MSER in subsequent images. The query MSER is compared to other extremal regions

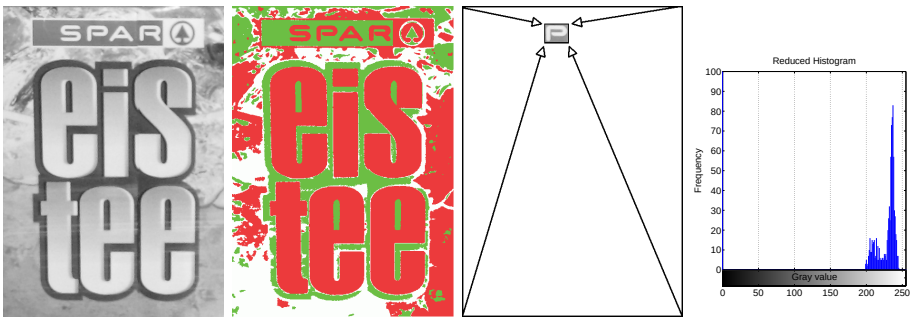
Property	Description
Area	The number of pixels included in this MSER.
Intensity	The minimum and maximum gray values of the MSER.
Center of Mass	The x- and y-position of the center of mass.
Stability	The criterion evaluating its relative change in size.
Bounding box	The position of the smallest surrounding rectangle.
Texture	The underlying raw image information.
Shape	The shape as defined by each included pixels.
Covariance	The statistical distribution of the shape.

Table 3.1: This table lists the properties of an MSER provided by the detection process.

having approximately the same size, location, intensity and stability. These approximations are part of a motion model which allows for more stable tracking. The best match is selected and used for comparison within the next frame. A threshold is applied to evaluate the fitness of the selected best match. If no best match could be obtained or the stability threshold is not achieved, the previous MSER is reused in the next frame with a looser set of restrictions. This again makes the tracking more stable in general since temporary unstable matches due to occlusion, noise or other distortions are not affecting the tracking. More details are discussed in Section 3.1.2.

The MSER tracking notion thus ensures robust tracking of all MSERs identified in the initial image by considering all extremal regions of an image as tracked representations. Additionally, there are three speedups which are proposed to allow faster tracking. First, only one type of detection – either `MSER-` or `MSER+` – is performed. Second, only an adaptive part of the analysis is performed through limiting the gray value range of the image. And third, the search area within the image for matching possible extremal regions is reduced significantly. Figure 3.3 shows these speedups where b) illustrates the difference between the two MSER detection types.

Building only part of the component tree is achieved through limiting the gray value range of that image. Each MSER which represents an extremal region is attributed two specific gray values defined by the brightest and darkest pixel included in its region. Each layer represents a gray value intensity and provides the hierarchical inclusion of extremal regions within larger regions. Any extremal region is enclosed by its lowest and highest intensity level. Depending on the homogeneity of the MSERs fewer or more intensity levels are required to encompass all of its pixels. To optimize the processing only extremal regions are calculated within a range of the analyzed gray value levels. An example estimation by Donoser and Bischof [9] shows this can lead to an improvement resulting in one fifth of the computational time. Figure 3.3 d) illustrates this in an histogram where only a small part of the



(a) Input (b) Type (c) ROI (d) Histogram

Figure 3.3: Illustration of the MSER tracking advantages: a) shows an example image to be analyzed for MSERs b) shows the `MSER-` and `MSER+` in green and red color respectively. c) shows the ROI which is analyzed for each next analysis based on the previous MSER. d) is the histogram of the previous MSER showing the reduced gray values range used for MSER tracking. The initial image contains 304575 pixels and due to the restrictions the final analysis only considers 814 pixels resulting in a speedup of 374x while at the same time improving the accuracy of matching as well.

gray value range is extracted.

Reducing the search region for any consequent extremal regions similarly produces a valuable speedup. The idea is that each corresponding MSER is located near its predecessor and thus provides a proximity measure. Using a motion estimation or a general non-directional parameter a region of interest (ROI) is defined – see Figure 3.3 a) and c) – to significantly reduce the area size and thus processing time.

These speedups have further advantages besides the obvious initial reduction of computational time. Since the search space is smaller, also fewer extremal regions are considered while searching for the best match. Further, the component tree is only build for already similar visual information, i.e. the same extremal regions at similar intensity levels in near proximity and size. Thus, the constraints not only improve computation time but also the quality of tracking.

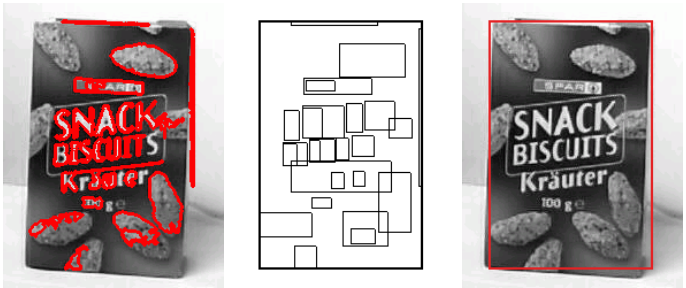
There are no steps to include affine invariance for the tracking since the regions are distorted in a similar way in each frame. A change in perspective view results in only a small change between two consecutive frames provided the frame rate is large enough to capture the motion relatively smooth. For the feature description process steps to ensure affine invariance are implemented.

3.1.2 Multiple MSER Tracking

The MSER tracking thus finds a best fit for each previously detected MSER in a greatly reduced search space. This allows for a small execution time for a single MSER detection and its matching step. Since a single MSER is not sufficient for good recognition, multiple MSERs are considered using an extension known as *compound MSER tracking*. This method provides significant advantages over the previously known *single MSER tracking* and *color MSER tracking* [10]. These methods fail on inhomogeneous objects since these are not always robustly detected. The resulting match is considered unstable and its incorrect segmentation leads to time-consuming and unwanted analysis of the background.

The *compound MSER tracking* is suitable for tracking multiple MSERs simultaneously. This approach detects and matches multiple smaller MSERs directly. It is an extension of the *single MSER tracking* to a compound analysis. Each tracked MSER is only analyzed in an image region around its previous center of mass and a range in gray value intensities both tightly restricted by its predecessor. This provides the full benefits since it speeds up the detection, increases the accuracy of the matching process, and third, only analyzes the image regions of interest.

The main drawback of this approach is the lack of an encompassing shape which reflects the objects shape directly. Although segmentation is not required for learning and recognition, it provides a valuable separation between the object of interest and background. In *compound MSER tracking* the bounding boxes of the individually tracked MSERs are combined to a global bounding box which is then used as restriction for the following robust tracking, as illustrated in Figure 3.4. Further robustness achieved through evaluation of the behavior in motion and properties of the tracked MSERs.



(a) Input (b) Compound (c) Global Box

Figure 3.4: In *compound MSER tracking* individually detected MSERs (a) and their bounding boxes are combined a global bounding box (b) which is used as a restriction for next tracking steps and redetections (c).

Two more aspects of multiple MSER tracking lie in the evaluation of the tracked MSERs and detection of new features. First, the evaluation of matching stability is a vital part to ensure robust tracking. Once a feature on the object is no longer visible or cannot be reliably be matched for several frames, its trajectory should be terminated. Second, in order to optimally learn 3D objects all sides must be visible during tracking and new features must be detected. These aspects are discussed in the next sections.

Evaluation of Stable Matching

Similar to global shape matching as used in *single MSER tracking* smaller MSERs also vary in stability and are subject to occlusion and distortion. Another new problem is the close proximity of other MSERs such as other letters in a word. To cope with unstable detections the matching between a previously identified MSER and a new image region is comprised of the following three steps.

First, a motion model is used to derive a specific direction of motion. Due to the locality more MSERs of a similar small size exist instead of a single global one. For example, individual letters of a text may get confused when matching in a general region around the previous location. To avoid this effect a more precise restriction of the search region is applied. As before the image region is restricted to a ROI of the previous stable match given by its bounding box. This reduced image area is extended only in the direction of the detected motion. The two-dimensional direction is calculated by

$$motion_x = center_x(i) - center_x(i-1) \tag{3.2}$$

$$motion_y = center_y(i) - center_y(i-1) \tag{3.3}$$

where $center_x(i)$ and $center_x(i-1)$ are the location of the center of mass of the currently tracked MSER and its predecessor respectively. The same calculation is done for the y-direction and then incorporated into the defined ROI. This individual motion estimation provides a closer search region inside the image space than a non-directional extension. As consequence less neighboring MSERs are incorrectly matched and confused with one another. Of course, this first-order model does not cope well with sudden change to an opposite direction.

Second, after the best available match is determined through comparing the vector of MSER properties by an Euclidean distance, the stability of the new MSER is evaluated. A threshold is applied to verify that its stability value – the relative change in size – is sufficiently small for a stable tracking. If the minimally required stability is not reached, the previous MSER is reused instead of the new unstable one. However, if only unstable or no best matches at all could be found for a number of tracks, the MSER is dropped from tracking and its trajectory ends. The measure used is a robustness

counter which is increased each time the stability is insufficient. This mechanism allows for small repairs during tracking. Figure 3.5 gives an overview of the length of trajectories and the effects of repairing. To minimize the effect of various motions and their frame rates it shows the frequencies of track length on average for several sequences. When no robustness limit is applied there is a large number of trajectories each with very short track lengths. This means no repair is performed and the tracking fails to robustly match MSERs over longer periods of time. A selection of increasing limits in Figure 3.5 shows the respective increase in track lengths. The spikes indicate when the robustness limit effects the natural length.

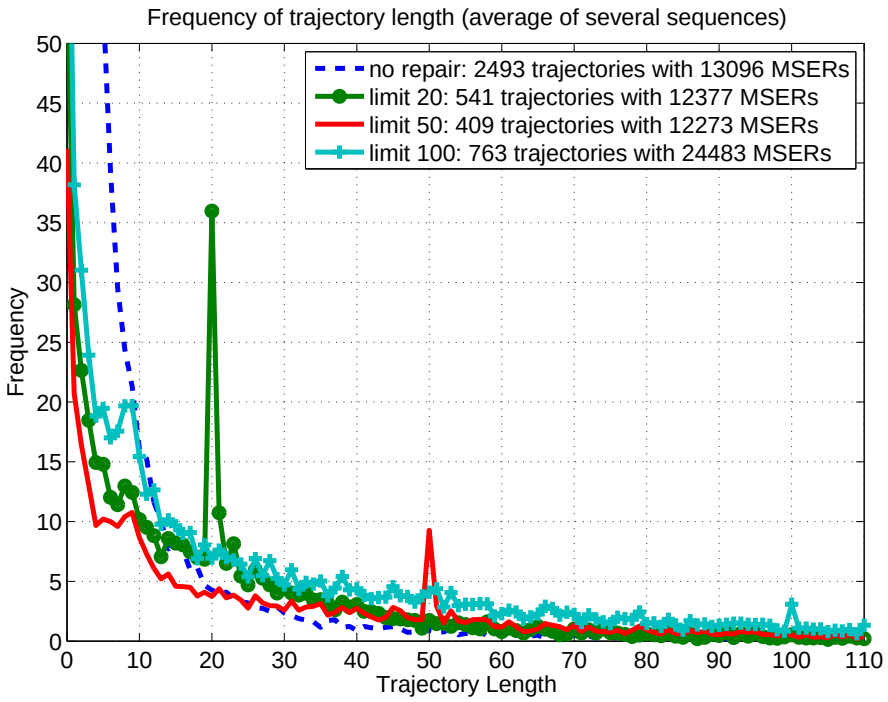


Figure 3.5: Comparison of the track length frequencies: The robustness limit is set to various values showing its effect on repairing trajectories by accepting unstable matches which would otherwise terminate trajectories immediately (no repair). The robustness limit evaluates the changes in tracking stability. Trajectories are longer when this limit is set to a higher value by repairing unstable periods during tracking. Please note, for better comparison the maximum frequency shown is 50 whereas no repair results initial frequencies up to 1000 but quickly drops to 50, as shown here.

The essence of this figure is to show the need for trajectory repair. The problem with a high limit is that a tracked MSER is allowed to remain un-

stable for long time. This deteriorates the trajectory and the tracking quality. Due to a limited number of concurrently tracked MSER, any prolonged tracking of an unstable MSER prevents the detection of new MSERs. This is also visible in Figures 3.5 by examining the total number of trajectories. For example, the total number drops on average by roughly 2000 trajectories between no limit and the first limit illustrated. For the high limit, the number of trajectories increases again due to unstable tracking of interest points on the background producing many more MSERs unrelated to the object itself.

The third step is a further evaluation on the behavior of the trajectories. The aim is to quickly terminate trajectories when the tracked MSER has suddenly become very unstable while still maintaining a good stability value, i.e. a small change in relative size. This step thus includes a set of rules evaluated frame by frame. These include maximum limits on absolute size, relative size increase, change in location as well as a check for duplicates.

Due to the motion and fine scale of the multiple MSERs the best-fit matching may determine a feature on the object which is already being tracked by another MSER. In this case, two MSERs track the same feature without any additional benefit. If such duplicates are detected, the trajectory with a shorter track length is terminated.

3D Learning requires Redetection

One benefit of tracking is the ability to follow an object and learn more views of it. This requires that newly appearing MSERs are detected on sides of the object not seen before. In principle this should be done every frame to ensure no information is lost. However, due to online performance requirements the number of redetections and the number of concurrently tracked MSERs is limited.

Active trajectories are frequently terminated because of losing stable matches. This provides an opportunity to efficiently combine redetections with a low number of active trajectories. If this number drops to zero, no MSERs are tracked and a redetection from scratch would be required to continue learning the object from other viewpoints. A full frame redetection is costly in terms of computation time and does not take into account that the object has been tracked sofar. The idea is to use the bounding box of the currently tracked stable MSERs to provide a ROI for new detections. When the total number of tracked MSERs has decreased to a certain limit, a redetection is performed on the reduced image space, as indicated by the sudden increase of tracked tracked MSERs in Figure 3.6. The threshold of active trajectories is set to a balance between retrieving new features frequently and an acceptable processing time.

