Lappeenranta
University of Technology

Jukka Lankinen

LOCAL FEATURES IN IMAGE AND VIDEO PROCESSING - OBJECT CLASS MATCHING AND VIDEO SHOT DETECTION

Thesis for the degree of Doctor of Science (Technology) to be presented with due permission for public examination and criticism in the Auditorium 1382 at Lappeenranta University of Technology, Lappeenranta, Finland on the 22nd of May, 2014, at noon.

Acta Universitatis
Lappeenrantaensis **576**

Supervisor
Professor Joni-Kristian Kämäräinen
Machine Vision and Pattern Recognition Laboratory
Department of Mathematics and Physics
Lappeenranta University of Technology, LUT Kouvola Unit
Finland

Reviewers
Reader Dr. Krystian Mikolajczyk
Faculty of Engineering & Physical Sciences
University of Surrey
United Kingdom

Docent Jorma Laaksonen
Department of Information and Computer Science
Aalto University
Finland

Opponents
Reader Dr. Krystian Mikolajczyk
Faculty of Engineering & Physical Sciences
University of Surrey
United Kingdom

Docent Esa Rahtu
Department of Computer Science and Engineering
University of Oulu
Finland

ISBN 978-952-265-592-9
ISBN 978-952-265-593-6 (PDF)
ISSN-L 1456-4491
ISSN 1456-4491

Lappeenrannan teknillinen yliopisto
Yliopistopaino 2014

Preface

This dissertation is the culmination of over four years of scientific research work, carried out at the Machine Vision and Pattern Recognition Laboratory at Lappeenranta University of Technology. The majority of the work has been done at the LUT Kouvola Unit and in Automatic Video Summarization (CAMERA2DVD) project. There is several people who have contributed in the work and I wish to express my deepest gratitude to those who made this possible. This thesis would have never been possible without the brilliant guidance of my supervisor Joni Kämäräinen, who tirelessly motivated me through the hard parts of the research work.

I thank my reviewers, Krystian Mikolajczyk and Jorma Laaksonen, who provided critical review of my thesis manuscript. Jorma Laaksonen, in particular, made extensively detailed review of the thesis that helped me to increase the quality of the overall thesis significantly. I would also like to thank the anonymous reviewers of the articles sent for the peer review.

I would like to thank the whole MVPR laboratory at Lappeenranta University of Technology, both former and present members. The group continuously provided insight into the topics also included in this thesis. I want to thank Lauri Laaksonen, Ekaterina Riabchenko, Dr. Nataliya Strokina, Dr. Teemu Kinnunen, Dr. Jonna Laaksonen, Dr. Leena Ikonen, Dr. Jarmo Ilonen, Dr. Tommi Tykkälä, Professor Ville Kyrki and Assistant Professor Arto Kaarna. Special thanks to Tarja Nikkinen, Riitta Laari, and Ilmari Laakkonen for technical and organizational support.

I also want to thank the Kouvola unit and the personnel at Kymenlaakso University of Applied Sciences. The Camera2DVD project provided versatile environment for professionals of different areas and it was enjoyable to work with the video and usability specialists. I want to thank Jori Pölkki for providing excellent feedback for the shot detection and summarization framework.

The success of the project lies on the shoulders of the excellent student works by Otto Närhi and Petteri Heikkilä. The graphical user interface for the presented web application was designed by talented students Jaakko Inkinen and Lasse Laukkanen with supervision by Suvi Pylvänen. The usability evaluation by Sanna Haapanen and her students were important for the quality of the end-user experience. Special thanks to Sanna Tomperi for support at Kouvola Unit.

For financial support I would like to thank CAMERA2DVD project partners, Lappeenranta University of Technology, Kymenlaakso University of Applied Sciences, Kouvola City and European Union.

Last, but not least, I want to thank my parents and family. Especially I want to thank my beloved wife Nadja for patience during the stressful times.

Lappeenranta, May 2014

Jukka Lankinen

Abstract

Jukka Lankinen

Local Features in Image and Video Processing - Object Class Matching and Video Shot Detection
Lappeenranta, 2014
97 p.

Acta Universitatis Lappeenrantaensis (576)
Diss. Lappeenranta University of Technology

ISBN 978-952-265-592-9
ISBN 978-952-265-593-6 (PDF)
ISSN-L 1456-4491
ISSN 1456-4491

The usage of digital content, such as video clips and images, has increased dramatically during the last decade. Local image features have been applied increasingly in various image and video retrieval applications. This thesis evaluates local features and applies them to image and video processing tasks. The results of the study show that 1) the performance of different local feature detector and descriptor methods vary significantly in object class matching, 2) local features can be applied in image alignment with superior results against the state-of-the-art, 3) the local feature based shot boundary detection method produces promising results, and 4) the local feature based hierarchical video summarization method shows promising new new research direction. In conclusion, this thesis presents the local features as a powerful tool in many applications and the imminent future work should concentrate on improving the quality of the local features.

Keywords: Local features, image processing, video processing, object class matching, shot detection, pattern recognition, computer vision

UDC 004.93'1:004.932:51.001.57

Abbreviations and symbols

2-D	2-Dimensional
3-D	3-Dimensional
BoW	Bag-of-Words
BoF	Bag-of-Features
BRIEF	Binary Robust Independent Elementary Features
CAD	Computer-aided Design
DLT	Direct Linear Transform
DoG	Difference-of-Gaussian
FAST	Features from Accelerated Segment Test
GMM	Gaussian Mixture Models
HMM	Hidden Markov Models
HOG	Histogram of Oriented Gradients
LBP	Local Binary Patterns
LoG	Laplacian-of-Gaussian
MSER	Maximally Stable Extremal Regions
NIST	National Institute of Standards and Technology
ORB	Oriented BRIEF
PCA	Principal Component Analysis
RANSAC	Random-Sample-Concensus
rBRIEF	Rotated Binary Robust Independent Elementary Features
RGB	Red-Green-Blue Color Space
RMS	Root mean square
SIFT	Scale-Invariant Feature Transform
SLAM	Simultaneous Localization and Mapping
SURF	Speeded-Up Robust Features
SVM	Support Vector Machine
TRECVid	TREC Video Retrieval Evaluation
UI	User Interface
VOC	Visual Object Categorization
VSLAM	Visual Simultaneous Localization and Mapping

σ	Gaussian kernel size
λ_1	Eigenvalue calculated from $\mathbf{M}$

λ_2	Eigenvalue calculated from $\mathbf{M}$
α	A balancing constant in Harris detector
τ	threshold value
$\tau(x, y)$	Binary test in BRIEF/ORB
$\in_0$	Overlap error
$\theta(x, y)$	The gradient orientation
d_{eye}	Distance used in the face detection
D	Approximation of gaussian derivatives
$D(\mathbf{x}, \sigma)$	Difference-of-Gaussian image of $I(x)$ in scale-space
$D_x(\mathbf{x})$	Partial derivative in x direction of $D(x, \sigma)$
$D_y(\mathbf{x})$	Partial derivative in y direction of $D(x, \sigma)$
$D_{xx}(\mathbf{x})$	Second-order partial derivative in x direction of $D_x(x)$
$D_{xy}(\mathbf{x})$	Second-order partial derivative in y direction of $D_x(x)$
$D_{yy}(\mathbf{x})$	Second-order partial derivative in y direction of $D_y(x)$
$\boldsymbol{D}_{N \times M_i}$	Distance Matrix of N feature candidates
$g_n(p, \sigma)$	Steered BRIEF/ORB definition for patch p
$G(x, y, \sigma)$	Gaussian kernel
$\mathbf{H}$	Hessian matrix
$I(\mathbf{x})$	Image intensity function
$I_x(\mathbf{x})$	Partial derivative in x direction of $I(x)$
$I_y(\mathbf{x})$	Partial derivative in y direction of $I(x)$
$I_{xx}(\mathbf{x})$	Second-order partial derivative in x direction of $I_x(x)$
$I_{xy}(\mathbf{x})$	Second-order partial derivative in y direction of $I_x(x)$
$I_{yy}(\mathbf{x})$	Second-order partial derivative in y direction of $I_y(x)$
$\mathbf{M}$	Second moment matrix
$I_{yy}(\mathbf{x})$	Second-order partial derivative in y direction of $I_y(x)$
k	Separation factor in DoG scale-space
K	Number of best matches in alignment
L	Image convoluted by a Gaussian kernel
L_b	Number of landmarks in alignment
$m(x, y)$	The gradient magnitude
n	Number of points selected in FAST algorithm
N	Number of correct matches in Coverage-N measurement
p	Patch in BRIEF/ORB
$\mathbf{p}$	Candidate points in FAST algorithm

R	Number of random iterations in RANSAC as part of alignment
R_μ	Detected regions
$\boldsymbol{s}_N$	Score vector
S	Patch size in BRIEF/ORB
t	Thresholding value in FAST algorithm
$\mathbf{v}$	SURF feature vector

Contents

Chapter I

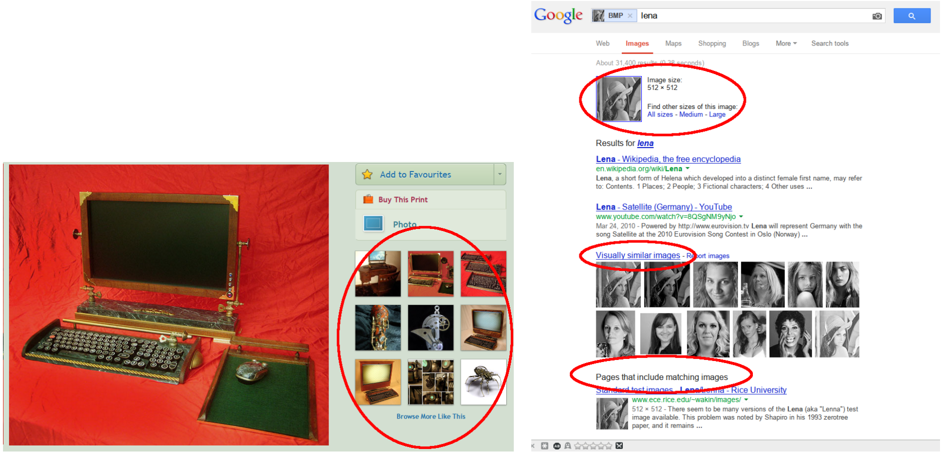
Introduction

The usage of digital content, such as video clips and images, has dramatically increased during the last decade. This is due to the increasing popularity of camera-ready cellphones which are able of producing high-quality images and video content. As the digital images increase in resolution and size, they produce even larger amounts of data to process. Digital photography itself has taken steps forward by introducing new high-speed cameras and large digital storages for the consumer market. A single image can have over ten million pixels and require over ten megabytes of memory even while being aggressively compressed.

Not only the capabities and digital capacity have increased, but also the networking is many times more efficient. Online services such as Flickr, contains billions of images and Youtube has billions hours of video submitted by its users. People watch 6 billion hours of video at Youtube every month. Even more images are stored on users' home computers and are not uploaded to the online services. Not only is there a lot of data but the data is also of a very high resolution when taken with the latest consumer-level devices.

There is a serious problem for presenting and processing an enormous amount of image data. Many services provide basic search tools for finding visual content based on the data given by the user, such as title and description. Recently, there have been some advancements in the search tools to utilize image content information to some extent. For example, as seen in Fig 1.1, sites such as *Google Image Search* and *DeviantArt* utilize the combination of metadata and image information, such as color histograms, to find similar images a user is currently viewing.

The local image features, which are studied in this thesis, have been applied to popular image services such as Google Image Search. It is common for the local feature based applications to perform very well with near identical images, but to have problems with drawn images and with insufficient text-based metadata such as title and description. It is possible for the metadata to be incorrect and cause declining performance. However, it is possible to improve the quality of visual content driven approaches. This thesis is



About 31,400 results (0.38 seconds)

Image size:
512 × 512

Find other sizes of this image:
All sizes - Medium - Large

Results for *lena*

Lena - Wikipedia, the free encyclopedia
en.wikipedia.org/wiki/Lena
Lena, a short form of Helena which developed into a distinct female first name, may refer to: Contents. 1 Places; 2 People; 3 Fictional characters; 4 Other uses ...

Lena - Satellite (Germany) - YouTube
www.youtube.com/watch?v=8QSgNM9yNjo
Mar 24, 2010 - Powered by http://www.eurovision.tv Lena will represent Germany with the song Satellite at the 2010 Eurovision Song Contest in Oslo (Norway) ...

Visually similar images - Report images

Pages that include matching images

Standard test images - Lena/Lenna - Rice University
www.ece.rice.edu/~wakin/images/
512 × 512 - There seem to be many versions of the Lena (aka "Lenna") test image available. This problem was noted by Shapiro in his 1993 zerotree paper, and it remains ...

(a) DeviantArt

(b) Google Image Search

Figure 1.1: Examples of the existing content aware systems a) DevianArt related images and b) Google Image Search for similar images.

about improving the object detection aspect of local features and identifying problems with the current approaches.

1.1 Objectives

The goal of this thesis is to study the properties of local features in image and video processing tasks: Object alignment, video skimming and shot boundary detection. The object alignment is an important aspect of many computer vision tasks such as object detection and recognition. However, it usually requires manual annotation of the images. Effectively, the questions are answered in this thesis:

- "How to align the selected images automatically using local features?"
- "How to apply local features in video shot boundary detection effectively?"
- "How to create sensible video skims using local feature based method?"
- "How well the local features perform in evaluation which targets object detection?"

The topics discussed in this thesis are evaluated against the state-of-the-art methods if freely available. The datasets are widely used and accepted by the scientific community. The only exception being the video skim generation for which no quantitative evaluation can be easily conducted.

1.2 Contribution and publications

This thesis studies applications of local features, which is a topic of a wide range of different appliable methods. In this thesis, methods in question are object alignment, video skimming and video shot boundary detection. Many publications here mention the term “object class images” which refers to images which contain the same object of the same class such as *Cars* or *faces*. A list of the contributions of this thesis is as follows:

Local feature detector and descriptor comparison for object categorization
The first contribution introduces an expansion for the existing local feature detector and descriptor evaluation framework. It is shown that when using object-class images, the results for the local features are different from the wide baseline matching based evaluation framework. In this work, the results show that in object-class image categorization, the number of possible feature match candidates is an important factor. The results were published in [61].

Local feature based alignment of object-class images
The second contribution is a novel local feature based alignment method for images containing object-class image examples. The algorithm is based on the standard visual object categorization (VOC) tools: local feature detectors and descriptors, correspondence based homography estimation, and random sample consensus (RANSAC) based spatial validation of local features. The experiments compare different local feature detectors and descriptors for local feature based alignment and select the best performing detector-descriptor pair with optimal parameter values. The experiments show state-of-the-art results compared to the pixel-based congealing methods. The results of this research were published in [59]. Moreover, the alignment procedure is utilized in a novel way in spatial feature matching in [54] and in color normalization [95].

Video shot boundary detection using visual Bag-of-Words
In the third contribution, the problem of shot boundary detection is solved by using a popular image analysis (object detection) approach: visual Bag-of-Words (BoW). The baseline approach for the shot boundary detection has been colour histogram and it is at the core of many top methods in the field, but our BoW method of similar complexity in the terms of parameters clearly outperforms color histograms. Interestingly, an “AND-combination” of color and BoW histogram detection is clearly superior indicating that color and local features provide complimentary information for video analysis. The results were published in [60].

Hierarchical video summarization for home videos
Video skimming methods are often tied to a single application. As such, they usually focus on features which they deem important for the application. Here, after exploring the shot boundary detection, hierarchical Bag-of-Features frame difference data is used to produce high-quality video summarizations. The results demonstrate the properties of the shot-based skimming approach and show that the shots, as used in many works on the field, bring not only the relevant video data but also notable amount of redundant material. It was shown that the redundant material can be reduced by providing more sub-shots to the user. Additionally an online application was developed as a part of this work. This work and results are under preparation for submission.

In addition to the published results, the source code and documentation related to these topics will be made available online.

1.3 Outline of the thesis

Chapter 2 is an introduction to different popular interest point detectors used in the area of object detection, revealing how and why they work. In this chapter, the evaluation framework for object detection is presented which is the base for Chapter 3 and Chapter 4.

In **Chapter 3**, the comparison of the local feature detectors and descriptors are made between in-class images. This evaluation work is to extend the original work [83] to take this important aspect of VOC into account.

A novel unsupervised object alignment method is presented in **Chapter 4**. The chapter shortly introduces homography transformations and the Random spatial sampling method which are an integral part of the method. Experimental results on object alignment are shown.

Chapter 5 introduces the core concepts of the video processing work related to shot detection and summary generation. This chapter also represents the state-of-the art methods, data and benchmarks in the field. The topics of this chapter are the base for the following Chapter 6.

The local feature based video processing framework for shot detection is introduced in **Chapter 6**. This chapter demonstrates the usage of the Visual Bag-of-Words in this context and presents results while utilizing a codebook of different sizes and of different input data. Additionally, in this chapter the hierarchical video summarization method is presented which is a direct continuation of the presented shot detection method. This chapter presents the application and gathered results of the proposed summarizations.

Finally, **Chapter 7** discusses what was achieved in this thesis and using the methods it presented. In this chapter, the strengths and weaknesses of the proposed methods are discussed, and ways to bring them to the next level in future work are explored.

Chapter II

Local Feature Detectors and Descriptors

This topics of this thesis are based on local feature detectors and descriptors. The popular local feature detectors and descriptors are presented with detail in this chapter. For the terminology, in the literature *local features* are also referred to as *interest points*, *interest regions* and *keypoints*. Although all the mentioned terms are correct, the term *local feature* is used to unify the meaning of different detector implementations as points or regions are not necessarily found for all feature detectors. For descriptors the terminology is more straightforward as most of the publications use local feature *description* or *descriptor*. The term "local feature descriptors" will be used througout the thesis.

The use of local features has become very popular during the past years in wide-baseline stereo vision and object detection. The increased computational facilities makes the use of these relatively complicated methods even more useful in practical applications. In Fig. 2.1, the extraction and matching of local features are demonstrated. The local features are areas detected from the images by using "good guesses", e.g. edges or borders. The areas are then converted into a vector presentation called "local feature descriptor" which can be compared against other similarly extracted descriptors by simply calculating the distance between the descriptors. The local features were originally used for 3D reconstuction, namely wide-baseline stereo matching for finding correspondency points between images of the same scene. As such, the features were required to be tolerant against perspective change, blur and visual noise produced by cameras. For the wide-baseline stereo matching, the local features need to be accurate and produce a minimum number of outliers. Less outliers provide better matches between images and ultimately depth maps with less error.

Another interesting application for the local features is object detection, a method attempting to determine if a specific object appears in an image. These methods sometimes accompany a part-based method utilizing the local features for recognition. The main advantage of the part-based approaches is that the whole object is not required to be visible to be detected. While object detection approaches detect the object in image, the visual object categorization (VOC) is defining the type of the object. In VOC, the problem is to find categories for a given image set based on the visual content of the images.

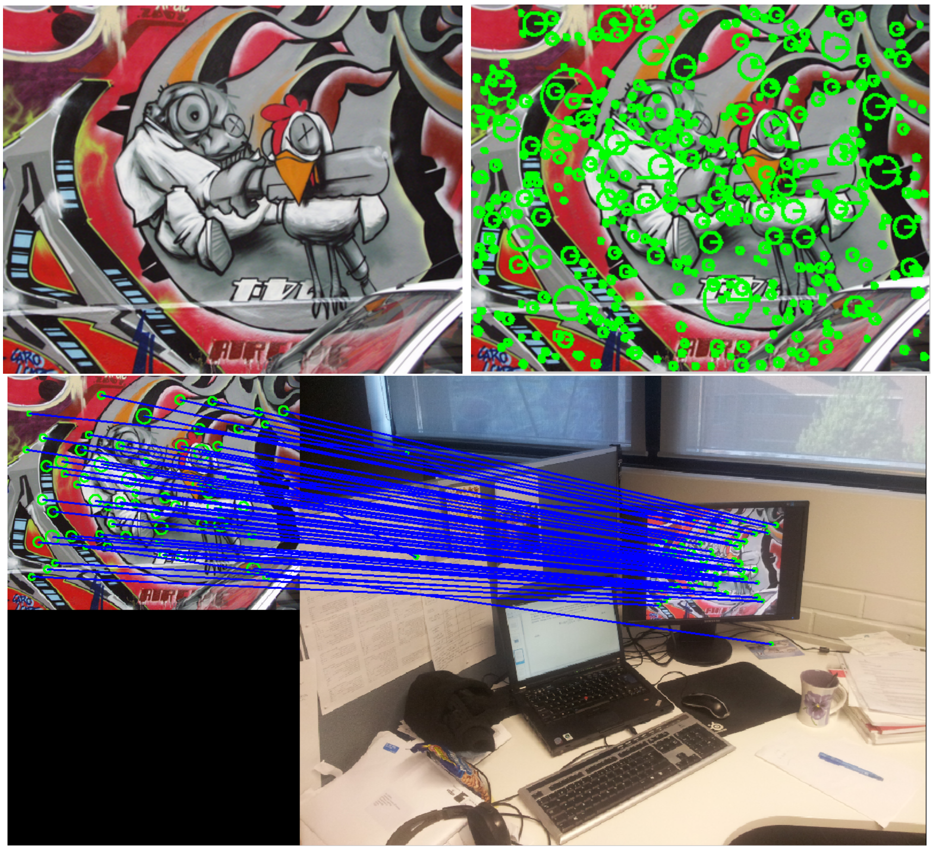


Figure 2.1: Feature extraction example: extracting the local features (SIFT) from both images and matching.

The methods can be supervised or unsupervised if the system is given examples for how to categorize the content. One of the supervised methods is visual bag-of-words (BoW) which is more closely studied in Chapter 6. New local features are presented from time to time to address the shortcomings of the existing approaches e.g. speed or accuracy. To evaluate new local features, a popular evaluation framework from Mikolajczyk *et al.* [82] is used, which is introduced in this chapter.

2.1 Local feature detectors

Here, the most commonly used detectors are briefly described. The presented local features are selected due to the historical impact or due to unconventional approaches. The feature detectors can be used to complement each other in order to produce more complete detections (such as in [14]). Various feature detectors are showcased in Fig. 2.2.

The local feature detectors which are discussed here are algorithms which provide a set of areas from which the local feature descriptors can be computed. The overall procedure

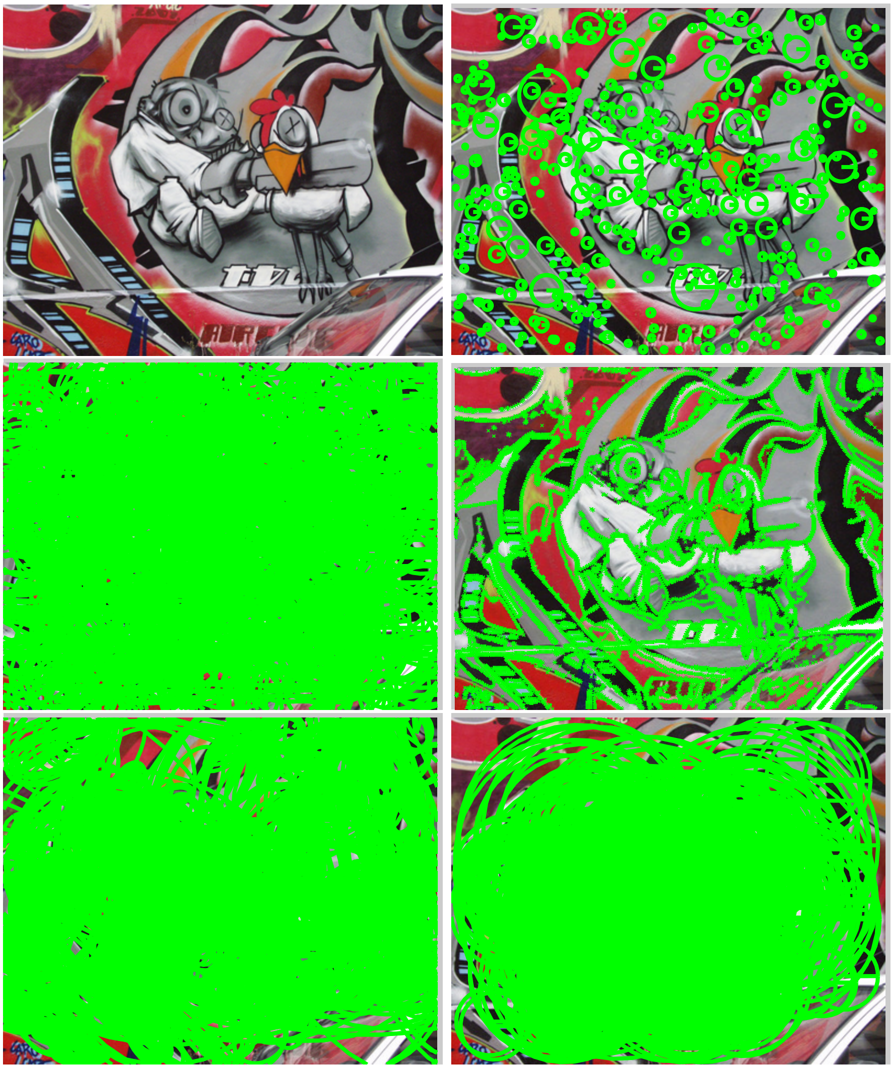


Figure 2.2: The local feature detectors discussed in this chapter. From left to right and from top to bottom: reference, DoG (SIFT), SURF, MSER, Hessian-Affine and oriented FAST

for the local feature detection can be described as follows:

1. Find a set of distinctive, stable local features
2. For each local feature, define a region with scale or affine invariance
3. Provide region content for the descriptor generator

In many cases, the local feature detection and description are procedures tightly coupled together. The local descriptor generation requires extra information from the local feature detector. For example, the SIFT local feature descriptor requires not only the local feature region, but the region must also provide orientation and scale. Likewise, MSER local feature descriptors can be generated from the oriented ellipse regions but are designed to handle arbitrary regions. Luckily, most of the local feature detectors provide oriented ellipses which are applicable in the majority of the local feature descriptors. Sometimes the descriptor generation utilizes the processing data from the local feature detection e.g. in SIFT the DoG octaves need to be calculated only once for the detector and the descriptor.

2.1.1 Hessian-based detectors

The first local feature detector discussed here are the Hessian-based detectors: the original Hessian keypoint detector, Hessian-Laplace, and Hessian-Affine. The original Hessian detector was introduced by Beaudet *et al.* [6] in 1978 and is used to search image locations with strong derivatives in two orthogonal direction. The Hessian detector is based on the *Hessian* matrix $\mathbf{H}$:

$$\mathbf{H} = \begin{bmatrix} I_{xx}(\mathbf{x}) & I_{xy}(\mathbf{x}) \\ I_{xy}(\mathbf{x}) & I_{yy}(\mathbf{x}) \end{bmatrix}. \tag{2.1}$$

Where the terms I_{xx}, I_{xy} and I_{yy} denote the second-order partial derivatives of $I(\mathbf{x})$ at location $\mathbf{x} = (x, y)$. The second-order partial derivatives can be computed from the image gradient of the input image (image intensity function $I(\mathbf{x})$). For example, I_{xx} is calculated from the image gradient in x direction. To find the keypoints, the determinant of the Hessian is computed:

$$\det(\mathbf{H}) = I_{xx}I_{yy} - I_{xy}^2 \tag{2.2}$$

The determinant of the Hessian is then computed for every pixel in the input image. Now, a 3×3 window is is used for every pixel and the determinant values around it are checked. If the pixel under inspection has the largest determinant compared to the neighboring values, it is stored. After filtering, all values that are higher than the predefined threshold τ_h are selected as keypoints.

However, this approach is not scale nor affine invariant as the extracted keypoints are computed from a single scale. With scale invariance, the regions from the same image, but with different scale (focal length) would cover the same region/object size-wise. The scale invariant local features are traditionally obtained by searching stable features in the scale space presentation [126]. The scale space presentation is produced from responses of a local kernel with varying scale parameter σ_h. Lindeberg [67] demonstrated how the scale space presentation can be used to find the characteristic scale for a local feature. He showed that the Gaussian kernel is the only operator that is able to fulfill the requirements of the scale space presentation. As such, Lindeberg proposed the Laplacian-of-Gaussian (LoG) detector:

$$L(x, y, \sigma) = \sigma_h^2(I_{xx} + I_{yy}) \tag{2.3}$$

Where σ^2 is the factor used to normalize LoG across scales and making the measurements comparable. The "characteristic scale" which defines the scale of the local feature is found using the local maximum of the Laplacian. The local maxima is checked for point $\mathbf{x}$ by 3×3 window from the scales above and below. Additionally, the pixels around the current point are checked. The LoG detected points can be used for feature detection as-is or they can be bundled together with more discriminative detectors, such as the Hessian detector. For Hessian-Laplace, the separate scale spaces are defined for the Hessian function and for the Laplacian. The candidate points are computed by the Hessian detector for each scale level. Then, the points for which the Laplacian is at local maxima are selected.

