Doctoral Thesis

Czech
Technical
University
in Prague

F3

Faculty of Electrical Engineering
Department of Cybernetics

Relevant Shot Detection with Local Features for Video Re-ranking

Javier Alejandro Aldana Iuit

Supervisor: Prof. Jiři Matas, Ph.D.
Supervisor–specialist: Assoc. Prof. Ondřej Chum, Ph.D.
Ph.D. Programme: Electrical Engineering and Information Technology
Field of study: Artificial Intelligence and Biocybernetics
Prague, October 2019

Acknowledgements

I am grateful to my family, my mother Genny, my father Javier and my siblings, to all of them. For the endless support and long distance cheerleading. Nothing would be possible without their love and understanding.

Thanks to all my friends who help my in the hard journey of the PhD. Your advises as colleagues and friends were extremely useful and energizing.

With a special mention to Jiři Matas and Ondřej Chum, my advisors. Thanks for all the patience you had for me, and let me learn a lot from you.

I want to sincerely thanks to my friend (and colleague) Dmytro Mishkin for the invaluable support and precise help when it was most required.

Thanks a lot to the staff of the Center for Machine Perception (CMP) providing me with all the help I could need during my long stay at the CMP.

A very special gratitude goes to the Consejo Nacional de Ciencia y Tecnología (CONCYTEY-CONACYT) for granting me with the PhD scholarship 216786 and the Czech Technical University student grant SGS15/155/OHK3/2T/13 for providing the funding for the work.

Finally, to all those special persons who were part of this very long journey, not only academic, more over a life experience.

Abstract

This thesis addresses the problem of retrieving videos that contain a specific object. To avoid indexing visual content of a fixed corpus of videos, a two-stage approach is adopted. First, a short-list of videos is obtained from a video sharing site, then, the short-list is re-ranked based on the visual content. In particular, given a name or phrase specifying an object, images and videos are collected from the Internet possibly depicting the object using a textual query on their name or annotation. Each video is divided into shots. Novel shot detector based on fast wide-baseline matching is proposed. A visual model of the object is constructed from the images. Video frames of each shot are efficiently matched to the visual model through local image features. Shot relevance is defined as the duration of the visibility of the object of interest.

To evaluate the video re-ranking task, a novel fully annotated multimedia dataset, called Specific Object Search dataset, is introduced. The dataset contains videos and images of 10 specific objects such as building landmarks, art paintings, architectonic monuments, etc. Additionally, confuser videos are collected. These videos do not contain the query object but are returned by video-sharing server when queried by the object identifier string. All videos contain frame-based annotation. The implementation of the proposed method runs at 208 frames per second. Averaged over the ten landmarks, it achieves the 0.95 recall, 0.65 mean precision 0.65, and mean Average Precision of 0.92.

In multiple stages of the approach, local features are exploited. Extraction and matching of those features is the most time consuming step of the whole pipeline. A novel similarity-covariant feature detector that extracts points whose neighborhoods, when treated as a 3D intensity surface, have a saddle-like intensity profile is proposed. The saddle condition is verified efficiently by intensity comparisons on two concentric rings that must have exactly two dark-to-bright and two bright-to-dark transitions satisfying certain geometric constraints. Saddle is a fast approximation of Hessian detector as ORB, that implements the FAST detector, is for Harris detector. Novel matching strategy called the first geometric inconsistent with binary descriptors is proposed. This matching strategy is suitable for our feature detector, including experiments with fix point descriptors hand-crafted and learned.

Experiments show that the Saddle features are general, evenly spread and appearing in high density in a range of images. The Saddle detector is among the fastest proposed. In comparison with detector with similar speed, the Saddle features show superior matching performance on number of challenging datasets. Compared to recently proposed deep-learning based interest point detectors and popular hand-crafted keypoint detectors, evaluated for repeatability in the Apollo Scape dataset, the Saddle detectors shows the best performance in most of the street-level view sequences a.k.a. traversals.

Keywords: Video re-ranking, relevant shot detection, shot boundary detection, object specific search, interest points, fast detectors, image matching.

Supervisor: Prof. Jiři Matas, Ph.D.
Karlovo namesti 13,
121 35, Prague 2.
Prague, Czech Republic.

Abstrakt

Práce se zabývá problémem vyhledávání videí obsahujících konkrétní objekt. Abychom se vyhnuli indexování vizuálního obsahu neměnného korpus videí, používáme dvoufázový přístup. Nejprve je získán krátký seznam videí z webu pro sdílení videí, a tento seznam je poté seřazen podle vizuálního obsahu. Podle názvu či fráze specifikující objekt se z internetu získají obrázky a videa pravděpodobně znázorňující objekt, a to na základě textového dotazu na jejich název či anotaci. Každé video je rozděleno na jednotlivé záběry. Práce navrhuje nový detektor záběrů založený na wide-baseline matching. Z obrázků je sestrojen vizuální model objektu. Snímky videa z každého záběru jsou efektivně porovnány s vizuálním modelem pomocí lokálních příznaků obrazu. Relevantnost záběru je pak definována jako doba viditelnosti objektu zájmu.

K vyhodnocení úlohy řazení videí představujeme novou, plně anotovanou, datovou sadu: Specific Object Search dataset. Datová sada obsahuje videa a obrázky 10 konkrétních objektů jako významné budovy, malby, architektonické památky, atd. Součástí sady jsou i "matoucí"videa, která neobsahují dotazovaný objekt, ale byla nalezena na serveru pro sdílení videí podle textového popisu objektu. Všechna videa obsahují anotaci jednotlivých snímků. Implementace navržené metody zpracuje 208 snímků za sekundu. Průměrně (přes všech 10 objektů) dosahuje 0.95 úplnost, 0.65 mean přesnost a 0.92 průměrný přesnost.