A novel solution to merging currently active trajectories with new detections is efficiently solved by ignoring active trajectories during the detection.

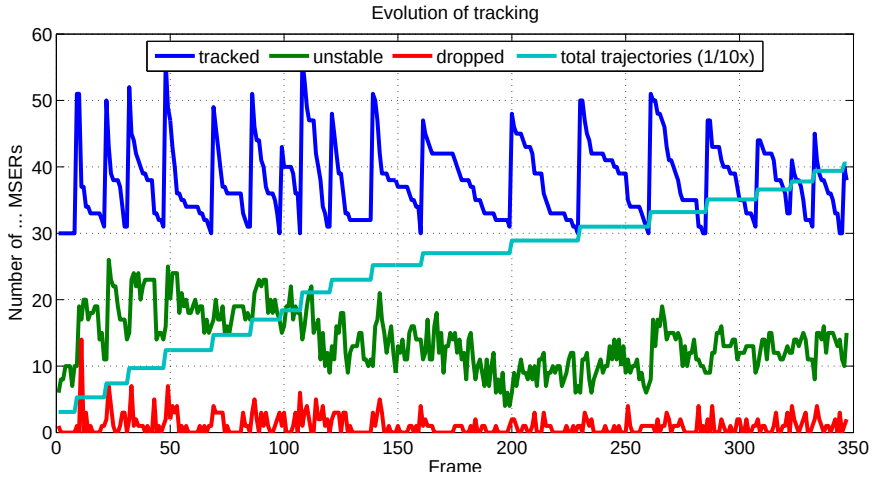


Figure 3.6: Evolution of tracked MSERs: The drop of stable tracked MSERs is shown in relation to the number of unstable matches and terminated trajectories due too sudden or prolonged instable matching.

(a) Frame (b) MSERs (c) Reduced (d) New MSERs (e) Merge

Figure 3.7: Illustration of new detection: a) shows the current input frame. The previously tracked MSERs b) are removed from the input to a reduced image c). Newly detected MSERs d) are thus guaranteed to be non-overlapping with previously tracked MSERs. They provide new interest points and their trajectories are efficiently merged e).

The process involves removing the shapes of current MSERs by skipping their pixels during the analysis for new MSERs. Figure 3.7 illustrates this process and its steps. First, the image is cropped to the global bounding box of interest. Second, the previous MSERs as shown in b) are subtracted from the image resulting in a reduced image such as c) where ignored pixels are shown in black. Third, the reduced area is analyzed for MSERs. Any

new detections d) are guaranteed to be non-overlapping with the previous MSERs, see e). That is, the new MSERs are not tracked sofar and describe new interest points.

Thus a time-consuming comparison and merge algorithm was replaced by a reduction of search space. This speedup also provides less duplicate detections and ensures that other previously uncovered regions of an object are also tracked.

3.2 Object Representation

Tracking provides trajectories which describe the motion and appearance of each MSER on the object. This information is used to build a robust and compact object representation. The online learning and recognition requires a representation to be repeatable and small to provide a good recognition as well as swift processing.

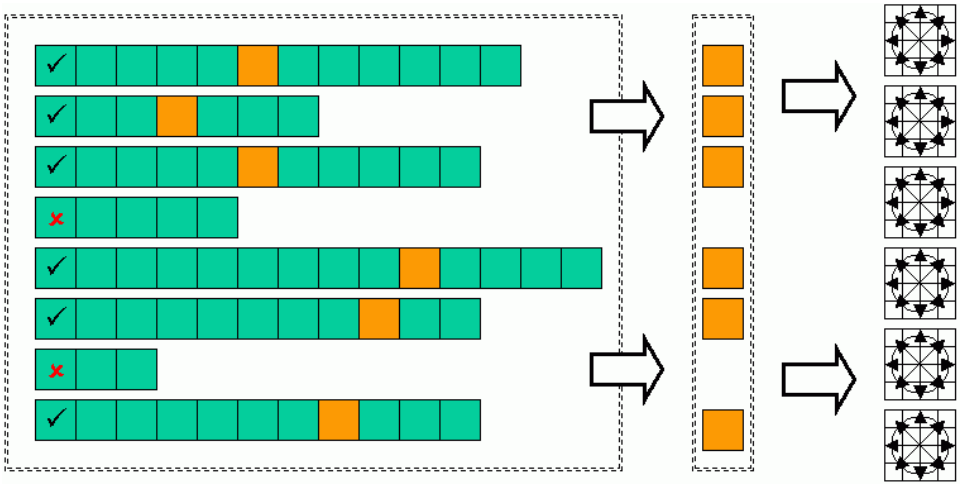


Figure 3.8: Building a compact object representation: First, a robust subset of the trajectories is selected. Second, the robust trajectories are summarized by a single *frontal MSER* providing the most fronto-parallel representation of this trajectory. Third, the *frontal MSERs* are described by SIFT descriptors providing the final robust, distinctive and invariant object representations. This process reduces the wealth of information collected during tracking to a reliable subset for online learning and recognition.

The information acquired through tracking is enormous and redundant. Every trajectory contains every MSER tracked through the length of its active trajectory. Refer to Figure 3.5, there exist on average 500 trajectories per sequence and the total number of tracked MSERs is roughly 12000. Although these values are only averages and strongly depend on the object

and its motion, they clearly show how much information is collected. The goal of the object representation is to reduce this wealth of information and provide a robust and compact subset.

Figure 3.8 illustrates the overall process which is discussed in the next sections. For clarification each step is discussed separately, but all steps are performed online during tracking without any additional post-processing. First, the trajectories are evaluated to select a set of robust trajectories. Second, each robust trajectory is summarized to a single representative known as *frontal MSER*. A description process such as SIFT is used to describe this final subset of *frontal MSERs*. The resulting descriptors make up the robust and compact object representation which is small enough and sufficiently distinctive to successfully apply it for online learning and recognition.

3.2.1 Robust Trajectories

A trajectory holds all information about a feature collected during the tracking. Since MSER tracking is used in combination with a compound tracking method and efficient merging of new detections, the trajectories are constructed inherently without the need for time-consuming matching between detected features.

The tracking provides an ongoing evaluation of the trajectories and ensures that only stable trajectories are pursued. Due to online performance the tracking only contains few MSERs simultaneously. If no evaluation is present, the tracker is quickly trapped with instable trajectories and is not able to track new MSERs. Thus it is a requirement of the tracker itself to perform an evaluation which is also used the robust selection for object representation.

The final robust subset is selected based on the quality of the trajectory. This measure defined as

$$quality = \frac{\#\ stable\ matches}{tracking\ length} \tag{3.4}$$

where the stable matches represents the number of successful and accurate matches between frames and the tracking length provides normalization. A threshold for this quality value is set to 0.5 and determines that at least half the trajectory must be comprised of MSERs which have been robustly matched between frames.

3.2.2 Compact Representation

The next step involves finding a suitable compact representation for a trajectory. In a single image system there exists only features without trajectories. In a multiple image system a larger number of unconnected features are present and require expensive correspondence matching. The benefit of

tracking and building trajectories is to collect more information about the features detected on an object. This redundant information is then used to find a meaningful compact representation of the entire trajectory.

Previous work by Grabner [22, 23] used the relative change of a feature descriptor to detect a minimum where the trajectory contains the best representation. In controlled rotation the minimum is detected by a quadratic fit ignoring outliers. For arbitrary rotation or any uncontrolled movement a stable minimum is harder to detect. Further, it is computationally too expensive to calculate descriptors for all MSERs of a trajectory online.

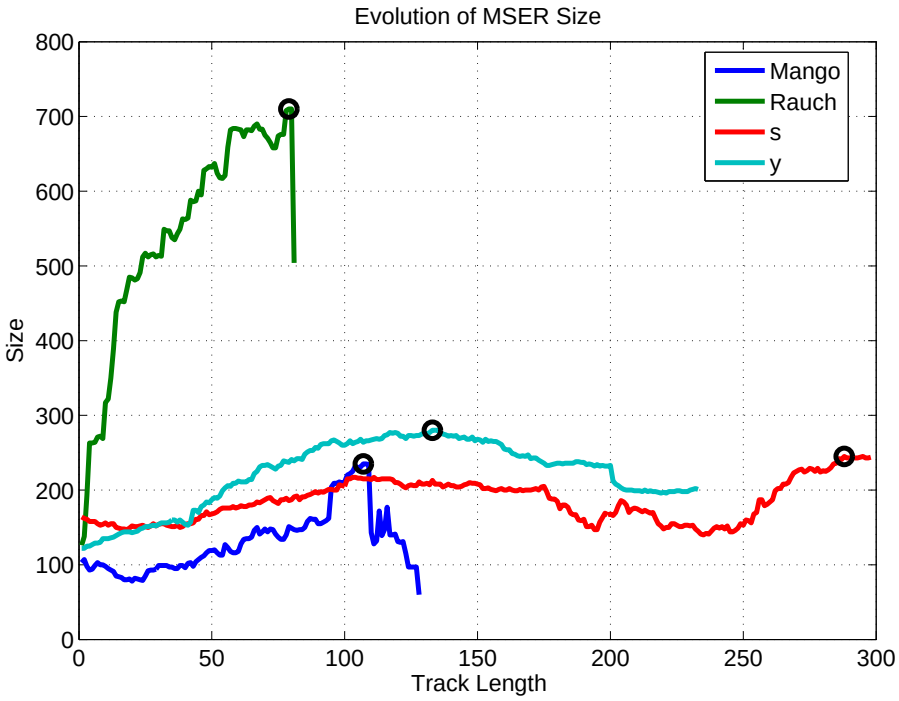


Figure 3.9: Illustration of *frontal MSER* selection as graph: The evolution of the size of the MSERs from Figure 3.10 are drawn. The circle indicates the chosen maximum size where the MSER is in its most frontal view.

Since we are using MSERs as interest regions, more information than just position and orientation is available, remember Table 3.1. An MSER has an arbitrary shape which reflects the perspective distortion in which it is viewed. The most suitable view for a compact representation is described by Grabner as the one which is fronto-parallel to the viewing plane. When a feature is parallel to the camera, it does not contain perspective distortion. The feature – or in our case the MSER – is thus the biggest MSER. If viewed at a different angle, the distortion decreases the size of the MSER.

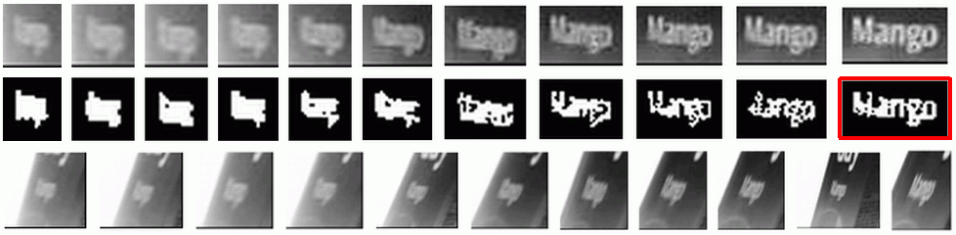
This property is used to select the *frontal MSER*, i.e. the MSER providing the frontal view, based on its size. The trajectory is analyzed and the largest MSER is selected. Scale change leads to incorrect *frontal MSER* selections and maybe counteracted by a normalization such as a filling factor defined by

$$filling\ factor = \frac{size\ of\ MSER}{area\ of\ bounding\ box} \tag{3.5}$$

where the area of the determined bounding box is used to normalize the size and derive a value how much of a bounding box is covered by the MSER. Instead of selecting the largest MSER the *frontal MSER* would be the one with the largest filling factor. However, in this thesis we are concentrating on maximum size selection.

Figure 3.9 shows a graph of the trajectories and their evolution in size over track length. For all four trajectories the maximum is clearly detectable, as indicated by the circle. This selection is used to identify the *frontal MSER*.

In Figure 3.10 the evolution of these trajectories is shown visually. The subset shows an image patch, the binary representation of the MSER, the affine normalization and size of the MSER. This illustration supports the choice of the MSER with the maximum size as suitable representation for its trajectory. The increase in size reduces not only the perspective distortions but also improves the quality of the underlying image data. Figure 3.10 b) shows that due to the motion a larger more clearly recognizable *Rauch* logo is tracked and detected as *frontal MSER*. The affine normalization patches in Figure 3.10 also show the value of the tracking and frontal view selection. When the feature is parallel to the camera, its image representation is the least distorted.



(a) Mango

(b) Rauch

(c) S

(d) y

Figure 3.10: Illustration of *frontal MSER* selection: a) to d) each show a subset of the information collected during tracking. Each example is illustrated by the underlying image patch, the MSER in its binary representation and the affine normalized image patch used for description. The *frontal MSER* – indicated by the red border – is selected based on the maximum size shown in Figure 3.9.

3.2.3 Description

To complete the object representation the selected best feature is described in an invariant, robust, repeatable and distinctive fashion. The first step in this process is the normalization of the MSER to achieve affine invariance as an approximation to the perspective invariance. The second step is a Scale Invariant Feature Transform (SIFT) description process introduced by Lowe [48, 49].

Since an MSER is by design affine-covariant only few steps are required to transform it into isotropic normalized region. Most of the affine normalization is adapted from the Local Affine Frames (LAF) approach by Obdržálek and Matas [65, 66].

The invariance of affine illumination changes is already partially handled by the MSER detection itself due to its inherent covariance of intensity hierarchies. Thus the same extremal regions are extracted with only their gray-value intensities globally skewed due to the photometric variations. This remaining effect is dealt with during the SIFT description.

The affine normalization uses the covariance of pixel locations in the binary MSER to derive a statistical measure for its distribution. Instead of an iterative approximation method as proposed by Mikolajczyk and Schmid [54, 57] where the inverse of the square root of the covariance matrix is used, the process is reduced to a single step. This is achieved by an efficient Cholesky decomposition of the 2x2 covariance matrix.

For some MSERs this results in problems due to their arbitrary shape. There are five categories which are described below and shown in Figure 3.11 with prominent examples.

Pure elongation describes a single thin MSER where either the height or width is a large multiple of the other.