However, the scale invariant Hessian-Laplace detector is unable to tolerate changing viewpoint and is not affine invariant. The affine invariant features can be constructed by utilizing the properties of the second moment matrix [80, 81, 113]. By utilizing the second moment matrix $\mathbf{M}$ (see 2.1.2) and its eigenvalues, the affine shape of the region is estimated. The affine shape is then normalized into the circural presentation and another detection is made in the normalized image. If the eigenvalues of the second moment matrix $\mathbf{M}$ are not equal with the previous detection, the affine shape is re-evaluated. This is repeated until the affine shape is found or a given number of iterations are exceeded. In the end, a set of elliptical regions is obtained, which is tolerant against deformations caused by viewpoint change.

Overall, Hessian detectors provide regions with high texture variation. Additionally, updated versions of the detectors (Hessian-Laplace and Hessian-Affinen) are provided by Mikolajczyk *et al.* [81].

2.1.2 Harris-based detectors

Here, the following Harris-based detectors are described: the original Harris detector, Harris-Laplace and Harris-Affine. The Harris corner detector was proposed by Harris and Stephens [44]. The basic assumption of the Harris detector is that at a corner, the intensity of an image will change in multiple orthogonal directions. The detector is based on the *second moment matrix* $\mathbf{M}$ that describes the intensity change in the local neighborhood at location $\mathbf{x} = (x, y)$.

$$\mathbf{M} = G(\sigma) \begin{bmatrix} I_x^2(\mathbf{x}) & I_x I_y(\mathbf{x}) \\ I_x I_y(\mathbf{x}) & I_y^2(\mathbf{x}) \end{bmatrix} \tag{2.4}$$

$$I_x(\mathbf{x}) = \frac{\partial}{\partial x} I(\mathbf{x}) \tag{2.5}$$

$$I_y(\mathbf{x}) = \frac{\partial}{\partial y} I(\mathbf{x}) \tag{2.6}$$

$$G(\sigma) = \frac{1}{2\pi\sigma^2} e^{-\frac{x^2+y^2}{2\sigma^2}}. \tag{2.7}$$

Where the derivatives $I_x(\mathbf{x})$ and $I_x(\mathbf{x})$ are computed in x and y directions. Finally, the derivatives are smoothed using Gaussian window $G(\sigma)$ of size σ. Practically, it sums over the pixels around point $\mathbf{x}$ and weights them based on the distance to the center point. The corners can be found in an image where the signal change is significant in both directions, i.e. the points where both eigenvalues are large. The eigenvalues λ_1 and λ_2 are calculated from $\mathbf{M}$. Harris proposed the *cornerness* measure R, that describes the cornerness of the local neighborhood of a point. R is computed using the trace and determinant of the matrix $\mathbf{M}$:

$$\operatorname{tr}(\mathbf{M}) = \lambda_1 + \lambda_2 \tag{2.8}$$

$$\det(\mathbf{M}) = \lambda_1 \lambda_2. \tag{2.9}$$

The *cornerness* can then be derived into:

$$R = \det(\mathbf{M}) - \alpha \operatorname{tr}^2(\mathbf{M}) \tag{2.10}$$

Here, the need of computing exact eigenvalues is avoided and the constant α is used for balancing the terms in the equation. Typical values for α are in the range of $0.04 - 0.06$ [113]. If only one of the eigenvalues is significantly larger than zero, an edge is found. A corner is considered to be found only if both eigenvalues are significantly larger than zero. Eventually, local maxima of R above the given threshold are considered as found Harris corners and thus, detected local features.

By default, Harris detector is not scale nor affine invariant. Most of the limitations are solved by Mikolajczyk *et al.* [80, 81] by introducing Harris-Laplace detector which is scale invariant and Harris-Affine which is also an affine invariant detector. The scale invariant local features are obtained by searching stable features in scale space presentation [126]. The overall procedure for Harris-Laplace is same as for Hessian-Laplace (see 2.1.1). Similarly, for Harris-Laplace, the separate scale spaces for the Harris function and for the Laplacian are evaluated. The candidate points are computed by the Harris detector for each scale level. Then, the points for which the Laplacian is at local maxima are selected. The Harris-Affine uses the same iterative approach as Hessian-Affine by evaluating the eigenvalues of transformed circles until no difference is detected or the maximum number of iterations are reached. Overall, Harris provides features which are, not surpisingly, very specific to corners in images.

2.1.3 Difference-of-Gaussian (DoG)

Scale Invariant Feature Transform (SIFT) was originally introduced by Lowe in 1999 [69]. It is regarded as distictive and relatively fast compared to the methods presented at the time. SIFT uses a Difference-of-Gaussian (DoG) filter for detection, which is a direct improvement to Laplacian-of-Gaussian (LoG) method [67]. The main improvement to LoG is that DoG is faster to compute and is an adequate approximation of the original Laplacian-of-Gaussian detector. The main reason is faster than LoG is the way the scale space is formed. The first step in DoG is to build a pyramid of filtered images for the scale-space analysis. The pyramid consists of octaves which are images subsampled by the factor of 2. Instead of directly subsampling the original image, a Gaussian kernel is used. Basically, an octave consists of images convolved with the Gaussian kernel using increasing values of σ_n which are separated by a factor of k_n. Layers of a scale-space are

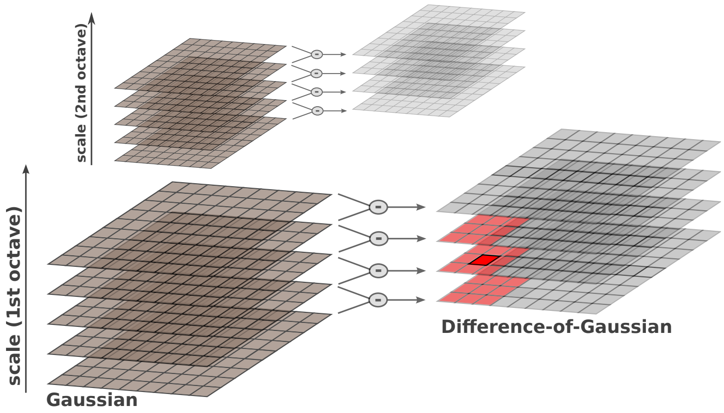


Figure 2.3: The scale-space pyramid with the Difference-of-Gaussian.

illustrated in Fig. 2.3. The edges, which are used for the detection, are found in images by subtracting images with different gaussian blur of scale $k\sigma$ and $(k-1)\sigma$. The pyramid of filtered images is shown in Fig. 2.3. The approximation of Laplacian-of-Gaussian can be written as

$$G(x, y, k\sigma) - G(x, y, \sigma) \approx (k-1)\sigma^2\nabla^2 G \tag{2.11}$$

where $k = \sqrt{2}$ when there are 2 intervals in scale space pyramid. The edges can be identified as local maxima in the scale-space $D(x, y, \sigma)$:

$$D(x, y, \sigma) = (G(x, y, k\sigma) - G(x, y, \sigma)) * I(x) \tag{2.12}$$

After constructing the scale space pyramid, the local maxima are found by pixel comparisons. Each pixel in each DoG-layer is compared to 9 surrounding pixels on the layer above, 8 pixels on the same layer and 9 neighboring pixels on the layer below. If the pixel has higher value than any of its neighbors, it is selected as a local maximum. By default, the difference-of-Gaussian function has a strong response on edges, making it unstable in the presence of noise. DoG is made tolerant against noise by applying 2 × 2 Hessian matrix $\mathbf{H}$ to the Gaussians at location $\mathbf{x} = (x, y)$:

$$\mathbf{H} = \begin{bmatrix} D_{xx}(\mathbf{x}) & D_{xy}(\mathbf{x}) \\ D_{xy}(\mathbf{x}) & D_{yy}(\mathbf{x}) \end{bmatrix}. \tag{2.13}$$

Where the derivatives are estimated by using the differences of neighboring sample points. To filter out the keypoints which do not follow the principal curvatures of D, the eigenvalues are estimated in same fashion as with Harris detector (see 2.1.2). As with Harris detector, with DoG, the threshold parameter needs to be selected. The final step for DoG method presented by Lowe, is to apply rotation invariance by detecting the most stable orientation. This is achieved by utilizing the first derivative image (Gaussian blurred image) L with the characteristic scale. For each point $L(x, y)$ the gradien magnitude m and orientation θ can be computed by pixel differences:

$$m(x, y) = \sqrt{(L_{x+1,y} - L_{x-1,y})^2 + (L_{x,y+1} - L_{x,y-1})^2} \tag{2.14}$$

$$\theta(x, y) = tan^{-1}(L_{x,y+1} - L_{x,y-1})/L_{x+1,y} - L_{x-1,y})) \tag{2.15}$$

An orientation histogram containing 36 bins is formed to cover gradient directions and magnitudes in sectors surrounding a feature. The whole 360 degrees around a feature is covered, using 10 degrees for each bin. Gradient samples are weighted by their distance to the origin using Gaussian-weighted circular window with $\sigma = 1.5$ and by their magnitude m. Each of the 36 bins contains a measurement of intensity change in one direction. The highest of those measurements is selected as dominant orientation of the region. If other peaks with a value over 80% of the highest are found, they are considered additional regions and corresponding dominant orientation is used for detecting another region at the same location.

Although the algorithm is straight-forward, the SIFT detector has a multitude of parameters which can be set. The default values for DoG algorithm change a lot depending on the implementation. The parameters such as the number of octaves and thresholds for local maxima detection. In Chapter 3, three different implementations of SIFT detector were evaluated. To demonstrate the difference between these detectors' default output, see Fig. 2.4. It is not defined how an image should be preprosessed before the SIFT detector.

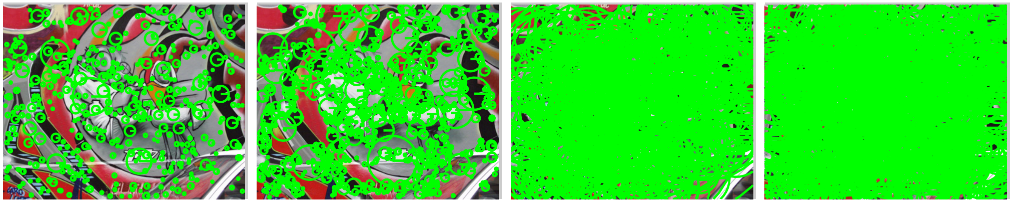


Figure 2.4: The difference of different implementations of the SIFT detector. From left-to-right: Original, VLFeat, Vireo, Featurespace

2.1.4 Maximally Stable Extremal Regions (MSER)

Maximally Stable Extremal Regions (MSER), was introduced by Matas *et al.* [75]. Unlike the approaches such as LoG and DoG, MSER does not start by selecting keypoints. Instead, MSER is fundamentally a segmentation algorithm applied for local feature detection. The algorithm finds areas where intensity change is minimal, i.e. areas that are constantly brighter or darker than an outer boundary of a region. The algorithm utilizes sequential thresholding of the image with all possible values, i.e. $\mathcal{S} = \{0, ..., 255\}$ for an 8-bit gray-scale image. The regions selected by the algorithm are the regions with the least amount of change (in pixels) over as many different thresholds as possible.

The computational complexity of the original algorithm is $O(n\ \log(\log(n)))$. A more efficient version with the worst-case complexity of $O(n)$ was proposed by Nister and Stewenius [87]. The interesting aspect of MSER is the potential complexity of found regions. Typically, found regions are converted to ellipses and information about shapes is lost as seen Fig. 2.5. In the example pair it is visible that some of the regions can be represented correctly using ellipses and some can not. However, in the matching task,

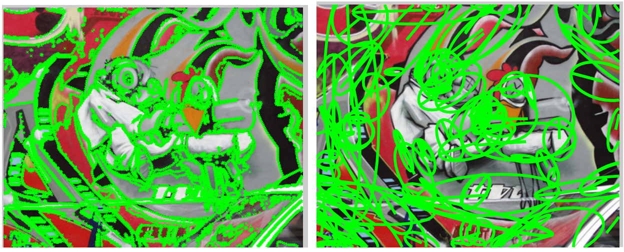


Figure 2.5: MSER finds areas of minimal intensity change. From left to right: a) MSER regions, b) estimated ellipses (only 10% of ellipses are shown here for clarity).

the most important property is spatial stability of the fit, i.e. the ellipse description should be invariant to affine transformation.

2.1.5 Speeded-Up Robust Features (SURF)

Speeded-Up Robust Features (SURF) was introduced by Bay *et al.* [5]. The SURF detector was designed to be very fast, but still not sacrifice any detector and descriptor performance. SURF uses *integral images*, presented by Viola *et al.* [122], which results in a notable performance boost. The integral images provide a way to calculate responses for box-type filters in constant time. After building the structure, the response for any boxfilter of any size inside the image can be built in constant time by only four operations inside any rectangular image. SURF utilizes an approximation of the Hessian matrix for detection ("Fast-Hessian Detector"):

$$\det(\mathbf{H}_{approx}) = D_{xx}D_{yy} - (wD_{xy})^2. \tag{2.16}$$

Where D_{xx}, D_{yy}, D_{xy} are approximations for Gaussian second order derivatives with the lowest scale (similar to SIFT in section 2.1.3). The Gaussian derivatives are approximated by the simple box filters (9×9), illustrated in Fig. 2.6.

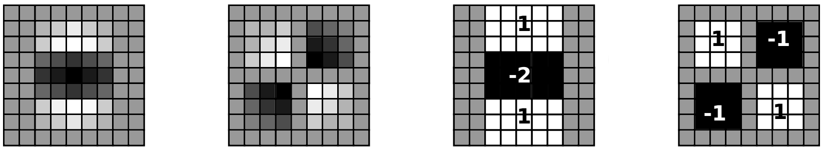


Figure 2.6: From left to right: the discretized second-order Gaussian derivatives L_{yy} and L_{xy} and SURF approximations for L_{yy} and L_{xy}. [5]

SURF detector also provides scale invariance by utilizing scale space presentation. Like with the DoG detector, the octaves are images with increasing Gaussian kernel size. This way, the filter is fast to calculate as the box filter is scaled instead of the image. The scale space can be constructed in parallel, although the original publication did not utilize it. Finally, the local features are selected as local maxima in $3 \times 3 \times 3$ neighborhood in the scale-space. A fast method for non-maximum suppression proposed by Neubeck and Van Gool is used to locate these extrema points [86] and thus the keypoints are found for SURF descriptor generation.

2.1.6 Center Surround Extremas (CenSurE)

Center Surround Extremas (CenSurE) features were introduced by Agrawal *et al.* [2]. The main emphasis in CenSurE is to provide computationally feasible scale invariant local features by replacing the scale-space pyramid (e.g. in SIFT and SURF) with Center-Surround bi-level filters. In CenSurE detector, the Center-Surround filters are calculated at all locations and all scales. The original publication lists a few different options for filters, for boxes being the simplest. The filter shape defines how well the detector captures the rotation invariance. The filter shape used for performance was octagon (CenSurE-OCT) with seven different block size n.

The non-maximal suppression works in similar fashion as with SURF: $3 \times 3 \times 3$ neighbourhood is used. The filter magnitude indicates the strength of the feature and the weak features are omitted by a threshold. As with SIFT, there is need for filtering the features along the edges as they are not stable. With CenSurE the second moment matrix described in equation 2.4 is used to filter out the responses along the edges. The integral images are used to speed up the filter responce calculation as with the SURF detector.

The descriptor used with CenSurE is slightly modified SURF descriptor called Modified Upright SURF (MU-SURF) which is described in descriptor section.

2.1.7 Features from Accelerated Segment Test (FAST)

Features from Accelerated Segment Test (FAST) were originally developed by Rosten *et al.* [97] to enable local feature detection for real-time applications. The FAST detector is actually a combination of a corner detector and a machine learning algorithm. The first step is to do the *segment test* for all corner candidates p.

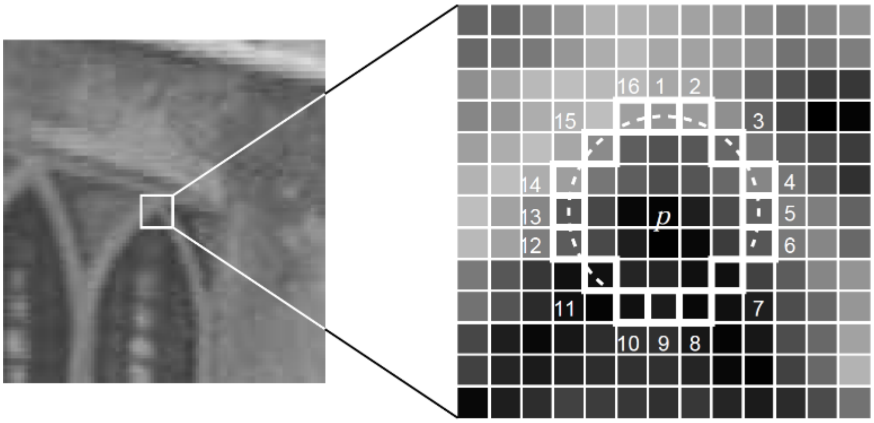


Figure 2.7: The detected corner and the tested pixels in FAST.[97]

The segment test criterion is a circle of sixteen pixels around the candidate point illustrated in Fig 2.7. There are two requirements for this candidate point to be a corner.

1. a set of continuous pixels on the edge of the circular area are all brighter than the intensity of the candidate point: $I(p) + t$, where t is a threshold for detection.
2. a set of continuous points that are darker: $I(p) - t$.

If the previous requirements are true, the corner is detected. In the high-speed test only 4 pixels are tested at first so a large amount of non-corners can be filtered out. The second step for the FAST detector is to utilize the machine learning approach to address several problems in the FAST algorithm. The algorithm does not scale well with criterian smaller than $n = 12$, where n is the number of bright pixels on the edge of the circular area.

Additionally, the choice and ordering of the test pixels in the high-speed test makes strong assumptions of the location of the pixels and might not be optimal. To address these problems, they utilized a decision tree algorithm which was trained with the candidate points with the full circle test. The generated decision tree is then converted into C-code with a long string of nested if-else-statements. As this is a result of a learning algorithm, it is not precisely the same as the original segment test detector.

The FAST detector was further improved by the authors of the ORB descriptor [98] by introducing multi-scaling and orientation. The introduced FAST detector was presented as oFAST (FAST Keypoint Orientation). They employed a scale space pyramid of the image, and produced FAST features at each level of the pyramid. They also solved the orientation by evaluating the corner orientation through intensity centroid (similarly to the DoG detector).

2.2 Local feature descriptors

Local feature description is the backbone of many computer vision applications, such as object recognition and 3D reconstruction. The regions found by the local feature detectors described in the previous section, need to be encoded into mathematical presentation for matching purposes. With the local feature descriptors, the similar visual parts of two separate images can be found. The majority of the applications utilizing local features are reliant on the fact that the extracted features can actually be reliably matched between images. An ideal local feature descriptor has several properties:

- *Distinctiveness:* Low probability of mismatch
- *Efficiency:* Must be efficient to compute
- *Invariance to common deformations:* Matches should be found even if several of the common deformations are present: image noise, changes in illumination, scale, rotation, and skew.

Distinctiveness is important for local features, as mismaches (matching features which are not visually similar) cause unwanted error especially with wide-baseline matching applications. Most of the local feature detectors (see 2.1) are resistant against most of the common deformations (such as affinity, scale and noise). However, the descriptor properties, especially how the detected regions are encoded, may affect the invariance. In this section, the most common descriptors are presented. The discussed descriptor generators assume that the regions have been already found by a local feature detector.

2.2.1 Scale-Invariant Feature Transform (SIFT) descriptor

The original SIFT algorithm by Lowe [68] utilized DoG detector (see 2.1.3) to detect scale level, rotation and location of local features. In DoG, the detected feature belongs to some level σ of the scale space pyramid and to a pixel location (x, y) of the scale space $D(x, y, \sigma)$.

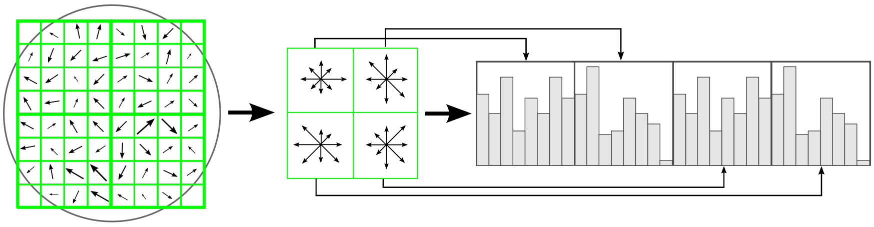


Figure 2.8: An illustration of SIFT descriptor construction.[68]

The descriptor construction is illustrated in Fig. 2.8. The arrows in the first part of the image represent the gradient magnitudes and orientations calculated earlier. They are later rotated according to the dominant orientation. The circle around the local pixel neighborhood illustrates the Gaussian weighting of gradients to make the nearby gradients more significant. In the middle of the figure, a 2×2 SIFT descriptor is shown. Each of the four cells contains accumulated gradients to 8 directions calculated from a 4×4 sample array. Although other sizes can be used, 4×4 sample arrays are usually used with SIFT, as they are reported to give the best results [68]. A SIFT descriptor is a vector with 128 dimensions. All vector components are normalized into 8-bit unsigned integers, i.e. range of values $\mathcal{S} = \{0, ..., 255\}$.

2.2.2 Speeded-Up Robust Features (SURF) descriptor

The SURF descriptor is a rotation and scale invariant local feature descriptor by Bay *et al.* [5]. The rotation invariance is achieved by finding reproducible orientation for the local neighborhood of a keypoint. When the scale of a detected keypoint is s, the Haar wavelet responses in both x and y direction are calculated in the circular neighbordhood of size $6s$. After calculating the filter responses, the local neighborhood is weighted with a Gaussian with $\sigma = 2$ to make the nearest intensity changes the most significant. In practice, the calculated wavelet responses are handled as points in 2-D space, X and Y axes represent responses in horizontal and vertical directions, respectively. A sliding "orientation window" (a sector) of size $\frac{\pi}{3}$ is used around the keypoint surroundings to calculate the sum of horizontal and vertical responses. Sums of responses are then used to calculate a local orientation vector for each direction. The longest such vector is finally selected to represent the dominant orientation of a descriptor.

Also, the responses of Haar wavelets are used in building the actual local feature descriptor. The first step is to select an area around the keypoint (detected in scale s) of size $20s$. The region is split up into 4×4 sub-regions and for each sub-region, Haar wavelet responses are calculated for 5×5 blocks from the grid of sample points. In practice, to decrease computational complexity, the wavelet responses are calculated using an unrotated image and then approximated with the rotated box filters for various descriptor orientations as needed. The responses around the keypoint are Gaussian-weighted ($\sigma = 3.3s$) to increase robustness towards geometric deformations and localization errors. Each of the 4×4 sub-regions contains 2×2 smaller regions where response strengths are summed. A feature vector $\mathbf{v}$ calculated from these response strength sums of sub-regions is then:

$$\mathbf{v} = (\sum d_x, \sum d_y, \sum |d_x|, \sum |d_y|). \tag{2.17}$$

Where d_x, d_y are the wavelet responces in horizontal and vertical directions. When the 16 vectors are combined, $16 \times 4 = 64$ dimensional vector is formed. By definition, SURF sums are invariant to illumination changes. To be invariant to contrast (scaling factor), a feature vector is turned into a unit vector, i.e. the vector is divided by its length:

$$\hat{\mathbf{v}} = \frac{\mathbf{v}}{\|\mathbf{v}\|}. \tag{2.18}$$

The integral images are exploited in descriptor building to boost the calculation of filter responses. Additionally, the original paper by Bay *et al.* [5] reports results for different sizes of SURF descriptor, showing that the SURF descriptor is 5×5 faster to calculate and provides a more than 10% better recognition rate than the SIFT descriptor.

2.2.3 Efficient Dense Descriptor (DAISY)

DAISY is a local feature descriptor presented by Tola *et al.* [108]. It is designed to be efficient to compute densely especially for wide-baseline tasks. Instead of calculating the descriptor at separate detected interest points, the descriptor calculation is formed so that it can be calculated for every pixel. To achieve this, the descriptor uses convolutions of the gradients instead of weighted sums as with SIFT descriptor. They found out that this gives similar invariant characteristics as with SIFT but was much faster to compute densely.

First, the orientation maps are computed. Orientation maps are image gradient norms at location (u, v) for each direction iH. The orientation maps are then convolved with Gaussian kernels of different Σ values. This produces *convolved orientation maps* which are used to produce the final descriptor. The descriptor itself is calcuated from a fixed shape arrangement with different values for Σ and normalized to unit form. The DAISY descriptor is produced of the Σ-convolved normalized orientation maps with different distances from the original origin point (u, v). The parameters for the method were chosen empirically using wide-baseline testing procedure.

The performance evaluation was made with dense sampling comparison between DAISY, SIFT, SURF, NCC and direct pixel difference. The presented results show notable improvements over previous methods and the method is advertised to be able to handle lower resolution video frames easily with a proper GPU implementation.

2.2.4 Binary Robust Independent Elementary Features (BRIEF/ORB) Descriptor

The binary descriptors discussed here are Binary Robust Independent Elementary Features (BRIEF) and Rotated BRIEF (rBRIEF or ORB). The BRIEF descriptor was originally introduced by Calonder et al. [12] to compete against the SURF descriptor in both speed and storage. One practical difference between the binary descriptors and the other descriptors presented in this chapter, is that with the binary descriptors, the feature matching requires distance calculation bitwise instead of the traditional Euclidean distance over real numbers.

The original BRIEF descriptor is a bit string description of a detected image patch. The bit string is defined by a set of binary tests on a smoothed image patch with Gaussian kernel of size σ. The binary test τ on patch p of size $S \times S$ can be defined as:

$$\tau(x, y) := \begin{cases} 1 & \text{if } p(x) < p(y) \\ 0 & \text{otherwise} \end{cases}$$

where $p(x)$ is the pixel intensity in a smoothed version of image patch, p at point x. The descriptor itself is defined as a vector of n binary tests:

$$f_n(p) = \sum_{1 \leq i \leq n} 2^{i-1} \tau(x, y) \tag{2.19}$$

where $n = 256$ with the traditional 32 bytes long BRIEF descriptor. The spatial arrangement of the test pairs (x_i, y_i) was examined by Calonder *et al.* [12]. It was found that the test pairs sampled from an isotropic Gaussian distribution produced the best results (Gaussian$(0, 1/25S^2)$, where S was the size of the image patch). They also presented performance results against the popular SURF descriptor. The construction of the BRIEF descriptor was 37 times faster than SURF and the matching the features was 6 times faster with BRIEF. Howerer, it is evident that the BRIEF descriptor does not support any affine invariance, which was noted by the authors. Rublee *et al.* [98] presented rBRIEF, the rotation invariant version of BRIEF to improve this aspect of the descriptor.

Instead of computing the BRIEF descriptor multiple times with different rotations and scales, they steered the BRIEF operator according the orientation of keypoints, i.e. rotate the binary tests. The matrix $\mathbf{S}_\sigma$ is the list of binary tests with corresponding rotation σ:

$$\mathbf{S}_\sigma = \mathbf{R}_\sigma \begin{bmatrix} x_i & \dots & x_n \\ y_i & \dots & y_n \end{bmatrix} \tag{2.20}$$

Now, the steered BRIEF can be defined as:

$$g_n(p, \sigma) := f_n(p) | (x_i, y_i) \in S_\sigma \tag{2.21}$$

The angle σ is discretized to increments of $2\pi/30$. It was noticed that the steered version did not have high variance in terms of how each test contributed to the descriptor. The low variance causes the descriptor to become undiscriminative. To recover the loss in variance, they utilized a learning method for selecting a good set of binary tests. They computed keypoints drawn from PASCAL image set [30] and enumerated every possible binary test. The tests are sorted by their distance from a mean 0.5. Now, they greedily select the best 256 binary tests which are as "diverse" as possible. These selected binary tests are then used to create the binary vector forming the rBRIEF descriptor (ORB).

2.2.5 Local Intensity Order Pattern for Feature Description (LIOP)

The increased usage of binary based descriptors has brought up interesting alternatives. Local Intensity Order Pattern for Feature Description (LIOP) by Wang *et al.* [132]. The proposed descriptor was designed to encode both local and overall intensity information to provide improvements over the existing state-or-the-art. LIOP descriptor divides the detected region into subregions called ordinal bins. For each ordinal bin, the LIOP descriptor is constructed by comparing the intensities of the center point with the N neighbouring sample points. The final LIOP descriptor is then constructed by accumulating the LIOPs of points in each ordinal bin and concatenating them.

For experiments they used Harris-Affine, Hessian-Affine, MSER, EBR and IBR region detectors. The presented results show that LIOP descriptor outperforms SIFT and DAISY descriptors especially with the illumination changes.

2.3 Summary

This chapter discussed the previous work on the popular local feature detectors and descriptors. The local features presented in this chapter contain very different chracteristics both in detection and in descriptor generation. These detectors and descriptors can be easily evaluated with the presented approaches, which translate well to the problems, such as wide-baseline matching [111, 115].