V několika úrovních navržené metody se využívá lokálních příznaků. Extrakce těchto příznaků a hledání jejich korespondencí je časově nejnáročnějším krokem celé metody. Práce navrhuje nový detektor podobnostně-kovariantních příznaků, který extrahuje body, jejichž okolí - vnímané jako 3D intenzitní profil - má sedlový profil intenzity. Tato sedlová podmínka je efektivně ověřována porovnáváním intenzit dvou soustředných kruhů, které musí mít přesně dva přechody tmavá->světlá a světlá->tmavá splňující určitá geometrická omezení. Saddle je rychlá aproximace Hessian detektoru, podobně jako je ORB (která implementuje FAST detektor) pro Harris detektor. Práce navrhuje novou strategii hledání korespondencí zvanou První geometricky nekonzistentní pomocí binárních descriptorů. Tato strategie je vhodná pro náš detektor příznaků, včetně experimentů s ručně navrženými a naučenými deskriptory fixních bodů.

Experimenty ukazují, že Saddle příznaky jsou obecné, rovnoměrně rozšířené a objevující se ve vysoké hustotě v řadě obrázků. Saddle detektor se řadí mezi nejrychlejší. V porovnání s detektory srovnatelné rychlosti ukazují příznaky Saddle lepší výsledky v hledání korespondencí na několika náročných datových sadách. Ve srovnání s nedávno navrženými detektory významných bodů založenými na hlubokém učení a s populárními detektory ručně navržených příznaků, které bylo vyhodnoceno na Appolo Scape datasetu, dosahuje Saddle detektor nejlepších výsledků na většině uličních sekvencí (známých jako "traversals").

Klíčová slova: Přeřazování videí, detekce relevantních záběrů, detekce hranic záběrů, vyhledávání specifických objektů, body zájmu, rychlé detektory, srovnávání obrázků.

Contents

Figures

Tables

Chapter 1

Introduction

1.1 Motivation

The multimedia content shared over the Internet is growing very fast in last years. According to YouTube statistics [209], over 6 billion hours of video are watched each month and 100 hours are uploaded every minute in 2017. Sharing photos is the most popular activity on the social networks, for instance, according to *Instagram* 60 million photos are uploaded daily to its servers [69] in 2014. *Flickr* reports 1.6 millions of new photos every day [114] in 2014. These numbers have awakened the interest of the scientific community to design efficient ways of querying web-scale image and video databases.

The problem of searching multimedia documents in large databases is called *retrieval* or *indexing*. In image/video retrieval *content-based* methods analyze the contents of the images rather than the metadata (keywords, tags, or descriptions) associated with them. In this context, the image *content* involves colors, shapes, textures, or any other information that can be derived from the pixels. Even with manually annotated images, the search can be very time-consuming and the annotation may not capture the desired description of the image. The evaluation of the effectiveness of keyword image search is subjective and has not been well-defined [46].

Popular video search engines like YouTube, *Dailymotion*, *Yahoo Screen*, *Bing video*, *AOL video*, *eHOW*, *MeFeedia*, *360daily*, *Veoh*, *Vimeo*, *Vevo* among others, do not perform content-based search to query the videos. The input of the search is text, hence full-text search is performed over the filename, description inserted by the owner of the video, and the comments written by other users of the website. Engines like YouTube use different factors to categorize and rank the videos. Typically, videos are ranked by the *on-page* and *off-page* Search Engine Optimization (SEO) factors. By SEO factors, the website and its content are designed to be highly relevant for both search engines and searchers. Basically, websites are indexed for the appropriate keywords, as well as marketing-focused tasks [47]. In addition to the number of links that reference the video and the video's age, YouTube also weights the video scores based on the number of views, rating, comments, and more [83]. Therefore, there is no guarantee that videos in the top of the list depict the object of interest and, moreover, the list may not be sorted so that the scores reflect the actual interest and number and length of relevant shots contained in the videos.

Indeed, Wikipedia, Google Images, Flickr, among others, are strong candidates to be source of data to link text with images. Such databases are coarse and diverse due to their public domain. As an example, Wikipedia is a free and open on-line encyclopedia which in its English version contains 5.8 millions of articles in 2019. The documents are created and maintained through the collaborative effort of a community of users [197]. However, the documents in these databases may not be reliable.

The documents do not include references to make the authors publicly accountable for the contributions they did to the corresponding articles. Moreover, any user can edit any article independently of his/her expertise. In some categories of the encyclopedia, the subjectivity in the item descriptions (images, tables, etc.) turns data selection, to ensemble a dataset, into a very complex task. Such task requires clever file filters to avoid the noisiness of the documents and their sections, to finally construct a correctly annotated dataset of images and text suitable to train machine learning classifiers.

1.2 Problem definition

In this thesis we address the problem of retrieving videos that are relevant with respect to an object of interest specified by the user. The query consists in a textual identification of the object (name or short description) and a set of images that potentially contain different views of the same object. The use case of our pipeline is intended to be real-life applications, which implies that latency is a major constraint with low tolerance.

1.3 Proposed method

In our proposed method, we are not interested in indexing a large corpus of videos because, at least for us, it is not realistic to index all video content. Instead, we relay on the short-list retrieved by text-based search engines and massive storage capabilities of video sharing web sites e.g. YouTube. The text search allows to retrieve documents with low reliability due to noisy labelling or annotation done by the contributors to the corpus. An efficient visual content-based matching is applied to verify and re-rank the initial short-list. This document focuses on the object model building from a set of images and on efficient on-line detection of the object in videos. The method is summarized in Fig. 1.1.

Our work proposes to access the visual content of the videos to improve the quality of the text-based search, avoiding the subjectivity and noisiness of textual human annotations. The proposed approach works on top of the list of retrieved files returned by the video search engine of the video sharing web page. There is a additional scoring step of the videos with a relevance metric that allows to re-rank the videos in the list. The relevance assessment involves an efficient detection of the query object specified by the user. The applicability of the proposed method ranges from individual user searches for relevant videos to systematic augmentation of Wikipedia (or similar) pages with relevant video documents. This proposal relies on the massive storage facilities of third-party video servers, it does not require to download, process and store all publicly available videos.
The processing of the images and videos in our approach is done *on the fly*. There is no allocation of preprocessed data and/or the videos themselves, since the demand of storage capabilities becomes intractable very fast. Our approach relies on the infrastructure of big multimedia sharing companies to deal and support with this kind of issues.