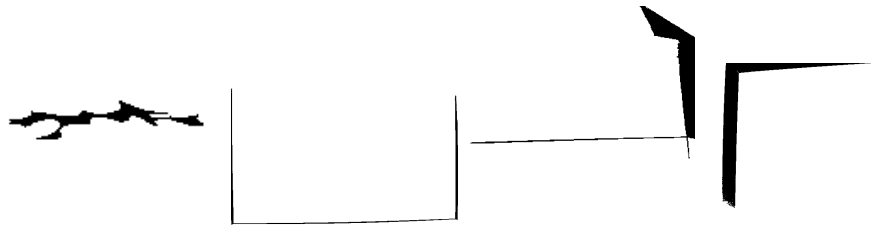
Mixed elongation describes an MSER which contains thins lines in two directions.

Appendices are problematic extensions to MSERs.

Close to border describe MSERs which may extend outside image to when normalization is performed.

Large MSERs have a large size before and a very large area size after normalization.

The main difficulty with most of these MSERs of problematic shape is that their normalization increases their image patch size by a large factor. The example in Figure 3.11 b) produces a new image which tries to thicken the thin horizonal line. As a consequence the two vertical lines are equally lengthened and the resulting image patch is impractically large. This requires immense



(a) Pure elongation (b) Mixed elongation (c) Appendices (d) Close to border

Figure 3.11: Illustration of the problematic arbitrary MSER shapes which result in image patches many times the original size and much higher computational effort.

computation during transformation. MSERs with appendices such as shown in 3.11 c) constitute a similar problem since the appendix is statistically insignificant due to its minute size compared to the rest of the MSER. A process which widens such an MSER inadvertently widens the entire MSER and the corresponding image patch.

The second problem arises with MSERs close to the image border as shown in 3.11 d). Here the normalization equally produces a large image patch but also requires image data from the original patch which lies outside its definition. It requires such unavailable image data due to an arbitrary rotation introduced by the affine normalization as well as the support region padding.

The solution implemented contains four countermeasures. First, eccentricity of the binary MSER is evaluated. If this value exceeds the threshold of 0.97 the affine normalization is skipped. Second, the covariance matrix is analyzed for typical values connected to the shape of the problematic MSERs. If such a configuration is found, the normalization is also skipped. Third, the maximum size for MSERs is generally limited to about 30 % of the image size. This prevents impractical computation efforts. Forth, when the outside of an image becomes part of the affine normalization, the underlying image data is mirrored.

The final step in the object representation process is the SIFT description [48, 49]. This method by Lowe creates an invariant, robust, repeatable and distinctive 128-dimensional description vector which is used to identify the underlying image patch.

The high-dimensionality of the SIFT descriptor provides a challenge when comparing to other description vectors. However, the gain in distinctiveness outweighs the loss in computational effort compared to lower dimensional descriptors.

Due to the design each step of the SIFT process merges efficiently with consecutive steps, for example the Gaussian smoothing to detect interest

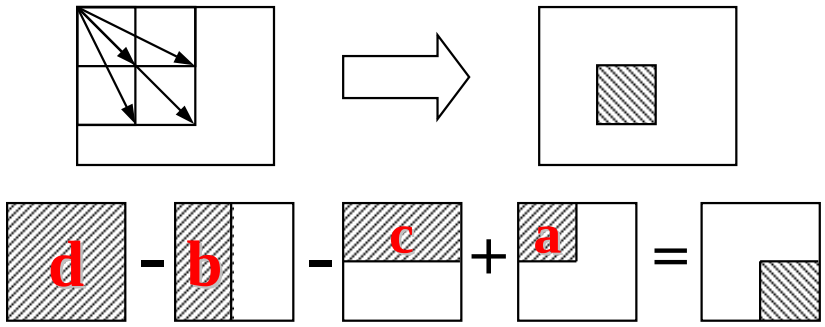


Figure 3.12: Illustration of the benefit of integral histograms: To obtain the histogram of region with its corner at the image corner only one lookup is required. For other regions the histograms of the four regions a, b, c, and d are combined. Thus the histogram of a rectangular region requires a runtime complexity of $O(4)$.

points is recycled for the gradients and orientation assignment. This delivers a fast detection and description process when used together.

To decrease the computation time of the original SIFT, extensions such as the *Fast Approximated SIFT* by Grabner et al. [24] and *Speeded Up Robust Features* (SURF) by Bay et al. [3] are investigated. Both make use of integral images [77, 86] and integral histograms [69] to provide significant computational improvements. *Fast Approximated SIFTs* provide a speed up of at least factor eight and SURFs a factor five while approximately maintaining similar recognition results.

This is based on several simplifications such as skipping the doubling of the image beforehand and using a difference-of-mean (DOM) as interest point detector. However, one of the main benefits is due to the amortization costs of using integral histograms [24]. The process involves incrementally calculating the histogram information. Each location in the integral version contains the final values up to this location. Figure 3.12 shows this concept where the histogram of an image patch is determined. Thus the histogram of any rectangular region is calculated by four simple additions – provided it has been build for the entire image data.

Since the MSER detection already provides interest points, only a description process is required and its benefits may be used. However, the main advantage of integral histograms cannot be used and would further prove impractical due to other reasons.

First, when individually normalizing the shape of an MSER, each MSER region undergoes a different individual normalization step. The integral information is lost since it is not applicable to another MSER. As consequence the cost of building an integral histogram does not reach the point of amortization.

Second, the cost for building an integral histogram pays off according to Grabner et al. after describing about 200 interest points [24]. Even if a global integral histogram could be used, the generally low MSER detection rate especially during the tracking prevents crossing the point of amortization. Further, the *frontal MSERs* selected for object representation are highly unlikely to all originate from the same frame. Thus the underlying image data is definitely different and again provides no common grounds in terms of image area for building integral histograms.

Thus the ideas described by Grabner et al. cannot be combined with the MSER detection, tracking and its individual affine normalization. The case is similar for the SURF process by Bay et al.. Therefore a plain SIFT description consisting of solving rotational ambiguity through orientations histograms and description is implemented.

The scale-invariance due to the scale-space normalization is missing but is replaced by the adaptive size of the orientation histograms. The first idea is to normalize the image patch to a fixed resolution. However, this step requires additional computation. The second idea is to tweak the affine normalization to include a scale normalization into the process. Yet this is step is not required since regardless of the size of the image patch always an equal number of pixels are placed in each description bin. Thus the SIFT description process is itself a form of scale invariance – given the details of higher resolution image patches do not introduce further strong diverting gradients.

One idea which is used from the *Fast Approximated SIFT* concept is the Sobel operator [11] as replacement of the difference-of-gaussian (DoG) for gradient calculation. This discrete approximation provides sufficient detail and satisfies the requirement for low computation time.

3.3 Object Recognition

At this stage an object and its trajectory have been tracked and a robust, distinctive and compact object representation exists. This description is used in training to learn an object representation and in testing to identify an unknown object. The identification requires that the SIFT descriptors are compared to all other object representations providing a distinguishing vote for the desired object.

For this purpose the vocabulary tree by Nistér and Stewénius [63] is used to store and retrieve the descriptors and object information. This approach generates a tree data structure which borrows ideas from text retrieval systems. The two main benefits of the hierarchical structure are the minimal computation requirements for inserting new objects and for matching of unknown objects against the entire vocabulary tree. Second, the number of objects stored in the data structure does not affect the recognition time sig-

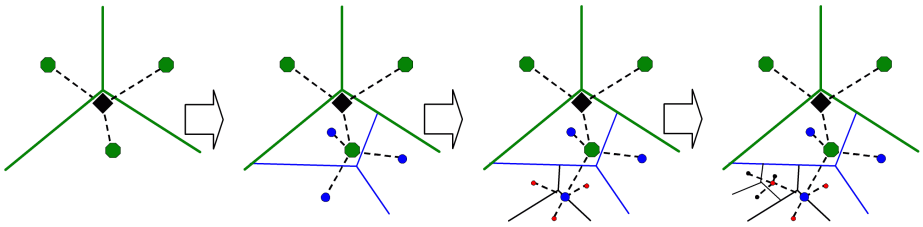


Figure 3.13: An example vocabulary tree for three cluster centers and a depth of four levels. Each hierarchical level contains a part of the previous data and refines the clustering detail [63].

nificantly. Thus the same approach is ready to be extended to a much larger learning and recognition system containing thousands of objects without performance loss.

3.3.1 Vocabulary Tree

The vocabulary tree is an efficient representation of the clustering of description vectors. The approach uses k-means clustering for each level of the tree. This achieves a hierarchy of clusters which again is used to efficiently traverse the vocabulary tree and find matching cluster centers.

In its definition a vocabulary tree is a data structure of k cluster centers and a depth of l levels. Figure 3.13 shows an illustration from Nistér and Stewénius of a vocabulary tree built for three cluster centers and a depth of four levels. For each new level the data clustered to the number of centers and divided. A new level of clustering provides more detailed quantization of the descriptors.

The cluster centers are referred to as nodes of the tree and the nodes at the last level are known as leaves. Each of these nodes contains an inverted file list. This list maintains an index to the objects whose descriptors are included in the respective nodes. So instead of holding the actual descriptors themselves, only a correspondence between best matching node and object identifier is available.

Further each node contains a weight based on entropy. The more objects are included in a node the less distinctive it becomes. Nistér and Stewénius define various voting strategies for retrieval. First, the *flat strategy* defines a scoring where only the leaf nodes are used. If a descriptor of an object matches to a node in the lowest level, its weight is included in a sum later normalized by the number of descriptors in total. Second, the *hierarchical strategies* define scoring based on how many levels upwards from the leaf level are also considered during scoring. The first strategy is fast, while the second one improves the recognition rate.

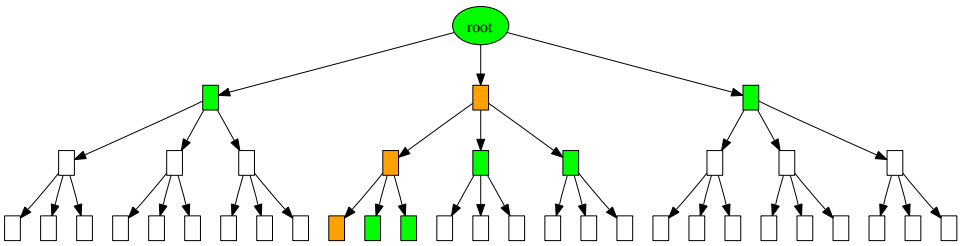


Figure 3.14: Illustration of the hierarchical traversal through a vocabulary tree: Only the nodes in green are considered at each level and the orange nodes indicate the best matches. These determine the k of k^{l+1} cluster nodes which are considered at the next level $l + 1$.

The weight w_i of a node is defined as

$$w_i = ln(\frac{N}{n_i}). \tag{3.6}$$

where the total number of objects N in the vocabulary tree and the number of objects n_i which are contained in a node i are used as entropy measure.

The final score s is determined by the sum over all nodes where the query descriptor matches this node. The frequency of matches for each descriptor is used and normalized by the total number of descriptors – for the query object and the already known objects in the vocabulary tree. The final score is then defined as

$$s = \sum_i \frac{w_i * q_i * d_i}{Q * D} \tag{3.7}$$

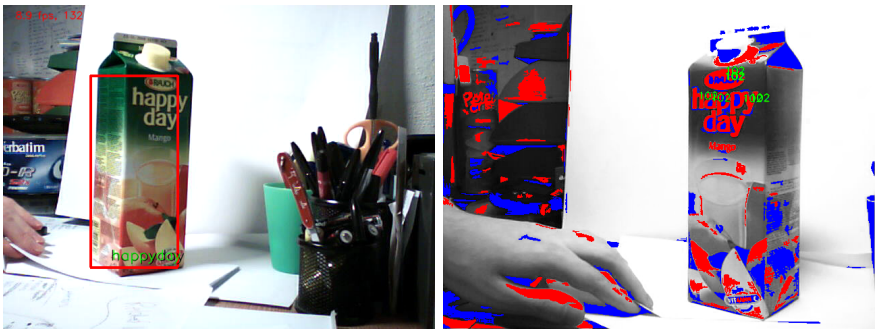
where w_i represents the weight of the current node, q_i and d_i the number of times a descriptor for a query or database object passed through the current node, and Q and D are the total number of query or database descriptors respectively.

3.3.2 Online Insertion

The hierarchical design of the vocabulary tree allows for a fast insertion of new objects. For each of its description vectors the top nodes and their cluster centers are matched. Only the children of the best matched cluster center are then matched again. This reduction of search space allows for a complete search of the vocabulary in k * l comparisons. Thus for a structure of ten cluster centers and six levels searching the one million leaf nodes for a best match only requires 60 comparisons, see Figure 3.14 for an illustration of this process.

During the insertion of a new object this advantage is used to find the best matching leaf node quickly. For each of the SIFT descriptors such a match is sought. Then, a new object identifier is included into the nodes' inverted file lists and their weights are updated. No further steps are required.

(a) Scene A (b) Scene B



(c) Scene B (d) Full Frame

Figure 3.15: Illustration of recognition examples: a) Tracking of an object showing the individual bounding boxes around the identified MSERs. b) and c) are two frames of a recognition test sequence where initially the object is unknown but then is correctly identified once enough feature information has been collected. In the example d) the localization is shown by a full scene analysis (both types of MSER indicated in colors) and the correctly matched MSERs highlighted by the numerical values.

3.3.3 Online Retrieval

The same hierarchical matching is used to determine the best matching nodes for retrieval. Due to the lower computational expense only *flat scoring* is used and no levels other than the leaf nodes are considered during scoring. This provides a less accurate score at a much faster speed.

The retrieval result is a list of objects which matched the query object in respect to the nodes the objects share. If a query object matches a node, all

objects in its inverted file are possible retrieval matches and are considered. The list of objects contains the final score over all matched nodes.

Figure 3.15 shows some recognition examples where either tracking or full screen recognition is applied. Figure 3.15 a) demonstrates the tracking including the individual bounding boxes combined to the global one. Figure 3.15 b) and c) are two frames during tracking. First, the object in the scene is known and in the second frame it is correctly identified when enough feature information has been collected. Figure 3.15 d) is an example of a full frame recognition. MSERs in the entire scene are detected and their successful matches to previously learned objects highlighted by numerical values.