A good descriptor needs to be able to tolerate a wide range of different geometric and photo-metric anomalies such as zoom, blur, rotation and illumination changes. Many of these problems are usually averted by making sure the environment minimizes the problems with imaging (e.g. by providing sufficient lighting).

The local feature detector LoG, used by the popular SIFT descriptor, was originally designed to provide accurate matches between images. However, many detectors and descriptors presented later has been criticizing the computation speed of feature detection and the large storage size of the descriptor. Different authors have tried to improve the computational speed by providing faster and simpler detectors and descriptors. The latest binary descriptors show that applications such as SLAM [39] can be done effectively in real-time with local features which perform as well as SIFT, but are many times faster to compute.

Chapter III

Detector and Descriptor Evaluation for Visual Object Classes

The popular local features were presented in the previous chapter and now the evaluation frameworks by Mikolajczyk *et al.* [83, 82] will be introduced and extended. It is important that standard ways to evaluate the local feature detectors and descriptors are established so that fair comparisons can be made between them. The standard ways to evaluate detectors and descriptors are already well established [83, 82], but the evaluation strongly reflects its origins in the wide-baseline matching [115] and applications which use images of the same scene, such as robot localization [101] and image stitching for panoramic views [11]. In these, correspondences are sought between different views of a same scene and the results in [83, 82] help to select the most suitable method. The color properties of the common local feature variants have been evaluated by Sande *et al.* [117]. Noteworthy, they evaluated the local features with the popular object detection data sets and showed the best results with a color sensitive variant of SIFT.

It is not evident how the detectors and descriptors behave when the target application is object detection, which does not deal with images of a scene but with images of one object class. It is already shown that the local features work very well in this category as shown in the results of annual PASCAL VOC Challenge [30].

Here, the detectors were evaluated by their repeatability rates and the total number of correspondences for different view points of several views and with various imaging distortions. The descriptors were evaluated by their matching rates for the same scenes. In this chapter, the state-of-the-art detectors and descriptors are evaluated in the visual object categorisation context. The detector repeatability evaluation procedure in [83] for object categories is extended so that the intra-class number of correspondences and repeatability rates are reported as performance numbers. The evaluation method also extends the descriptor matching evaluation in [82] for object categories. The intra-class match counts/rates are shown in the experiments section of this chapter.

It is fair to believe that the evaluation principles in [82, 83] also hold in the visual object categorisation context:

1. The local feature detectors which return the same object regions for category examples are good detectors. This means *detection repeatability.*
2. The local feature descriptors which match the same object regions between category examples are good descriptors. This leads to *matching score* for descriptors.

Success in the final task, categorisation, is important for the final application, and therefore, Zhang et al. [130] compared different detectors and descriptors using a baseline bag-of-features (BoF) method. In the baseline method, codebooks are constructed from the raw descriptors and then images are categorised according to their codebook histograms.

For the feature detection evaluation in object recognition, Mikolajczyk*et al.* [79] were more specific by measuring average precision of the feature clusters to represent a single class. Therefore, they measured the entropy of the spatial location distribution produced by a single cluster (ideally compact) and the complementarity of different detectors. Furthermore, in [79] the authors did not find any evidence for any detector or descriptor being more suitable for one particular category of objects, which is not true for the experiments presented in this chapter.

Both of these evaluations, the cluster entropy and the complementarity are biased by the fixed approach of visual Bag-of-Words. Furthermore, they adopt the original evaluation principles and thus obtain quantitative performance in the general and intuitive terms used in the original works, and are not tied to any specific approach.

3.1 Previous work

The evaluation method presented in this chapter is basically an extension on the evaluation framework by Mikolajczyk *et al.* [82]. The framework presented a fair way to evaluate a large set of different local features as it allowed different blob detectors to be added to the evaluation. This evaluation framework evaluates the overlap of the detected areas of interest (detector test) as well as how well these regions actually match (descriptors). They also provide the evaluation framework source code, images and ground-truth data for making any other evaluations in the future.

The dataset for the evaluation framework uses a small set of real images with different photometric and geometric transformations. In total, six image transformations are evaluated: rotation, scale, viewpoint, blur, compression, and illumination. The real images chosen contain homogenous regions with distinctive edges such as buildings and graffiti. The set also contains repeated textures of different forms to comfort structured versus textured scene situations.

The real images taken with a camera have different scale ($\times 1 - \times 4$) and blur by changing zoom and focus respectively. The viewpoint is evaluated by different fronto-parallel views and views with strong foreshortening (60 degrees) to the camera. The rotation is acquired by simply rotating the camera around the center of the view. The JPEG compression test is done by applying different compression parameters (from 40 to 2 percent) with xv program[1]. All the images are approximately 800×640 pixels and there are six images in each test sequence.

The ground-truth for the plane projective transformations is acquired by doing a homography estimation between the roughly aligned image pairs. The rough approximation of the transformation is acquired by annotating the images with correspondent points. This approximation is further improved by baseline homography estimation which includes

[1] http://www.trilon.com/xv/

hundreds of automatically detected interest points. This gives an accurate homography between the reference image and the other image.

3.1.1 Local feature detector evaluation

To evaluate local feature detectors, the overlap comparison was presented. The overlap metric introduces the repeatability and the accuracy of localization. The repeatability is the average number of the correspondent image regions detected between the reference image and the other image (after geometric and photometric transformations). When local feature regions from two images are compared, an exhaustive search is done to find the overlapping regions (ellipses). The two regions are matching if the overlap error is sufficiently small. The overlap error is estimated as:

$$1 - \frac{R_{\mu_a} \cap R_{(H^T \mu_b H)}}{(R_{\mu_a} \cup R_{H^T \mu_b H})} < \in_0 \tag{3.1}$$

Where R_μ are the detected regions and H is the homography relating the two images. Mikolajczyk *et al.* [82] set 40% overlap error. In overlap measurement, the sizes of ellipses have an effect on results. The bigger the ellipses are, the smaller is the overlap error in the measurement. For that reason, all the ellipses are normalized to a radius of 30 pixels before calculating the overlap error. When the number of correspondences is known, the ***repeatability rate*** can be written as:

$$repeatability\, rate = \frac{\#\, of\, correspondences}{min(\#\, of\, reg\, in\, img\, A, \#\, of\, reg\, in\, img\, B)} \cdot 100\%. \tag{3.2}$$

Only regions present in both images A and B are included. The known homography is used to project the ellipses from the reference image to the transformed image. This causes some of the features to not necessarily be present in both images and affect the overall rating score.

3.1.2 Local feature descriptor evaluation

As stated in Chapter 2, it is preferred for a local feature descriptor to provide good matches. Thus, the evaluation on local feature descriptor is basically local feature matching estimation. In the matching estimation, the found correspondences between regions are used as ground-truth for the descriptor evaluation. The descriptor matches are sought by using a few different distance metrics.

Mikolajczyk *et al.* [82] tested three different selection rules. The first, simple threshold approach, calculated the distances between descriptors of two images. If the distance was less than the specified threshold, a match was registered. In the second, the nearest-neighbor approach, only the shortest descriptor distance with a distance less than the decided threshold was registered as a match. However, the task of choosing a good threshold value is very difficult or even impossible because the distances between descriptors contain a lot of variation. This problem was addressed by Lowe *et al.* [68] for SIFT descriptor matching. They solved it by utilizing a relative threshold using the ratio of distances between the nearest and the second nearest neighbors. For every descriptor, distances to all descriptors from the other image are calculated and sorted. If the distance to the nearest descriptor is smaller than 1.5 times the distance to the second-nearest, a match is not accepted.

The *matching score* is calculated as the ratio between the number of correct matches and the number of detected regions: Here, if the source features (ellipses) of matching descriptors overlap, a match is considered to be correct. In case the regions do not overlap, the match is not detected.

3.2 Data and ground-truth

The experiments for both detector and descriptor evaluation were conducted using the popular Caltech-101 dataset [33]. The selected dataset is widely used for benchmarking purposes in object categorization. For compactness and clarity, the results are reported for the following ten categories which represent the overall performance variation well: *watch*, *stop_sign*, *starfish*, *revolver*, *euphonium*, *dollar_bill*, *car_side*, *airplanes*, *Motorbikes* and *Faces_easy*. Caltech-101 provided information for segmenting objects from the background and this information was used to mask out local features detected in the backgrounds.

Affine correspondences between the examples were established by manually annotating at least 5 object landmarks and by estimating the pair-wise transformations with the direct linear transform [45]. Example images with landmarks are shown in Fig. 3.1. For the experiments, 25 random pairs of images were used from each category, resulting in the total of 500 images. The standard deviations of the projection errors are reported. The error is evaluated by calculating the standard deviation for each landmark in pixels and normalizing the pixel error with image diagonal.

3.3 Comparison of local feature detectors

A good detector should detect local points or regions at the same relative locations, “object landmarks”, on every class example. This criterion differs from the evaluation by Mikolajczyk *et al.* [83] in the sense that here the detectors are evaluated over different instances instead of different views of the same instance. For this reason, the variation of visual appearance is expected to be much larger.

3.3.1 Selected detectors

The comparison includes nine publicly available and popular detectors, which were introduced in Chapter 2. The majority of these detectors have been evaluated in earlier studies, e.g. Mikolajczyk *et al.* [83]. The selected detectors and the implementations are described in Table 3.1. In total, there are three implementations of Hessian-affine, two implementations of Difference-of-Gaussian (DoG), one implementation for SURF, Laplacian-of-Gaussian and Harris-Laplace and MSER each. The Hessian-affine detector presented in [80] performed well in the comparison by Mikolajczyk *et al.* [83]. For this reason, the Mikolajczyk’s original (*hesaff-alt*) and a more recent implementation (*hesaff*)

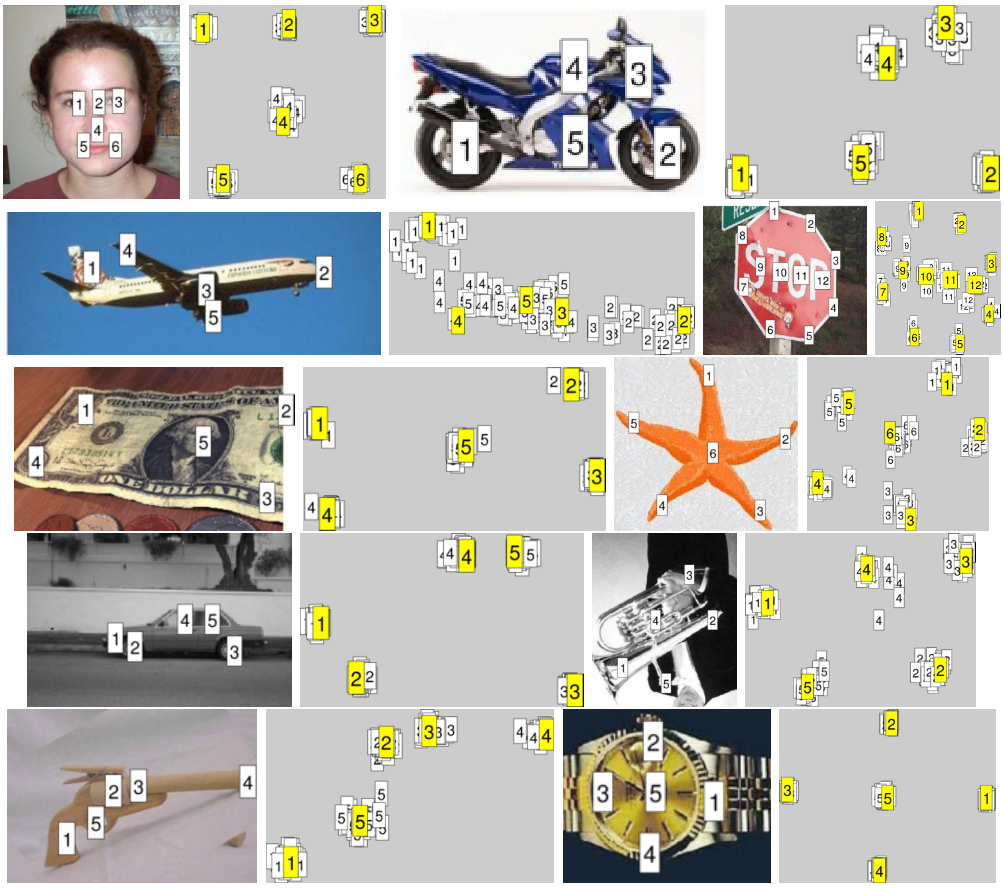


Figure 3.1: Selected object classes with annotated landmarks and their "canonical spaces" where all landmark tuples are projected onto the first example (denoted by the yellow tags). The two standard deviations of the image diagonal normalised projection errors are (from left to right and top to bottom): 0.0158, 0.0297, 0.1701, 0.0460, 0.0304, 0.0373, 0.0194, 0.0641, 0.0486, and 0.0177.

were selected [78]. An alternative implementation by Zhao [131] (*hesslap-vireo*) with a slightly different algorithm and different default parameters was also selected.

The set of "fast detectors" consisted of Difference-of-Gaussian (DoG) by Lowe [68] (*sift*), Zhao's implementation of DoG (*dog-vireo*) and speeded-up robust features (SURF) by Bay *et al.* [5] (*surf*). In addition, Zhao's implementation of Laplacian-of-Gaussian (LoG) (*log-vireo*), Harris-Laplace (*harlap-vireo*) and Maximally Stable Extremal Regions (MSER) by Matas *et al.* [75, 119] (*mser*) were included.

There are many more detectors, but the aforementioned are the ones that performed best in earlier studies and the ones that performed well in the preliminary tests. Moreover, all experiments were conducted using the available implementations and their default parameters. The implementations made by Zhao are an exception to this rule as all

Table 3.1: The detectors selected for the evaluation.

Name	Method	Implementation
hesslap-vireo	Hessian-Affine	FeatureSpace [78]
hesaff-alt	Hessian-Affine	robots.ox.ac.uk [123]
hesaff	Hessian-Affine	FeatureSpace [78]
sift	Difference-of-Gaussian	VLFeat [119]
dog-vireo	Difference-of-Gaussian	Vireo [131]
harlap-vireo	Harris-Affine	FeatureSpace [78]
log-vireo	Laplacian-of-Gaussian	Vireo [131]
mser	MSER	VLFeat [119]
surf	SURF	ETH [5]

detectors are configured to enable affinity detection by default, i.e. all detectors are affine invariant.

3.3.2 Performance measures and evaluation

For the detector performance evaluation, the test protocol is similar to the approach presented in Section 3.1.. In the same fashion, the interest points and/or regions are first extracted from images. Only points with their centroid in the object area (Caltech-101 foreground) are selected for the evaluation procedure. The evaluation procedure is in Alg. 3.1.

Algorithm 3.1 The detector evaluation procedure

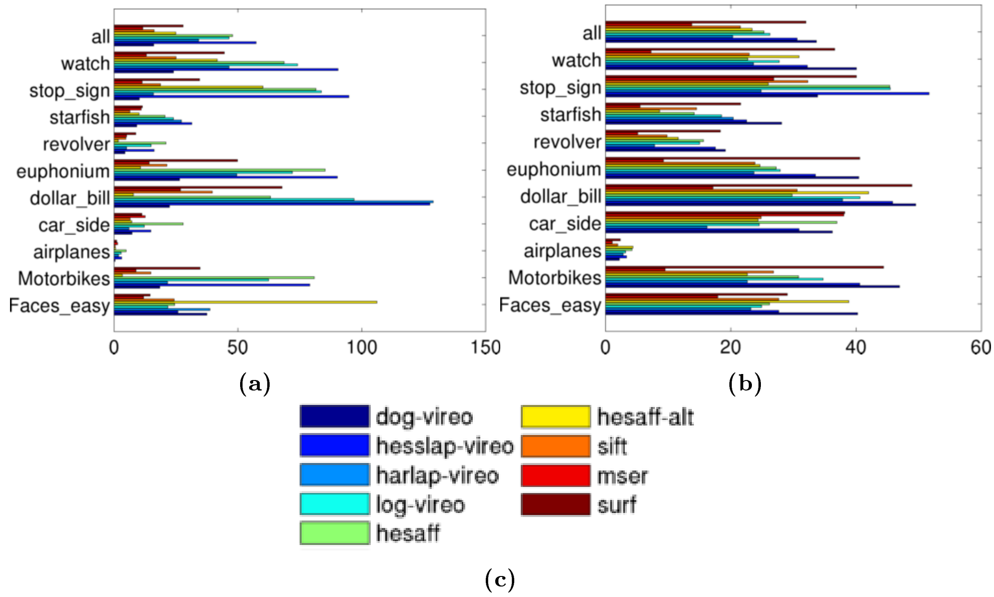
```
Extract local feature regions from all images
Filter out local features in the background
for all image pairs (indexed with $i$) do
    Estimate 2D homography $H$ from the first image to the second image
    Transform all detected regions onto the second image using $H$
    for all regions do
        if Overlap is more than threshold $t$ then
            Increment the correspondency score $\boldsymbol{c}_i$
        end if
    end for
end for
Return the correspondency result $\boldsymbol{c}$
```

For each image pair, points from the first image are projected onto the second image. The projection is an affine transformation estimated using the annotated landmarks. The landmarks projected on the first example of each category are demonstrated in Fig. 3.1 with the two standard deviations corresponding to the 95% error distributions.

As explained in Section 3.1., the 40% overlap threshold and normalisation of the ellipses to the radius of 30 pixels is used. Similarly, the reported performance numbers are the average number of corresponding regions between image pairs and the repeatability rate, i.e. the ratio between the corresponding regions and the total number of detected regions.

3.3.3 Results

Results are gathered in Fig. 3.2 with the numerical results being in Table 3.2. There are significant differences between the different categories. *Dollar_bill* and *stop_sign* are generally the easiest, as expected due to lower variability in their visual appearance, while the *airplanes*, *car_side* views and *revolvers* are the most difficult. For the airplanes this can be explained by the fact that one of the landmarks is on the wing resulting in 3D pose changes instead of 2D affinity. The numbers for all categories are by order of magnitude smaller than for the fixed scenes in [83], being tens of correspondences instead of hundreds of them.



(a) (b)

(c)

Figure 3.2: Detector evaluation: (a) average number of corresponding regions, (b) average repeatability rate, and (c) colour coding of the method names.

The following three methods have good repeatability ratio: *hesslap-vireo*, *dog-vireo* and *surf*. They frequently detect regions from the same relative positions in all examples. On the other hand, VOC methods require a sufficient number of correspondences to form a discriminative description. In this sense, hesslap-vireo (≈ hessaff), is very good. Its repeatability rate is the third best (30%) and it provides more correspondences (57) on average. Dog-vireo (≈ sift) has the best repeatability (33%), but its average number of correspondences (16) can be too low for categorisation tasks.

Surf detector has the second best repeatability (32%) and it provides 27 correspondences on average. Since the repeatability rates for the three best methods are very similar, the selection is based on the preferred number of correspondences. Surprisingly, the MSER detector (*mser*), which was among the top performers in the original study, was now clearly the worst. It is furthermore interesting, that Zhao's recent implementations provide better results than the original methods (hesslap-vireo ↔ hessaff/hessaff-alt and

Table 3.2: Detector evaluation: The overall results in numerical form. Top-3 performers are written in bold.

Method	Avg # of corr.	rep. rate
dog-vireo	16.0	**33.7%**
hesslap-vireo	**57.4**	**30.6%**
harlap-vireo	34.2	20.3%
log-vireo	**46.5**	26.3%
hesaff	**47.8**	25.3%
hesaff-alt	25.0	23.4%
sift	16.2	21.5%
mser	11.7	13.8%
surf	27.9	**32.0%**

dog-vireo ↔ sift).

In summary, the recent implementations, hesslap-vireo, hessaff and log-vireo perform best, ≈ 50 corresponding regions on average with repeatability rate of 25-30%.

3.4 Comparison of local feature descriptors

A good region descriptor for the problems in the area of VOC should be discriminative to match only correct regions, but also tolerate small appearance variation between category examples. These requirements are general for feature extraction in computer vision and image processing.

The original method for evaluating the local feature descriptors was presented in Section 3.1. In the case of visual object categorization, the descriptor matches are expected to be weaker due to increased variance in regions' visual appearance. For example, scooters and roadbikes both belong to the Caltech-101 motorbikes category, but their pair-wise similarity is much weaker than that of two scooters or two roadbikes. Moreover, there is natural variation in the spatial configurations of the regions (constellation deformation). Therefore, the original evaluation method was tailored to cope with these two sources of variation. This comparison also proposes an alternative measure, *coverage*, which measures the number of images "covered" with at least the given number of correspondences N (coverage-N).

3.4.1 Selected descriptors

Out of the many available descriptors, the most frequently used and best performing were selected for evaluation. The selected descriptors and the implementations are described in Table 3.1. In the original comparison [82], the SIFT descriptor by Lowe [68] and its extension, the gradient location and orientation histogram (GLOH) [82], obtained the best results. SIFT was selected for this comparison.

As with detectors, descriptors for this evaluation were selected using the existing knowledge and preliminary tests on their performance. As the chosen detector also has impact on the performance of the descriptor, some descriptors occur in the descriptor evaluation more than once combined with various detectors.

A more recent descriptor SURF, in addition to being very fast detector, is a robust descriptor and conceivably more tolerant to at least moderate amount of noise than SIFT [5]. In most works, the descriptors used are chosen from these three.

In addition, the in-house implementation of one "traditional" descriptor Gabor, namely response vector of oriented linear filters, which was included in the original work [82] but did not place among the best performers, was included in the evaluation because in the VOC context, the better generalisation and weaker specificity could be advantageous.

Table 3.3: The descriptors selected for evaluation.

Name	Method	Implementation
gabor	Gabor	MVPR
sift	SIFT	VLFeat [119]
sift-vireo	SIFT	Vireo [131]
surf	SURF	ETH [5]

For the SIFT descriptor, two implementations were selected: the close approximation (by VLFeat) of the original by Lowe [68] (*sift*) and a more recent by Zhao [131] (*sift-vireo*). The oriented linear filter descriptor (*gabor*) was an in-house implementation. For SURF, the implementation by its original authors [5] (*surf*) was used. Ideally, we should combine these descriptors with all three best detectors, but the following best combinations were selected according to our preliminary tests:

1. *hesslap-vireo+sift-vireo* (≈ hesaff+sift),
2. *dog-vireo+sift-vireo* (≈ sift+sift),
3. *hesaff+sift*,
4. *surf+surf*,
5. *hesaff+gabor* and
6. *sift+sift*.

It should be noted that the available executables do not allow arbitrary combinations, thus forcing the evaluation to use the implementation of the descriptor by the same author as the detector.

3.4.2 Performance measures and evaluation

The main work flow is similar to the original work described in Section 3.1. Similarly, descriptors are computed for all detected regions (foreground only). Images are processed pair-wise and best matches for the regions sought by computing one-to-all distances and selecting the closest match. It is noteworthy that the rule proposed in [68] for discarding "bad regions" (distance between the first and the second best is less than 1.5×) must not be used in VOC. This rule is default in many implementations, but it causes the matching performance to collapse. The reason is that matches are rarely as good as in the wide-baseline matching case. The evaluation procedure is shown in Alg. 3.2.

Algorithm 3.2 The descriptor evaluation procedure

```
Extract local feature regions for all images
Filter out local feature regions in the background
for all image pairs (indexed with $i$) do
    Estimate 2D homography $T$ from the first image to the second image
    Transform all detected regions onto the second image using $T$
    for all regions do
        Find the $N$ closest descriptors associated with the region
        if Any of the $N$ closest descriptors is closer than threshold $t$ then
            Increase the number of matches for $\boldsymbol{m}_{di}$
        end if// Alternatively:
        if Any of the $N$ closest descriptors with region has overlap more than threshold $t$ then
            Increase the number of matches for $\boldsymbol{m}_{oi}$
        end if
    end for
end for
Return the coverage value $\boldsymbol{c}$ for $N$ closest matches
```

The spatial verification stage differs from [82] by being less strict since the original rule provides only a few matches for the most pairs. In the original rule, the regions were described by ellipses, and for the spatial verification the ellipses were projected onto each other using the estimated affine transformation. If a sufficient overlap occurred for the ellipses, the match was accepted. The results using the original overlap rule are presented in Section 3.4.5.

Furthermore, in this case, the categories have natural variation in their spatial structure. This natural variation cannot be exactly encoded into affine transformation, and therefore, the matches are not exact even for the ground truth landmarks as demonstrated in Fig. 3.1. The two standard deviations vary between 0.0158 (Faces_easy) and 0.0641 (euphonium). The airplanes category is an outlier due to the one non-2D landmark on the wing tip (0.1701). However, for ellipse overlap computation, even a small difference in the ellipse centroid may have an enormous effect on the overlap area [83] as will be seen in the results in Section 3.4.5.

For VOC evaluation we needed to replace the ellipse overlap rule with a distance threshold between the ellipse centroids. For resolution independence, the distances were normalised with the image diagonal, and in the evaluation, the matches were discarded if the distance was greater than threshold value $t = 0.05$. This threshold covers the two standard deviations of the ground truth landmarks and over 95% of the landmarks are within this distance.

The presented performance numbers are the *average number of matches* and *median number of matches*. The median is used to suppress the effect of several too well matching pairs (a same person, two very similar stop signs, *etc.*) It is also important to know how many matches are guaranteed to be found. This can vary between images and is not uniform if the difference between the average and the median is large.

Table 3.4: Descriptor evaluation overall results table. The number of matches can be larger than the number of correctly detected regions due to the new matching criterion not using the ellipse overlap rule (see Sec. 3.4.2 for details).

Method	**Avg #**	**Med #**
hesaff+gabor	**58.9**	**39.0**
hessaff+sift	**66.1**	**46.0**
dog-vireo+sift-vireo	18.7	15.0
surf+surf	12.3	9.0
sift+sift	9.5	6.0
hesslap-vireo+sift-vireo	30.9	22.0

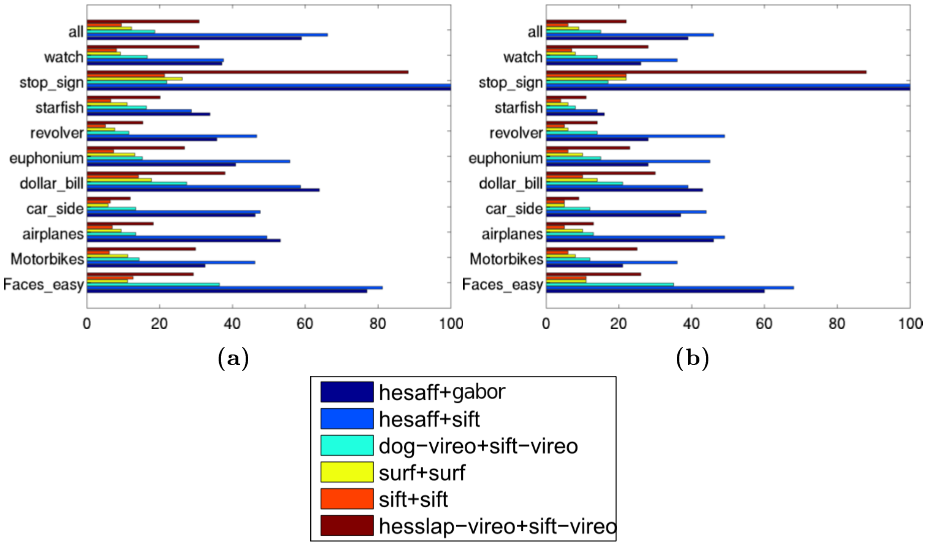
3.4.3 Coverage-N performance

In this work, an additional measure is introduced: *coverage-N*. For some applications, it is important to know how many matching features are guaranteed to be found. Thus, the term "coverage" to show in how many of the image pairs descriptor matches were found. *Coverage-N* then shows the pairs where N or more matches are found. It can reveal some properties of descriptors as some are able to obtain a few matches even in very challenging image pairs in which no matches can be found using most descriptors. It is important to note that the detector used affects this as it determines the spatial locations in the image that descriptors are calculated for.

3.4.4 Results using the distance rule

The major difference from the original descriptor evaluation approach presented in Section 3.1 was the spatial verification stage. Here, the regions are only presented as coordinates with Euclidean distance instead of ellipse overlap. The distance approach is less strict as it does not require matched features to have the same shape as long as the descriptors match. The data for this experiment were the same as for the detector comparison. The average and median number of matches are shown in Fig. 3.3 with the numerical results being in Table 3.4. The results verify the findings in the earlier works: the Hessian-Affine and SIFT detector-descriptor-pair leads to the largest number of matches. Overall, these results also seem to verify the important finding by Nowak *et al.* [88] that detector-descriptor combinations with a detector providing a larger number of correspondence candidates perform well. *hessaff* and *hesslap* based methods clearly outperform those using *sift* (*dog*) and *surf* (see also Fig. 3.2(d)). The weaker performance of hesslap-vireo+hesslap-sift can be explained by the fact that the vireo code does not do full affine normalisation (only one iteration), which seems to degrade matching with the SIFT descriptor.