The thesis provides complete pipeline for content-based video re-ranking multiple components with independent tasks and specific performance constrains - speed, memory footprint, and complexity, for example. In addition, we deal with the fact that processing must be done *on the fly*, due to limitations on storing preprocessed data. These factors lead us to use and propose efficient algorithms with optimized implementations in the multiple components of the pipeline.

The method was implemented, weaknesses were identified and addressed. The core contribution - Saddle - has impact on various problems in computer vision [74, 104, 213, 175, 37], by far exceeding the video re-ranking task.

Visual representation of the query requires images used as visual description of the search object to be related semantically by textual information. This assumption

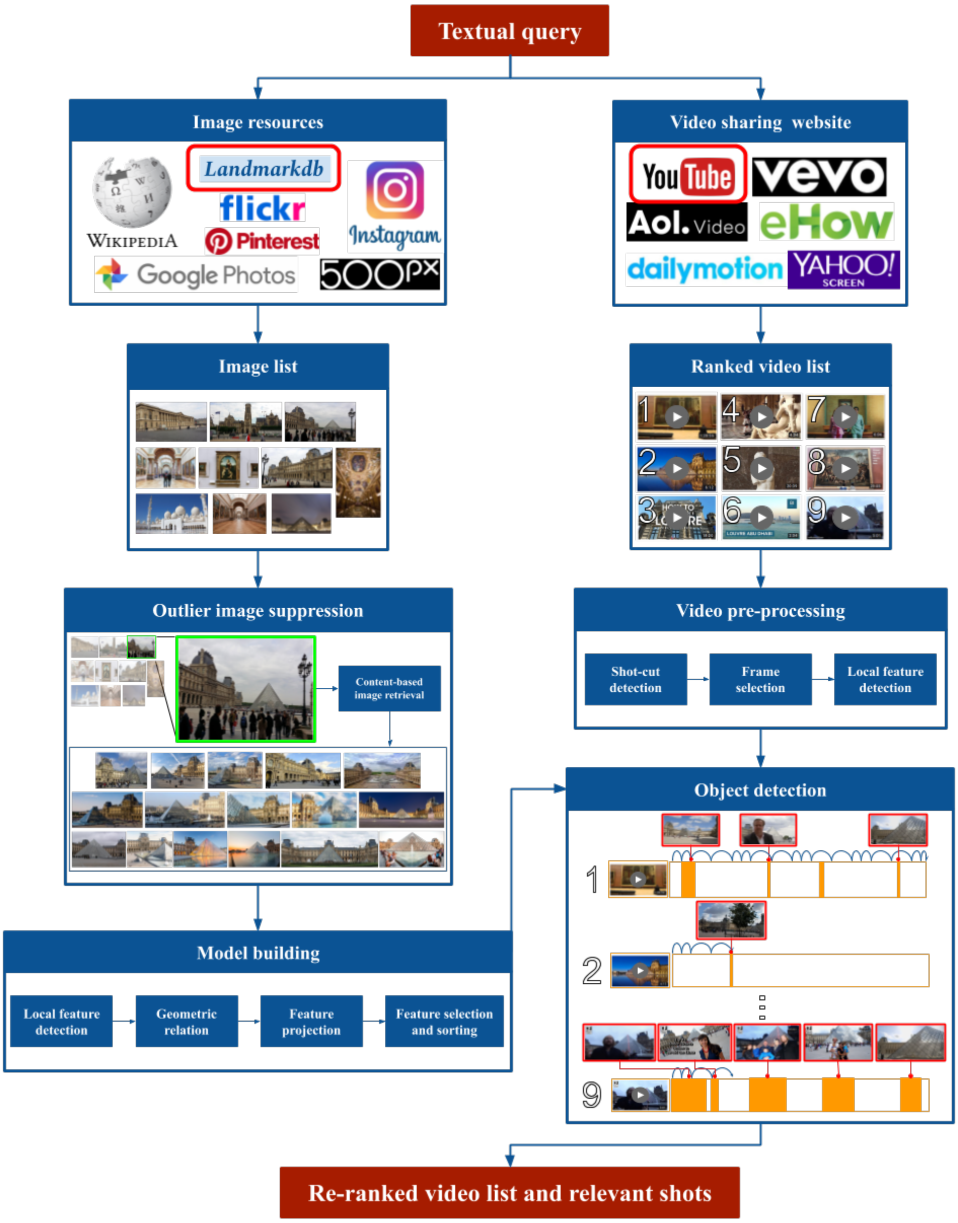


Figure 1.1: Block diagram of the workflow of our approach for relevant shot detection and video re-ranking based on visual content. The left column shows the construction of the visual models and the right column shows the modules related to the video processing, relevant shot detection and content-based video re-ranking. Our approach uses *Landmarkdb* and *YouTube* as image and video sources, respectively, marked with red rounded rectangles.

allows us to select a subset of images from a large collection to set up a model or query. The text linked with the selected images indicates that they potentially depict the object, building or scene of interest, i.e. the query of our search. Even though some stages of the pipeline check the actual content of the images before including them in the model construction step, it is important that the textual information has some degree of confidence in order to not run expensive and unnecessary checks on images that are out of the context, i.e. there is no relation with the object of interest. The desirable scenario is that all pictures in the selected set are linked to the name of the landmark shown in them.

Landmarkdb dataset [72] is image source for the visual models. The dataset was collected from public domain sites for knowledge contributors by means of a concise entity definition, presented as a list of constrains, for document classification. A broader description and details about the Landmarkdb dataset is presented in Sec. 4.2. Once the text-image dataset is collected, our proposed solution performs a text search of the query string to obtain a set of images presumable depicting the corresponding object or landmarks. Later local image features are extracted from the image set and post-processed to efficiently search the model along the video sequences. The final result of the approach is the initial list of videos is sorted with respect to the relevance of the videos.