3.3.4 Confidence Measurement

The final step of the recognition system is evaluating the score retrieved through the vocabulary tree. The score provides a measure how many nodes and SIFT descriptor are successfully matched for each considered object. Most of the objects in the vocabulary tree are also in the final list. This means at least one node's cluster center is matched in whose inverted file list this object resides. While a simple maximum score selection is the straightforward approach, it is not expedient in this case since we seek a measure showing the accuracy of the recognition decision.

Due to the tracking more information is acquired the longer an object is tracked. Initially only a few features are tracked and described. Thus the retrieved score is based on low number of features as well. The two benefits of tracking now show their effect in score. First, the longer the tracking, the better the selected *frontal MSER* and its compact the object representation. Second, the longer the tracking the more features are visible resulting in more *frontal MSERs*. This is discussed in detail during the experiments in Section 4.

The idea is to create a confidence measure which evaluates how stable and accurate the recognition is based on the score retrieved from the vocabulary tree. The proposed measure is defined as

$$confidence = \frac{highest\ score}{second\ highest\ score} \tag{3.8}$$

where the highest and second highest score are selected from the list of scores returned by the matching process of the vocabulary tree.

This distance ratio determines how similar the top two scores are. If there is enough distance between these the recognition is highly likely to be correct. To determine such a threshold the experiments includes a setup where this confidence measure is evaluated. In those cases the highest score is taken to be correct score. Thus a confidence value larger than one means the recognition decision is correct.

Chapter 4

Experiments

In this chapter the benefit for learning through the tracking approach is evaluated. The tracking and building of object representations is compared to single view approaches and various motions to prove the increased performance for robust online recognition. The outline of the experiments are as follows.

In Section 4.1 the configuration of the evaluation framework which is part of the online learning and recognition system is described along with the way learning and recognition are handled. In Section 4.2 the videos for training and testing as well as the ground truth for the vocabulary tree are presented. Sections 4.3 to 4.6 describe the four types of sub-experiments. First, the recognition capabilities of the trajectory tracking is compared to a single view full frame test scene and a region of interest (ROI) of the test scene. Second, the progress of the recognition score during the course of tracking is evaluated to derive a measurement for confidence. Third, the confidence decision is evaluated on the test scenes. Finally, the online performance in terms of execution time of the learning and recognition system is analyzed.

4.1 Evaluation Framework

The robust online learning and recognition system introduced in Section 3 is used in an offline fashion to ensure repeatability. The processes remain the same except for the image source which is provided through pre-recorded video files. For each task a video of an unknown object is presented together with an initialization bounding box which defines the ROI around the object. During the course of the experiment the same videos with the identical bounding boxes are used for each type of experiment. More details about the training and testing setup is discussed in Section 4.2.

4.1.1 Learning

During the tracking of the video all detected MSERs and their trajectories are recorded. *Frontal MSER* selection during the tracking provides the compact summarization as well as affine normalization. In the learning task one entire video is analyzed per object and at its end there is a one time insertion step. Here the SIFT descriptors of the final robust subset of summarized trajectories are inserted into the vocabulary tree as a new object. Details for the construction of the vocabulary tree are discussed in Section 4.2.

4.1.2 Recognition

In the recognition task the entire video is equally analyzed, however at certain intervals a recognition step is performed. Depending on the required response rate these intervals range from once per frame to every 20th frame. The recognition step matches SIFT descriptors against the learned objects contained in the vocabulary tree. Depending on the type of recognition approach the SIFT descriptors are obtained through the proposed tracking system, or in the case of one-shot recognition by a one time MSER detection and SIFT description step.

The resulting score is based on the weighting scheme of the vocabulary tree introduced in Section 3.3.1 as combination of weighted nodes and their matching frequencies. This value is further evaluated and converted to a confidence measure which determines the certainty of the recognition result.

4.2 Training and Testing Data

The set of videos used for learning and recognition show five different objects. Figure 4.1 shows the first frames of several sequences for the five objects. The entire set consists of 34 videos of which five are used for training (one per object) and 29 for testing. The five videos used to track the unknown objects for training are similar to the remaining videos in respect to arbitrary motion, lighting conditions and sequence length.

Each of the objects has a range of unique visual features. However, at the same time the objects share similar aspects such as letters, symbols and shapes. All objects contain text on their surfaces and some objects share the same letters in a similar font. Further object 1 as shown in Figure 4.1 a) to d) contains the same brand name as object 4 shown in Figure 4.1 m) to p) and object 5 shown in Figure 4.1 q) to t).

Figure 4.1 also illustrates the range of viewing and lighting conditions. In each sequence the object undergoes motion in an arbitrary way including a combination of rotation around the y-axis and in-plane, translation and shearing. Scaling is intentionally avoided to ensure the concept of proper

frontal MSER selection. Thus the motion is performed at a similar distance from the camera position.

Due to the goal to evaluate the combination of tracking and trajectories only a small number of objects are investigated. A data structure such as vocabulary tree however is designed for a large number of objects. To approximate more realistic test conditions the vocabulary tree is clustered and filled with random objects from the UK Bench database [63]. In total 100 images are used which correspond to 25 objects with four different viewpoints each. This does not provide an optimal setup. However, the focus is set on evaluation of the trajectories and not the size of the database.

Each image provided on average 230 MSERs and 450 SIFT descriptors. Similar values – 210 *frontal MSERs* with 330 descriptors – are retrieved on average for each object during tracking. The vocabulary tree structure is build for nine clusters and four levels. Using the *flat scoring strategy* an average top score of 3.54 (88.5% correctly recognized objects) is achieved which is similar to results by Nistér and Stewénius for a database of this size.

4.3 Experiment 1 - Recognition Methods

This experiment evaluates the recognition performance comparing the results obtained by tracking objects to single frame recognition. Each of the five test objects has been tracked through one video, summarized, described and inserted into the ground truth vocabulary tree.

The recognition rate is evaluated in five setups where the tracking results are compared to four single image based variants. The only difference is the method of interest point extraction. The same tracking and trajectory summarization is used to extract the features on the remaining 29 videos.

Tracking (EOF) - Recognition is carried out after the entire video.

Tracking (100th) - Recognition is performed after the first 100 frames.

Frontal frame - A full frame of the object in its dominant frontal position.

Frontal ROI - The same frame is cropped to a ROI around the object.

Non-frontal frame - A full frame with a non-frontal view is used.

Non-frontal ROI - A ROI around the object in the same non-frontal view.

The frames are extracted at random from the 29 videos with the only distinction between frontal and non-frontal views. The comparison of full frame to a cropped ROI has the main intend to provide an equal basis to the tracking approach. Since both of these are initialized with a bounding box roughly separating the object from its background.

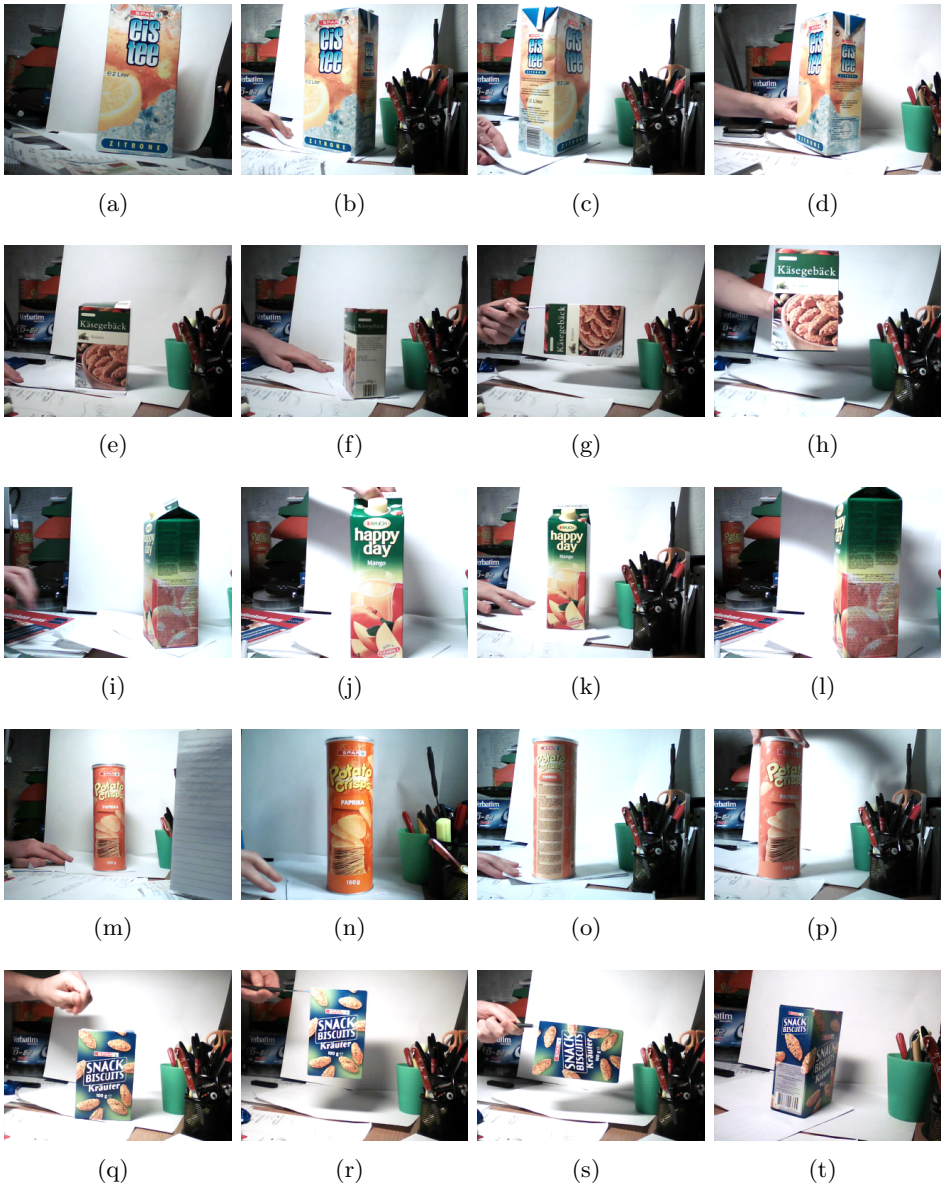


Figure 4.1: First frames of a subset of 34 the video sequences used in training and testing. For each object one video is used to train its object representation and the remaining 29 videos are used in testing.

	Tracking		Frontal		Non-frontal	
Objects	EOF	100th	Frame	ROI	Frame	ROI
eistee	100%	83%	100%	100%	25%	25%
geback	100%	100%	50%	50%	0%	0%
happyday	86%	83%	75%	75%	0%	0%
pringles	100%	100%	0%	0%	50%	50%
snack	100%	57%	100%	100%	100%	100%
Total	97%	83%	69%	69%	42%	42%

Table 4.1: A comparison of tracking against single frame recognition. The percent of correct recognition is shown for the full video (EOF), after the 100th frame, for a full frame and a ROI of a frontal view and a non-frontal view of the object.

Table 4.1 summarizes the results of this experiment. The recognition rate is determined by the relative number of correctly identified objects as best match. The columns from left to right represent the results for tracking through an entire video of roughly 300 frames (EOF) and the recognition rate after 100 frames (100th). The next four columns show the performance for the frontal or non-frontal views provided by single images.

The advantage of the object tracking is clearly visible as 97% of the presented sequences are correctly identified using the entire video. The analysis of frontal views shows that enough information is available to match 69% of the test scenes. Since learning only uses the *frontal MSERs* to create an object representation, the dominant frontal view of an object provides the undistorted appearance of these MSERs even without tracking. However, features on other sides are hidden which explains the lower recognition rate compared to the tracking.

The second column for the tracking results shows the performance after 100 frames. This value was selected to provide a balance between those sequences starting with a dominant frontal view and those starting with a non-frontal view. For optimal performance the tracking approach requires the object to be visible in a close to frontal view. Sequences starting with such an approximated frontal view have a better initial recognition performance than those starting with non-frontal views. The effects of such motions is discussed in detail in the following experiments.

The low performance of the non-frontal views is explained by the lack of fronto-parallel features all together. The subset of test scenes such as Figure 4.1 d), f), i), o) and t) show the object from two sides, except for o) where only the backside is visible. In the other cases none of the visible features is in a frontal position and thus may only achieve a similar description by means of the affine normalization. However, this does not seem to be sufficient to recognize objects learned purely through *frontal MSERs* as less than half

are detected correctly.

Interesting are the identical scores of a full scene and a ROI selection of the object. For all test images the same recognition is achieved for both types. This may be explained by the lack of distinguishing features on the background. Typically only 20 more MSERs are detected between the two types. None of these additional MSERs show a significant effect on the recognition score.

As shown in this experiment tracking provides a valuable benefit when extracting and describing an object's features. More information is gained, the learning experience is improved and successfully used in recognition. The combination of tracking during learning and recognition provides a significant advantage during recognition while maintaining fast processing.

4.4 Experiment 2 - Recognition over Time

In this experiment the progress of the recognition score is investigated in terms of time. The expected result is that the longer an object is learned, the better is its recognition score.

The goal of an online robust learning and recognition system is the ability to cope with arbitrary motion of the object. To reflect this situation the video sequences have been recorded in a similar fashion. As consequence there are many fluctuations in the resulting score due to the motion.

The following sections provide a detailed analysis of the main evolution variants of recognition score over time in four sub-experiments. First, a motion to and from a frontal view; second, a motion starting at a frontal position and third, a motion towards a frontal position are shown. Forth, an analysis of unstable matching with few trajectories is discussed.

The purpose of these experiments is to show the effect of learning through tracking in detail. The previous section showed that tracking provides a significant advantage. Now the question is investigated how much tracking is required to learn an object sufficiently. For this, the experiments demonstrate the evolution of the score and the selection of *frontal MSERs* over time.

4.4.1 Pure Frontal Rotation Motion

In the sequence illustrated in Figure 4.2 there exists only a rotation of the object from a non-frontal to a frontal and again to a non-frontal view. The evolution of the recognition score is shown in Figure 4.3 a). The general progress of the score is as expected and the final score is more than twice that of the second best score.