It is noteworthy, however, that in the VOC context even the more traditional descriptors, such as our gabor descriptor, are almost equal to the SIFT when paired with the Hessian-affine detector. This was not the case in the original work in the context of wide-baseline stereo matching.



(a)

(b)

(c)

Figure 3.3: Descriptor evaluation results with distance rule: (a) average number of matches per class, (b) median, and (c) colour coding of the method names.

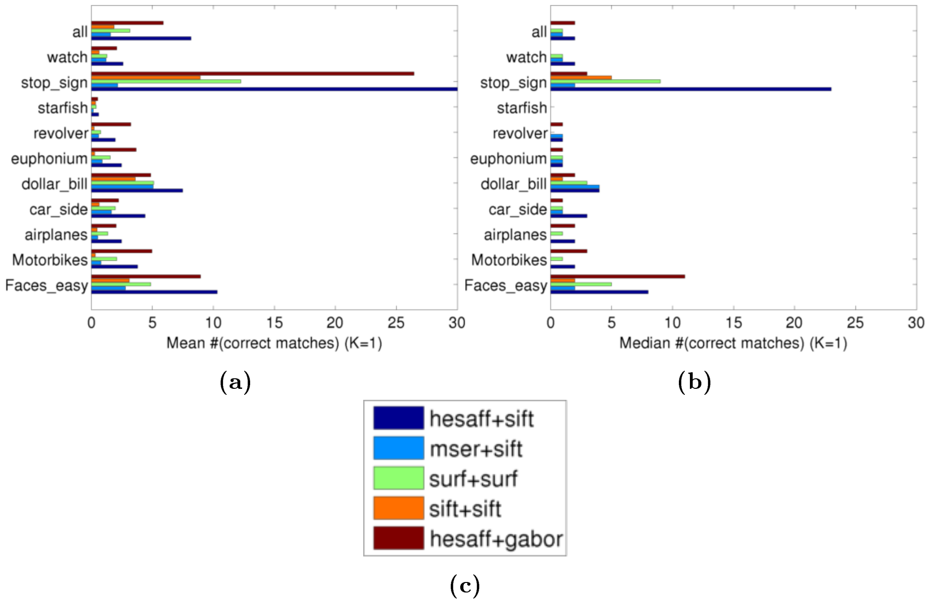
3.4.5 Results using the overlap rule

If the original ellipse overlap rule (with 40% overlap requirement) is applied with the proposed method, the results drop drastically. The results can be seen in Fig 3.4.

It is evident that the local features, when matching, might not be visually same. This is strongly visible when there are a lot of features, but with many different scales. Eventhough there is a high amount of overlapping regions presented in detector evaluation, it is shown here that most descriptors are unable to match the potential regions. However, the overall results compared to the distance based metric did not change. Hessian-Affine coupled with any descriptor was the best performer, followed by SURF and SIFT respectively.

3.5 Discussion

Based on the results in the Sections 3.4.4 and 3.4.5, it seems straightforward to claim that the best detector-descriptor-combination for VOC is the Hessian-affine detector with the SIFT descriptor (or with Gabors). However, if the matching ratio of the matches are plotted with respect to all detected regions, shown in Fig. 3.5, it is clear that this combination is not much better than any other. The high number of matches is due merely to a larger number of possible candidates. It is evident that there is a clear correlation between the number of detected feature regions and the performance in the object-class descriptor evaluation. This leads to the question how to decrease the number of regions without sacrificing the matching performance.

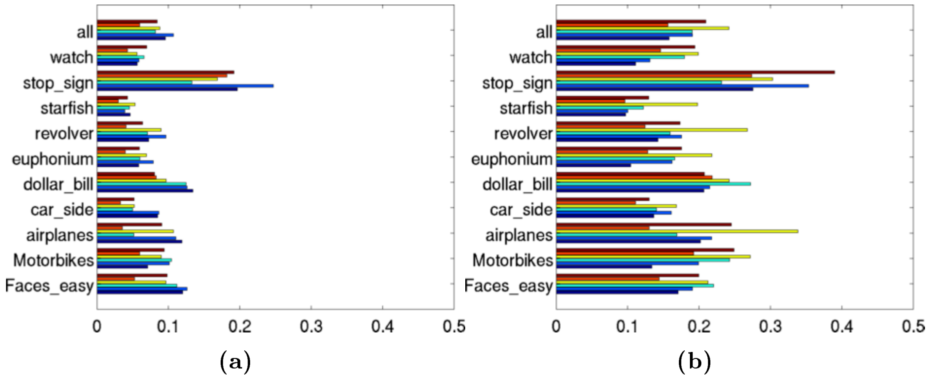


(a) (b)

(c)

Figure 3.4: Descriptor evaluation results with overlap rule: (a) average number of matches per class, (b) median, and (c) colour coding of the method names.

Intuitively, the best matches are not always correct between two object class images, and thus, not only the best, but a few best matches can be used. This hypothesis was tested by counting matches as correct if they were within the K best and located within the normalised distance ($d_n < 0.05$). The matching ratios for $K = 5$ are shown in Fig. 3.5. The number of matches increases for all methods, but the effect is stronger for the detectors with a good detection ratio performance, SURF, in particular.



(a) (b)

Figure 3.5: The matching ratio for (a) the original experiment with the best match only ($K = 1$) and (c) for $K = 5$ best matches. Uses same labels as Fig. 3.3.

Hessian-affine results, on the other hand, become proportionally weaker compared to other methods. This means that whatever the heuristics for selecting interesting regions in the methods providing fewer candidates are, the same matches are more likely found in other class examples. This hypothesis can be further verified by measuring the *coverage-N* instead of the average or median number of matches or their ratio. The coverage-N measures the number of images for which at least N correct matches have been found. Coverage-5 and coverage-15 for $K = 1$ and $K = 5$ are shown in Fig. 3.6. By selecting $K = 5$ best candidates there is practically no difference between the detector-descriptor pairs.

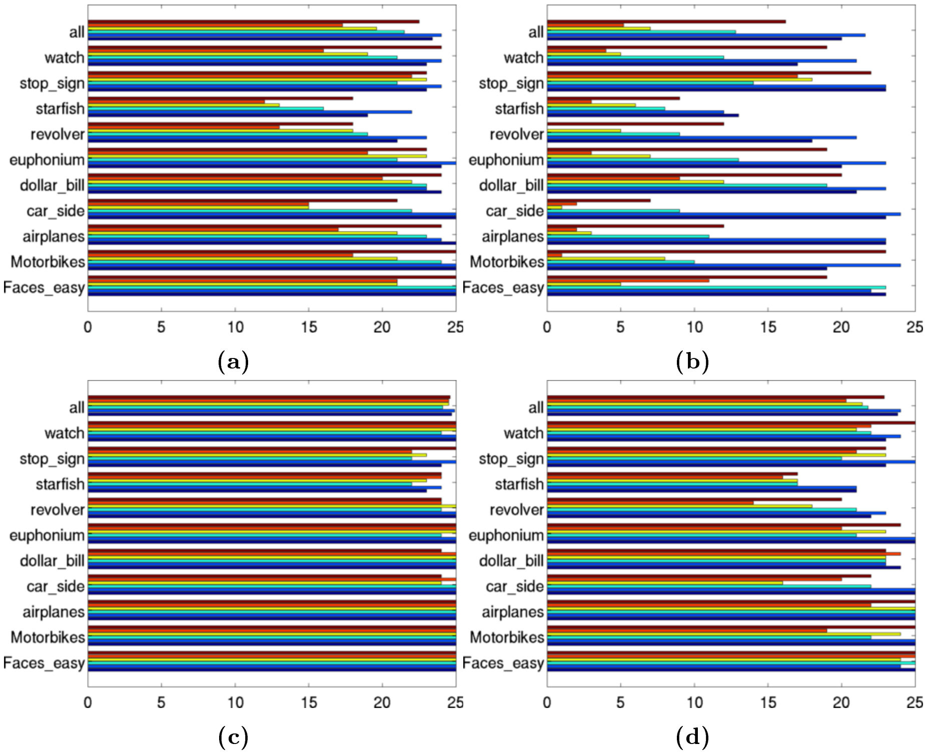


Figure 3.6: (a) coverage-5 for $K = 1$; (b) coverage-15 for $K = 1$; (c) coverage-5 for $K = 5$; (d) coverage-15 for $K = 5$. For "all" the value must be multiplied by 10 (25 pairs for ten classes). Uses same labels as Fig. 3.3.

3.6 Summary

In this chapter, the well accepted and intuitive interest point detector and descriptor performance measures by Mikolajczyk *et al.* [83, 82], repeatability and number of matches, were extended to measure intra-class performance with visual object categories. The most popular state-of-the-art detectors and descriptors were compared using the Caltech-101 data set. This work was motivated by the fact that the original performances were computed in a wide baseline setting, but the most popular use of the detectors and descriptors is in visual class detection and categorisation, which are tasks with clearly distinct requirements.

The detector experiment was yet another verification of the fact that SIFT and SURF are the most reliable in the terms of repeatability rate even on category level, but their marginal to the Hessian-affine is not significant and the Hessian-affine has the complementary feature of providing many more interest points (48 vs. 16 for the SIFT and 28 for the SURF, on the average). However, the most interesting result was spotted in the descriptor experiments. At first, the experiments showed that any descriptor paired with the Hessian-affine detector performs well, and that the recent implementation of Hessian-Affine+SIFT by Mikolajczyk *et al.* [79] is especially good. It was assumed that this verifies the finding by Nowak *et al.* [88] that the VOC performance gradually improves with more and more features – "the more the better". The matching ratio, however, goes down with this approach and finally most of the detected regions are just false alarms.

To avoid this problem, an alternative approach was presented to that of Nowak's: instead of having more features, we allowed multiple matches – "less is more". By accounting only $K = 5$ best matches, the matching approach turned out to be equally successful with the results of higher amount of features. Furthermore, by introducing a new performance measure, *coverage-N*, it was shown that all methods performed equally well for $K = 5$ best matches, thus giving preference to the methods generally producing less features.

The results justify two alternative directions for local feature based VOC methods. The first one is justified by Nowak *et al.* [88]: VOC performance increases by adding more and more patches and larger codebooks. This is the mainstream as dense sampling has replaced the interest point detectors in the top performing methods in the annual VOC challenge [30] and also "dense interest points" [110] have been proposed. The improvements, however, have not been that significant and with many features and large codebooks the methods start to overfit. The second direction, which is justified by the results in this work, is to have less detected regions, but assign each local feature to multiple best matches or codes. Several attempts of this approach, such as soft-assignment [1, 114], have been proposed. The soft-assignment, however, does not solve the problem properly as it does not consider multiple alternative assignments, but just embeds all into one, which magnifies noise.

Chapter IV

Local Feature Based Alignment of Object Class Images

As already mentioned in the previous chapters, one of the prominent uses of the local features is object recognition and detection. Object detection has strong emphasis on learning object classes from a set of images. The most influential approach has been visual bag-of-words (BoW) [103, 24] where image content is encoded into a histogram of local feature codes. The methods using spatial arrangement of the features in addition to BoW methods have been even more successful [34, 58]. A predominant problem in learning such part-based object models is that object poses vary. In 2D, this means changes in rotation, translation and scaling, and in 3D also rotations in depth.

The original BoW approach tolerates 2D variation and moderate 3D variation since it is a global descriptor, but lately it has been shown that class specific spatial configurations of the local features are a very important cue for detection [3, 15]. Efficient learning of the computational models for object detection would benefit from automatic alignment of training images. Virtually all proposed methods rely on manually annotated metadata, such as bounding boxes [48, 18] or object specific landmarks [76, 57].

Bounding boxes have become the de-facto ground truth format in the popular image datasets, such as Caltech-101 and ImageNet. Usually, the bounding boxes are sufficient for image alignment, such as bounding boxes in Caltech-101, but in more recent datasets the bounding boxes do not guarantee good alignment. From the Fig. 4.1 it is evident that when the objects are strongly rotated, alignment with bounding boxes fails.

It is clear that the learning of visual object classes would benefit from a method for unsupervised alignment of object-class examples: The learning method could use the aligned images to learn features from the same position in each image. In this chapter, the image alignment problem is introduced and the published solution presented [59]. The main contributions in this alignment work are:

- Proposing a novel local feature based congealing method for the problem of unsupervised image alignment. The method is experimentally shown to be suitable for unsupervised alignment of images containing object category examples.

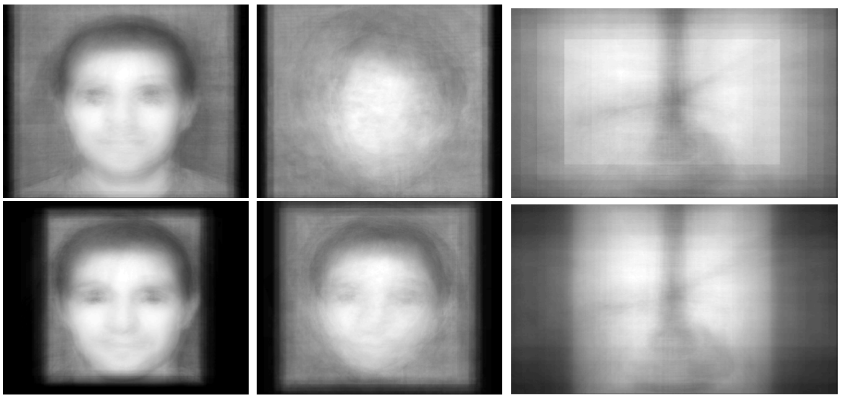


Figure 4.1: Average images of class examples without alignment (top) and aligned using bounding boxes (bottom). From left to right: Caltech-101 [32], r-Caltech-101 [53] and ImageNet [28]. Caltech-101 classes are recognisable from the average of the original images, for r-Caltech-101 bounding boxes are needed, but the ImageNet class still remains unknown (the class is revealed in Fig. 4.10).

- Comparing different local feature detectors and descriptors for local feature based congealing and selecting the best performing detector-descriptor pair.
- Seaching good default values for the parameters of our method to make it parameter-free and thus facilitate its usage in applications.
- Reporting qualitative and quantitative results for realistic and difficult images in the r-Caltech-101 data set.

Both qualitative (average images with and without aligning) and quantitative (alignment errors of manually annotated landmarks) performances are reported in Section 4.5. The experiments were conducted using the widely used Caltech-101 dataset, its randomised version r-Caltech-101 by Kinnunen *et al.* [53].

4.1 Previous work

The image alignment problem has been recognised in many works [7, 8, 57, 18, 107, 48], but none of them explicitly defines an alignment method nor reports quantitative performance of alignment or effect to overall performance. The alignment problem is illustrated in Fig. 4.2. Top level images are unaligned images with arbitrary position and rotation. The bottom level images are aligned. The images can be "stacked" to form an average image on the right. The unsupervised alignment has also been recognised as its own problem referred to as "spatial image congealing" [64].

A number of congealing methods have been proposed by various authors [37, 64, 120, 46, 22, 23]. These methods are mainly seminal work to Learned-Miller [84, 64] extending and improving the original algorithm. The main drawback of the congealing approach

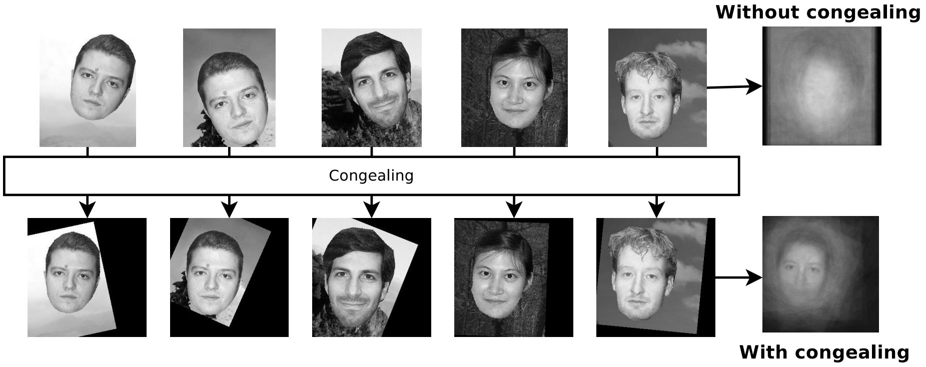


Figure 4.2: An illustration of automatic image alignment

is that its iterative optimisation on pixel-level requires at least moderately good initial alignment to converge.

There is another branch by Frey and Jojic [36, 37], but their approach is also similar to the works of Learned-Miller. The algorithms are iterative optimisation or search methods directly adjusting transformation parameters to maximise congealing quality measured by pixel-based errors. Huang *et al.* [46] extended the standard binary algorithm to gray-level images by replacing binary pixels with soft assignments to "codes" generated by clustering SIFT descriptors. However, all these methods share the same main limitations:

1. *slow convergence* (limiting the size of an ensemble)
2. *limited tolerance to imaging effects* (background, illumination, occlusion, etc.)
3. *parameter drift* (error functions prefer transformations vanishing objects away)
4. *sensitivity to initialisation*

The fundamental drift problem has been solved by heuristic correction terms in the error function or by re-formulation of the target function [120]. The slow convergence of the original algorithm was addressed by Cox *et al.* [22] who re-formulated the entropy-based error function. The tolerance to illumination changes was addressed in [46] where the pixel-based measure was changed to a soft SIFT codebook. Efforts are continuously made to improve and extend the original algorithm [23], but there is space for new approaches as well.

The algorithm presented in this chapter deviates from the congealing works by the fact that the algorithm presented in this chapter utilises the well established local features instead of pixel level processing. This "feature-based congealing" is similar to the spatial scoring procedures used in specific object search [91, 20], but, in this work, the alignment algorithm is explicitly defined and measured. Additionally, the feature-based approach is known to overcome problems with occlusion and background clutter and is more robust to imaging variations. The "feature-based congealing" of n images requires an estimation of

$n-1$ transformations to a single randomly selected "seed image" which fixes the canonical pose. Instead of parameterising the transformation [64], the alignment is treated as a homography estimation problem based on point correspondences between object class specific local features.

4.2 Landmark selection

The presented alignment algorithm, "feature-based congealing", assumes that there are certain category specific local features, whose saliency triggers interest point detectors at same locations and which appear sufficiently similar to match their descriptors between images. Manual or automatic identification of these local features enables joint-alignment (registration) of category images and, therefore, they are referred to as "object landmarks". Furthermore, it can be assumed that the alignment transformations are sufficiently rigid to be represented by 2D homography.

Under these assumptions, it is possible to define an interest point based algorithm for unsupervised alignment of category images. The algorithm is based on the following principle: if there exists a transformation T which transforms a set of object landmarks P_i' from an image I_i' into spatial correspondence with the corresponding landmarks P_i'' in an image I_i'', then the transformation T is a spatial alignment operator between I_i' and I_i''. Congealing algorithms estimate the transformations T_i for all images I_i. The algorithm is based on automatically found local features. The algorithm selects a single "seed" image, identifies landmarks within the set of found features, and then, estimates the operators T_i from other images to the seed.

The initial step is to find putative matches for N local features F_s extracted from a seed image $\mathcal{I}_s$. From another image $\mathcal{I}_i$ total of M_i local features F_i are extracted. Using the computed descriptor distances, distance matrices $\boldsymbol{D}_{N \times M_i}$ between the seed and $i = 1, \ldots, I$ other images are computed. Here, not only the best matches for each seed feature are found but also all distances are computed. This is justified since local appearance variation makes even the best matches not perfect and thus a small number of the K best matches needs to be retained (the correct match is expected to be among the K closest matches).

4.3 Image alignment using feature scoring

The basic idea is to select the seed features which match the similarly detected features in other images well. Overall performance can be computed using, for example, the SIFT descriptor distances. The simplest solution is to sum the seed feature distances and select the ones with the K smallest sums.

That solution, however, is not robust to missing or occluded landmarks and a single image may have an undesirably strong effect on the sum. Therefore, we replace the sum with a ranking-based accumulation: our algorithm accumulates scores of the K best matches per image. Hypothetically, the best landmarks should finally appear as the top scoring seed features. The algorithm is given in Alg. 4.1.

The next step in the image alignment is to find the transformations between the images. This can be achieved by using the spatial scoring approach instead of simple closest

Algorithm 4.1 Landmark selection by feature scoring.

```
Select the seed image and remove it from the image set.
Extract seed interest points and form their descriptors (tot. of $N$).
Initialise score vector $\boldsymbol{s}_N$for $N$ candidates.
for all images (indexed with $i$) do
    Extract interest points and form their descriptors (tot. of $M_i$).
    Compute distance from each seed point to each image point: distance matrix $\boldsymbol{D}_{N \times M_i}$.
    Increment scores for the $K$ best matching seed features in $\boldsymbol{s}_N$.
end for
Return coordinates and descriptors of the $K$ best scoring seed interest points.
```

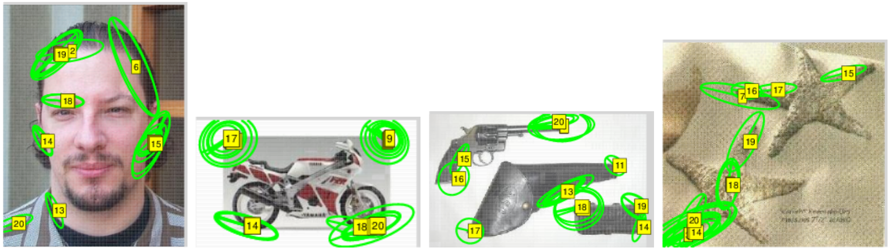


Figure 4.3: Twenty best landmarks found for different categories using Algorithm 4.1.

neighbor detection for the landmarks which are required by the alignment method. The straightforward approach would use the distance matrices $\{D_{N \times M_i}\}$ directly to select the best seed features, for example, by accumulating scores for the K closest matches over all images. However, this approach leads to a selection of general image features representing strong frequency components, such as corners and edges, and not class specific landmarks. This effect can be seen in Fig. 4.3.

4.4 Image alignment using spatial scoring

The main problem in the feature match approach is that it does not use spatial configuration of local features. That contradicts the proposed correspondence principle. The spatial dimension can be introduced by scoring points which match under the selected homography: *isometry*, *similarity*, *affinity* or *projectivity*. Isometric transformation preservers the distance between the transformed points. Similarity is a transformation that preserves the distance and also adds a scaling factor. Affinity adds skewing and projectivity allows projective transformations. Between point correspondences a homography can be estimated using the standard linear methods: Umeyama [116] for isometry and similarity and a restricted version of the DLT for affinity (the standard DLT accounts for projectivity [45]).

The algorithm used in this work is as follows: Over R iterations the minimum number of correspondences are randomly selected, the minimum being two for similarity and three for projectivity. In every iteration, the algorithm estimates a homography transformation and transforms all image points to the seed image and accumulates scores of putative features within a pre-set distance threshold τ. The procedure is similar to the Random Sample Consensus (RANSAC) [35] robust estimation, except that in this case, the algorithm does not seek for a single solution, but accumulates scores over a number of random iterations. The pseudo code of this approach is given in Alg. 4.2. The main computational factor is the number of random iterations R. The other parameters are the match threshold τ, the number of best matches K, and the total number of best features selected L_b (class landmarks).

Algorithm 4.2 Class specific landmark selection by spatial scoring.

```
Pick a seed image and remove it from the image ensemble.
Extract seed interest points and form their descriptors (tot. of $N$).
Initialise score vector $\boldsymbol{s}_{i \times N}$ of $N$ seed feature candidates.
for all images (indexed with $i$) do
    Extract $M_i$ interest points and form their descriptors.
    Compute the distance matrix $\boldsymbol{D}_{N \times M_i}$.
    Initialise image-wise score vector $\boldsymbol{v} \leftarrow \mathbf{0}$.
    for $R$ random iterations do
        Select two random seed features.
        Select random correspondences within the $K$ best matches in $\boldsymbol{D}_{N \times M_i}$.
        Estimate 2D homography from the image $i$ to the seed space (Umeyama [116]).
        Transform all image features to the seed space.
        for all seed features $j$ (excluding the selected two) do
            if the seed feature $j$ has matches closer than $\tau$ within the $K$ best in $\boldsymbol{D}_{N \times M_i}$ then
                Increment the seed feature score: $\boldsymbol{v}(j) \leftarrow \boldsymbol{v}(j) + 1$.
            end if
        end for
    end for
    Sort $\boldsymbol{v}$ and increment the $L_b$ highest seed scores in $\boldsymbol{s}_N$. // All images have equal contribution.
end for
Return coordinates and descriptors of the $L_b$ best scoring seed interest points.
```

The spatial scoring algorithm outputs the best L_b landmarks based on the top scores. The top scoring seed features represent parts which have been independently verified by other features in a similar configuration in other images. The found seed features using the spatial scoring algorithm are illustrated in Fig. 4.4. The selected features represent landmarks inside the object area and thus encode object class specific local parts and their configuration.

With the best scoring landmarks $\boldsymbol{s}_1, \ldots, \boldsymbol{s}_{L_b}$ the alignment procedure itself is straightforward. For a number of random iterations, the minimum number of seed landmarks are selected, then homography is estimated to randomly selected correspondence points within the best matches, and finally, the transformation that produces the highest number of inliers is selected and the transformation re-estimated using all inliers. This stage is very similar to stereo and baseline matching, except that instead of a single best match, a few best matches must be used for robustness. The alignment is demonstrated in Fig. 4.5.

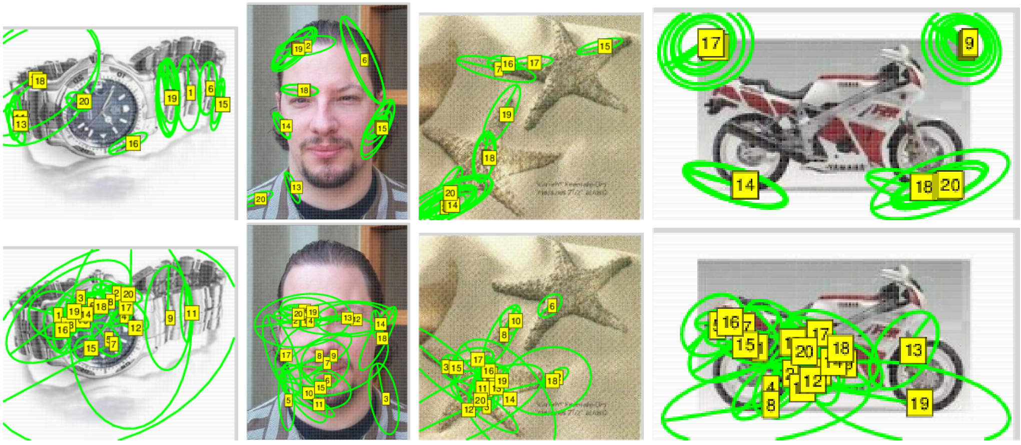


Figure 4.4: The results with the different landmark selection approaches. The top row: watch, Faces, starfish and motorbikes with the feature scoring (Alg. 4.1). The bottom row: landmarks selected utilizing the spatial scoring (Alg. 4.2).

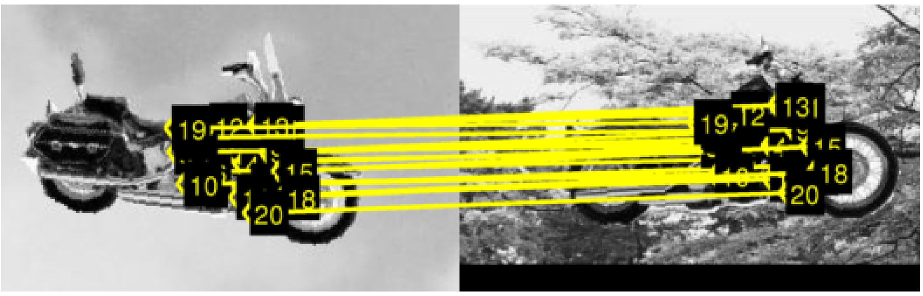


Figure 4.5: Image alignment by finding the best sub-set of landmark matches.

4.5 Performance evaluation

The experiments were conducted using classes from Caltech-101 [32], r-Caltech-101 [53] and ImageNet [28] datasets. The used performance measures were average images (qualitative) and a quantitative measure from Cox *et al.* [22]. The results are also compared to a state-of-the-art congealing method for natural images by Huang et al. [46]. The various popular interest point detectors and descriptors presented in Chapter 2 were tested. Additionally, the dense sampling was included in tests which does SIFT feature detection on a regular grid. The dense sampling has gained momentum in the recent works of object class detection [31].

The object detection methods which use metadata or implicit alignment [7, 8, 57, 18, 107, 48] only report detection performance and no analysis on the alignment or its effect are given. The congealing works [37, 64, 120, 46, 22, 23] report results for simple or artificial images, such as binary images of digits, for which averages of aligned images are reported as qualitative results. As an exception, Cox *et al.* [22] used face images in the MultiPIE dataset that contains manually annotated landmarks. They measured errors between true and congealed landmarks and plotted a cumulative RMS point error graph as the quantitative performance. We adopt their performance measure, but normalise pixel errors by the image diagonal to make it resolution independent.