1.4 Evaluation

In order to measure the performance of video re-ranking with a scoring strategy by relevance assessment, we require an *ad hoc* fully annotated dataset with images and videos of specific objects of interest. Currently there is no publicly available dataset to benchmark searching algorithms for specific object. Some popular datasets are designed to test categorical classification/detection of objects or activities [85, 142]. As a contribution of this work we present our *Specific Object Search* (SOS) database that, to the best of our knowledge, is the first one with textual labels, images and videos of specific objects. Fig. 1.2 shows four objects included in the SOS benchmark.
The dataset contains frame-based annotations and a set of images with multiple views of the object. The data collected corresponds to 10 specific objects among building landmarks, art paintings, architectonic monuments, etc. Additionally, there are confuser videos retrieved querying the same identifier string to the video server but they do not contain the object in none of the frames. For more details about SOS dataset see Sec. 4.9.1. Fig 1.3 presents a few keyframes from two videos of the SOS benchmark. The text-based search engine of YouTube retrieves both videos with query *Notre Dame*, however, the distractor does not contain the query in any frame.

To test specifically the performance of the new proposed feature detector, we evaluate on standard benchmarks listed in Tab. 3.1.

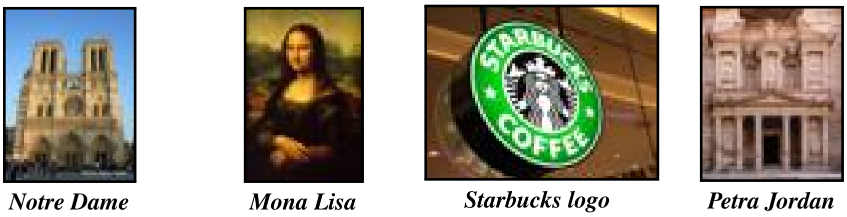


Figure 1.2: Examples of objects included in the *Specific Object Search* benchmark. The names of the objects/queries are at the bottom of each image.

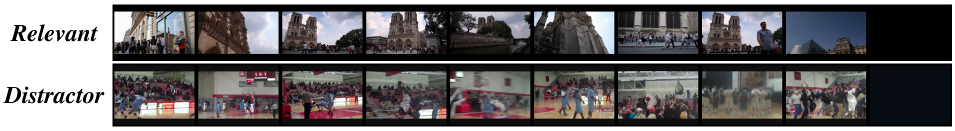


Figure 1.3: Examples of relevant and distractor videos from the *Specific Object Search* benchmark for the query *Notre Dame*. Both videos are retrieved by YouTube with the same textual query. The distractor does not depict the query even it is relevant with respect to text-based search. Only a few keyframes are shown.

1.5 Publications

The list of publications produced during the PhD project, the directly related to the dissertation topic and additional ones, are listed below:

1.5.1 Publications related to the dissertation topic

- "***Saddle: Fast and repeatable features with good coverage***". Javier Aldana-Iuit, Dmytro Mishkin, Ondrej Chum and Jiri Matas. Image and Vision Computing (IMAVIS), Elsevier journal. Accepted date: 20.08.2019.
- "***In the Saddle: Chasing Fast and Repeatable Features***". Javier Aldana-Iuit, Dmytro Mishkin, Ondrej Chum and Jiri Matas. Proc. 23rd International Conference on Pattern Recognition (ICPR). 2016.
- "***Relevance Assessment for Visual Video Re-Ranking***". Javier Aldana-Iuit, Ondrej Chum and Jiri Matas. Proc. 11th International Conference On Image Analysis and Recognition (ICIAR). 2014.
- "***Wide-baseline Stereo Matching for object detection on videos***" Javier Aldana-Iuit. Proc. International Student Conference on Electrical Engineering (POSTER). 2014. Best poster and presentation award.

1.5.2 Other publications

- "*MaxNet: Neural network architecture for continuous detection of malicious activity*". Gronát, Javier Aldana-Iuit and Martin Bálek. Proc. 40th IEEE Symposium on Security and Privacy. 2019.

1.6 Thesis contributions

In particular, the main contributions of this thesis are as follows:

Re-ranking approach to improve retrieval of video sharing websites We present a scoring approach to rank full videos by relevance with respect to a query. A given query is encoded with visual information that actually improves the results of retrieval compared to full-text search based methods. Currently, video search in video sharing websites is based on full-text search with high number of false positives caused by poor quality in the textual annotation. Even that video retrieval approaches index databases with robust and compact spatiotemporal descriptors the processing is slow and requires preprocessing steps prohibited for real-time applications. Our proposal is an in-line approach free of precomputation.

We present a modular architecture of the relevant assessment for video re-ranking, that provides flexibility and enables with a set of state of the art detectors and descriptors for local image regions. Performance metrics like speed, precision and recall are dependent on the setting of the matcher and the query modeling approach. This architecture is convenient for scalability since additional modules can be added to improve the performance without adapting or modifying the whole pipeline.

We introduce a web-interface that provides an on-line service for querying the YouTube text-based search engine and boosts the results by our visual relevant assessment. The system is provided with a video player, ranked list of the retrieved videos with respect to the visual content, time markers located at relevant shots, and metadata of the search performed by the video server. The time stamps allow fast access to shots that contains similar frames that are encoded in the query model. This is the first web service allowing fast access to shot under the specification of the target object (query).

To the best of our knowledge, this is the first *relevant shot detection* approach for fast specific object detection and visual content-based video re-ranking. Even though similar approaches are reported in the literature for automatic creation of dataset for video activity classification dataset, this is the first attempt to detect relevant shots to increase the accuracy of retrieval engines that are not aware of the visual content of the documents, i.e. videos. In addition, the Saddle detector not only speeds up the feature detection step of our pipeline, it also works well in medical images, specifically in retinal images [155, 31], and it gets state-of-the-art results for street level view video sequences for autonomous driving vehicles [67]. For more details see Section 2.5. More details about this contributions are in Cha. 4.