There are three interesting parts in Figure 4.3 a). First, the spike at frame 100 and the subsequent drop to a much lower score. This is explained through Figure 4.3 b) which shows the evolution of the size of the tracked MSERs. For illustration purpose only the 25 trajectories with a minimum

(a) 0 (b) 50 (c) 100 (d) 150 (e) 200 (f) 250 (g) 300 (h) 350

Figure 4.2: Pure Frontal Rotation Motion: Selected frames of the video sequence showing the rotation. Around frame 200 the object is presented at its dominant frontal view.

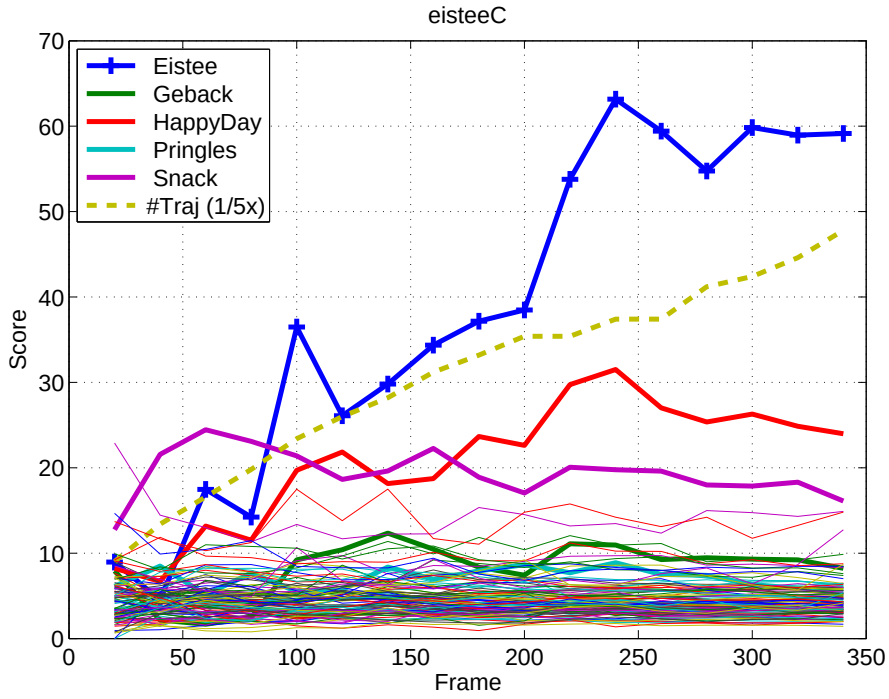
length of 150 are drawn, which are roughly 15% of the considered robust and 3% of all trajectories. At frame 70 two trajectories arrive at their maximum size and thus frontal position, as indicated by the circle. Additionally, two more trajectories commence their tracking, as indicated by the cross. At frame 100 these two new MSERs arrive at a relatively stable size. That means, at least four new frontal MSERs are available for recognition. This boosts the score to the new peak.

The slight drop afterwards is due to new trajectories which do not resemble the best frontal view but which are taken into account when normalizing. Another aspect may be the case when a good frontal MSER is incorrectly matched and the new frontal MSER is no longer part of the representation of the learned object. This is discussed in detail in Section 4.4.3. In Figure 4.3 a) this effect is seen at frame 120 when the correct score drops and the score of another object increases suddenly.

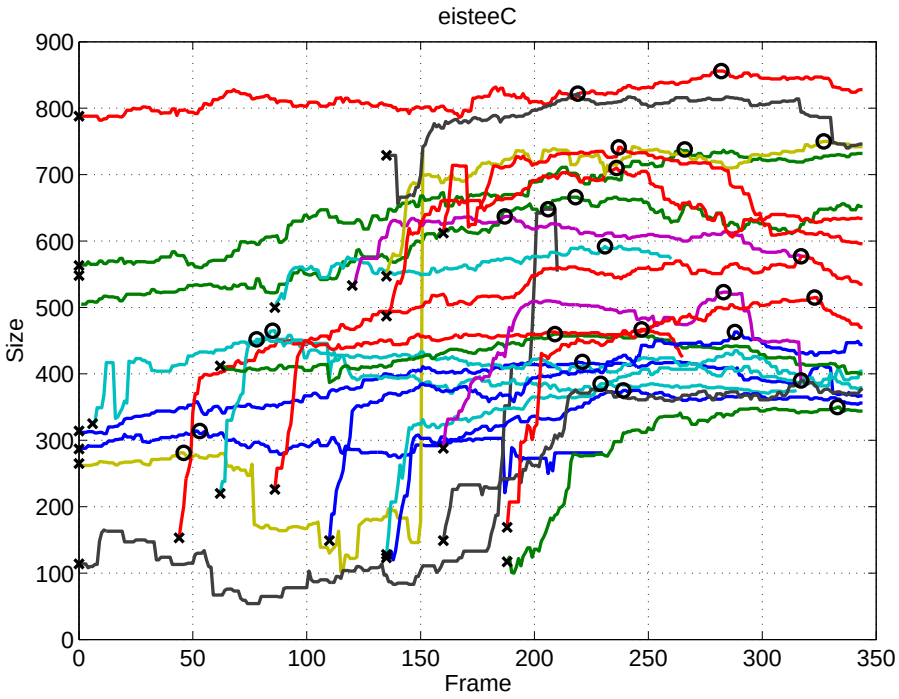
The second interesting part of Figure 4.3 a) is the distribution of scores of the unrelated objects. The thick lines which are also shown in the legend are the new objects learned through trajectory summarization. The thin lines indicate the scores for UK Bench images. When the number of robust trajectories is still low, as illustrated by the dashed line, the few matches which occur during the vocabulary tree matching process have a much greater influence on the score. This explains why another object has a higher score than the correct object up to about 100 robust trajectories. This is also a common situation in other video sequences where the correct recognition also receives the highest score after at least 100 robust trajectories are available.

The third effect which is visible in Figure 4.3 a) is the clustering of the UK Bench images at the lower spectrum of the score while four out of the top ten scores belong to the newly learned objects. The content of the two image types varies greatly and thus provides an advantage.

As shown in this experiment, the learning through tracking effect is very prominent. However, too few or non-robust trajectories have a negative effect on the final recognition score. The tracking approach thus requires a certain minimum of robust trajectories to successfully recognize the object.



(a) Score



(b) Size

Figure 4.3: Pure Frontal Rotation Motion: a) The recognition score and b) the size of the tracked MSER both in relation to the frames. *Frontal MSERs* and new detections are indicated circles and crosses respectively. The concentration of *frontal MSERs* between frames 200 and 250 causes the increase in score.

(a) 50 (b) 100 (c) 150 (d) 200 (e) 250 (f) 300 (g) 350 (h) 400

Figure 4.4: Rotation Motion Beginning At Frontal: Selected frames of the video sequence showing the initial dominant frontal view up to frame 150 and then a side view.

4.4.2 Rotation Motion Beginning At Frontal

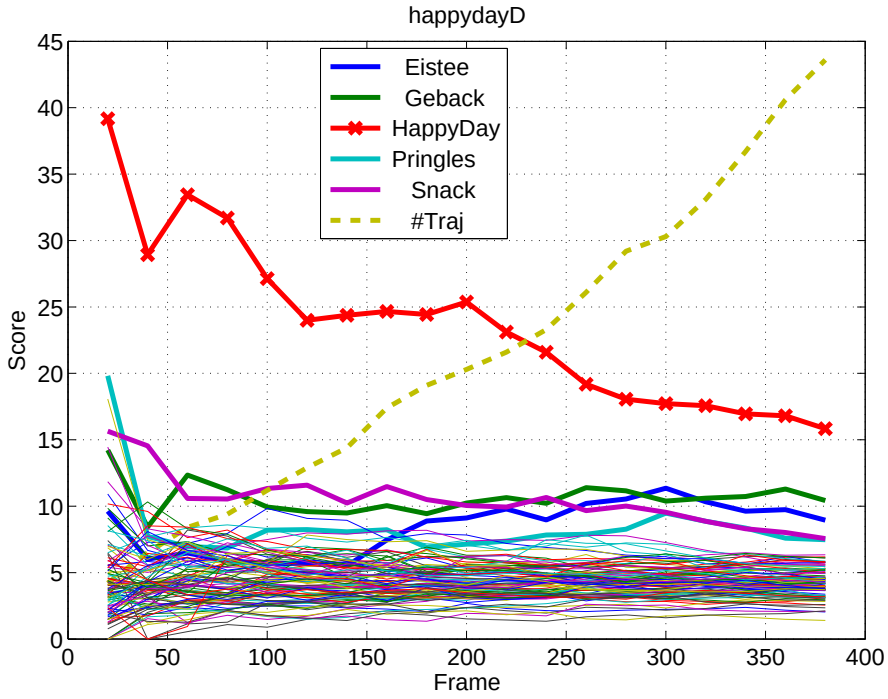
In the sequence illustrated in Figure 4.4 a frontal view is present from the start of the sequence for 150 frames undergoing only translation. Afterwards the object is rotated and a side view is parallel to the camera. The distribution of the recognition score is shown in Figure 4.5 a). The progress of the score is opposite of what the learning through tracking concept states since the recognition score seems to drop the more frames are seen. Please note that Figure 4.5 a) shows the recognition score normalized by the number of SIFT descriptors and Figure 4.5 b) shows the raw score without normalization. Here the benefit of learning is more clearly demonstrated. The more time is spent learning, the more information is collected and consequentially the recognition score improves.

The disconcerting decrease of the score is explained by the steady rise of robust trajectories and SIFT descriptors. Since the scoring scheme within the vocabulary tree includes a normalization by the number of descriptors for each object, the more trajectories are used the smaller is the normalized score.

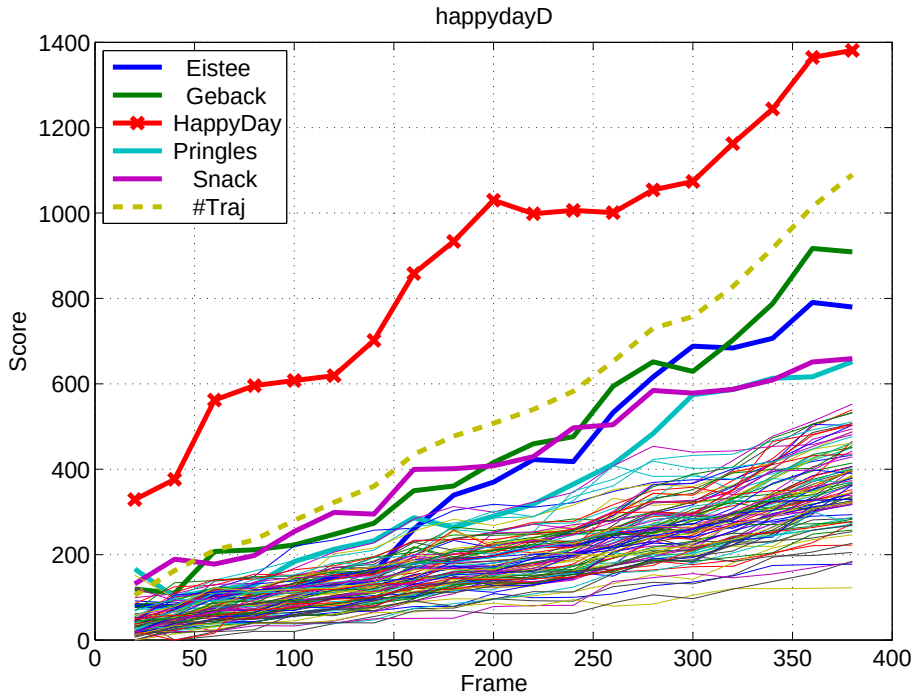
Between frames 120 and 200 the raw score as well as the trajectories increase resulting in a steady normalized score. The reason for the increase in raw score between these frames is the slight rotation which brings the object in the ideal frontal view exactly parallel to the camera. Before these frames the view was also mostly frontal but at a slight angle.

The further decrease in the normalized score after frame 200 is due to the new trajectories which do not provide much distinctive power during the recognition. This is visible in Figure 4.5 b) where the raw score remains unchanged after frame 200. This again makes sense since the next 100 frames of the sequence mainly show one side of the object which contains many paragraphs of text. As discussed in Section 2.1.2 if text is too small, its individual letters cannot be identified [19]. The text is thus not robustly detected and tracked resulting in unreliable *frontal MSERs*.

As shown in this experiment, it is the normalization which produces most



(a) Normalized Score



(b) Raw Score

Figure 4.5: Rotation Motion Beginning At Frontal: a) The recognition score after normalization by the number of trajectories and b) the raw score is increasing because of the better object representations.

(a) 25 (b) 35 (c) 45 (d) 55 (e) 65 (f) 75 (g) 85 (h) 95 (i) 105 (j) 115

Figure 4.6: Unstable Matching with few trajectories: The MSER is tracked robustly at first, then at frame 95 its stability is no longer sufficient. However, at frame 115 a slightly different MSER is matched again which produces a very different affine patch and thus SIFT descriptor.

of the decreases in score due to the detection and tracking of features which are not distinctive during recognition. These types of less repeatable MSERs arise due to too small text. In this case recognition is neither advanced nor hindered by them. However, the next experiments provide a detailed analysis of such unstable *frontal MSERs*.

4.4.3 Unstable Matching with few Trajectories

This experiment shows the effect when too few trajectories are used for recognition. This is the case when only trajectories are selected which have a high tracking quality. The following is an in depth analysis of the scoring within the vocabulary tree which produces this effect.

The distribution of the recognition score is shown in Figure 4.7 a) with four interesting sudden changes. At the frames 73, 98, 103 and 117 the score increases by 30 and 22 units; then decreases again by 25 and 23 units respectively. The remaining distribution is slightly fluctuating as usual. What causes this effect is visible in Figure 4.7 b). A single trajectory is responsible for the drop of the score in frame 117. The evolution in size and a rate of size change are shown. At frame 117 the tracked MSER recovers from a period of unstable matches, as shown in Figure 4.6, from frames 92 to 116 indicated by the zero change in size. The change in size at frame 117 moves the *frontal MSER* from the previous location at frame 73 to this frame, as indicated by the black circles. The gain at frame 73 and loss at frame 117 of this valuable *frontal MSERs* – and similar for frames 98 and 103 with another MSER – causes changes in score by 70%.