Here, the main focus is on the natural visual classes and, thus, we selected the Caltech-101 [32] dataset which has been a popular benchmark for visual object class detection. Caltech-101 contains object classes with natural visual appearance variation and with varying background. For object detection, Caltech-101 is considered outdated since the objects are mainly in the image centre, pose variation is very limited, and backgrounds provide unrealistically strong cues for detection [92]. However, for unsupervised alignment, the dataset still provides a challenging test bench. To make the problem even more challenging, experiments are made with the recent “randomised” Caltech-101 (r-Caltech-101) [53], where the backgrounds have been replaced with random Google landscape images and the objects are transformed to random poses (scale, translation, rotation).

For all classes for which the quantitative results are reported, 5-12 landmarks (see Fig. 3.1) were manually annotated.

Alignment results using classes from Caltech-101

In Fig. 4.6, the average images of selected classes are shown with and without alignment. Quantitative results for the same classes are shown in Fig. 4.8. The y-axis in the cumulative RMS error graphs illustrates how many of 50 random images were aligned to the given precision in x-axis. The normalised distance corresponds roughly to the d_{eye} distance used in the face detection where 0.05 is considered excellent localisation, 0.10 as good and ≥ 0.25 as failure [42]. With the regular Caltech-101 dataset, the images are already well aligned, and thus, the proposed method does not show as significant improvement as with other datasets. The differences between various datasets can be seen in Fig. 4.3. The average images show that r-Caltech-101 is a much more challenging dataset than the original Caltech-101.

For r-Caltech-101, the results are shown in Fig. 4.7 (The average images of selected classes and without alignment). Quantitative results for the same classes are shown in Fig. 4.8. The method proposed by Huang *et al.* [46] completely failed with r-Caltech-101 since it requires good initial alignment to converge properly. Note that significant improvements for our method are reported in the next sections where the parameters are optimized.

4.6 Selecting method Parameters

It is clear that the feature-based alignment uses a few parameters which can dramatically affect the performance of the alignment. From the computational point of view, it is justified to seek good default values for the parameters. During the experiments in Section 4.5, the initial set of parameter values were found and the values were selected at value ranges around them. The selections are shown in Table 4.1. The method was run using all possible parameter combinations. The results for 6 classes are shown in Fig. 4.9: *Airplanes*, *Dollar_bill*, *Faces*, *Motorbikes*, *car_side* and *stop_sign*. To tune the method

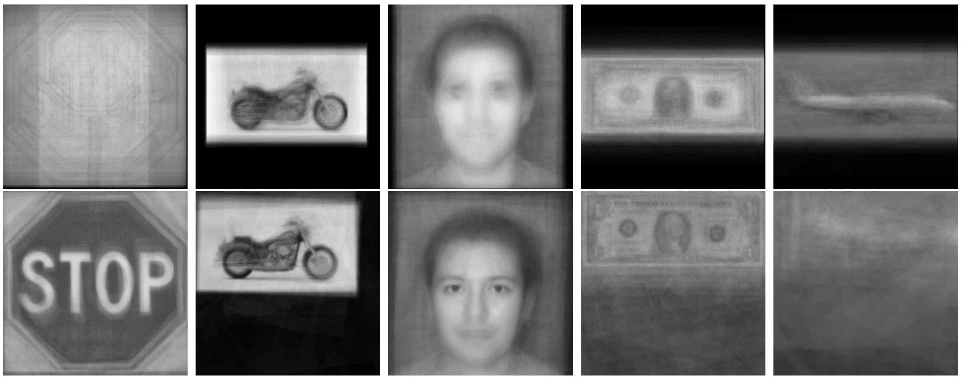


Figure 4.6: Top: avg. images of Caltech-101 classes. Bottom: our method. The class difficulty increases from left to right (stop signs: 84% aligned to 0.05 normalised Euclidean distance accuracy, airplanes: 10%; for 0.10 the numbers were 100% and 36%, respectively).

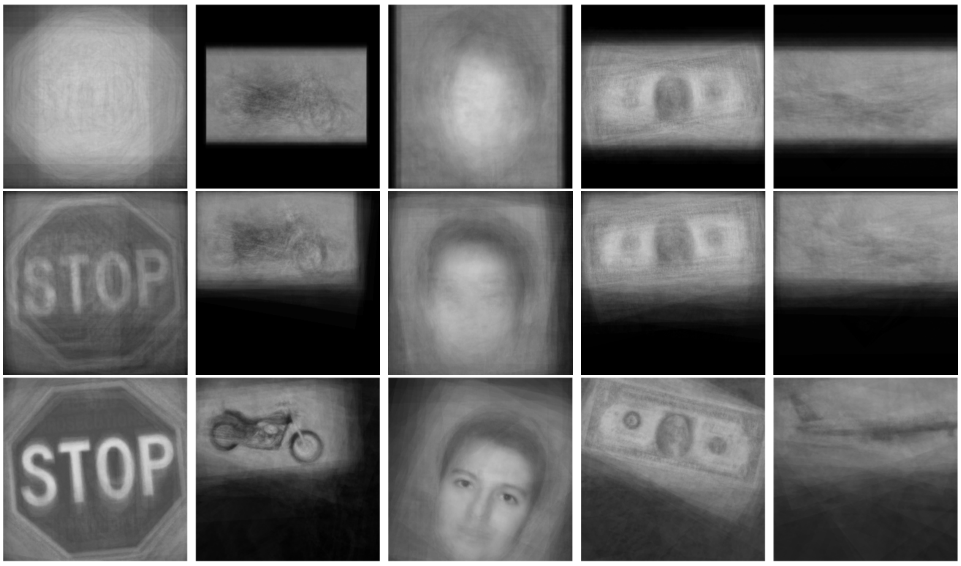
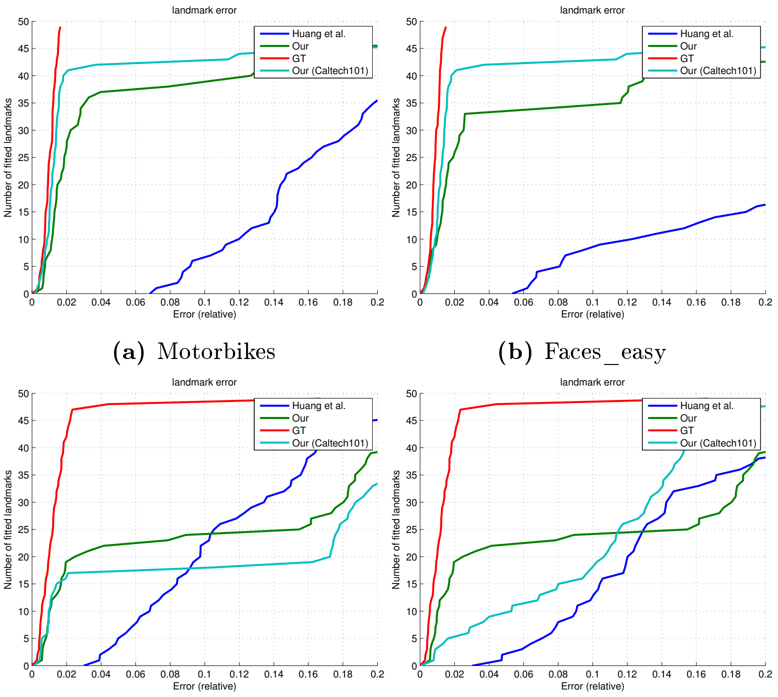


Figure 4.7: Top: avg. images of r-Caltech-101 classes. Middle: Huang et al. [46]. Bottom: our method. Class difficulty increases from left to right (stop signs: 94% aligned to 0.05 normalised Euclidean distance accuracy, airplanes: 26%; for 0.10 the numbers were 100% and 54%, respectively).

parameters, 50 randomly selected images from ten different classes in Caltech-101 image set were used. The selected images were then used as seeds in turn and the alignment method is applied to the selected images. This procedure was repeated with all chosen parameter combinations.



(a) Motorbikes **(b)** Faces_easy

(c) Dollar_bill **(d)** Airplanes

Figure 4.8: Quantitative results for the alignments in Fig. 4.7 (stop signs omitted). Graphs represent cumulative error curves for ideal (red), Huang et al. [46] (blue), feature-based alignment (green) and the feature-based alignment for the original Caltech-101 (cyan).

It is important that the number of regions selected from a seed image and other images is not exhaustive and there are many correct matches within these regions. That depends solely on the used local feature detector and descriptor. The widely used detectors and descriptors were introduced in Chapter 2 and an evaluation of the local features was provided in Chapter 3. As shown in local feature evaluation, the frequently cited evaluations of detectors [83] and descriptors [82] by Mikolajczyk *et al.* evaluate only the case of wide baseline matching, where scenes remain the same while in this case, the "scene" (object category) varies and thus the validity of their results is unclear. To find the best detector for this alignment method, the following popular and well performing detectors were tested: Hessian-Affine (*fs_hessaff*) [82] (Mikolajczyk's recent implementation at FeatureSpace), maximally stable extremal regions (*fs_mser*) [74] (FeatureSpace implementation), SIFT detector (*vl_sift*) [68] (VLFeat implementation) and two versions of dense sampling (*vl_dense* and *vl_densems*). The dense sampling was included because it is very popular within the recent top performing visual object categorisation methods [31]. The two versions of dense sampling are: single scale of radii 10 patches spaced by 10 pixels (*vl_dense*) and multiple scales of radii 12, 24 and 36 spaced 12, 24 and 36, respectively (*vl_densems*).

The results for the selected detectors and the SIFT descriptor are given in Fig. 4.9. It is clear that there are two detectors that perform moderately well for all classes: Hessian-Affine and dense sampling. The five best parameter combinations for the Hessian-affine and multi-scale dense detectors and the best common setting for both Caltech-101 and r-Caltech-101 datasets are shown in Table 4.2. In general, dense sampling outperforms Hessian-affine, but needs more landmarks and less strict thresholds. This can be explained by the fact that with a fixed grid, the object landmark locations are not accurate and affect both to the matching distance (τ) and to the required number of landmarks (L). The best common parameter settings for the two datasets are ($K = 2$, $\tau = 0.02$, $L = 20$) for the Hessian-affine detector and ($K = 5$, $\tau = 0.04$, $L = 80$) for multi-scale dense sampling.

Table 4.1: Parameter selection experiment: tested values for the method parameters. Used for both spatial scoring and candidate selection.

Number of best matches[1] K:	1	2	5	10
Spatial match threshold τ:	0.01	0.02	0.04	0.08
Number of landmarks L:	20	40	80	160

Table 4.2: Five best parameter settings for the two best detectors and the both datasets (including the best common setting). The last line refers to the bounding box based pre-processing.

	Caltech-101				r-Caltech-101			
	K	τ	L	avg# (0.05,0.10,0.25)	K	τ	L	avg#
fs_ hessaff	2	0.01	40	(34,36,48)	1	0.02	80	(31,33,47)
	2	**0.02**	**20**	**(33,36,48)**	**2**	**0.02**	**20**	**(31,33,45)**
	2	0.02	40	(33,36,48)	2	0.02	40	(30,33,47)
	5	0.02	40	(32,35,48)	2	0.02	80	(30,32,46)
	2	0.01	80	(32,35,47)	1	0.02	160	(29,31,47)
fs_ densems	10	0.04	160	(41,46,49)	**5**	**0.04**	**80**	**(36,41,48)**
	2	0.04	160	(41,45,49)	2	0.04	80	(34,40,47)
	1	0.01	160	(41,43,48)	10	0.04	80	(34,41,49)
	5	0.04	160	(40,45,49)	10	0.04	40	(34,40,49)
	5	0.08	40	(40,45,49)	5	0.02	160	(34,40,47)
(12th)	**5**	**0.04**	**80**	**(39,44,49)**	-	-	-	(36,41,48)
densems+BB	5	0.04	80	(44,47,49)	n/a	n/a	n/a	n/a

Automatic seed selection (ImageNet)

The automatic seed selection is the main drawback of this method. To obtain the best possible alignment the seed image which best represents the class needs to be found. The basic solution is to try all images in turn and select the one which provides the best average image (qualitative evaluation). This is a supervised approach. However, by a

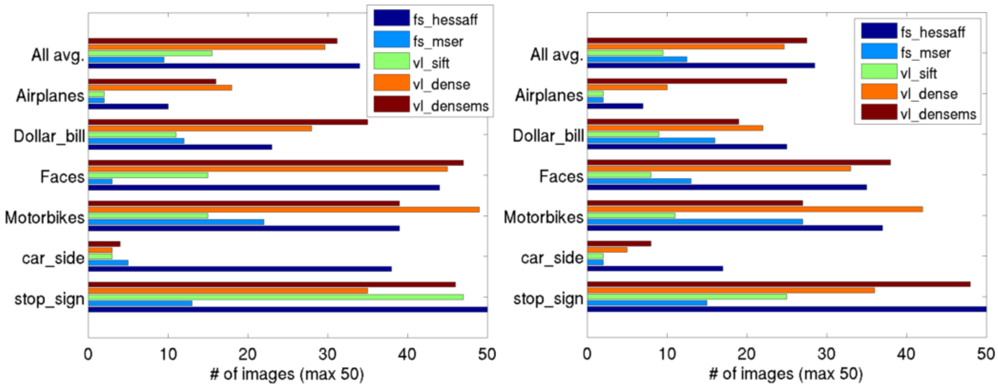


Figure 4.9: Alignment performance bars (numbers of correctly aligned images within normalised landmark distances ≤ 0.05) for Caltech-101 (left) and randomised Caltech-101 (right).

proper seed selection procedure, the method would be completely unsupervised requiring only a set of images to be aligned. A partial solution would be utilising bounding boxes. The object detection rule was used in the Pascal VOC competition (see, e.g., [31]), which computes the detected region (bounding box) overlap and the image with the best average overlap would be chosen as the seed image.

Using this technique, it was possible to reveal the true class of the ImageNet example in Fig. 4.1 (see Fig. 4.10) in a completely unsupervised manner (multi-scale dense grid and the default parameters for the approach).

4.7 Summary

In this chapter, the problem of aligning images in arbitrary poses was addressed. The previous work on the subject was based on the pixel level optimization approach which did not perform particularly well with images of arbitrary poses in r-Caltech-101. The problem was solved by introducing a local feature based method for image alignment. The presented method was shown to be relatively simple, utilizing the well established methods widely used in wide-baseline stereo approaches, such as RANSAC and local features. The alignment was done using the “seed image” which is used to define the space in which all other images are aligned. The landmarks used for the alignment task were selected by applying a voting method with RANSAC. For each feature, a spatial score is calculated and the features which gained the highest score were used for alignment.

The experiments showed that the multiscale dense SIFT features provided the best average performance. However, with r-Caltech-101, the performance dropped as class examples were in arbitrary orientations, since a dense grid and fixed orientation do not provide invariance to rotation changes. Furthermore, the method uses a single global seed based on which all other images are aligned. However, it may happen that no single seed works well (especially in the case of sub-classes within examples) and seed selection

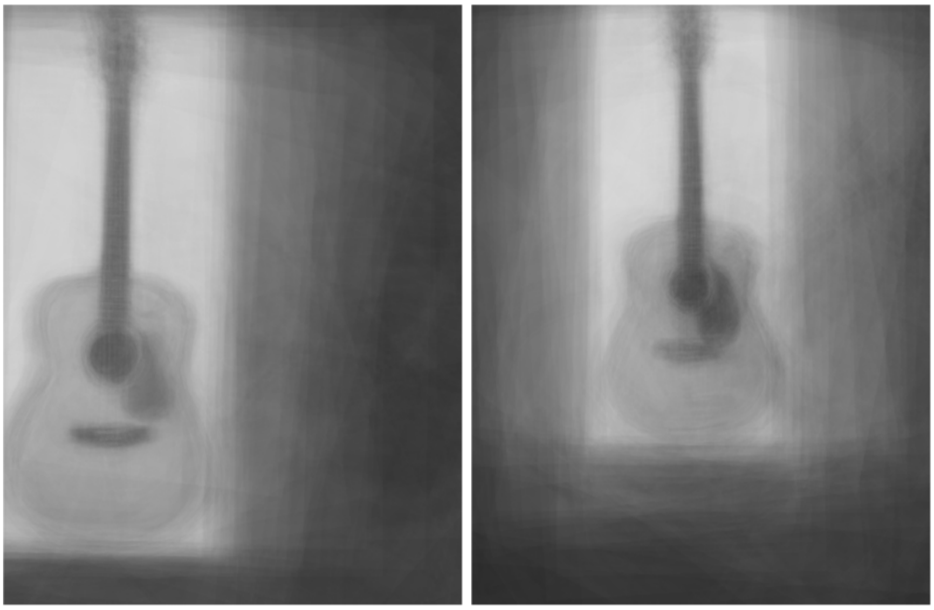


Figure 4.10: Automatically aligned ImageNet examples (see Fig. 4.1) with (left) and without bounding boxes (right) (for both the optimal seed was selected using the overlap criterion, the average bounding box overlap is 51%).

may be problematic or at least ad-hoc. This can be improved by using a local approach instead of a global seed, for example, an "alignment tree" built from pair-wise matches, so that all images are aligned using a single "root" via pair-wise alignment paths. A similar approach was successfully used in [85] for the detection of common landmarks. In conclusion, this approach clearly outperformed a state-of-the-art congealing method. The visual object class learning and detection methods can benefit from a method which automatically aligns class examples making the learning procedure more straightforward.

Chapter V

Shot Detection and Video Summarization

In this chapter, the local features are introduced in the domain of video summarization, which is one of the important topics of this thesis. In addition to the video summarization, which will be explained shortly, the area of shot boundary detection is discussed. The shot boundary detection is considered a preprosessing step for the video summarization and is also closely examined in this chapter. In the following chapter a novel local feature based shot boundary detection is presented. In addition, a video summarization application is introduced, using the presented local feature based shot boundary detection method.

In Fig. 5.1, the overall hierarchy of frames, shots and scenes is illustrated. Video is, by definition, a set of consecutive images (*frames*) which, when displayed rapidly, form illusion of motion in human brains. Videos add a temporal dimension which can be utilised. Frames can be combined together to form a logical set of consecutive frames without any visually abrupt moments called a *shot*. A set of shots is usually taken in the same environment, thus forming a *scene*, in which the content is filmed. The whole video, such as a feature film, is built from the different scenes, which together form a semantically coherent story.

In many applications, such as in video summarization [109] and content based retrieval [103], video shot boundary detection is the first step before higher level processing. For analysis, the shots are usually considered the basic units and thus the success of the boundary detection affects the whole processing pipeline. Shot detection has been studied within specific applications as its own problem and a wide variety of proposed methods exist.

5.1 Shot boundary detection

Shot boundary detection, as the name suggests, is a research field attempting to find the boundaries between two different shots. These shots are usually generated by pausing the camera or made afterwards by editing the video to a coherent show or movie. A half an hour of video, in a television show for example, may contain hundreds of shots, depending how the show was recorded and edited.

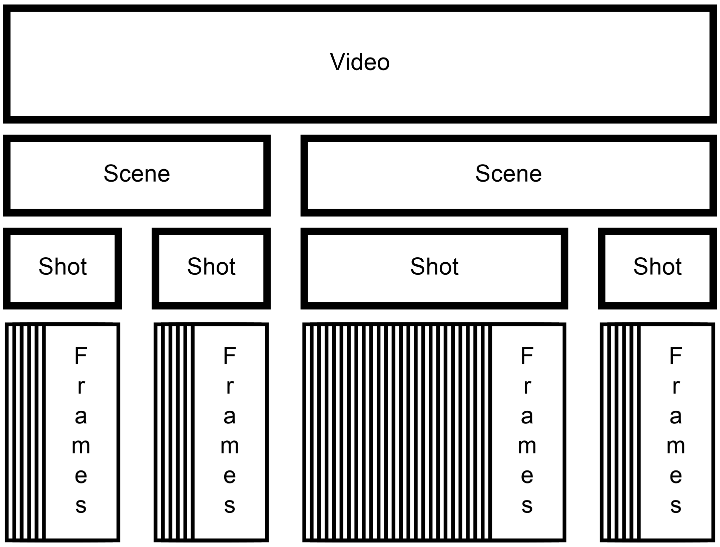


Figure 5.1: The hierarchy of video segments.

Usually, the shot boundary methods compare consecutive frames by using a distance metric. The baseline method for comparing the frames is considered to be histograms. The histogram-based methods utilise some distance-metric between the histograms of two consecutive frames which measures how content has changed. For example, high difference peaks on the time-line may denote hard cuts and sequences of smaller consecutive changes may denote fade-outs. The colour histogram based shot boundary detectors are fast and accurate when accompanied by heuristics for all transitions types [106, 49, 73].

In 1996, Boreczky *et al.* [9] compared different shot boundary detection techniques, such as full image pixel difference, color histograms, edge tracking and motion vectors from MPEG compressed video sequences. The pixel difference approaches compared pixel values directly between the consecutive frames by counting the number of pixels that change in value more than some threshold. Another threshold was used to determine if the shot boundary was found. The direct pixel differences are usually accompanied by a filtering step to reduce the error rate. The histogram based approaches use exactly same approach, but instead of direct pixel differences the histogram of choice is used as data and histogram difference distance funtion is applied to determine the difference between the frames. In edge tracking the percentage of edges that enter and exit between the consecutive frames was computed. This was found to be effective with transitions. Due to the computational restrictions the features in MPEG compression have been successfully used. The block motion vectors can be directly extracted from the MPEG data. The region-based pixel difference calculation is another way to obtain the motion vectors. Boreczky *et al.* [9] showed that, in general, simpler methods outperformed the more complicated algorithms. In tests, the more complex methods were sensitive to the threshold and parameter settings.

Furthermore, some of the techniques are evaluated by Gargi *et al.* [40] in 2000. They showed that the threshold selection for histogram-based methods (such as [49, 73]) is critical and the performance changes significantly depending on the selected color space. The Munsell (MTM) space, RGB, HSV, XYZ, LAB, LUV and OPP were evaluated.

The most block-motion matching algorithms were found to be slow and did not perform well in the evaluation. However, MPEG-based methods showed to be fast but did not perform as well as the color histogram based methods. It is noteworthy that the MPEG-based methods are reliant on the encoder output, thus causing variation in the performance (based on the bitrate they were encoded with). The histogram intersection method in the Munsell color space (MTM) was the best performer [40]. A more recent study by Smeaton *et al.* [104] shows that, based on the results of the latest TRECVid activity, simple methods such as color histograms are accompanied by modern higher level machine learning algorithms such as GMM (Gaussian Mixture Models) [50] or HMM (Hidden Markov Models) [94]. There are also methods to detect pauses in speech to specify shot boundaries, dividing video sequences into event and non-event segments and evaluating dominant image motion.

Local features have been used in video shot boundary detection. Li *et al.* [66] presented a local feature based (SIFT) shot boundary detection algorithm which utilized SVM to decide the frame similarities. Another approach was presented by Luo *et al.* [71], which used the SIFT Flow algorithm coupled with clustering and Kernel PCA to produce video summarizations. An accumulation algorithm similar to the shot boundary algorithm what will be presented in next chapter, is used and presented by many authors [70, 124, 106] for MPEG videos. They, however, had a lot of additional processing to reduce many artifacts in the old filming techniques, such as gamma correction and motion compensation due to the nature of the accumulation algorithm.

5.1.1 Scene detection

The scene detection algorithms are approaches which usually utilize shot detection as a pre-processing step. As such, they often contain ideas which can be applied in both shot detection and scene detection. Here, some previous work on the subject is discussed and important findings and ideas are presented.

One way to do scene detection is to apply graph-based presentation [100] or use the sub-image correlation approaches [133]. These methods rely heavily on clustering. Another approach is to use audio cues to detect scene changes. Velivelli *et al.* [121] combined audio cue score with the visual score to improve the performance of the scene detections in documentaries. Similarly, Chen *et al.* [17] used the sampled audio data to identify “silent frames” and overall changes in the audio features such as *volume*, *power* and *spectrum*. It has been shown that the audio based approaches considerably improve the scene/shot detection performance. However, combining the visual and audio features has been shown to be tricky, e.g. Velivelli *et al.* minimized a cost function to define the correct weights for both types of features (0.7 for audio, 0.3 for visual). Additionally, Chen *et al.* considered the audio features to be complementary and added the values equally, still gaining over 0.9 in both precision and recall.

5.1.2 TREC Video Retrieval Evaluation (TRECVid)

The shot boundary task was included in TRECVid (TREC Video Retrieval Evaluation)[1] as an intruductory problem from which the results are commonly used for more high-level tasks such as video summarization. TRECVid itself is a large-scale, worldwide benchmarking activity, concentrating on research related to content-based retrieval on digital video. The competition is running annually just like the Pascal VOC [29] for the visual object categorization and was started in 2003. TRECVid contains very thorough evaluation procedure for evaluating the performance of various shot boundary detection tasks. TRECVid provides the participants a large-scale video collection for testing and means to evaluate the results. Furthermore, a forum is available for the organizations for comparing results and for submitting the results of the experiments back to the coordinator, the National Institute of Standards and Technology (NIST). NIST collects all the submitted result for automatic evaluation and eventually the overall results are made publicly available on the TRECVid webpage. Later, the detailed performance figures are sent to the participants for further analysis.

The data provided for this competition (by NIST), has increased over years to increase the difficulty of the common tasks, i.e. shot detection. Smeaton *et al.* stated that the origins and the genre of the data has changed a lot since 2001, as editing styles and thus the shot sizes and transition types change over time. The shot boundary test data set has been always 6 hours of video material, e.g. in TRECVid 2007 the length of the data set was 6 hours with 17 different videos. Furthermore, the data used in TRECVid 2007 stands out from the previous data sets for its longer shot durations and lack of gradual transitions. A few frames from three different videos are presented in Fig. 5.2. The ground-truth for the shot boundary data was done by an anonymous researcher at NIST by manually annotating the shot boundaries with a freely available tool, VirtualDub. The annotations contained the shot boundaries as frame number as well as the transition type such as hard cut or dissolve. The same annotator was used over the years (2001-2007) to make sure the annotations had the least amount of inconsistency possible. As such, there is no annotator-oriented analysis such as annotator variation.

The TRECVid shot boundary task is to find each shot boundaries in the video data collection and identify it correctly as abrupt (hard cut) or gradual transition. The different gradual transitions between the successive shots include *dissolve*, *fade-in/out*, and *wipe*. The test data used in TRECVid 2007 [104] contained six hours of video material, most of which was from different documentaries, news and educational material. The submissions were compared to the reference data using a slightly modified test protocol by Ruiloba *et al.* [99]. The NIST evaluation software calculates: *inserted transition count*, *deleted transition count*, *correction rate*, *deletion rate*, *insertion rate*, *error rate*, *quality index*, *correction probability*, *recall*, and *precision*. Of these the **recall** and **precision** are the primary measures for TRECVid. In addition, F-measure was also introduced, which combines the recall and precision measures:

$$\mathbf{F} = \frac{2 \times Recall \times Precision}{(Recall + Precision)} \tag{5.1}$$

[1] `http://trecvid.nist.gov/`

Figure 5.2: Examples of the video data provided by NIST for TRECVid. Top: BG_2408, Middle: BG_11362, Bottom: BG_37822.

Furthermore, the detection criteria required only a single overlapping frame between the estimation and ground-truth transition, as it was important to make the detection independent of the accuracy of the detected boundaries. This means that hard cuts are supposed to provide not only the last frame before the cut, but also the first one after the cut. To accomodate the differences in frame numbering by different decoders, the cuts were also expanded by five frames before matching. Additionally, the shot boundaries with gradual transition shorter than five frames were considered as hard cuts in evaluation.

5.2 Video summarization pipeline

Video summarization/skimming is a research area which has been actively investigated for over a decade. In the most simple case the video skims can be fast-forwarded versions or uniformly subsampled versions of the video. These approaches are not usually good for having a coherent understanding of the original video. Thus, the majority of video skims retain the same frame rate as the original and only use selected parts of the video in the final video summary. There are many applications for video skims/summaries such as news, sports, surveillance, overall visualization and so on.

Truong *et al.* [109] detected the most important attributes which affect on how the skim is generated: *skim length*, *perspective*, *mechanism*, the used *features*, *data domain* and the overall *generation process*. In this chapter, the main emphasis is on the skim generation process and on the mechanisms used to generate the video summaries. **The skim length** has two possible alternatives: the length of the skim is known as a priori or is unknown and given as a posteriori. When the length of a skim is known, the underlying shortening mechanism is modified so that the result will retain as much information as it can due to the time limitation. If the skim length is unknown, the shortening process decides the length by maximizing information while minimizing the time presenting it. **The data domain** represents the "topic" of the video data. The majority of the techniques used in skim generation are domain dependent. Thus, different videos, such as sports, documentaries and home videos are processed with different approaches or by using different parameters. **The selected features** are information extracted from the input data. Features can be visual, such as local features or histograms presented in the previous chapters. Features such as edges and textures with their spatial configurations are important for detecting the desired situations in video data such as closeups. The audio track and the extracted audio cues such as tempo are also usable features. Some methods utilize text extracted from the subtitles or from the audio track to evaluate the redundancy [43]. The general camera motion, if known, can be used to improve the quality of the generated skim [26]. Furthermore, even 3-D depth maps have been used to extract additional information from the scenes. In turn, the skim generation algorithm itself is affected by **the persperctive**, which defines the general goal of the video skim:

- Information coverage / maximization: The video skim should contain as much information as possible.
- Important events / highlights: The skim should seek for the interesting moments based on the engineered "built-in" detector.
- Query context: The skim is generated by the user preferences e.g. preferred features such as loud noise

After identifying the target domain of the skim generation, a suitable method is selected. The state-of-the-art methods combine multiple features to achieve the best performance and avoid certain video anomalies such as flash detection. On a conceptual level a method has **a mechanism** which it uses during the skim generation process. The mechanisms are tightly coupled with the selected perspective, such as highlight detection and redundancy elimination.