A state of the art feature detector in multiple vision tasks The evaluation of our video re-ranking by relevance assessment reveals the bottleneck of the processing pipeline is the keypoints detection in images (also video frames). In order to speed up the detection without hurting the performance of the system, we propose the Saddle detector which is ranked with the best performance in a stability evaluation of interest point detectors on a street-level view dataset [67]. In addition, the experiments presented in this thesis show that Saddle has significantly better repeatability, precision in location, coverage that despicable faster detectors, comparable or even better performance than more complex detectors including learned approaches.

In the literature are some reports of authors finding applications where Saddle performs better than traditional or standard methods. The domain of medical images is not explore with enough extent in this thesis, nevertheless, in Section 2.5 we present publications about Saddle used in retinal images. In the same section, we introduce a publication concerning a challenging dataset proposed to compare vision systems for autonomous driving vehicles. Saddle poses the state of the art performance for given task. More details about our feature detector are in Cha. 3.

A multimedia dataset for specific object search The evaluation of the re-ranking approach requires a dataset with specific characteristics like images and videos for object specific queries with fully annotated frames. Such dataset was not available publicly therefore we designed and collected a novel benchmark for specific object detection with a challenging multimedia dataset with images and videos of 10 objects that go from brand logos to landmark buildings, with manually annotated labels. The dataset can be used in different computer vision tasks like wide-baseline stereo, feature learning, shot boundary detection, etc. The dataset is called *Specific Object Search* (SOS) benchmark. Sec. 4.9.1 presents a detailed description of the benchmark.

Shot boundary detector by wide-baseline stereo We present a video shot boundary detector which similarity metric is the number of correctly match pairs of features by wide baseline stereo matching. Detection of the boundaries is defined as a search problem with two phases, first, the *forward* phase finds shot discontinuities with a line search algorithm (typically used in optimization problems), and second, the *backwards* phase refines the estimate of the forward stage with a more accurate location in time of the boundary by binary search. Previous similar methods search exhaustively without optimizations approaches to increase the efficiency. Sec. 4.7 presents more details about our shot boundary detector.

A performance metric for spatial coverage of local features We present a novel metric for testing the coverage of local image feature detectors in the image. The metric is up to the number of invariants computed by the detector, i.e. it can be calculated with different geometries from similar to affine covariant features. The homogeneously spatial distribution is a desirable property for points used in 3D reconstruction and SLAM systems. Good coverage is a desirable feature of the detector in multiple vision task and this thesis introduces the first metric to measure it. See Sec. 3.3.2.

A novel matching strategy for binary descriptors We introduced a novel strategy for computing point in correspondence using binary descriptors. The main inspiration is taken from the strategy called First Geometric Inconsistent [133] for image matching with view synthesis with floating point descriptors. The experiments show that our strategy out performs the standard methods based on hard thresholding distance. This contribution is described in detail in Sec. 3.3.6.

1.7 Authorship

I hereby certify that the results presented in this thesis were achieved during my own research, in cooperation with my thesis advisor Jiři Matas and my specialist advisor Ondřej Chum, published in [7, 6, 4, 3], with Dmytro Mishkin, published in [7, 6], and with Petr Gronát and Martin Bálek, published in [56].

1.8 Structure of the Thesis

This thesis is organized as follows.

Chapter 2 presents an overview of the state of the art regarding topics that are involved in this dissertation. More precisely, the approach proposed in this thesis requires a wide range of methods and algorithms for different stages of the pipeline, therefore we present the methods that have the best performance reported in the literature. The literature review includes the following topics local image feature detectors and descriptors, methods for image registration, shot detection, image retrieval, video representation and classification, among others. All these topics are tightly related to our approach in specific modules of the processing workflow.

Chapter 3 introduces one of our main contributions to the state of the art, the Saddle detector. Saddle is a similarity covariant feature detector proposed as a fast approximation of the Hessian detector for points located where the Hessian of the image has negative determinant. An extensive set of experiments shows the properties, capabilities, and limitations of the detector. The performance of Saddle is demonstrated in multiple real applications in computer vision. The detector gets a significant speed-up of the process and it shows to have general applicability.

Chapter 4 introduces our method for solving the relevant shot detection and content-based video re-ranking problems, defined in Sec. 1.2. This chapter describes in details all the modules of our pipeline for improving the full-text search engine of a video sharing website by means of incorporating visual information in the query. The full set of modules comprises video acquisition from the video server, decoding and accessing to the keyframe classification, extraction and description of the local features detected. In addition, this chapter introduces 1) the workflow to build the query model, 2) the indexing of the images database to enable full-text search, 3) the implementation of our image retrieval engine with global deep features to suppress outliers from the set of relevant images, 4) the matching strategy between model and video, and 5) the metric for grading the relevance of the shots.

Chapter 5 concludes the thesis and outlines future work on the subject of video re-ranking and relevant assessment in video shots as well as on our *Saddle* feature detector.

Chapter 2

State of the art

Let us review the existing work related to the topic of this thesis in the fields of relevant shot detection, video classification, object recognition and local feature detectors. First, we provide some context via the overview of the structure of the chapter in the sake of clarity and justifying the content since the number of related fields is high.

The elementary units used to represent an image or a decoded video frame in our framework are the keypoints or interest local regions detectable in them, therefore in Section 2.1 we present an overview of some state-of-the-art feature detectors that are suitable for our applications. The performance of the relevant assessment is highly dependent on the selected detector due to some of them relay on complex and expensive numerical processes to improve the precision in locations and scale selection, or additional steps to increase the number of geometric invariants covered in the normalization of the image patch for description. As a follow-up of the detection, we discuss in Section 2.2 some descriptors used in different tasks in computer vision. This is another important factor that reverberates in the global performance of our pipeline, thus this step requires a careful selection of the algorithm, in the case of hand-crafted descriptors, or architectures and training strategies, in case of the learned descriptors. Since our method for identifying the object in the images is based on matching local features, we present possible variants of the image matching task in Section 2.3 where we also define the problem of image registration for different camera configurations. Later, in Section 2.4 we introduce the related work to our framework for selecting shots in video, the methods are previously reported and are references to show our contributions to the state-of-the-art in this area. Finally, in Section 2.5 we discuss some publications that are relevant to our proposed Saddle detector where the authors show additional domains for the detector to out performed standard and state-of-the-art approaches.