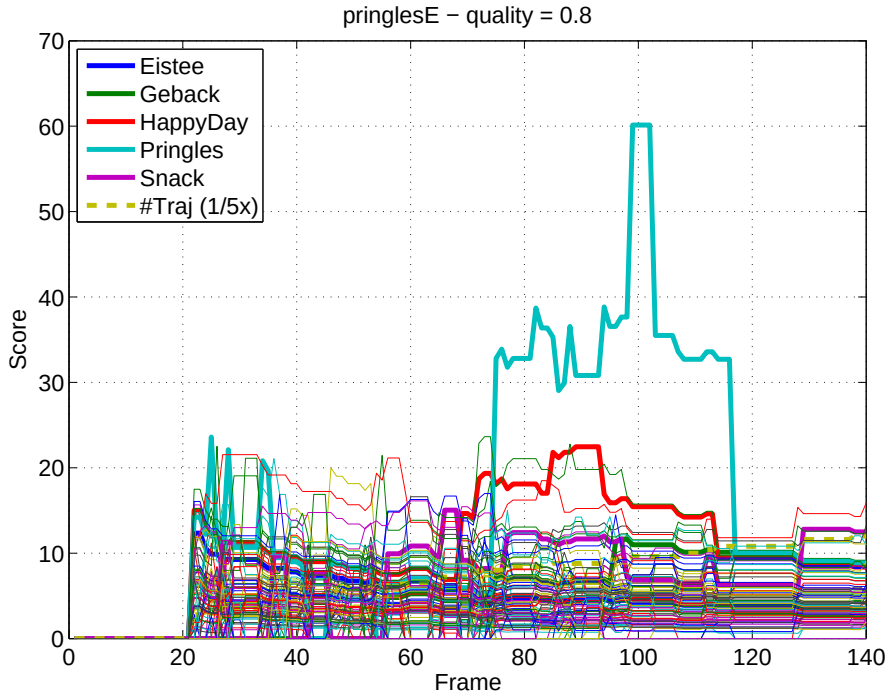
A further analysis reveals two problems causing this drop. First, only nine of the 162 descriptors at frame 116 are matched to nodes corresponding to the correct object. That means less than 6% of the SIFT descriptors are successfully matched within the vocabulary tree. Second, the single unstable match between frame 116 and 117 makes up 30% of the score. This is due to – in this example – an unusually high concentration of descriptors in this

node. The other descriptors spread to six other nodes while three matches are in the same node. This combined with four matches of the correct object's learned descriptors becomes an unhealthy mix. The previous weight of 12 (4 database descriptors x 3 matches) is decreased to 8 (4 database descriptors x 2 matches) for frames 116 and 117 respectively. Since the number of images in these nodes is roughly the same, the entropy weighting is equally similar. The final score is 18x the nodes' weight for frame 116 and 12x for frame 117. This enormous difference creates the 30% drop in score.

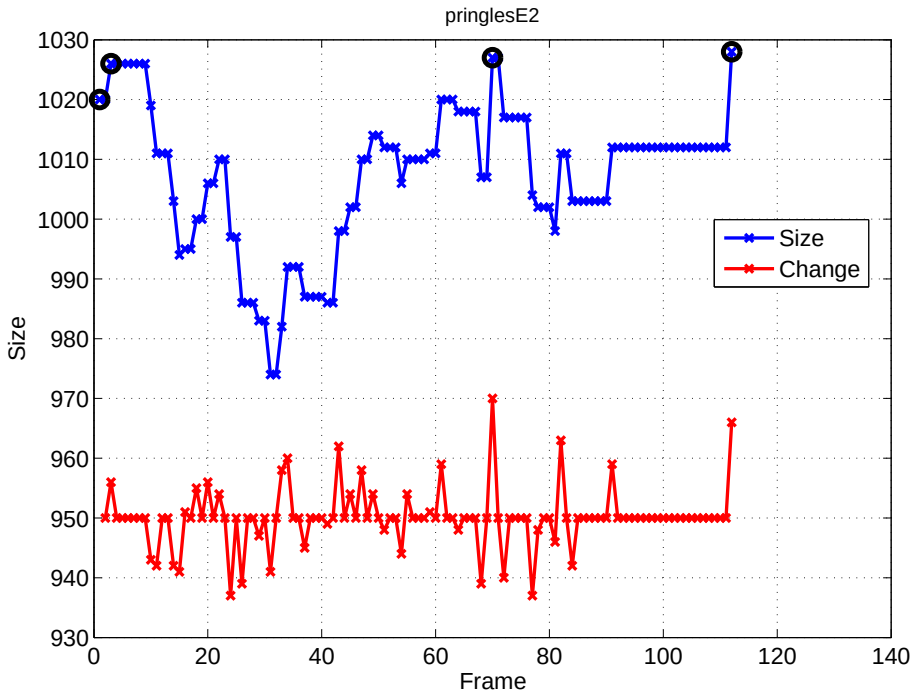
Now, this highly instable unwanted behavior is caused by the two underlying factors. First, only very stable trajectories were used in constructing trajectories resulting in far shorter less continuous trajectories. Figure 4.8 a) and b) show the same video sequence including more trajectories by changing the quality threshold from 0.8 to 0.5 in a) and even to 0.1 in b). This results in a large increase of trajectories and consequentially much better recognition. In these graphs the same increases and decreases in score are still present but their effects are diminished. And second, sudden changes in *frontal MSERs* are not wanted and additional quality evaluations should be introduced to ignore such changes at the end of a trajectory.

This experiment showed the adverse effect an unstable *frontal MSER* selection may have if combined with few robust trajectories and very few SIFT descriptor matches in the vocabulary tree. The current selection based on maximum size provides a good base for frontal selection, but its stability is not optimal and may be increased by averaging the binary MSER representation of the neighboring frames. This moderates the effect small changes have on the affine normalization resulting in more stable underlying image patches for SIFT description. Please note that these are individual cases and occur rarely in comparison to the next issue.

The low percentage of matches between the descriptors in the vocabulary tree is not rare. Schindler et al. [73] addresses this issue proposing a greedy algorithm to evaluate multiple nearest matching cluster centers at once. This approach provides a significant increase in recognition rate and its principle is similar to the 'best-bin-first' mechanism used by Beis and Lowe in their approximate nearest neighbor search [4].

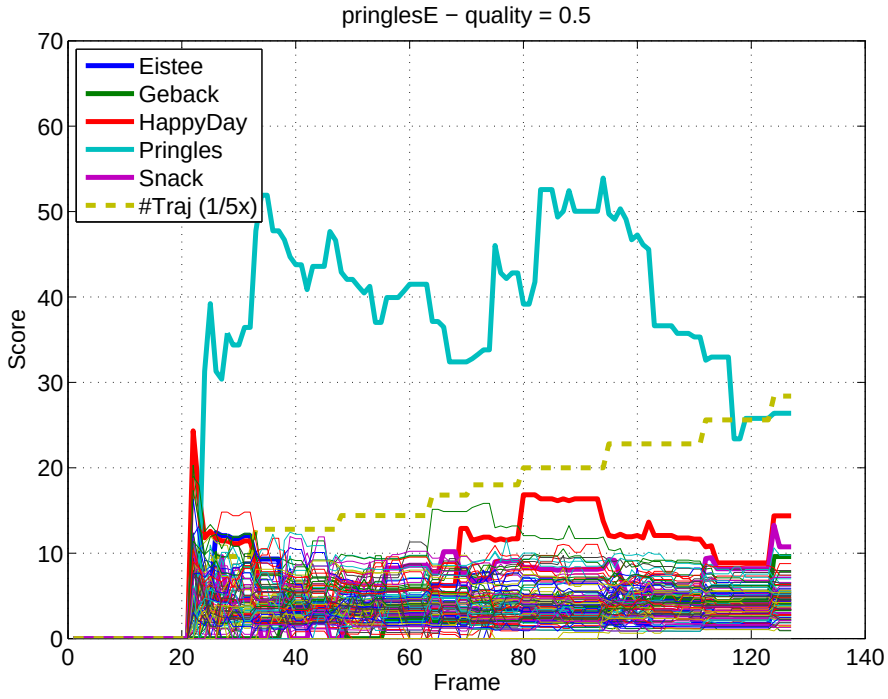


(a)

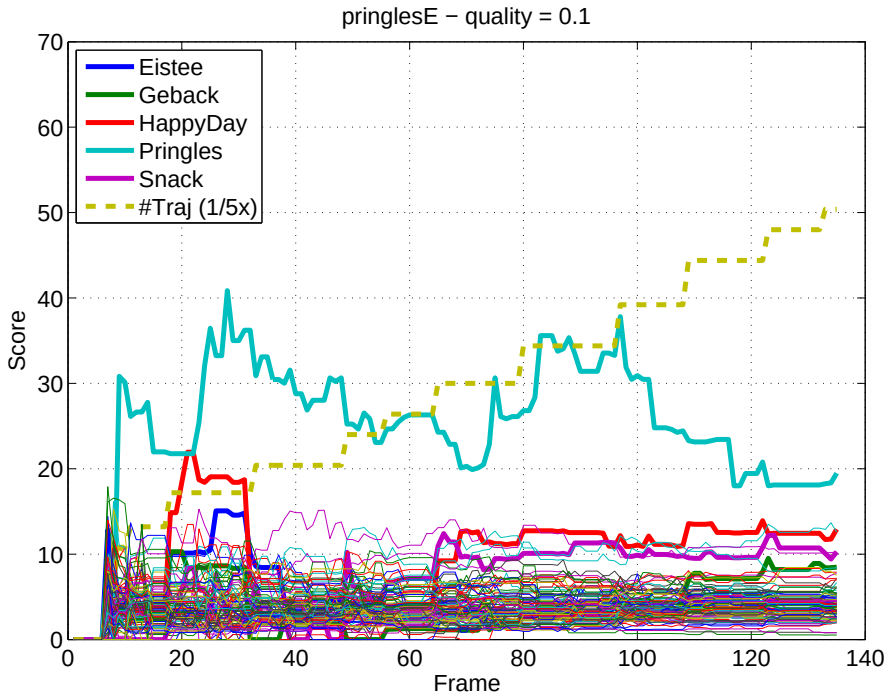


(b)

Figure 4.7: Unstable Matching with few trajectories: a) shows the recognition score using only tracks with at least a quality of 0.8 b) the size analysis of a single MSER trajectory responsible for the large changes in score. This MSER is also illustrated in Figure 4.6.



(a)



(b)

Figure 4.8: Unstable Matching with few trajectories: a) and b) again show recognitions scores using all tracked MSERs with at least a quality of 0.5 or 0.1 respectively. The effect diminishes since more trajectories are used.

(a) 0 (b) 50 (c) 100 (d) 150 (e) 200 (f) 250 (g) 300

Figure 4.9: Towards Frontal View Rotation: Selected frames of the video sequence showing the initial view and the rotation towards the dominant frontal view around frame 150.

4.4.4 Towards Frontal View Motion

In this experiment the sequence as shown in Figure 4.9 starts off with a non-frontal view and after half of the view the dominant frontal view is shown. After this, only a smooth translation motion is performed.

The eviolution of the score as illustrated in Figure 4.10 reflects this motion very closely. The initial view does not provide distinctive *frontal MSERs* for recognizing this object. The score behaves similarly to the preloaded UK Bench objects while other learned objects receive a slightly higher score. Starting with the detection of MSERs from the dominant frontal view, the score steadily increases up to a steady level.

This experiment showed the need for *frontal MSERs* during tracking to achieve comparable object representations. If features are initially already shown fronto-parallel, they are selected as *frontal MSERs*. If such an optimal view is not available only non-frontal features are selected and further approximated by affine normalization. These may be sufficient to correctly recognize an object. As demonstrated here the recognition only commences to be successful once the selected *frontal MSERs* are shown in a view which can be approximated by affine normalization to the optimal frontal view. Even though these MSERs were initially detected around frame 100, they are only shown in such a similar view around frame 140. This causes the rise in score at this point and is so prominent due to the low discriminative text on the initially visible side of the object.