Truong *et al.* [109] presented four different steps for skim **generation process** utilizing excerpts:

- *Excerpt segmentation* (the video is divided into processable units)
- *Excerpt selection* (the units are selected)
- *Excerpt shortening* (the units are used to shorten the video material)
- *Skim assembly* (assemble the video after the shortening step)

It is possible to generate skims without excerpts and the whole process is done in the skim assembly. In the following sections a short description of each step is given in addition to references to the different approaches to solve the problem.

5.2.1 Excerpt segmentation

In segmentation, the whole video sequence is divided into separate units, some of which will be included in the skim. These units are usually shots taken from the original video, but because this is not always true, a more general term *excerpt* is used. The segmentation step is required, but is not always considered as part of the actual skim generation. Segmentation is done in various different ways either utilizing the audio, video or other metadata such as captions or timestamps. For example, Rui *et al.* [99] consider segmentation as dividing the input video into event and nonevent segments.

The shot boundaries are a commonly used type of excerpt. Most of the detection algorithms are developed to be used in video retrieval methods and they are not directly meant for skim generation. However, they provide useful excerpts to be selected in skim generation. [109]

5.2.2 Excerpt selection

In the selection step, we need to select the excerpts which are included in the skim. This step has the highest influence to the coverage of the skim and can be applied in many ways. This step is sometimes included into the shortening step [109]. Some methods utilize shot clustering or selecting only the most relevant events for the skim (similar to the scene detection). Usually the skim length and other priori information needs to be taken into account when selecting excerpts for the skim. The graph-based algorithms (e.g. [19]) usually operate in this step. This is because the graph-based algorithms only select those excerpts from the graph which fulfil the requirements of the skim (such as the total length).

5.2.3 Excerpt shortening

In shortening, one of the methods to shorten the excerpts is applied. Shortening usually tries to maximize information coverage while preserving certain viewpoint. Shortened excerpts need to be done so that no inappropriate cut points are generated such as cutting in the middle of speech. The simplest way to achieve this is to pick predefined portion of each excerpt to generate a skim. Some more sophisticated methods utilize similarity of video, selecting only those parts from excerpts that are similar (and may ignore temporal order of the shots). Other methods include attention models, usage of metadata or other higher level features (e.g. faces). The sound information can also be utilized as one can detect e.g. music or speech and some methods remove silent shots as they most likely do not contain any information. This step can also be ignored if the generated excerpts are short enough or the skim length is not defined a priori. [109]

5.2.4 Excerpt assembly

In Assembly, the excerpts are brought together and assembled into a skim. Here, video summary can have asynchronous and synchronous integration of video and audio information. When the video and the audio are synchronized, the video corresponds to what we hear. It's also possible to use an asynchronous method when the audio might not directly correspond to what we see (such as documentaries). The simpliest way for assebling excerpts into a skim is simply join them in their temporal order. In addition, one can apply a gradual transition to the shortened excerpts. [109]

5.3 Video summarization approaches

Truong *et al.* [109] identified three major underlying approaches for skim generation:

- Redundancy elimination
- Event or highlight detection
- Skimming curve formulation

The approaches are briefly discussed in this section of the thesis.

5.3.1 Redundancy elimination

Redundancy elimination is strongly associated with the excerpt shortening step as the goal in redundancy elimination is to retain data enough for comprehension. Cooper *et al.* [21] presented a matrix factorization method to remove redundancy by selecting only the continuous frames with the least difference to the entire video sequence. Ma *et al.* [72] performed redundancy elimination in the exactly opposite fashion by selecting the most interesting parts of video, i.e. the ones containing the most frame-by-frame difference. Clustering is also widely applied to detect redundancy by selecting the maximum of one shot from a cluster of visually similar shots [19, 41]. Some methods use audio to remove redundancy by removing parts of video which contain noise or silence. Ngo *et al.* [19] proposed a method where only the most relevant scenes are selected through graph modeling. The graph modeling of the detected shots in video content aim to represent the shots as a temporal graph. The temporal graph is an interpretation of how the scenes change and reappear in the video. The illustration of a temporal graph is shown in Fig. 5.3. Each state (A-D) are states for shots and B1, B2, B3 are subshots of a scene.

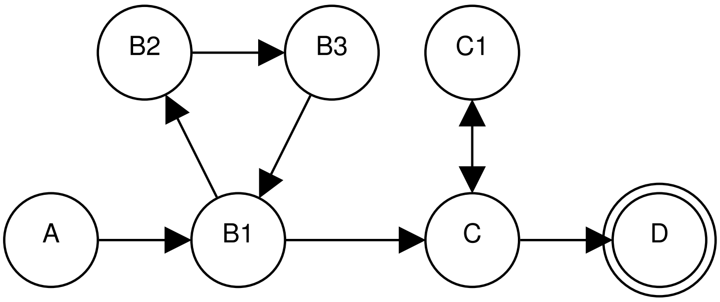


Figure 5.3: An example of a temporal graph of a video.

Many approaches can be used in redundancy elimination as the model allows selecting the parts of the video which are the most important for information coverage. Additionally, the motion attention model makes it appliable in highlight detection, which is another popular approach used in summarization. There is also progress in the field of stereo vision. Fu *et al.* [38] presented the first multi-view video summarization approach to improve results over graph and user attention models. Machine learning is also applied in video summarization by several authors [4]. Here, the machine vision system learns the underlying skimming approach by using the original videos and the user-provided skims for the original videos. The learned model for cutting videos is then used for future videos to provide high-quality skims.

5.3.2 Highlight detection

To preserve certain events in video skim, the highlight detection is applied. Traditionally, this approach concentrates heavily on the segmentation and selection steps, as the algorithms are required to detect certain events from the excerpts. Highlight detection should be applied if the application and the skim target domain can be accurately defined. The methods which fall in this category are generally application oriented. Chang *et al.* [16] used the Hidden-Markov Model to detect the different events in videos, thus requiring classification of video excerpts. Other approaches use the audio track, shot templates or temporal logic to detect the interesting events in videos. Many of the approaches try to detect certain highlights, such as:

- Replays in sports [55, 56, 125]
- Applauses, cheering [128]
- Excited speech (especially in sports) [26]
- Dominant camera motion [26, 72]
- The most viewed patterns [129]

Furthermore, the replay detection in sports has also a general framework presented by Han *et al.* [43], which can be applied for all types of sports. Another interesting approach is to produce fusions of the video data. Pritch *et al.* [93] created synopsis of the surveillance videos by detecting moving objects and their movement “tubes” in the scene. The overall skim is then produced by the objects moving along these tubes and thus, compressing the action in the scene. The approach only works with inactive surveillance videos, but is a novel way to visualize interesting moments in the video data.

5.3.3 Skimming curve formulation

The curve formulation is an approach that computes a score that directly associates an excerpt to be included in the skim to the selected perspective. The curve is a set of scores calculated from the excerpts. Finally, the video skim is generated by thresholding this curve and selecting those parts of the original video which have a score higher than the threshold as seen in Fig. 5.4.

The curve formulation is defined upon the decided *base unit.* The base unit is a primitive and modal-dependent. It can be, for example, color histogram difference between the consecutive frames. Visual summaries use frame or shot based approaches as video skims based on audio are usually dependant on the length of the base unit. The base units in skimming curve formulation are often resolved in segmentation and may contain higher

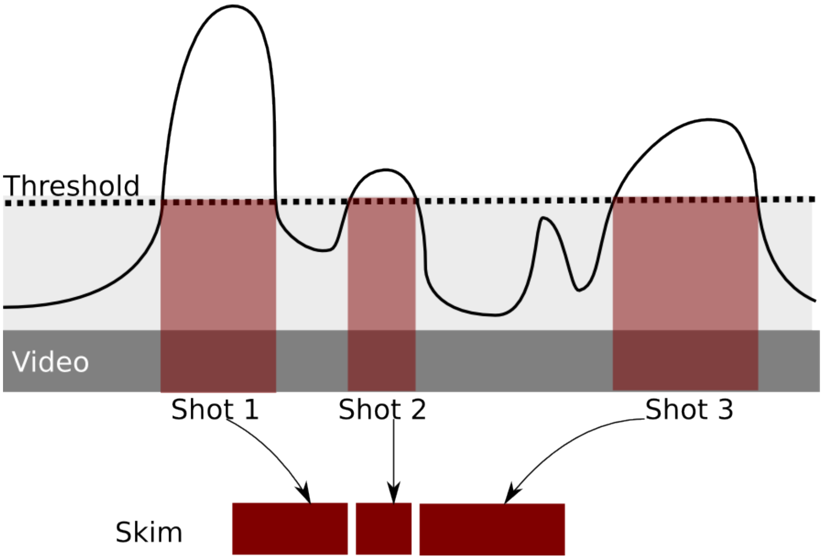


Figure 5.4: The curve formulation with an effective thresholding and selected shots from the original video.

level features to improve the quality of the skim (such as ignoring shots to be cut in the middle of a sentence).

After deciding the base units, the appropriate way to produce the score curve with the base units is defined. Especially, if there are multiple base units, it is time consuming to find the correct weight for the each unit. Mei *et al.* [77] identified many aspects of the camera work done in the video and produced multiple analysis which are eventually brought together with SVM approach and turned into a score curve. Furthermore, in highlight detection, the user needs to decide a thresholding value to use with the skim generation, which is usually problematic.

5.4 Evaluation of video summarization

Evaluating video skimming is difficult due to the lack of an objective ground-truth. It is also difficult for humans to decide if one video skim is better than another. Additionally, skims are usually application dependent, thus having different metrics for evaluation. Most of the publications evaluate the skimming results by using user studies. Some objective metrics has been used but they are easily biased towards a certain kind of skimming result. Truong *et al.* [109] address most of the problems in their survey and propose ways to evaluate the summarization methods. They propose recommendations for video summarization evaluation by creating ground-truth/common dataset and setting application-dependent evaluation perspectives.

One popular way to evaluate the summaries is through user studies and through user attention model. Ngo *et al.* [19] evaluated the skims by two criteria: informativeness and enjoyability presented by the user attention model by Ma *et al.* [72]. The evaluation was done by showing a group of people summaries of different length and then the original video. Afterwards, the users were required to give informativeness and enjoyability ratings (from 0 to 100) for each summary and also for the original video. They reported encouraging results of dropping 75% of content and only losing 20% of informativeness and enjoyability as a result with various types of videos.

A human driven experience model was proposed by Peng *et al.* [90]. They followed the viewer's eye movement and other behaviour to automatically evaluate user experience and created a video summarization system based on the collected data. Smeaton *et al.* [105] gave an introduction to the information retrieval (IR) evaluation from both the user and the system point of view. They presented TRECVid as an example of a system for evaluation benchmark. TRECVid had a video summarization evaluation task called "Rushes summarization" which contained raw film material from BBC which was used to create the finished product. The data contained a lot of stock footage where the actors are only sometimes present. The summaries were evaluated by how well the information in the videos was presented with as few frames as possible. Furthermore, the participants were allowed to use some creativity in the way summaries represented the original video, e.g. have picture-in-picture and split screen. Over *et al.* [89] demonstrated the evaluation process used in the TRECVid 2008 BBC Rushes summarization task. They had three human judges to evaluate the performance of the summarization in addition to the objective measures: length, processing time, and evaluation time (by human).

5.5 Summary

In this chapter, the basic tools for video analysis in the context of video shot boundary detection and video summarization, were presented. Although many authors have tried utilizing more complicated approaches for shot boundary detection, it has been always found that the simple color histogram-based method was fast and easy to parameterize. Many of the complicated methods were deemed to be slow to compute and performed poorly in the evaluations. The results presented in the TRECVid competition show that the best performing methods almost always used color histogram as the key component upon which the rest of the system was built. These systems frequently used different machine learning systems (e.g. SVM) to decide on the shot boundary type. Evidently, the shot detection is considered as a solved problem with over 80 percent accuracy with both hard and transitional cuts. The skimming has shown to be a much more difficult problem as there is no good general solution. Most of the approaches are strongly application driven with emphasis on speed and accuracy.

Skim generation is often built so that the output of the shot detection is used as a basic element for shortening. The shots, or more generally, the excerpts, are selected after the detection. The target of the selection step is to find the excerpts which most likely contain important content. The selection is done using the application specific appoach after which the shortening is applied to acquire the final excerpts. The shortening is very often associated with redundancy elimination which maximises the "interesting" content in videos.

Finally, the skim is assembled by using the shortened excerpts. Interestingly, the evaluation of the skims is problematic. There is no established evaluation framework and most of the evaluation is done through user studies. Even online monitoring of the users has been used to detect the important parts of the video. The TRECVid competition has partial tools for evaluating video summarizations and annotated data by several volunteers by NIST organization. The participant submitted summaries were compared against the ground-truth summaries, which is problematic as the summarization process, when conducted by humans, is a subjective topic at best. To migitate this problem, the submitted summaries were also subjected to a standard human evaluation.

Chapter VI

Video Shot Boundary Detection Using Bag of Features

The problem setting in image and video analysis problems are almost equivalent, but the adopted approaches have been divergent due to per frame processing required in many video processing tasks, such as in video shot boundary detection. For example, one hour of video contains approximately 100,000 frames, and the processing time of one second per frame would take 27 hours in total. In these kinds of tasks, “fast to compute features”, such as colour histograms, have typically been used. On the other hand, benchmark databases for image analysis have also become very large. For example, there are nearly 15 million annotated images in the ImageNet[1]. This has set new demands on approaches, and development has not only produced new techniques, but also more efficient implementations of the existing ones.

In this thesis, the state-of-the-art BoW method is adopted for processing massive amounts of images: dense SIFT for feature detection and representation, k-means clustering for codebook generation, L1-normalisation of codebook histograms, and the Euclidean distance for code matching. Our main contribution is to apply this method to shot boundary detection. In addition, we compare video specific codebooks, generated from the local features extracted from an input video to a “general codebook” generated from the ImageNet descriptors used in [27]. Moreover, the effect of varying the codebook sizes is studied, which is the most important parameter of BoW. The experiments are performed using the TRECVid 2007 shot boundary detection competition data.

In Chapter 5, the commonly used shot boundary detection methods were presented. The presented approach is compared against the frame windows method [106] which was among the top performers in [104] and can be considered the baseline method for shot boundary detection.

6.1 Visual Bag-of-Words

The seminal works of the visual bag-of-words (BoW) are [103] and [24]. The codebook generation and utilization with the visual Bag-of-Words in VOC case can be seen in Figure 6.1. In BoW, the salient local image features (interest points) are extracted with a special detector (e.g. SIFT) or fixed size patches are selected using dense sampling

[1] http://www.image-net.org/

on a regular grid. Then, these “keypoints” are described with a descriptor, the SIFT descriptor being the most popular. In the training phase, a codebook is generated by clustering the descriptors into a fixed number of codes. In the matching phase, the best matching code is assigned for each descriptor.

An image feature is generated by computing the histogram of the codes appearing in the image. Matching can be performed by histogram similarity between two images (frames).

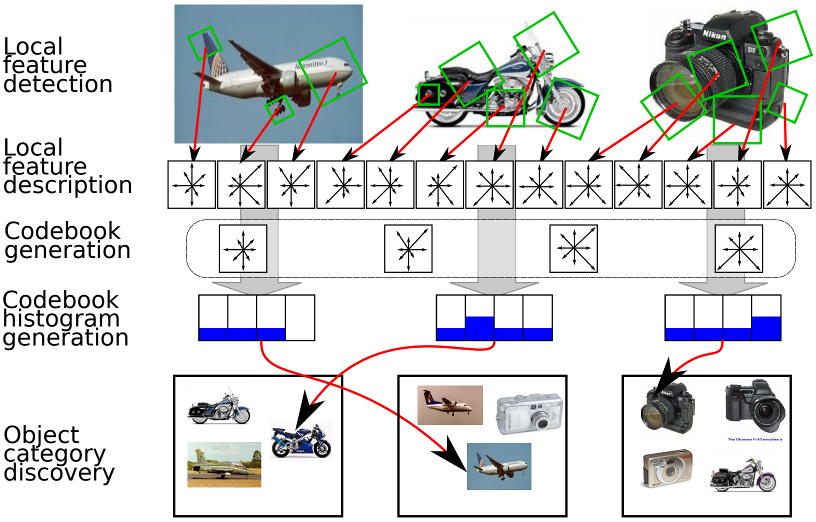

Local feature detection
Local feature description
Codebook generation
Codebook histogram generation
Object category discovery

Figure 6.1: The Process with the Visual Bag-of-Words. [52]

The shot boundary detection methods using local features are only a few. Li *et al.* [66] computed SIFT regions and descriptors, but did not utilise a codebook. They searched for SIFT matches directly between consecutive frames. A similar approach for content analysis was proposed by Sivic and Zisserman [103] who used a codebook. Both of these techniques, however, are extremely slow due to random sampling based matching. Sivic and Zisserman run the matching only for key frames of every shot as their application was content retrieval and Li *et al.* [66] did not report the computation times for their method. To the authors best knowledge, this work is the first to propose the bag-of-words approach to video shot boundary detection.

6.1.1 Selection of local features

As for the local features, the descriptor of choice is SIFT, the codebook is generated using the k-means clustering and the feature histograms are L1-norm normalised. Additionally, it was decided to use the color histogram presented in [106].

6.1.2 Codebook generation

There is a huge number of variants and extensions of the baseline BoW method (e.g., [63, 65, 13]), but often the basic method performs the best [112] and for large scale problems the most efficient discriminative methods are no longer feasible [27]. The recent implementation in [27] was selected for the method presented here. For feature detection, the method uses dense sampling on a regular grid, which has lately replaced the interest point detection methods in most visual object classification methods [31]. The grid size for the dense feature detection was 20×20 pixels and the radius of the circular SIFT features was 10 pixels.

The codebook was generated by first extracting 100,000 features from the frame data extracted from the TRECVid 2007 shot boundary data set. The "general" codebook was created from the randomly selected 100,000 ImageNet images. The k-means clustering was applied to the extracted features. The implementation of the k-means was provided by the widely used OpenCV library. The k-means was run with varying sizes of codebook: $K = 10$, 100, 1,000 and 10,000.

6.2 Shot boundary detection

The main parts of the Bag-of-Words implementation are similar to the approach presented by Deng *et al.* [27]. This is particularly the case with a general codebook generated from two million features extracted from ImageNet. It is interesting to study how well the general codebook performs as compared to a specific codebook, which is re-generated for every input video. Specific codebooks are generated using features extracted from selected frames (one frame per second in the implementation) and using the k-means clustering method. Next, the shot boundary detection algorithm is given in Alg. 6.1. It is noteworthy, that the only parameter for the method is the detection threshold τ which is equivalent to the colour histogram detection threshold. The other inputs are video and a pre-computed codebook.

Algorithm 6.1 Video shot boundary detection (BoW).

```
Load codebook $cb$.
for all Frames $i$ in video do
    Init $\boldsymbol{v}(i) \leftarrow 0$.
    Extract dense interest points and form their descriptors.
    Search the best matches in the codebook $cb$ for every extracted descriptor using
    the fast KD-tree search.
    Form the code histogram $\boldsymbol{h}$ using the codes.
    L1-norm normalise the histogram and compute the Euclidean distance $d_{curr}$ to the
    previous frame histogram.
    Calculate the distance difference (derivative) $d' = d_{curr} - d_{prev}$.
    If $d' \geq \tau$, then mark a shot boundary to the current frame: $\boldsymbol{v}(i) \leftarrow 1$.
    Set $d_{prev} = d_{curr}$
end for
Return the vector of shot boundaries $\boldsymbol{v}$.
```

6.3 Experiments

The experiments were conducted with the TRECVid 2007 Shot Boundary data set which contains over 6 hours of human annotated videos, 637,805 frames with 2317 transitions.

In the evaluation, the TRECVid evaluation protocol was used with the provided data and ground-truth. The operating point is set by the difference threshold τ. Low values result in high recall, but low precision, and vice versa. The precision-recall evaluation curves were computed by iteratively testing all possible values of the threshold τ. The implementation was programmed by using C++ and OpenCV [10] with fast K-NN descriptor matching [102] provided by the library.

6.3.1 Optimal codebook size

The size of the codebook (the number of clusters in k-means) is one of the computational bottlenecks. In object classification, the codebook sizes vary between 1,000 and 100,000, but in this case a codebook as small as possible is preferred as it would lower the computational requirements. The precision-recall curves for the method here and with varying codebook sizes are shown in Fig. 6.2. It is evident that the boundary detection is a low level task which requires only moderate discrimination power from the codebook.

Already 100 codes performed very well and increasing the size did not improve the results. The method started to collapse with codebooks smaller than 10.

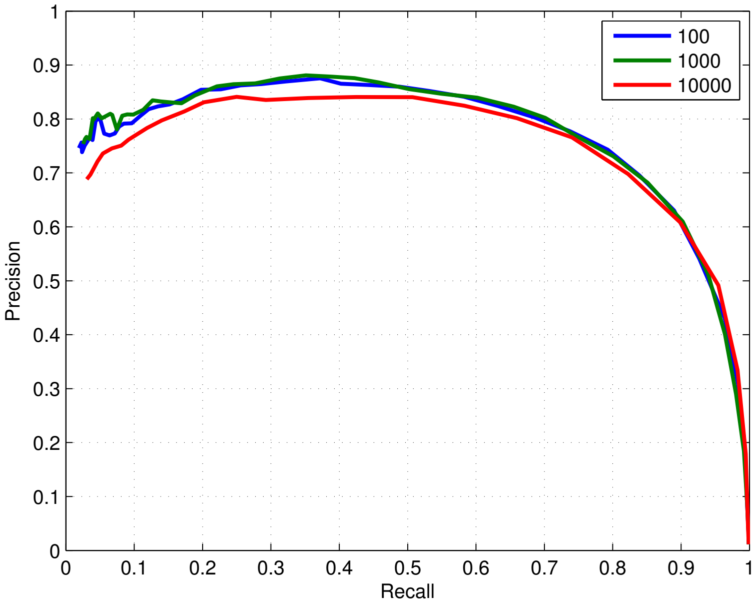
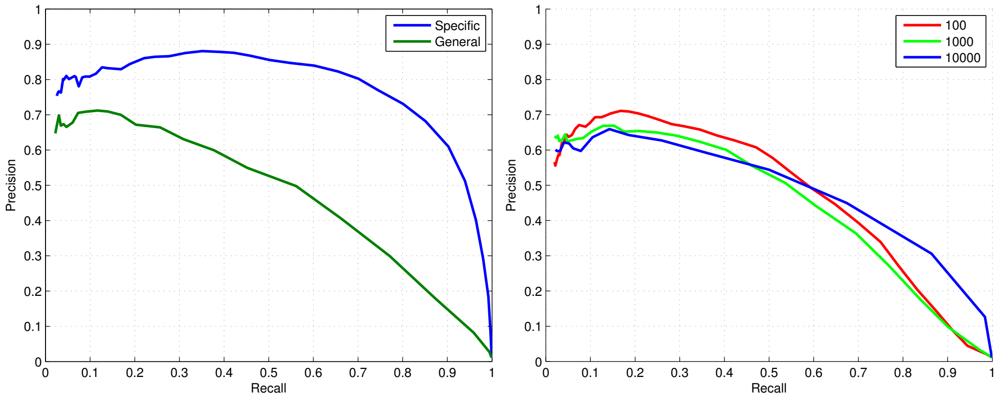


Figure 6.2: TRECVid Shot boundary detection precision-recall curves using the BoW method and different codebook sizes.

6.3.2 General vs. specific codebook

Based on the previous experiment, a generated codebook of size 100 performs well in shot boundary detection. That does not produce significant computational burden since the local features need to be extracted anyway. However, this could be improved if a general codebook would perform well, since the codebook generation step could be completely omitted. A general codebook was generated using the two million ImageNet features used in [27]. The shot boundary detection results and a comparison to a specific codebook ($k = 100$) are shown in Fig. 6.3. The results provide clear evidence that the general codebook does not perform well in this application and changing the codebook size does not help the situation. This result is quite surprising from object class detection point of view, but for shot boundary detection, video specific codebooks should be used.



(a)

(b)

Figure 6.3: TRECVid precision-recall curves. a) general and specific codebooks, b) general codebook results with varying size.

6.3.3 Baseline comparison

In the last experiment, the method was compared to the tailored RGB colour histogram method in [106]. The complementarity of the colour histograms and BoW histograms were also studied. This was achieved by first running both methods and then combining the output binary vectors (1's denoting a cut and 0's no cut). For combining, the logical AND and OR rules were tested. The AND rule should mainly improve the precision and the OR rule the recall. The results are shown in Fig. 6.4.

All the possible combinations of the thresholds τ_{BoW} and τ_{RGB} were tested and the highest precision for each recall point was selected. The BoW method clearly outperforms the baseline method using RGB colour histograms. However, it is evident that combining the two still improves detection performance remarkably.

The most problematic situations are presented in Figs. 6.5 and 6.6. Although most of the TRECVid test material performed well in practice, there were a few videos with a few serious misclassifications. The missed real cuts in Fig. 6.5 show similar scenes with a lot of similar features. In a) The bewel area causes major similarities between frames so that the contents are ignored, in b) the similar features such as text, person and background. In c) and d) the majority of the features are located on the borders and thus cause a missed cut.

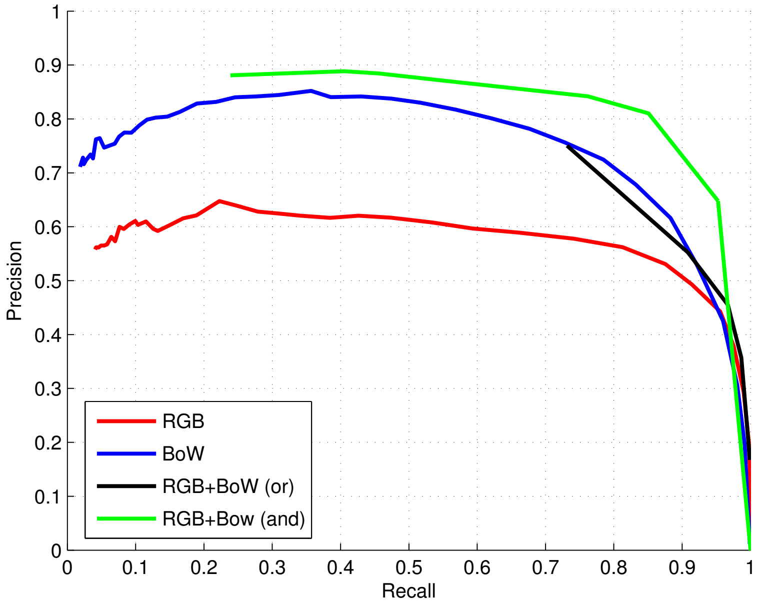


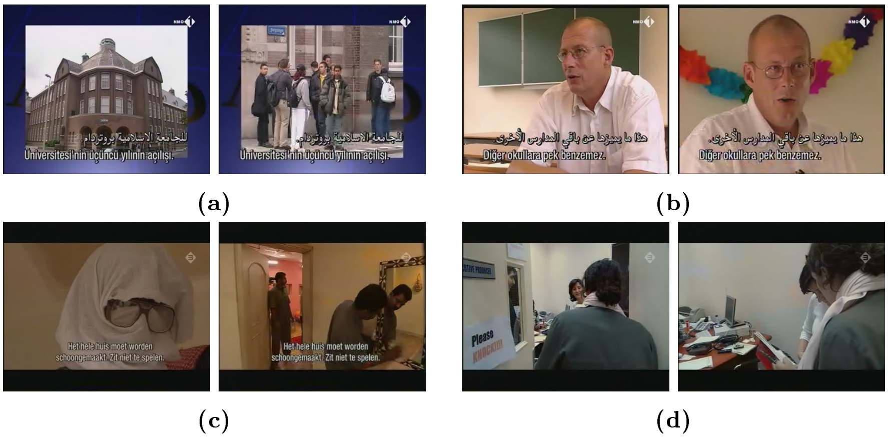
Figure 6.4: TRECVid comparison with the baseline method and with the hybrid of the two methods.