2.1 Keypoint detection

The task is to find sub-regions or local regions in the image that are useful for posterior stages of the processing pipeline. This regions are the primitives used by plenty approaches in computer vision such as object recognition, image registration, 3D reconstruction, etc.
This regions must have different appearance from surrounding (close) regions and allow robust detection under different nuisance factors. In the literature we can find multiple names for these entities, for instance, local image regions, local feature, interest points, keypoints, covariant regions, among others.

Another selectivity criterion of the point is repeatability, namely, the points must be detected in different images with viewpoint changes for the same scene, i.e. they must cover the same physical surface of the scene. A good interest point must be distinguishable at least from its immediate regions, i.e. the image patch (group of pixels in a section of the image) centered in the interest point must be dissimilar to neighboring patches [59].

The location of interest regions in the image coordinate frame corresponds to the location of the local maxima of hand-crafted functions that measure the distinctiveness of local regions related to a specific morphology in the 2D intensity space [59, 178, 170, 106]. Some morphologies have statistical and geometric properties that prove they are good regions to be selected as features for a given vision task. Examples of such regions are corners, blobs, ridges, ellipses and intensity saddle points. An overview of state-of-the-art feature detectors (hand-crafted and learned ones) is presented in Section 2.1. Fig. 2.1 shows some interest regions posed at local maxima and minima of the determinant of the Hessian (Eq. 2.1) with the elliptical shapes approximated by the Baumberg [17] iteration. These examples are intended to give a visualization of these entities that are distributed in the images and therefore used to represent it. The two images shown in Fig. 2.1 belong to a standard dataset [120] used for benchmarking image registration approaches. Fig. 2.1(a) is a target image to be matched with the reference one shown in Fig. 2.1(b). Both images are related by a geometric transformation that models the perspective change. The colored ellipses indicate some detected keypoints that are set in correspondence along the pair of views, that is to say, pair of points in the images that correspond to the same 3D point in the scene. The change in the viewpoint is modeled as an homography which is used to project the regions on the reference image into the coordinate frame of the target image and compute the accuracy of the detection by the average reprojection error (see Eq. 3.2). The close overlap of the shapes shows that the appearance of the regions changes with respect to the transformation, i.e. they are covariant with change of perspective. Due to the affine transformations being first order Taylor approximations of the homography at given points [35], locally the transformations are modeled with an affinity and the elliptical local regions are

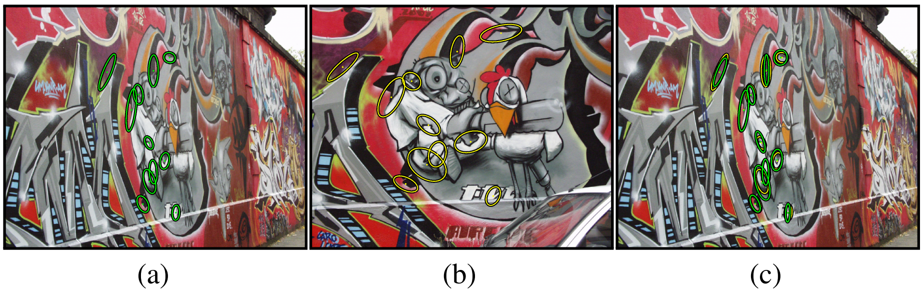


Figure 2.1: Local covariant features detected in two images of the *graffiti* sequence in the *OxAff* dataset[121, 120]. (a) is the target image with the shapes of some interest points in **green** and (b) is the reference image with local features in yellow. The image pair is related by a geometric transformation that modify the appearance of the local regions. In (c) the regions of the reference images are projected into the target image showing the tight overlap between shapes and the local affine covariant behavior with respect to the viewpoint change.

called *affine covariant regions* or *local affine frames*.

$$\mathbf{H} \simeq \mathbf{H}(\mathbf{x}, \sigma_D) = \begin{bmatrix} h_{11} & h_{12} \\ h_{21} & h_{22} \end{bmatrix} = \begin{bmatrix} I_{xx}(\mathbf{x}, \sigma_D) & I_{xy}(\mathbf{x}, \sigma_D) \\ I_{xy}(\mathbf{x}, \sigma_D) & I_{yy}(\mathbf{x}, \sigma_D) \end{bmatrix} \tag{2.1}$$

Once the distinctive points of the image are selected/detected, the neighboring pixels (known as patches) that surround the points are used to compute a signature or descriptor of the region. The descriptors are high-dimensional vectors that allow to recognize the same 3D point projected into another image by comparing descriptors in the Euclidean space. The approaches to compute the descriptors are very diverse nowadays (see Section 2.2 for a more detailed overview), from binary descriptors based on pair-wise questions over intensity pixels or patches [29, 98], minimizing objective functions to find embeddings that maximize the similarity between matching patches while minimize it for pairs of non-matching descriptors [189], tuning kernel descriptors [22, 139], deep convolutional neural network architectures [127, 184] and generative adversarial networks [214] to learn compact but discriminative representations either floating-point or binary. Overall, finding a high number of similar regions is a strong indicator that the object of interest appears in the image under test, assuming that we have a precomputed database of described regions belonging to the query.