4.5 Experiment 3 - Confidence

In this third experiment the influence of the tracking length on recognition is analyzed. This is done by evaluating the confidence measure introduced in Section 3.3.4. The analyzed confidence value indicates how much higher the correct score is with respect to the second highest score. If the confidence falls

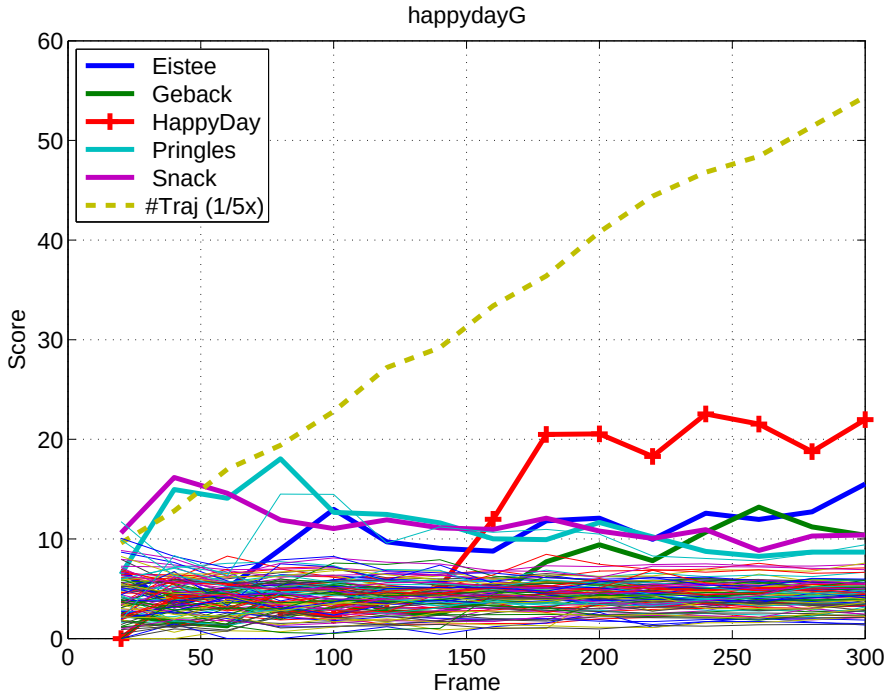
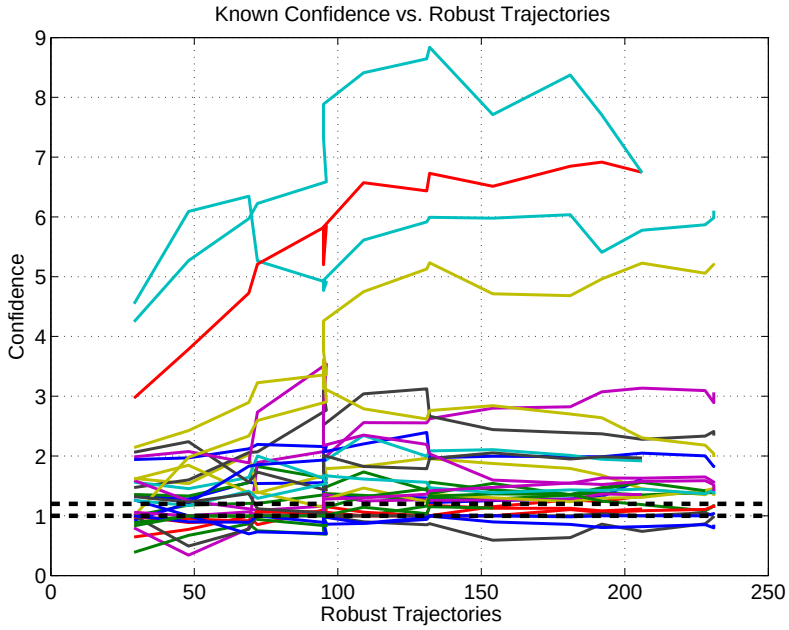


Figure 4.10: Towards Frontal View Rotation: The recognition score with the dominant frontal view around frame 180.

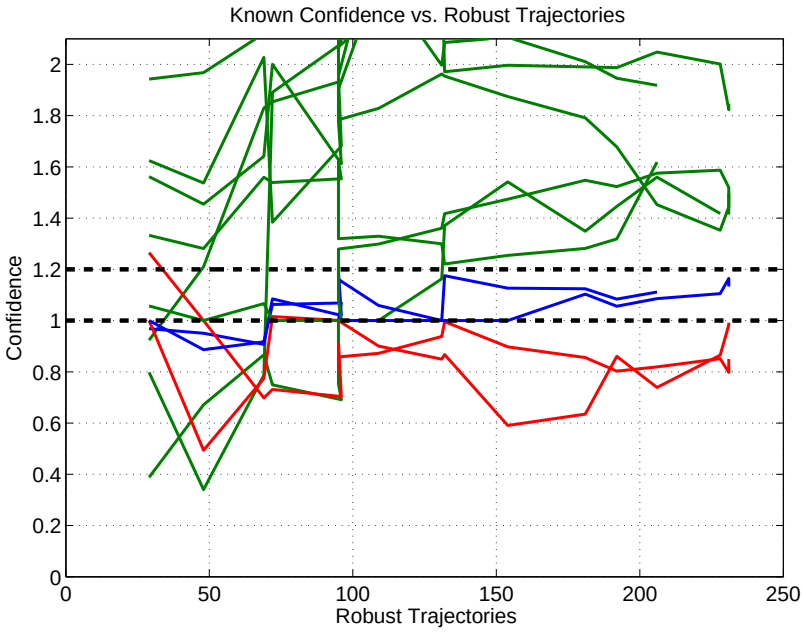
below the value of one, the recognition is incorrect since another object has a higher score. This level is considered the minimal confidence for recognition.

Figure 4.11 a) shows this confidence against the number of robust trajectories for all 34 video sequences, whereas b) provides a characteristic subset for more detail. Two main aspects are visible. First, most of the test sequences already achieve the minimal confidence leading to correct recognition within the first 30 frames. Second, when at least 100 robust trajectories have been tracked the confidence rises for all except for two test sequences to above the minimal confidence, as indicated by the dashed line. This indicates that enough feature information is collected to successfully identify the objects. The reason that two test sequences are not identified is due to their video quality. In these sequences the object is rotated much more quickly than in the other sequences. This results in motion blurring and decreases the number of detected and correctly selected *frontal MSERs*.

Due to the previous experiments it is also known that a certain minimum number of trajectories is required to provide a reliable basis for recognition. This experiments shows that 100 trajectories provide a very sound basis which is not required for every sequence. Thus for a robust online recognition



(a) All test sequences



(b) Characteristic subset

Figure 4.11: Confidence vs. Robust trajectories: These graphs show the confidence measure in relation to the number of robust trajectories. a) for all 34 video sequences and b) for characteristic subset. For these video sequences 100 robust trajectories are enough to correctly recognize all except two objects (red lines).

two conditions are evaluated, of which one must be achieved.

- a strict confidence value of at least 1.7, or
- a minimum confidence value of 1.2 and at least 100 robust trajectories.

The first condition enables recognition when a very high score is retrieved and distance measure between the top scores is large. The second condition ensures a stable basis for the recognition decision is available and only requires a confidence slightly higher than the minimal for correct recognition. Many other confidence measures are available and it would be valuable and interesting to derive a second measure evaluating how many of the robust trajectories have reached their globally optimal *frontal MSER*. This would relinquish the need for a content-independent minimum number of trajectories and focus more on the rate of change within the robust trajectories.

However, other measures are not required as this method provides the necessary decision power to create a robust online learning and recognition system.

4.6 Experiment 4 - Execution Performance

This last section provides insight into the execution time of each task during the recognition phase. The profiling details are shown in Table 4.2 and illustrate the frame rates achieved during tracking. The measurements are an average and provide an estimation for each task. Three types of frame retrieval (A) are used. First, the OpenCV library [27] requires the most time. Second, the *Extremely Simple Capture API (ESCAPI)* [37] provides much faster access to the frames. Third, for offline testing pre-recorded videos have been used which have the fastest retrieval time. Interesting is that retrieval of the next image frame is the single most costly operation ranging from at least 46.5% to 72.3% of the respective total execution time for each of the retrieval types.

The other tasks are the *compound MSER tracking* (B) which is comprised of MSER tracking, evalutation and redetection. Building of compact object representations (C) includes the *frontal MSER* selection, affine normalization and SIFT description. The recognition task itself and the visual overlaying (D) of information are the final two aspects considered in this profiling.

Section	Task	Duration	
A (1x)	Frame retrieval (OpenCV)	120 ms	72.3 %
A (1x)	Frame retrieval (ESCAPI)	60 ms	56.6 %
A (1x)	Frame retrieval (AVI)	40 ms	46.5 %
B (1x)	MSER tracking	15 ms	9.0 %
B (1x)	MSER evaluation	15 ms	9.0 %
B (1/20x)	MSER redetection	20 ms	0.6 %
C (1x)	Frontal MSER selection	2 ms	1.2 %
C (1x)	Affine normalization	3 ms	1.8 %
C (1x)	SIFT description	5 ms	3.0 %
D (1/5x)	Recognition	5 ms	0.6 %
D (1x)	User interactions	4 ms	2.4 %
OpenCV Sum:	6.0 fps	166 ms	100 %
ESCAPI Sum:	9.4 fps	106 ms	64 %
AVI Sum:	11.6 fps	86 ms	52 %

Table 4.2: Summary of the execution time for each task involved in the online recognition system. A) Frame retrieval alternatives, B) *Compound MSER tracking*, C) Building of compact object representation and D) recognition and visualization.

Chapter 5

Conclusion

In this thesis a robust online learning and recognition system is proposed which uses tracking to improves the learning experience [87]. The method of continuously collecting more information increases the recognition rate and the compact object representation by means of *frontal MSERs* speeds up the process. The significant gain in performance is demonstrated in the experiments.

The proposed tracking system allows semi-automatic learning. The concept of local features scattered over foreground objects as well as background delivers an implicit scene representation and is robust to occlusion [81]. The advantage of being able to neglect the semantic or in this case manual segmentation step is used when the recognition begins with a full scene analysis and initializes the tracking by the retrieved matching features. This form of online learning and recognition system guides us one step closer to the capabilities of human learning and incorporating them such into service robots assisting humans in everyday life.

The online processing is possible due to three main reasons. First, the selection of state-of-the-art technologies – MSER detection, SIFT description and the vocabulary tree for storage and retrieval – provide an ideal environment. Second, the concept of *compound MSER tracking* which allows tracking of multiple MSER in combination with on-the-fly trajectory construction is a vital part during the learning process. Third, the robust trajectory selection and most importantly the efficient and optimal summarization into *frontal MSERs* minimize the redundancy of information and hence the computational time.

Though the achieved stability through *frontal MSER* selection is sufficient for online recognition, it may be improved by the notion of multiple representatives [88]. Wallraven and Bülthoff employ additional key frames when large changes in the evolution of a trajectory are detected. Further, scale-invariance for *frontal MSERs* should be implemented and evaluations of the rate of change of *frontal MSERs* within its trajectory may provide

additional recognition confidence.

The main benefit will be achieved by raising the low percentage of matches between the SIFT descriptors in the vocabulary tree. Schindler et al. [73] addresses this issue proposing a greedy algorithm to evaluate multiple nearest matching cluster centers at once. This approach provides a significant increase in recognition rate and its principle is similar to the 'best-bin-first' mechanism used by Beis and Lowe in their approximate nearest neighbor search [4].

Future applications such as a global localization by finding correspondences of features [18] will benefit from the online processing. Such an extension will also make use of the vocabulary tree's ability to work with much larger database sizes.

Building true 3D models may be an interesting next idea. Local image patches and positioning information is available and may be used to create 3D models during the tracking.

Bibliography

[1] Amit, Y. and Geman, D. (1997). Shape quantization and recognition with randomized trees. *Journal of Neural Compututation (JNC)*, 9(7):1545–1588.

[2] Baumberg, A. (2000). Reliable feature matching across widely separated views. In *Proceedings of the Conference on Computer Vision and Pattern Recognition (CVPR)*, volume 1, pages 774–781.

[3] Bay, H., Tuytelaars, T., and Van Gool, L. (2006). SURF: Speeded Up Robust Features. In *Proceedings of the European Conference on Computer Vision (ECCV)*, volume 1, pages 404–417.

[4] Beis, J. and Lowe, D. (1997). Shape Indexing Using Approximate Nearest-Neighbour Search in High-Dimensional Spaces. In *Proceedings of the Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1000–1006.

[5] Canny, J. (1986). A computational approach to edge detection. In *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, volume 8, pages 679–698.

[6] Chetverikov, D. and Verestoy, J. (1998). Tracking feature points: A new algorithm. In *Proceedings of the International Conference on Pattern Recognition (ICPR)*, volume 2, pages 1436–1438.

[7] Couprie, M., Najman, L., and Bertrand, G. (2005). Quasi-Linear Algorithms for the Topological Watershed. *Journal of Mathematical Imaging and Vision (JMIV)*, 22(2-3):231–249.

[8] Deng, H., Zhang, W., Mortensen, E., Dietterich, T., and Shapiro, L. (2007). Principal Curvature-Based Region Detector for Object Recognition. In *Proceedings of the Conference on Computer Vision and Pattern Recognition (CVPR)*.

[9] Donoser, M. and Bischof, H. (2006). Efficient Maximally Stable Extremal Region (MSER) Tracking. In *Proceedings of Conference on Computer Vision and Pattern Recognition (CVPR)*, volume 1, pages 553–560.

[10] Donoser, M., Bischof, H., and Wiltsche, M. (2006). Color Blob Segmentation by MSER Analysis. In *Proceedings of International Conference on Image Processing (ICIP)*, pages 757–760.

[11] Duda, R. and Hart, P. (1973). *Pattern Classification and Scene Analysis*. A Wiley-Interscience Publication, New York: Wiley.

[12] Dufournaud, Y., Schmid, C., and Horaud, R. (2000). Matching images with different resolutions. In *Proceedings of the Conference on Computer Vision and Pattern Recognition (CVPR)*, volume 1, pages 612–618.

[13] Everingham, M., Sivic, J., and Zisserman, A. (2006). Hello! My name is... Buffy – Automatic Naming of Characters in TV Video. In *Proceedings of the British Machine Vision Conference (BMVC)*, pages 889–908.

[14] Ferrari, V., Tuytelaars, T., and Van Gool, L. (2006). Simultaneous Object Recognition and Segmentation from Single or Multiple Model Views. *International Journal of Computer Vision (IJCV)*, 67(2):159–188.

[15] Forssén, P. and Lowe, D. (2007). Shape Descriptors for Maximally Stable Extremal Regions. In *Proceedings of the International Conference on Computer Vision (ICCV)*.

[16] Fraundorfer, F. and Bischof, H. (2004). Evaluation of Local Detectors on Non-Planar Scenes. In *Proceedings of the Austrian Association for Pattern Recognition Workshop (ÖAGM/AAPR)*, pages 125–132.

[17] Fraundorfer, F. and Bischof, H. (2005). A novel performance evaluation method of local detectors on non-planar scenes. In *Proceedings of the Conference on Computer Vision and Pattern Recognition (CVPR)*, volume 3, pages 33–43.

[18] Fraundorfer, F. and Bischof, H. (2006). Global localization from a single feature correspondence. In *Proceedings of the Austrian Association for Pattern Recognition Workshop (ÖAGM/AAPR)*, pages 151–160.

[19] Fraundorfer, F., Winter, M., and Bischof, H. (2005). MSCC: Maximally Stable Corner Clusters. In *Proceedings of the Scandinavian Conference on Image Analysis (SCIA)*, pages 45–54.