The outliers (detected cuts which are not really cuts) shown in Fig. 6.6 for the shot boundary detector are mostly found in dark or exceptional scenes. In a) and b) the false positive detection was caused by the slow gradual transition. In c) there was a flashing logo on black background which was causing strong differences between frames. In d) the lack of features in a dark scene caused the small increase of non-dark elements to register as a cut.

Some of these problems were not affected by the RGB histogram approach due to the global nature of the feature. The RGB histograms were tolerant to local instability such as dark scenes and static backgrounds. However, the RGB approach produces a lot of outliers and lacks precision. As shown, combining the approaches improves the detection results remarkably.

6.4 Application: supervised video summarisation using bag of features

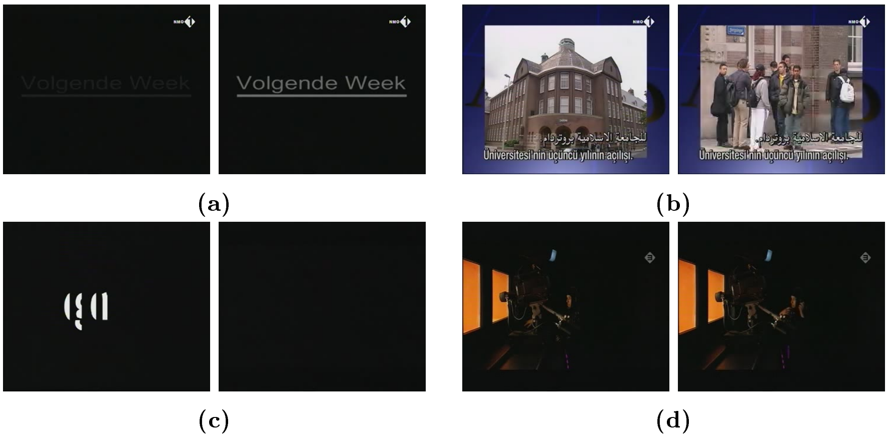
As explained in Chapter 5, the purpose of the video summarization is to provide a compact and informative representation of the original video. In this work, the main interest is in home videos. The algorithm presented is a hierarchical redundancy elimination method, where the redundancy is defined as short shots and major camera motion. In other words, the shortening algorithm avoids fast camera motion which usually means fast moving camera in home videos. It is presumed that the least motion means that


Üniversitesi nin üçüncü yılının açılışı.
Üniversitesi nin üçüncü yılının açılışı.
Diğer okullara pek benzemez.
Diğer okullara pek benzemez.
Het hele huis moet worden schoongemaakt. Zit niet te spelen.
Het hele huis moet worden schoongemaakt. Zit niet te spelen.
Please

(a) (b)

(c) (d)

Figure 6.5: Some missed cuts with the proposed method for the hardest videos: BG_14213 and BG_36628.


Volgende Week
Volgende Week
Üniversitesi nin üçüncü yılının açılışı.
Üniversitesi nin üçüncü yılının açılışı.

(a) (b)

(c) (d)

Figure 6.6: Outlier cuts with the proposed method for the hardest videos: BG_14213 and BG_36628.

something important is filmed. As such, the method does not work as well with professional and stable video material. The work is strongly application-oriented, by providing an online application to generate video skims. The subjectivity of the topic has been shown to be very challenging to be evaluated by many authors and usually the works are evaluated by lengthy user studies. The user studies help to verify and improve the algorithm, but do not work for comparisons against other methods in the field. Also, it is difficult to obtain a statistically large enough interviewing group. Due to the time limitation, only minimal subjective evaluation is presented in skimming results.

6.4.1 Overall workflow

The main skimming procedure presented follows the work of Cooper *et al.* [21] as the method proposed here is defining a similar redundancy minimization algorithm. However, the higher level definition differs and it is similar to the graph-based works by Ngo *et al.* [19]. Similar hierarchical works are done by Kang *et al.* [51], where a hierarchical model was applied, but they utilized GMM for learning the type of the camera motion.

The first step in the workflow is to acquire a set of useful features for making the decisions upon importance of different parts (shots) of the video. The features considered important in method are the parts with the least camera movement. This is due to the nature of home videos as people usually keep the camera steady while filming the important scenes. Here, the camera movement is estimated by comparing the feature histograms between frames. The presented method utilizes the output of the Bag-of-Words (BoW) based shot detection algorithm presented in Chapter 6. The method uses local features from each frame to determine the similarity between consecutive frames, detecting the abrupt changes and thus, the shot boundaries. The rationale behind the usage of local features in skimming is that local features, or more precisely, the BoW approaches are shown to be efficient in image classification and detection tasks. This is also important for future extensions, as local features can be utilized in many higher level image processing tasks, as face detection and recognition. The local features have been utilized in video summarization context by many authors [71, 25]. However, most of the work concentrates on the usage of low level color histograms [47, 127], edge detection [127], some form of motion information [47], or any combination of these features [127, 47].

The most similar approach to the method presented here is made by Cunha *et al.* [25]. They utilized three different methods to generate the initial summaries and finally combined them to create a summary which minimizes redundant information. The three methods were: BoW with SIFT, BoW with Hue-SIFT [118] and STIP [62]. However, they approached the redundancy elimination from a different perspective by selecting the relevant segments using the representative groups from k-means clustering.

6.4.2 Shot hierarchy

The presented method uses a hierarchical structure extracted from the local feature based shot detection algorithm. The approach for hierarchical skim generation is presented in Alg. 6.2. First, the BoW shot detection method is used to construct a hierarchical model. The different levels of the hierarchy are acquired by decreasing the threshold t_j. The shots are selected by thresholding the difference curve presented in Fig. 6.7.

The intervals between the highest peaks define the shots which are inserted into the hierarchy. The thresholding starts at the top which produces the least shots (minimum 2). The treshold levels are adjusted automatically by selecting the next threshold which produces two times fewer shots than the previous threshold. The top level is one shot, which is further divided until there is at least t_n shots to accommodate the required total length of the skim τ_l. Next, the shots are generated for all threshold levels in $\boldsymbol{t}$ and the hierarchical skim model is generated. The model generation is shown in Fig. 6.8.

After the generation of the hierarchical model, the overall process is simple. The first step is to find an appropriate set of shots from which the user can select the relevant

Algorithm 6.2 The hierarchical skim generation.

```
Load frame distances $\boldsymbol{d}$ using Alg. 6.1
Initialize the hierarchy vector $\boldsymbol{r_{it}}$.
for all Thresholds $\boldsymbol{t}$ do
    for all Distances $\boldsymbol{d}$ do
        if $d_i \geq t$ then
            Store the shot into $\boldsymbol{r_{it}}$.
        end if
    end for
end for
Initialize the selected shots set $\boldsymbol{s}$
Initialize number of shots $n_s$
Initialize thresholds: length $\tau_l$ and shots $\tau_s$
Acquire the user input: weights $\boldsymbol{w}$ for shots
Select the highest threshold $t$
while length $\boldsymbol{s} \leq \tau_l$ and $\tau_s \leq n_s$ do
    Set weighted shot: $wr_{it} = length(r_i) \times w_i$
    Select the longest weighted shot $max(wr_{it})$
    if the total length of skim is over $\tau_l$ then
        Next hierarchy level $t = t + 1$
        Select the next highest threshold $max(r_{it})$
    end if
end while
Generate the final skim by using the set of shots from $\boldsymbol{s}$
```

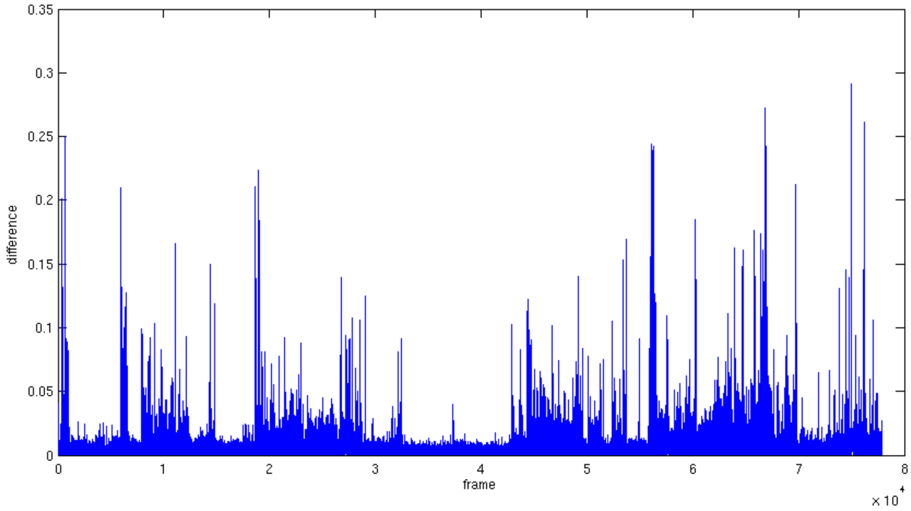


Figure 6.7: The difference curve for Proffessor's day video (78000 frames in total).

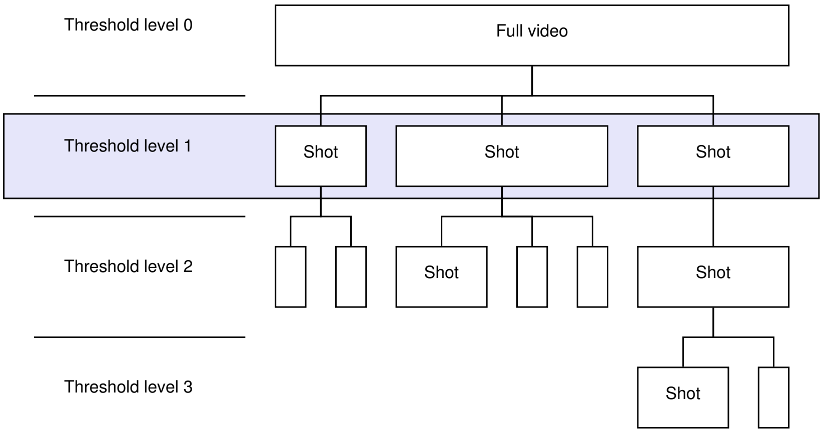


Figure 6.8: The hierarchical model used by the summarization method. The selected threshold level is highlighted.

shots to be included in the skimming process. This is achived by finding a threshold level which produces the required amount of shots. For example, in Fig. 6.8 the selected threshold level is shown with gray background. The threshold level of the hierarchical model is found by breadth-first search, limited by the depth of the model. Next, the shots are entered into the skimming process to check if the selected shots are short enough to generate the skim. If not, then the shot with the longest length is divided by the hierarchical model so that the next longest sub-shot can be selected. This process will be continued as long as there is a sub set which is shorter than the total length of skim τ_l.

When the selection of shots is finished, the summarization process is brought to the last step - rendering. The selected shots are rendered together using a user-specified transition such as fade-in/out. Other user-controlled options include different color modification schemes, similar to the popular online image service *instagram* [2].

6.4.3 Examples

The examples section contains experiments with the proposed home video video summarization method. Most of the topics discussed in this section are very subjective in nature. The experiments were conducted on a typical home video: unplanned and shaky filming, low-quality imaging (strong interlacing), large amount of transitions with a few spots of important content for the user. The visual overview of the Professor's Day video can be seen in Fig. 6.9. The video contains situations from the morning until the end of the day, filmed by Joni Kämäräinen from Lappeenranta University of Technology. Another video selected for the experiments was Wedding material produced by Kymenlaakso University of Applied Sciences. The wedding video material does not have any problems with shaky camera movement, but has a lot of dark scenes with flashing lights which are problematic.

[2] `http://instagram.com`

Table 6.1: The numerical representation of the contents of Proferssor's Day video.

Type of material	**Number of frames**	**Percentage**
Ignored/junk	27169	34.8%
Transitions/filler	22516	28.9%
Relevant/zoomed-in	28223	36.3%

The visual overview of the Wedding video can be seen in Fig. 6.10. The video contains basically three scenes: inside the church, outside the church and the reception.

For quantitative measurement, the videos were annotated manually. The annotation was made by utilizing the publicly available open-source video software *Avidemux* [3] and *Kino* [4]. The annotation is a list of frame ranges with a score on how relevant this moment is for the topic of the video.

The numerical representation can be seen in Table 6.1. The Professor's Day home video has 1/3 of redundant information which is not desired to be included in skims. The majority of the video material was transitions: walking/driving between places or content is relevant to the topic of the video: "A day in a professor's life". The relevant video material is not automatically an indication of "enjoyable" viewing experience: the relevant video material contains a lot of "still moments" such as watching the computer screen.

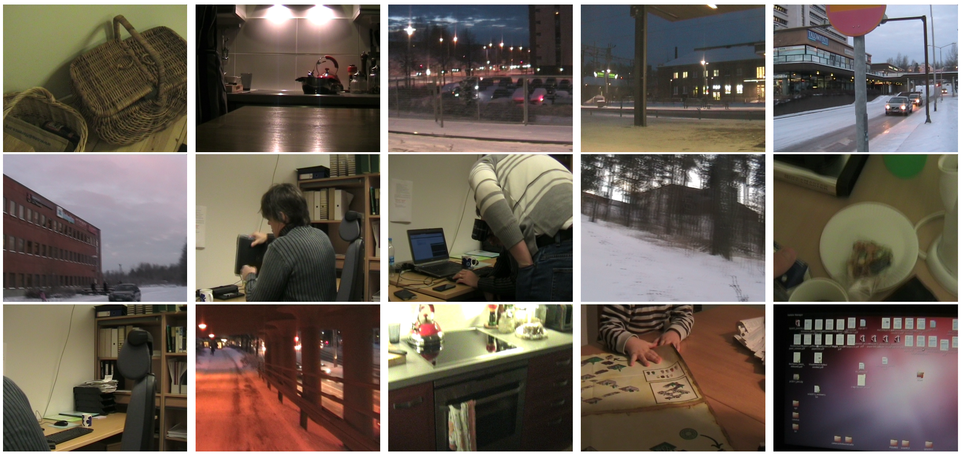


Figure 6.9: Overview of the contents of the professor's day video.

The proposed method uses the shots detected with the Visual Bag-of-Words as a starting point for the skim generation. The ideal situation for the skim generation is to have all relevant information within the skim. As such, the user needs to be able to select all shots containing important information. However, every shot selected for the skim generation

[3] `http://fixounet.free.fr/avidemux/`
[4] `http://kinodv.org/`

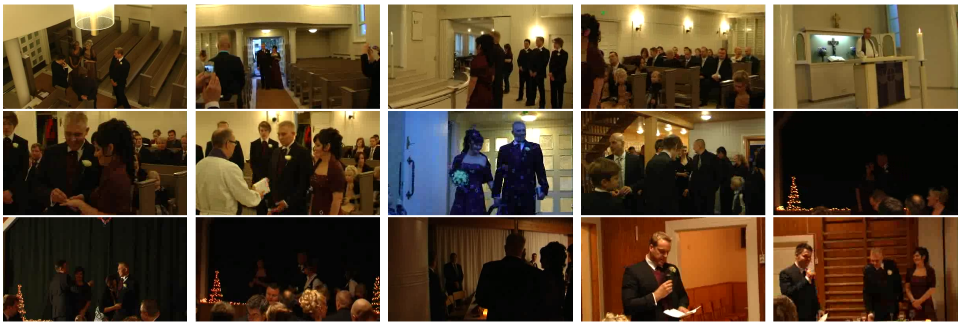


Figure 6.10: Overview of the contents of the wedding video.

is bound to have some redundant video material. An experiment with annotated data was made to estimate how many shots would be enough to provide as much relevant frames as possible. In Fig. 6.11, the ratio between the shots and redundant frames for the annotated Professor's Day video is shown. It can be seen from the figure, if the user is given 200 shots to select from and the user selects all shots containing relevant information for him, the user will receive all frames of interest, but also 10,000 redundant frames (12%). Interestingly, there is not much difference when selecting only relevant data or relevant data with transitions.

From the application point of view, the number of primary shots presented to the user is not allowed to be too high. Already 100 UI (User Interface) items is considered to be a cluttered view in UI design [96]. However, it is unlikely that the end-user wants every frame annotated as relevant, and thus, the resulting skim is forced to have some amount of irrelevant frames. This is visible in Fig. 6.11, as the number of frames with irrelevant data is never zero. Naturally, if 300 shots were provided for users it would be possible to select much better set of shots, but the user experice would suffer. Although justified, the presented numbers do not translate very well to real life situations, as they only present a subjective evaluation. During the practical evaluation, it was found out that 20 shots provide adequate results. This finding showed that the irrelevant content is seen as appropriate in many cases.

The video summarization produced by the method is illustrated in Fig. 6.12, with the parts removed from the video in Fig. 6.13. The 52-min long video was processed into a 60 seconds long summarization by providing the user 20 shots to select from. After processing the selected shots, the generated video summary contained: 41% relevant frames, 45% transitions and 14% unwanted frames. Another example, the Wedding video, was also processed into a 60 second long summarization. The original length of the Wedding video was 59 minutes and it contains 89151 frames. A subset of the frames included is shown in Fig. 6.14. The frames removed from the video are shown in Fig. 6.15. By providing more shots to select from, the quality of summarization can be improved. In addition, the users found that lowering the speed of the skim (about half-speed) and adding slow music produced the best result.

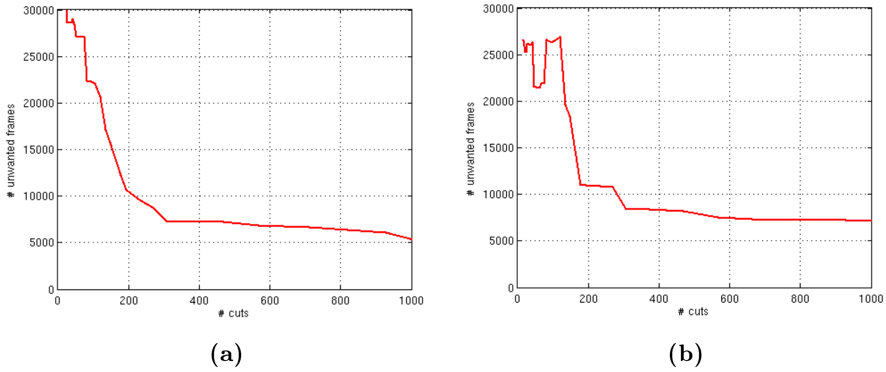


Figure 6.11: Irrelevant frame ratio. The data has a) all transitions and "filler" b) only "filler".

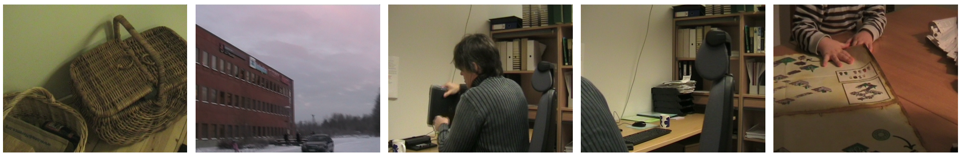


Figure 6.12: Example summary produced by the summarization method.

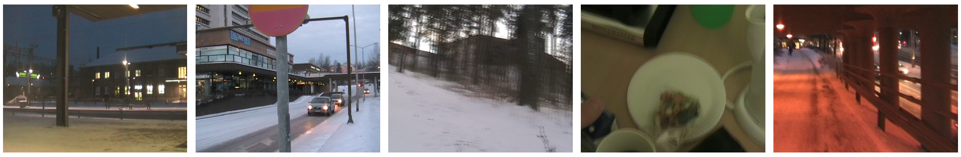


Figure 6.13: Samples of content which the method classifies as redundant.

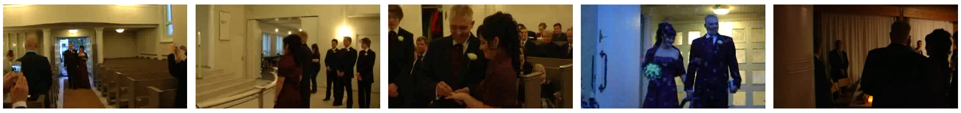


Figure 6.14: Example summary produced by the summarization method.

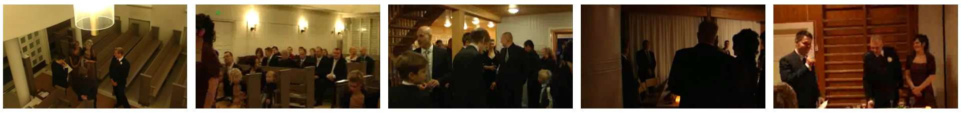


Figure 6.15: Samples of content which the method classifies as redundant.

6.4.4 Online application

To demonstrate the method, an online application was developed. The online application utilizes HTML5 document definition with Javascript language to communicate with the server software. The server software is a RESTful Web Service written in PHP5, which takes care of user logins and skim generation process. The application is shown in Fig. 6.16. The application allows users to upload videos to the server and run the proposed summarization method. After the upload, the shot detection algorithm is automatically run on the provided video. The overall process of the online application is:

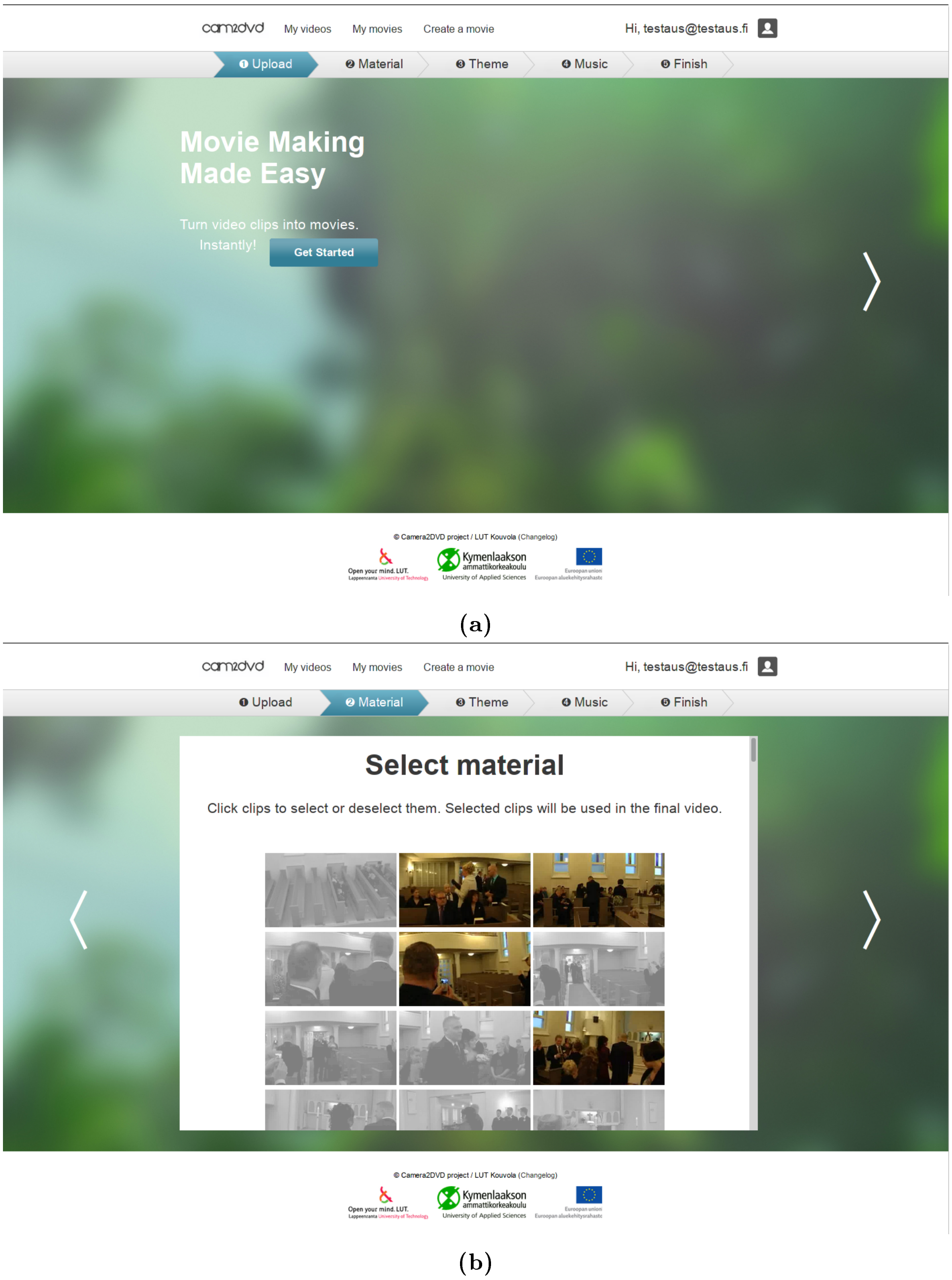
1. Login to the service
2. Upload video
3. Select the shots which are included into the video skim
4. Select the background music or the length of the video skim
5. The video skim generation process
6. Viewing the generated video (and possible rating)

The users are allowed to select the way the skim is generated: by selecting the shots, providing the background music or length of the video skim. By selecting any provided music track, the length is fixed and can not be changed. Some people have used the system for visualization purposes as it gives a good overview of the video content in the shot selection phase. The overall processing time for one hour of video is approximately 1.5 hours. The results are satisfactory but can be improved with higher level processing: e.g. object or face detection and color normalization.

6.5 Summary

In this chapter, a novel Bag-of-Words based approach for shot boundary detection was introduced. The well established TRECVid evaluation framework with over 6 hours of annotated videos was used for the experiments and the results showed improved results over the baseline color histogram approach.

The more interesting result arose from the fact that the local feature based approach (BoW) was complementary with the results by the global description, namely the color histogram. The best results were achieved by combining the local and global description by AND rule. This means that the high number of outlier results by color histogram method were "verified" by the SIFT features which had high precision but low recall rates. Further, the codebook related reseults are noteworthy as it is evident that using a codebook size higher than 100 is not recommended with just 6 hours of video. Furthermore, the results with the general codebook showed low performance rates, leading to the assumption that codes generated with random images are not enough to differentiate small changes between frames.



cam2dvd
My videos
My movies
Create a movie
Hi, testaus@testaus.fi
❶ Upload
❷ Material
❸ Theme
❹ Music
❺ Finish
Get Started
© Camera2DVD project / LUT Kouvola (Changelog)
Open your mind. LUT. Lappeenranta University of Technology
Kymenlaakson ammattikorkeakoulu University of Applied Sciences
Euroopan unioni Euroopan aluekehitysrahasto

Movie Making Made Easy

Turn video clips into movies. Instantly!

(a)

cam2dvd
My videos
My movies
Create a movie
Hi, testaus@testaus.fi
❶ Upload
❷ Material
❸ Theme
❹ Music
❺ Finish
© Camera2DVD project / LUT Kouvola (Changelog)
Open your mind. LUT. Lappeenranta University of Technology
Kymenlaakson ammattikorkeakoulu University of Applied Sciences
Euroopan unioni Euroopan aluekehitysrahasto

Select material

Click clips to select or deselect them. Selected clips will be used in the final video.

(b)

Figure 6.16: The online application. a) starting screen, b) shot selection phase.

The strong point of using local features is that the large amount of computer vision methods such as face and object recognition can help detect higher level features. As such, it is a feasible future work of this method.

Additionally, in this chapter, a general video summarization method was introduced. The method uses shot boundary detection algorithm based on the visual Bag-of-Words and generates video summaries by hierarchical approach. The hierarchy is utilized to generate a skim from the shots selected by the user. The number of shots given to the user is fixed by a given threshold level in the hierarchy. In the examples, the number of given shots was 20. The skim generation process uses the user selected shots to generate the video summarization. The algorithm selects the most fitting shot from the hierarchical model given the required time.

Based on the examples selected in this chapter, the resulting skims are forced to contain some redundant video material. The reason is that, the method uses shots as a basic unit in the skim generation and shots themselves are forced to contain some amount of redundant information. Based on the annotations made by a volunteer, the threshold level which produces 300 shots should proved the best results. This is problematic from the practical point of view, as the user studies and UI proposals suggested that selecting relevant moments in videos becomes cumbersome with more than 100 shots. It is important to mention that the numbers presented here are highly subjective and reflect the nature of the video. The video summarizations created with only 20 shots presented to users already provided satisfactory results.

Future work should concentrate more on user-surveys based on the enjoyability of the resulting videos. The initial work towards this goal has already been done as an easily accessible online application has been created. The resulting feedback is then used to further evaluate and improve the proposed method. Another interesting goal for future work is the utilization of local features. The method only uses local features for skim generation in the shot generation phase, but local features could be easily applied in face or object detection and recognition. The current method is able to detect the scene change, enabling scene related processing such as white balancing.

Chapter VII

Discussion

In this thesis local features and their various applications were investigated. Four major contributions were made: 1) evaluation of the local features, 2) image alignment, 3) shot detection and 4) skim generation. The commonly used local feature detectors and descriptors were presented in Chapter 2 and it was shown how the common local feature detectors and descriptors have evolved over the last decade. The current emphasis is in improving computational performance of the existing methods by applying GPU techniques and introducing low-lever binary features. In Chapter 3, previous work was presented and it was shown that there is a strong emphasis on the evaluation of the local features in wide baseline stereo applications. The evaluation framework in the object categorization context and the important findings regarding the evaluation are the first major contribution of this thesis. With the presented Coverage-N metric, important results were presented: not only the local features seem to perform better with more detected features, the local feature descriptors also have interesting properties. This direction is already seen in the mainstream VOC approaches as the methods use dense sampling. Also, the importance of the multiple assignment of features was demonstrated in the results - an aspect which is not widely utilized in the mainstream object detection approaches. Therefore, based on the findings in this thesis, the better matching local features (less false positives) or higher number of local features should be utilized and examined in the object detection tasks.