2.1.1 Hand-crafted detectors

The traditional interest points detectors rely on hand-crafted features, i.e. discrete version of analytic functions with the intensity of the pixels as domain. Given function by construction has local maxima or fires in the location of desired structures captured in the image, as an example, one of the most popular feature detector is the Harris detector [59] which finds locations where the local autocorrelation function is changing in all directions. These regions correspond to corners and edges where the *corner response* is locally maximum. The corner response is defined as $R(x, y) = det(\mathbf{C}) - \alpha\ \mathrm{trace}^2(\mathbf{C})$ where $\mathbf{C}$ is the auto-correlation matrix based on the first directional derivative. Structures like blobs or saddle points correspond to local maxima and minima of the Hessian response [19], respectively. Therefore, the Hessian detector's response (as the name suggest) is the determinant of the Hessian matrix [121], shown in Eq. 2.1, its extrema are regions that have strong derivatives in two orthogonal directions. Again the position of the keypoints is refined by a non-maxima suppression step. The baseline version of the Harris and Hessian feature detectors work well for stereo matching and visual object tracking, since objects appears in the same size. Nevertheless, the size of the regions must be taken into account when the viewpoint changes. Objects, and consequently local regions, have different size, maybe caused by a zoom in or zoom out in the camera or because the image were digitally resized. The local effect caused by perspective changes will be discussed later.

Several local feature detector extend its capabilities to be scale invariant by means of the scale-space analysis [102]. Basically, the feature detectors are applied to multiple versions (scales) of the same image. The original image is blurred with a Gaussian kernel convolution, of gradually increased standard deviation, and down scaled allowing the detector to focus on small details (high resolution) and large structures (low resolution). For a single keypoint, the characteristic scale selection mechanism [103] consists in detecting the local maxima of the feature responses overs scales.

The Harris-Laplace and Hessian-Laplace detectors [117] are scale aware versions, i.e. the scale selection is performed applying the Laplace operator in the potential points selected by the baseline detectors along the multiple scales. The local maxima across the scales indicates the size of the features corresponding to the standard deviation of the Gaussian used for blurring.

The perspective changes caused by the camera pose is normalized by estimating the covariant shape of the interest points. For the purpose of normalized local regions and get affine invariant descriptors, some detectors have an additional step for estimating the elliptical affine shape of the region [105]. This step is know as the Baumberg iteration [17] which measures the affine shape of the region with the eigenvalues of the second moment matrix of the image. Basically, the task is to find a linear transformation that projects the affine pattern into a another sub-image where the eigenvalues are equal, such transformation is the square root of the second moment matrix $\mathbf{M}^{1/2}$. Harris-Affine [118] and Hessian-Affine [121, 124] are examples of this affine covariant detectors.

Other approaches to detect corners use sliding windows techniques [177], pixels laying inside a given circular mask with fixed radius are classified as similar or not to the pixel located at the center of the mask. Similar pixels have brightness close to the central one up to a threshold, therefore the relative area of similar pixels is computed and compared to the *geometric threshold*, typically set to $\frac{3}{4}N$ where N is the number of pixels in the image patch. This approach is slow since all pixels in the patch are compared against the central one.

One of the most popular method reported in the literature is SIFT (Scale Invariant Feature Transform) [106, 107]. SIFT is presented as an integral solution for object recognition and image matching providing a keypoint detector and a powerful descriptor robust against partial occlusions, rotations, scale and intensity changes, and shows good performance against moderate affine transformations [120]. See Sec. 2.2.1 for more details about this descriptor.
The detection step of SIFT finds local extrema of the Difference-of-Gaussian (DoG) as the response function in a 3D space-scale structure, efficiently computed as an image pyramid [107]. The DoG function $D(x, y, \sigma)$ can be computed subtracting two adjacent (nearby) scales which are separated by a factor k as shown in Eq. 2.2,

$$\begin{aligned} D(x, y, \sigma) &= (G(x, y, k\sigma) - G(x, y, \sigma)) * I(x, y) && (2.2) \\ &= L(x, y, k\sigma) - L(x, y, \sigma), && (2.3) \end{aligned}$$

where $L(x, y, \sigma)$ is the image smoothed by a Gaussian kernel $G(x, y, \sigma)$ with standard deviation σ.

Finally, a region is classified as interest point if the 3D point ($[x, y, \sigma]^T$) is a local maximum or minimum with respect to its 26-connected neighborhood. Likewise others detectors described before, the scale selection is refined by computing the Laplacian response along the σ dimension [27] and introducing sub-pixel and sub-scale precision by quadratic curve fitting. Some interest points are rejected from the final feature set based on response (DoG) and *corneress* (Harris) thresholding, dropping homogeneous regions and edges which are affected by the aperture problem [113].

All feature detectors presented so far are too slow for real-time applications due to their computational complexity, from building the image pyramids to non-maximal suppression and sub-pixel precision accuracy refinement. The limitation in the use case assumes that no GPU or distributed processing system is available. The later was the motivation for approximated method with efficient implementations that allows reaching real-time ($>$ 30 fps) processing. One of the most relevant alternatives to SIFT is the well known SURF (Speeded Up Robust Features) [18] detector which uses and approximation of the Hessian-Laplace detector by box filters of the second-order partial derivative required to compute the Hessian response with I_{xx}, I_{yy} and I_{xy} (see Eq. 2.1). The significant speed up arises by using integral images to compute the box filters efficiently and using the same response for scale selection, as proposed in [193], for fast object detection with feature boosting.