[20] Freeman, W. and Adelson, E. (1991). The design and use of steerable filters. In *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, volume 13, pages 891–906.

[21] Friedman, J., Bentley, J., and Finkel, R. (1977). An Algorithm for Finding Best Matches in Logarithmic Expected Time. In *ACM Transactions on Mathematical Software (TOMS)*, volume 3, pages 209–226.

[22] Grabner, M. (2004). Object Recognition with local feature trajectories. Master's thesis, Technical University in Graz.

[23] Grabner, M. and Bischof, H. (2005). Object Recognition based on local feature trajectories. In *Proceedings of the International Cognitive Vision Workshop (ICVW)*.

[24] Grabner, M., Grabner, H., and Bischof, H. (2006). Fast Approximated SIFT. In *Proceedings of the Asian Conference on Computer Vision (ACCV)*, pages 918–927.

[25] Harris, C. and Stephens, M. (1988). A Combined Corner and Edge Detection. In *Proceedings of the Alvey Vision Conference (AVC)*, pages 147–151.

[26] Hartigan, J. (1985). Statistical theory in clustering. *Journal of Classification (JOC)*, 2:63—-76.

[27] Intel (2006). Open Source Computer Vision Library. Intel Corporation, Retrieved: March 26th, 2007, http://www.intel.com/technology/computing/opencv/.

[28] Johnson, A. and Hebert, M. (1999). Using spin images for efficient object recognition in cluttered 3D scenes. In *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, volume 21, pages 433–449.

[29] Jolliffe, I. (1986). *Principal Component Analysis (PCA)*. Springer Series in Statistics. Springer, 2 edition.

[30] Jones, R. (1997). Component trees for image filtering and segmentation. In *Proceedings of the Conference on Nonlinear Signal and Image Processing Workshop (NSIP)*.

[31] Jones, R. (1999). Connected filtering and segmentation using component trees. *Journal of Computer Vision and Image Understanding (CVIU)*, 75(3):215–228.

[32] Kadir, T., Boukerroui, D., and Brady, M. (2003). An analysis of the scale saliency algorithm. Technical report, Robotics Research Laboratory, Department of Engineering Science, University of Oxford. TR-2264-03, http://www.robots.ox.ac.uk/~timork/Saliency/TR-2264-03.pdf.

[33] Kadir, T. and Brady, M. (2001). Saliency, Scale and Image Description. *International Journal of Computer Vision (IJCV)*, 45(2):83–105.

[34] Kadir, T. and Brady, M. (2003). Scale Saliency: A novel approach to salient feature and scale selection. In *Proceedings of International Conference on Visual Information Engineering (VIE)*, pages 25–28.

[35] Kadir, T., Zisserman, A., and Brady, M. (2004). An Affine Invariant Salient Region Detector. In *Proceedings of the European Conference on Computer Vision (ECCV)*, volume 1, pages 228–241.

[36] Kim, H., Murphy-Chutorian, E., and Triesch, J. (2006). Semi-autonomous Learning of Objects. In *Proceedings of the Computer Vision and Pattern Recognition Workshop (CVPRW)*, pages 145–151.

[37] Komppa, J. (2007). Extremely Simple Capture API. Retrieved: March 26th, 2007, http://sol.gfxile.net/code.html.

[38] Lazebnik, S., Schmid, C., and Ponce, J. (2003). A Sparse Texture Representation Using Affine-Invariant Regions. In *Proceedings of the Conference on Computer Vision and Pattern Recognition (CVPR)*, volume 2, pages 319–324.

[39] Lazebnik, S., Schmid, C., and Ponce, J. (2004a). A sparse texture representation using local affine regions. In *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, volume 27, pages 1265–1278.

[40] Lazebnik, S., Schmid, C., and Ponce, J. (2004b). A sparse texture representation using local affine regions. Technical report, Beckman Institute, University of Illinois. CVR-TR-2004-01, http://www-cvr.ai.uiuc.edu/ponce_grp/publication/paper/pami04.ps.gz.

[41] Lazebnik, S., Schmid, C., and Ponce, J. (2005). A sparse texture representation using local affine regions. In *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, volume 27, pages 1265–1278.

[42] Lepetit, V. and Fua, P. (2004). Towards Recognizing Feature Points using Classification Trees. Technical report, Swiss Federal Institute of Technology (EPFL), Lausanne, Switzerland. IC/2004/74, http://cvlab.epfl.ch/~vlepetit/papers/lepetit-tr04.pdf.

[43] Lepetit, V. and Fua, P. (2006). Keypoint recognition using randomized trees. In *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, volume 28, pages 1465–1479.

[44] Lepetit, V., Lagger, P., and Fua, P. (2005). Randomized trees for real-time keypoint recognition. In *Proceedings of the Conference on Computer Vision and Pattern Recognition (CVPR)*, volume 2, pages 775–781.

[45] Lepetit, V., Pilet, J., and Fua, P. (2004). Point matching as a classification problem for fast and robust object pose estimation. In *Proceedings of the Conference on Computer Vision and Pattern Recognition (CVPR)*, volume 2, pages 244–250.

[46] Lindeberg, T. (1994). *Scale-Space Theory in Computer Vision*. Kluwer Academic Publishers, Norwell, MA, USA.

[47] Lindeberg, T. (1998). Feature Detection with Automatic Scale Selection. *International Journal of Computer Vision (IJCV)*, 30(2):77–116.

[48] Lowe, D. (1999). Object Recognition from Local Scale-Invariant Features. In *Proceedings of the International Conference on Computer Vision (ICCV)*, volume 2, pages 1150–1157.

[49] Lowe, D. (2004). Distinctive Image Features from Scale-Invariant Keypoints. In *International Journal of Computer Vision (IJCV)*, volume 60, pages 91–110.

[50] Matas, J., Chum, O., Martin, U., and Pajdla, T. (2001). Distinguished regions for wide-baseline stereo. Technical report, Center for Machine Perception, K333 FEE Czech Technical University, Prague, Czech Republic. CTU-CMP-2001-33, http://cmp.felk.cvut.cz/~matas/papers/matas-tr-2001-33.ps.gz.

[51] Matas, J., Chum, O., Martin, U., and Pajdla, T. (2002a). Robust Wide Baseline Stereo from Maximally Stable Extremal Regions. In *Proceedings of British Machine Vision Conference (BMVC)*, volume 1, pages 384–393.

[52] Matas, J., Obdržálek, S., and Chum, O. (2002b). Local affine frames for wide-baseline stereo. In *Proceedings of the International Conference on Pattern Recognition (ICPR)*, volume 4, pages 363–366.

[53] Mikolajczyk, K. and Schmid, C. (2001). Indexing based on scale invariant interest points. In *International Conference on Computer Vision (ICCV)*, volume 1, pages 525–531.

[54] Mikolajczyk, K. and Schmid, C. (2002). An Affine Invariant Interest Point Detector. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 128–142.

[55] Mikolajczyk, K. and Schmid, C. (2003). A performance evaluation of local descriptors. In *Proceedings of the Conference on Computer Vision and Pattern Recognition (CVPR)*, volume 2, pages 257–264.

[56] Mikolajczyk, K. and Schmid, C. (2004a). Comparison of affine-invariant local detectors and descriptors. In *Proceedings of the European Signal Processing Conference (EUSIPCO)*.

[57] Mikolajczyk, K. and Schmid, C. (2004b). Scale & Affine Invariant Interest Point Detectors. *International Journal of Computer Vision (IJCV)*, 60(1):63–86.

[58] Mikolajczyk, K. and Schmid, C. (2005). A performance evaluation of local descriptors. In *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, volume 27, pages 1615–1630.

[59] Mikolajczyk, K., Tuytelaars, T., Schmid, C., Zisserman, A., Matas, J., Schaffalitzky, F., Kadir, T., and Van Gool, L. (2005). A Comparison of Affine Region Detectors. In *International Journal of Computer Vision (IJCV)*, volume 65, pages 43–72.

[60] Mosorov, V. and Kowalski, T. (2002). The development of component tree structure for grayscale image segmentation. In *Proceedings of the Conference on Modern Problems of Radio Engineering, Telecommunications and Computer Science (TCSET)*, pages 252–253.

[61] Najman, L. and Couprie, M. (2004). Quasi-linear algorithm for the component tree. In *IS&T/SPIE Symposium on Electronic Imaging, Vision Geometry XII*, volume 5300, pages 98–107.

[62] Najman, L. and Couprie, M. (2006). Building the Component Tree in Quasi-Linear Time. In *IEEE Transactions on Image Processing (TIP)*, volume 15, pages 3531–3539.

[63] Nistér, D. and Stewénius, H. (2006). Scalable Recognition with a Vocabulary Tree. In *Proceedings of Conference on Computer Vision and Pattern Recognition (CVPR)*, volume 2, pages 2161–2168.

[64] Obdržálek, S. (2006). *Object Recognition Using Local Affine Frames*. PhD thesis, Czech Technical University in Prague.

[65] Obdržálek, S. and Matas, J. (2002a). Local Affine Frames for Image Retrieval. In *Proceedings of the International Conference on Image and Video Retrieval (CIVR)*, pages 318–327.

[66] Obdržálek, S. and Matas, J. (2002b). Object recognition using local affine frames on distinguished regions. In *Proceedings of British Machine Vision Conference (BMVC)*, volume 1, pages 113–122.

[67] Obdržálek, S. and Matas, J. (2005). Sub-linear Indexing for Large Scale Object Recognition. In *Proceedings of the British Machine Vision Conference (BMVC)*.

[68] Philbin, J., Chum, O., Isard, M., Sivic, J., and Zisserman, A. (2007). Object retrieval with large vocabularies and fast spatial matching. In *Proceedings of the Conference on Computer Vision and Pattern Recognition (CVPR)*.

[69] Porikli, F. (2005). Integral histogram: a fast way to extract histograms in Cartesian spaces. In *Proceedings of the Conference on Computer Vision and Pattern Recognition (CVPR)*, volume 1, pages 829–836.

[70] Roth, P., Donoser, M., and Bischof, H. (2006). Tracking for Learning an Object Representation from Unlabeled Data. In *Proceedings of the Computer Vision Winter Workshop (CVWW)*, pages 46–51.

[71] Salembier, P., Oliveras, A., and Garrido, L. (1998). Anti-extensive connected operators for image and sequence processing. In *IEEE Transactions on Image Processing (TIP)*, volume 7, pages 555–570.

[72] Schaffalitzky, F. and Zisserman, A. (2002). Multi-view Matching for Unordered Image Sets, or "How Do I Organize My Holiday Snaps?". In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 414–431.

[73] Schindler, G., Brown, M., and Szeliski, R. (2007). City-Scale Location Recognition. In *Proceedings of the Conference on Computer Vision and Pattern Recognition (CVPR)*.

[74] Schmid, C., Mohr, R., and Bauckhage, C. (1998). Comparing and evaluating interest points. In *International Conference on Computer Vision (ICCV)*, pages 230–235.

[75] Schmid, C., Mohr, R., and Bauckhage, C. (2000). Evaluation of Interest Point Detectors. *International Journal of Computer Vision (IJCV)*, 37(2):151–172.

[76] Sethi, I. and Jain, R. (1987). Finding trajectories of feature points in a monocular image sequence. In *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, volume 9, pages 56–73.

[77] Simard, P., Bottou, L., Haffner, P., and Lecun, Y. (1999). Boxlets: A fast convolution algorithm for signal processing and neural networks. In *Proceedings of the Conference on Neural Information Processing Systems (NIPS)*, pages 571–577.

[78] Sivic, J., Schaffalitzky, F., and Zisserman, A. (2006). Object Level Grouping for Video Shots. *International Journal of Computer Vision (IJCV)*, 67(2):189–210.

[79] Sivic, J. and Zisserman, A. (2003). Video Google: A Text Retrieval Approach to Object Matching in Videos. In *Proceedings of the International Conference on Computer Vision (ICCV)*, volume 2, pages 1470–1477.

[80] Tomasi, C. and Kanade, T. (1991). Detection and tracking of point features. Technical report, Carnegie Mellon University, Pittsburgh, PA. CMU-CS-91-132, http://www.ces.clemson.edu/~stb/klt/tomasi-kanade-techreport-1991.pdf.

[81] Tuytelaars, T. and Mikolajczyk, K. (2006). A Survey on Local Invariant Features: "What? Why? When? How?". Unpublished draft retrieved on September 12th 2007, http://homes.esat.kuleuven.be/~tuytelaa/survey_inv_features.pdf.

[82] Tuytelaars, T. and Van Gool, L. (1999). Content-Based Image Retrieval Based on Local Affinely Invariant Regions. In *Proceedings of the Conference on Visual and Information Systems (VIS)*, pages 493–500.

[83] Tuytelaars, T. and Van Gool, L. (2000). Wide baseline stereo matching based on local, affinely invariant regions. In *Proceedings of British Machine Vision Conference (BMVC)*, pages 412–425.

[84] Tuytelaars, T. and Van Gool, L. (2004). Matching Widely Separated Views Based on Affine Invariant Regions. *International Journal of Computer Vision (IJCV)*, 59(1):61–85.

[85] Vedaldi, A. and Soatto, S. (2005). Features for recognition: Viewpoint invariance for non-planar scenes. In *Proceedings of the International Conference on Computer Vision (ICCV)*, volume 2, pages 1474–1481.

[86] Viola, P. and Jones, M. (2001). Rapid object detection using a boosted cascade of simple features. In *Proceedings of the Conference on Computer Vision and Pattern Recognition (CVPR)*, volume 1, pages 511–518.

[87] Wallis, G. and Bülthoff, H. (2001). Effects of temporal association on recognition memory. In *Proceedings of the National Academy of Sciences (PNAS)*, volume 98, pages 4800–4804.

[88] Wallraven, C. and Bülthoff, H. (2001). Acquiring Robust Representations for Recognition from Image Sequences. In *Proceedings of the DAGM-Symposium on Pattern Recognition*, pages 216–222.

[89] Wishart, D. (1969). Mode analysis: A generalization of the nearest neighbor which reduces chaining effects. *A. J. Cole, editor, Numerical Taxonomy*, 2:282–319.