The second important contribution was the image alignment method presented in Chapter 4 utilizing the local features and the multiple assignment of the features. The relatively simple algorithm was able to align images based on the common local features found over a set of object-class images. Not only does the presented method provide good alignments, the general idea can also be utilized in other applications, such as improving the object detection approaches. The results show that the local feature based approach clearly outperforms the pixel-based method for being flexible with objects in arbitrary alignments. Future work on the local feature based alignment should concentrate on limiting the available transformations or improving the matching performance of the local features. By limiting the transformations, the overall precision should increase as the false positive feature matches would be ignored. Naturally, by increasing the quality of the features, the results of the automatic alignment would improve as the feature maches are more often correct.

In Chapter 6, the local features were applied to the temporal space of videos. In terms of computing power, the problems posed by videos are very different from those posed by images. In particular, with over 90,000 frames of image data to process, the performance of the selected method is important. The video shot boundary detection using the Visual Bag-of-Words was the third major contribution of this thesis. The Visual Bag-of-Words method, which is a widely utilized approach in object detection, was examined in shot boundary detection. Not only did the Visual Bag-of-Words provide notably improved results against the baseline color histogram based approach, but it also showed insight into the BoW method in this context: The experiments showed that a small codebook of just 100 features was enough although the mainstream object detection uses codebooks of size 1000 or larger. Another important result was the difference between a codebook generated using random images and the codebook generated using the video frames. The performance dropped dramatically when using the "general" codebook. The results showed that the small differences between the consecutive frames are impossible to detect with a general codebook. Another important finding was the complementary nature of the BoW and RBG histogram features in shot detection. The complementarity itself suggests possibilities of combining the global and the local image features to improve the precision of the existing object detection and classification methods.

Another important contribution, the skim generation, was presented as a part of Chapter 6. The presented method builds upon the BoW shot detection method introduced in Chapter 6 by utilizing the produced shots in a hierarchical manner. The method was supervised by the user to produce relevant summaries of the source material. The user has the possibility to ignore the shots containing too much redundant material and thus produce a higher quality skim. Eventhough the video frames included in the final skim are all relevant, the overall viewing experience might still be unpleasant. This aspect of the evaluation requires user studies and professional evaluation to be confirmed and it is the one lacking quality in the current evaluation practices. Nevertheless, the proposed method and the available online application can lower the threshold for creating summaries for home videos and the results have been confirmed by users even if the results are not perfect. The possible improvements for the skim generation algorithm contain many of the existing approaches which utilize the local features: e.g. motion detection and face/object detection/recognition. The usage of the audio is another important aspect of the skim generation which was only slightly explored during the implementation.

In conclusion, this thesis verified local features as a powerful tool in many applications. The most intriguing future work would be the improvement of the local features: Every main contribution in this thesis would benefit from higher quality local features for object detection. Another limiting aspect of the local image features shown in this thesis is the speed: the local features in video processing are slow to compute, which is unacceptable in consumer-level applications.

Bibliography

[1] Agarwal, A., and Triggs, B. Multilevel image coding with hyperfeatures. *International Journal of Computer Vision 78*, 1 (2008).

[2] Agrawal, M., Konolige, K., and Blas, M. R. Censure: Center surround extremas for realtime feature detection and matching. In *Proc. of European Conference on Computer Vision* (2008), D. A. Forsyth, P. H. S. Torr, and A. Zisserman, Eds., vol. 5305 of *Lecture Notes in Computer Science*, pp. 102–115.

[3] Ashraf, A., Lucey, S., and Chen, T. Fast image alignment in the fourier domain. In *Proc. of Computer Vision and Pattern Recognition* (2010).

[4] Basak, J., Luthra, V., and Chaudhury, S. Video summarization with supervised learning. In *Proc. of International Conference on Pattern Recognition* (2008), pp. 1–4.

[5] Bay, H., Ess, A., Tuytelaars, T., and Gool, L. V. Surf: Speeded up robust features. *Computer Vision and Image Understanding 110*, 3 (2008), 346–359.

[6] Beaudet, P. R. Rotationally invariant image operators. In *Proc. of International Joint Conference on Pattern Recognition* (1978), pp. 579–583.

[7] Belongie, S., Malik, J., and Puzicha, J. Shape matching and object recognition using shape context. *IEEE Transactions on Pattern Analysis and Machine Intelligence 24*, 24 (2002).

[8] Berg, A., Berg, T., and Malik, J. Shape matching and object recognition using low distortion correspondences. In *Proc. of Computer Vision and Pattern Recognition* (2005).

[9] Boreczky, J., , Boreczky, J. S., and Rowe, L. A. Comparison of video shot boundary detection techniques. 170–179.

[10] Bradski, G. The OpenCV Library. *Dr. Dobb's Journal of Software Tools* (2000).

[11] Brown, M., and Lowe, D. Recognising panoramas. In *Proc. of International Conference on Computer Vision* (2003), pp. 1218–1227.

[12] Calonder, M., Lepetit, V., Strecha, C., and Fua, P. Brief: Binary robust independent elementary features. In *Proc. of European Conference on Computer Vision* (2010), pp. 778–792.

[13] Cao, Y., Wang, C., Li, Z., Zhang, L., and Zhang, L. Spatial bag-of-features. In *Proc. of Computer Vision and Pattern Recognition* (2010).

[14] Cao, Y., Zhang, H., Gao, Y., Xu, X., and Guo, J. Matching image with multiple local features. In *Proc. of International Conference on Pattern Recognition* (2010), IEEE, pp. 519–522.

[15] Carbonetto, P., Dorko, G., Schmid, C., Kuck, H., and de Freitas, N. Learning to recognize objects with little supervision. *International Journal of Computer Vision 77* (2008), 219–237.

[16] Chang, P., Han, M., and Gong, Y. Extract highlights from baseball game video with hidden markov models. In *Proc. of IEEE International Conference on Image Processing* (2002), pp. 609–612.

[17] Chen, S.-C., Shyu, M.-L., Liao, W., and Zhang, C. Scene change detection by audio and video clues. In *Proc. of IEEE International Conference on Multimedia and Expo* (2002), IEEE, pp. 365–368.

[18] Chen, Y., Yuille, A., Zhu, L., and Zhang, H. Unsupervised learning of probabilistic object models (POMs) for object classification, segmentation and recognition. In *Proc. of Computer Vision and Pattern Recognition* (2008).

[19] Chong-wah Ngo, Y.-f. M., and jiang Zhang, H. Video summarization and scene detection by graph modeling. *IEEE Transactions on Circuits and Systems for Video Technology 15* (2005), 296–305.

[20] Chum, O., and Matas, J. Web scale image clustering – large scale discovery of spatially related images. Tech. rep., Center for Machine Perception, Czech Technical University in Prague, 2008.

[21] Cooper, M. D., and Foote, J. Summarizing video using non-negative similarity matrix factorization. In *Proc. of IEEE Workshop on Multimedia Signal Processing* (2002), IEEE Signal Processing Society, pp. 25–28.

[22] Cox, M., Sridharan, S., Lucey, S., and Cohn, J. Least squares congealing for unsupervised alignment of images. In *Proc. of Computer Vision and Pattern Recognition* (2008).

[23] Cox, M., Sridharan, S., Lucey, S., and Cohn, J. Least squares congealing for large number of images. In *Proc. of Computer Vision and Pattern Recognition* (2009).

[24] Csurka, G., Dance, C., Willamowski, J., Fan, L., and Bray, C. Visual categorization with bags of keypoints. In *ECCV Workshop on Statistical Learning in Computer Vision* (2004).

[25] Cunha, T. O., de Souza, F. G. H., de Albuquerque Araújo, A., and Pappa, G. L. Rushes video summarization based on spatio-temporal features. In *Proc. of South Atlantic Conference* (2012), S. Ossowski and P. Lecca, Eds., ACM, pp. 45–50.

[26] De, C. C., and Coldefy, F. Unsupervised soccer video abstraction based on pitch, dominant color and camera motion analysis. In *Proc. of ACM International Conference on Multimedia* (2004), pp. 268–271.

[27] Deng, J., Berg, A., Li, K., and Fei-Fei, L. What does classifying more than 10,000 image categories tell us? In *Proc. of European Conference on Computer Vision* (2010).

[28] Deng, J., Dong, W., Socher, R., Li, L.-J., Li, K., and Fei-Fei, L. ImageNet: A large-scale hierarchical image database. In *Proc. of Computer Vision and Pattern Recognition* (2009).

[29] Everingham, M., van Gool, L., Williams, C., Winn, J., and Zisserman, A. The PASCAL Visual Object Class Challenge 2007 (VOC2007) Results. http://pascallin.ecs.soton.ac.uk/challenges/VOC, 2007.

[30] Everingham, M., Van Gool, L., Williams, C. K. I., Winn, J., and Zisserman, A. The PASCAL Visual Object Classes Challenge 2010 (VOC2010) Results. http://www.pascal-network.org/challenges/VOC/voc2010/workshop/, 2010.

[31] Everingham, M., Van Gool, L., Williams, C. K. I., Winn, J., and Zisserman, A. The PASCAL Visual Object Classes Challenge 2011 (VOC2011) Results. http://www.pascal-network.org/challenges/VOC/voc2011/workshop/, 2011.

[32] Fei-Fei, L., Fergus, R., and Perona, P. Learning generative visual models from few training examples: an incremental bayesian approach tested on 101 object categories. In *CVPR Workshop on Generative-Model Based Vision* (2004).

[33] Fei-Fei, L., Fergus, R., and Perona, P. One-shot learning of object categories. *IEEE Transactions on Pattern Analysis and Machine Intelligence 28*, 4 (2006), 594.

[34] Felzenszwalb, P., McAllester, D., and Ramanan, D. A discriminatively trained, multiscale, deformable part model. In *Proc. of Computer Vision and Pattern Recognition* (2008).

[35] Fischler, M., and Bolles, R. Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography. *International Journal of Graphics and Image Processing 24*, 6 (1981).

[36] Frey, B., and Jojic, N. Transformed component analysis: Joint estimation of spatial transformations and image components. In *Proc. of Computer Vision and Pattern Recognition* (2000).

[37] Frey, B., and Jojic, N. Transformation-invariant clustering and dimensionality reduction using EM. *IEEE Transactions on Pattern Analysis and Machine Intelligence 25*, 1 (2003), 1–17.

[38] Fu, Y., Guo, Y., Zhu, Y., Liu, F., Song, C., and Zhou, Z.-H. Multi-view video summarization. *IEEE Transactions on Multimedia 12*, 7 (2010), 717–729.

[39] Galvez-Lopez, D., and Tardos, J. D. Real-time loop detection with bags of binary words. In *Proc. of IEEE/RSJ International Conference on Intelligent Robots and Systems* (sept. 2011), pp. 51 –58.

[40] Gargi, U., Kasturi, R., and Strayer, S. H. Performance characterization of video-shot-change detection methods. *IEEE Transactions on Circuits Systems for Video Technology 10*, 1 (2000), 1–13.

[41] Gong, Y. Summarizing audiovisual contents of a video program. *EURASIP Journal on Advances in Signal Processing 2003*, 2 (2003), 160–169.

[42] Hamouz, M., Kittler, J., Kamarainen, J.-K., Paalanen, P., Kälviäinen, H., and Matas, J. Feature-based affine-invariant localization of faces. *IEEE Transactions on Pattern Analysis and Machine Intelligence 27*, 9 (2005).

[43] Han, B., Yan, Y., Chen, Z., Liu, C., and Wu, W. A general framework for automatic on-line replay detection in sports video. In *Proc. of ACM International Conference on Multimedia* (New York, NY, USA, 2009), MM '09, ACM, pp. 501–504.

[44] Harris, C., and Stephens, M. A combined corner and edge detector. In *Alvey Vision Conference* (1988), pp. 147–151.

[45] Hartley, R., and Zisserman, A. *Multiple View Geometry in computer vision.* Cambridge press, 2003.

[46] Huang, G., Jain, V., and Learned-Miller, E. Unsupervised joint alignment of complex images. In *Proc. of International Conference on Computer Vision* (2007).

[47] Huang, S.-H., Wu, Q.-J., Chang, K., Lin, H.-C., Lai, S.-H., Wang, W.-H., Tsai, Y.-S., Chen, C.-L., and Chen, G.-R. Intelligent home video management system. In *Proc. of International Conference on Information Technology: Research and Education* (2005), IEEE, pp. 176–180.

[48] Jiang, T., Jurie, F., and Schmid, C. Learning shape prior models for object matching. In *Proc. of Computer Vision and Pattern Recognition* (2009).

[49] Joyce, R. A., and Liu, B. Temporal segmentation of video using frame and histogram space. *IEEE Transactions on Multimedia 8*, 1 (2006), 130–140.

[50] Kang, H.-W., and Hua, X.-S. To learn representativeness of video frames. In *Proc. on ACM International Conference on Multimedia* (2005).

[51] Kang, H.-W., and Hua, X.-S. To learn representativeness of video frames. In *Proc. of ACM International Conference on Multimedia* (New York, NY, USA, 2005), MULTIMEDIA '05, ACM, pp. 423–426.

[52] Kinnunen, T., Kämäräinen, J.-K., Lensu, L., and Kälviäinen, H. Bag-of-features codebook generation by self-organisation. In *International Workshop on Self-Organizing Maps (WSOM 2009)* (2009).

[53] Kinnunen, T., Kamarainen, J.-K., Lensu, L., Lankinen, J., and Kälviäinen, H. Making visual object categorization more challenging: Randomized Caltech-101 data set. In *Proc. of International Conference on Pattern Recognition* (2010).

[54] Kinnunen, T., Lankinen, J., Kämäräinen, J.-K., Lensu, L., and Kälviäinen, H. Unsupervised visual object categorisation with BoF and spatial matching. In *Proc. of Scandinavian Conference on Image Analysis* (Espoo, Finland, 2013).

[55] Kobla, V., Dementhon, D., and Doermann, D. Detection of slow-motion replay sequences for identifying sports videos. In *Proc. IEEE Workshop on Multimedia Signal Processing* (1999), pp. 135–140.

[56] Kobla, V., DeMenthon, D., and Doermann, D. Identifying Sports Videos using Replay, Text and Camera Motion Features. In *SPIE* (January 2000), pp. 332–343.

[57] Kokkinos, I., and Yuille, A. Unsupervised learning of object deformation models. In *Proc. of International Conference on Computer Vision* (2007).

[58] Kumar, M., Zisserman, A., and Torr, P. Efficient discriminative learning of parts-based models. In *Proc. of International Conference on Computer Vision* (2009).

[59] Lankinen, J., and Kamarainen, J.-K. Local feature based unsupervised alignment of object class images. In *Proc. of British Machine Vision Conference* (2011).

[60] Lankinen, J., and Kämäräinen, J.-K. Video shot boundary detection using visual bag-of-words. In *International Conference on Computer Vision Theory and Applications* (Barcelona, Spain, 2013).

[61] Lankinen, J., Kangas, V., and Kamarainen, J.-K. A comparison of local feature detectors and descriptors for visual object categorization by intra-class repeatability and matching. In *Proc. of International Conference on Pattern Recognition* (2012).

[62] Laptev, I. On space-time interest points. *International Journal of Computer Vision 64*, 2-3 (Sept. 2005), 107–123.

[63] Lazebnik, S., Schmid, C., and Ponce, J. Beyond bags of features: Spatial pyramid matching for recognizing natural scene categories. In *Proc. of Computer Vision and Pattern Recognition* (2006).

[64] Learned-Miller, E. Data driven image models through continuous joint alignment. *IEEE Transactions on Pattern Analysis and Machine Intelligence 28*, 2 (2006), 236–250.

[65] Leibe, B., Ettlin, A., and Schiele, B. Learning semantic object parts for object categorization. *International Journal of Image and Vision Computing 26*, 1 (2008), 15–26.

[66] Li, J., Ding, Y., Shi, Y., and Li, W. A divide-and-rule scheme for shot boundary detection based on SIFT. *International Journal of Digital Content Technology and Its Applications 4*, 3 (2010).

[67] Lindeberg, T. Feature detection and ridge detection with automatic scale selection. *International Journal of Computer Vision 30*, 2 (1998), 79–116.

[68] Lowe, D. Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision 60*, 2 (2004), 91–110.

[69] Lowe, D. G. Object recognition from local scale-invariant features. In *Proceedings of the International Conference on Computer Vision* (1999), pp. 1150–1157.

[70] Lu, T., and Suganthan, P. N. An accumulation algorithm for video shot boundary detection. *Multimedia Tools and Applications 22* (January 2004), 89–106.

[71] Luo, Y., Xue, P., and Tian, Q. Scene alignment by sift flow for video summarization. In *Proc. of International Conference on Information, Communications and Signal Processing* (Piscataway, NJ, USA, 2009), ICICS'09, IEEE Press, pp. 1117–1121.

[72] Ma, Y.-F., Lu, L., Zhang, H.-J., and Li, M. A user attention model for video summarization. In *Proc. of ACM International Conference on Multimedia* (2003), pp. 533–542.

[73] Mas, J., and Fernandez, G. Video shot boundary detection based on color histogram. In *TRECVid Workshop* (2003).

[74] Matas, J., Chum, O., Martin, U., and Pajdla, T. Robust wide baseline stereo from maximally stable extremal regions. In *Proc. of British Machine Vision Conference* (2002), pp. 384–393.

[75] Matas., J., Chum, O., Urban, M., and Padja, T. Robust wide baseline stereo from maximally stable extremal regions. In *Proc. of British Machine Vision Conference* (2002).

[76] M.C.Burl, M.Weber, and P.Perona. A probabilistic approach to object recognition using local photometry and global geometry. In *Proc. of European Conference on Computer Vision* (1998).

[77] Mei, T., Hua, X.-S., Zhu, C.-Z., Zhou, H.-Q., and Li, S. Home video visual quality assessment with spatiotemporal factors. *IEEE Transactions on Circuits Systems for Video Technology 17*, 6 (2007), 699–706.

[78] Mikolajczyk, K. Featurespace: Feature detectors and descriptors: The state of the art and beyond. `http://www.featurespace.org/`, 2010.

[79] Mikolajczyk, K., Leibe, B., and Schiele, B. Local features for object class recognition. In *Proc. of Computer Vision and Pattern Recognition* (2005).

[80] Mikolajczyk, K., and Schmid, C. An affine invariant interest point detector. In *In Proceedings of the 7th European Conference on Computer Vision* (2002), pp. 0–7.

[81] Mikolajczyk, K., and Schmid, C. Scale and affine invariant interest point detectors. *International Journal of Computer Vision 1*, 60 (2004), 63–86.

[82] Mikolajczyk, K., and Schmid, C. A performance evaluation of local descriptors. *IEEE Transactions on Pattern Analysis and Machine Intelligence 27*, 10 (2005), 1615–1630.

[83] Mikolajczyk, K., Tuytelaars, T., Schmid, C., Zisserman, A., Matas, J., Schaffalitzky, F., Kadir, T., and Gool, L. V. A comparison of affine region detectors. *International Journal of Computer Vision 65*, 1-2 (2005).

[84] Miller, E., Matsakis, N., and Viola, P. Learning from one example through shared densities of transforms. In *Proc. of Computer Vision and Pattern Recognition* (2000).

[85] Munsell, B., Temlyakov, A., Styner, M., and Wang, S. Pre-organizing shape instances for landmark-based shape correspondence. *International Journal of Computer Vision 97* (2012), 210–228.

[86] Neubeck, A., and Gool, L. V. Efficient non-maximum suppression. In *Proc. of International Conference on Pattern Recognition* (1998), pp. 230–235.

[87] NistÃľr, D., and StewÃľnius, H. Linear time maximally stable extremal regions. In *Proc. of European Conference on Computer Vision* (2008), pp. 183–196.

[88] Nowak, E., Jurie, F., and Triggs, B. Sampling strategies for bag-of-features image classification. In *Proc. of European Conference on Computer Vision* (2006).

[89] Over, P., Smeaton, A. F., and Awad, G. The trecvid 2008 bbc rushes summarization evaluation. In *Proc. of ACM TRECVid Video Summarization Workshop* (New York, NY, USA, 2008), TVS '08, ACM, pp. 1–20.

[90] Peng, W.-T., Huang, W.-J., Chu, W.-T., Chou, C.-N., Chang, W.-Y., Chang, C.-H., and Hung, Y.-P. A user experience model for home video summarization. In *Proc. of International Multimedia Modeling Conference on Advances in Multimedia Modeling* (Berlin, Heidelberg, 2008), MMM '09, Springer-Verlag, pp. 484–495.

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

[92] Ponce, J., Berg, T., Everingham, M., Forsyth, D., Hebert, M., Lazebnik, S., Marszalek, M., Schmid, C., Russell, B., Torralba, A., Williams, C., Zhang, J., and Zisserman, A. Dataset issues in object recognition. In *Workshop on Category Level Object Recognition* (2006), pp. 29–48.

[93] Pritch, Y., Rav-Acha, A., and Peleg, S. Nonchronological video synopsis and indexing. *IEEE Transactions on Pattern Analysis and Machine Intelligence 30*, 11 (2008), 1971–1984.

[94] Pruteanu-Malinici, I., and Carin, L. Infinite Hidden Markov Models for Unusual-Event Detection in Video. *IEEE Transactions on Image Processing 17*, 5 (2008), 811–822.

[95] Riabchenko, E., Lankinen, J., Buch, A., Kämäräinen, J.-K., and Krueger, N. Supervised object class colour normalisation. In *Proc. of Scandinavian Conference on Image Analysis* (Espoo, Finland, 2013).

[96] Rosenholtz, R., Li, Y., Mansfield, J., and Jin, Z. Feature congestion: A measure of display clutter. In *Proc. of ACM SIGCHI Conference on Human Factors in Computing Systems* (2005), pp. 761–770.

[97] Rosten, E., and Drummond, T. Machine learning for high-speed corner detection. In *Proc. of European Conference on Computer Vision* (2006), pp. 430–443.

[98] Rublee, E., Rabaud, V., Konolige, K., and Bradski, G. Orb: An efficient alternative to sift or surf. *Proc. of International Conference on Computer Vision 0* (2011), 2564–2571.

[99] Ruiloba, R., March, S., and Quenot, G. Towards a standard protocol for the evaluation of video-to-shots segmentation algorithms. In *Proc. of European Workshop on Content-Based Multimedia Indexing* (1999).

[100] Sakarya, U., and Telatar, Z. Graph-based multilevel temporal segmentation of scripted content videos. In *Proc. of International Conference on Graph-based Representations in Pattern Recognition* (Berlin, Heidelberg, 2007), GbRPR'07, Springer-Verlag, pp. 168–179.

[101] Se, S., Lowe, D., and Little, J. Global localization using distinctive visual features. In *Int'l Conf. of Intelligent Robots and Systems* (2002), pp. 226–231.

[102] Silpa-Anan, C., and Hartley, R. Optimised KD-trees for fast image descriptor matching. In *Proc. of Computer Vision and Pattern Recognition* (2008).

[103] Sivic, J., and Zisserman, A. Video Google: A text retrieval approach to object matching in videos. In *Proc. of International Conference on Computer Vision* (2003).

[104] Smeaton, A. F., Over, P., and Doherty, A. Video shot boundary detection: Seven years of TRECVid activity. *International Journal of Computer Vision and Image Understanding 114* (2010), 411–418.

[105] Smeaton, A. F., Over, P., and Kraaij, W. Evaluation campaigns and trecvid. In *Proceedings of ACM International Workshop on Multimedia Information Retrieval* (New York, NY, USA, 2006), MIR '06, ACM, pp. 321–330.

[106] Tahaghoghi, S., Williams, H., Thom, J., and Volkmer, T. Video cut detection using frame windows. In *Proc. on Australasian Computer Science Conference* (2005).

[107] Todorovic, S., and Ahuja, N. Unsupervised category modeling, recognition, and segmentation in images. *IEEE Transactions on Pattern Analysis and Machine Intelligence* (2008).

[108] Tola, E., Lepetit, V., and Fua, P. DAISY: An Efficient Dense Descriptor Applied to Wide Baseline Stereo. *IEEE Transactions on Pattern Analysis and Machine Intelligence 32*, 5 (May 2010), 815–830.

[109] Truong, B., and Venkatesh, S. Video abstraction: A systematic review and classification. *International Journal on Multimedia Computing, Communications and Applications 3*, 1 (2007).

[110] T.Tuytelaars. Dense interest points. In *Proc. of Computer Vision and Pattern Recognition* (2010).

[111] Tuytelaars, T., and Gool, L. V. Wide baseline stereo matching based on local, affinely invariant regions. In *Proc. of British Machine Vision Conference* (2000).

[112] Tuytelaars, T., Lampert, C., Blaschko, M., and Buntine, W. Unsupervised object discovery: A comparison. *International Journal of Computer Vision 88*, 2 (2010).

[113] Tuytelaars, T., and Mikolajczyk, K. Local invariant feature detectors: A survey. *Foundations and Trends in Computer Graphics and Vision 3*, 3 (2008), 177–280.

[114] Tuytelaars, T., and Schmid, C. Vector quantizing feature space with a regular lattice. In *Proc. of International Conference on Computer Vision* (2007).

[115] Tuytelaars, T., and van Gool, L. Matching widely separated views based on affine invariant regions. *International Journal of Computer Vision 1*, 59 (2004).

[116] Umeyama, S. Least-squares estimation of transformation parameters between two point patterns. *IEEE Transactions on Pattern Analysis and Machine Intelligence 13*, 4 (1991), 376–380.

[117] van de Sande, K., Gevers, T., and Snoek, C. Evaluation of color descriptors for object and scene recognition. In *Proc. of International Conference on Computer Vision* (2008).

[118] van de Sande, K. E. A., Gevers, T., and Snoek, C. G. M. Evaluating color descriptors for object and scene recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence 32*, 9 (2010), 1582–1596.

[119] Vedaldi, A., and Fulkerson, B. VLFeat: An open and portable library of computer vision algorithms. `http://www.vlfeat.org`.

[120] Vedaldi, A., and Soatto, S. A complexity-distortion approach to joint pattern alignment.

[121] Velivelli, A., wah Ngo, C., and Huang, T. S. Detection of documentary scene changes by audio-visual fusion. In *Proc. of International Conference on Image and Video Retrieval* (2004), pp. 227–237.

[122] Viola, P., and Jones, M. Robust real-time feace detection. *International Journal of Computer Vision 57*, 2 (2004), 137–154.

[123] Visual Geometry Group, Katholieke Universiteit Leuven, I. R.-A., and the Center for Machine Perception. Affine covariant region detectors and descriptors. `http://www.robots.ox.ac.uk/~vgg/research/affine/`.

[124] Volkmer, T., Tahaghoghi, S., and Williams, H. Gradual transition detection using average frame similarity. In *CVPR Workshop on Multimedia Data and Document Engineering* (2004).

[125] Wang, L., Liu, X., Lin, S., Xu, G., and yeung Shum, H. Generic slow-motion replay detection in sports video. *Proc. of IEEE International Conference on Image Processing 2004* (2004), 1585–1588.

[126] Witkin, A. P. Scale-space filtering. In *International Joint Conference on Artificial Intelligence* (1983), pp. 1019–1022.

[127] Wu, P. A semi-automatic approach to detect highlights for home video annotation. In *Proc. of International Conference on Acoustics, Speech, and Signal Processing* (2004), pp. 957–960.

[128] Xiong, Z., Radhakrishnan, R., and Divakaran, A. Generation of sports highlights using motion activity in combination with a common audio feature extraction framework. In *Proc. of IEEE International Conference on Image Processing* (2003), pp. 5–8.

[129] Yu, B., Ma, W.-Y., Nahrstedt, K., and Zhang, H.-J. Video summarization based on user log enhanced link analysis. In *Proc. of ACM International Conference on Multimedia* (New York, NY, USA, 2003), MULTIMEDIA '03, ACM, pp. 382–391.

[130] Zhang, J., Marszalek, M., Lazebnik, S., and Schmid, C. Local features and kernels for classification of texture and object categories: A comprehensive study. *International Journal of Computer Vision 73*, 2 (2007), 213–238.

[131] Zhao, W. LIP-VIREO local interest point extraction toolkit. `http://vireo.cs.cityu.edu.hk`.

[132] Zhenhua Wang, B. F., and Wu, F. Local intensity order pattern for feature description. In *Proc. of International Conference on Computer Vision* (2011), pp. 603–610.

[133] Zhu, S., and Liu, Y. Automatic scene detection for advanced story retrieval. *International Journal on Expert Systems With Applications 36* (April 2009), 5976–5986.