Another example of very fast detectors is the FAST (Features from Accelerated Segment Test) corner detector [159, 160] as a simplified version of SUSAN [178]. At a given pixel p, a test is performed to regard the potential corner as a feature point. The intensity of pixels q laying in the discrete circle of radius 3 (16 positions) with center at p are compared with the intensity $I(p)$. Pixels q_i are classified as *brighter*, *darker* or *similar* to p (up to a threshold). The point is considered a corner if a segment/arc of at least 12 pixels has the same label, such label must be brighter or darker. An efficient implementation of the test allows early dropping the point without exhaustive comparison of all pixels, given the shape of the pattern only 4 positions (top, bottom, left and right) need to be dissimilar to the center otherwise the test fails and there is no corner. In case the first condition is satisfied, the test continues with further comparisons using a decision tree leading to a very fast release of features. In the baseline implementation of FAST detector, there is no multi-scale detection and no non-maximal suppression step. A significant improvement of FAST is called AGAST (Adaptive and Generic Accelerated Segment Test) [110] that includes space-scale analysis and a more efficient decision tree that actually combines two trees by switching between them, adapting itself to the environment without training. These trees use two more classes (not brighter and not lighter) in the pixel classification. Additionally, 3D non-maximal suppression across the image pyramid and sub-pixel and sub-scale precision refinement is performed. This improvements over the scale selection is also used to build on top of AGAST detector in order to build a fast feature descriptor called BRISK [96].

One of the most used integral approaches for fast detecting and describing local features for mobile applications, where computing resources are constrained, is ORB [163]. It is composed of two main components: oFAST and rBRIEF. The first one stands for *oriented* FAST that computes the orientation of the feature point as the intensity centroid of the neighborhood of p using the moments of the image patch [158]. The pattern of binary comparisons used by BRIEF [29] descriptor is aligned according to the feature orientation to be rotation invariant. In addition, the descriptor is improved by reducing the correlation between the descriptor dimensions by selecting the binary tests from the most balanced features and aggregating the less correlated ones building the binary descriptor. ORB belongs to a class of similarity covariant features since it is normalized for rotation and scale. In the following paragraphs, the most relevant detectors which model the affine covariant behavior of local regions during detection are briefly described.

Local affine covariant regions can be detected under some assumptions, like the EBR (Edge Based Regions) detector [46] does on intersecting edges in the image. It assumes that a parallelogram can be defined with a corner point (detected by Harris) and the two closest edges. The two adjacent sides have as common vertex the corner point and their lengths are estimated by an optimization task, where the objective function is defined over the image region comprehended by such parallelogram. The function is the distance ratio of line segments defined between the corners and the gravity center weighted with the pixel intensities. Such distance ratio between intersecting lines is a geometric affine invariant. To this end, a function is evaluated over the parallelogram region that reaches its extrema for corresponding values of the affine invariant parameters of the edges. The region for which such function reaches a local extremum is selected.

A more general approach to get local coordinate frames in distinguishable regions, based on the affine invariants, is presented in [144]. The regions are extracted by the feature detector called MSER (Maximally Stable Extremal Regions) [111] very robust against illumination and perspective changes. Stable regions are detected by consecutive thresholding the image by all possible thresholds in the intensity dynamic range. Keeping a record of the connected components obtained after binarization. The components whose areas are the most stable across the multiple thresholds are selected as keypoints that are covariant with continuous deformations of the image. Since the whole intensity range is used for thresholding, the MSERs are invariant to affine transformation of pixel intensities and their computation is suitable for real-time application because the pixels are sorted and the list of components and areas is maintained with the efficient union-find algorithm [169].

2.1.2 Learned detectors

In previous section, we went through a significant set of local feature detectors which by construction fires in certain regions that fulfill some conditions, for instance, the region is located in a local extrema of a response function. For this section, we will explore a set of keypoint detectors based on high-dimensional regressors. Such regressors are trained minimizing customized objective functions that enforced some features to be improved in the local region selection, i.e. maximize the repeatability or accuracy in the spatial location. Some approaches use machine learning techniques to improve a specific aspect of the feature extraction task and others are end-to-end learned architectures.

TILDE (Temporally Invariant Learned DEtector) [191] is an approach to detect keypoints with high repeatability under extreme illumination changes which cause performance drops in hand-engineered detectors. TILDE is train to be robust against lighting conditions changes from day to night, weather and moreover across seasons of the year. AMOS dataset [71] is convenient for this task since consists of 540 static cameras across the United States. The dataset has more than 17 millions of images with surface orientation, weather and seasonal changes. The regressor is trained with positive samples which are SIFT (multi-scale DoG) features detected independently in each image of a given camera. The patches centered in the most persistent points (detections in barely the same location across the image stack) are selected as positive samples and patches for negative samples are located far away from the interest regions used as positive samples. The regressor is trained to return a score value for every patch in the image as map in order to extract features by a simple non-maximal suppression step. The regressor used by TILDE is a piece-wise linear regressor expressed in the form of Eq. 2.4 which is the representation of linear function using [171]. The Generalized Hinging Hyperplanes for classification has the form:

$$\mathbf{F}(\mathbf{x};\omega) = \sum_{n=1}^{N} \delta_n \max_{m=1}^{M} \mathbf{w}_{nm}^{\top}\mathbf{x}, \tag{2.4}$$

where $\mathbf{x}$ is the deep feature representation of a given image patch and ω is the parameter set (weights) of the regressor defined in Eq. 2.5.

$$\omega = [\mathbf{w}_{11}^{\top}, ..., \mathbf{w}_{MN}^{\top}, \delta_1, ..., \delta_N]^{\top} \tag{2.5}$$

Finally, the objective is formulated to integrate three constrains: The classification term which enforces the separation between positive and negative samples (max-margin loss [40]), a pick shape regularizer term that enforce the response to have local maxima at the positive sample location and low values for the negative ones, and finally a regularization term of the response over time, i.e. to enforce the repeatability of the detector by getting similar responses along the image stack. The resulting detector outperforms the state-of-the-art on different datasets.

Some specific parts of the local feature extraction pipeline are improved by deep CNNs, for instance, the estimation of the canonical orientation used to normalized the patches before the descriptor extraction is supervised learned in [208] using a Siamese network [25, 211, 172] instead of the standard dominant orientation of the gradient approach as in SIFT. The assignment of the orientation is proposed as an optimization task minimizing the distance between descriptors, then the orientation is defined as an implicit variable that is learned by the regressor during training. The objective function has the following form

$$\mathcal{L}(\mathbf{p}_i) = ||g(\mathbf{p}_i^1, f_{\mathbf{W}}(\mathbf{p}_i^1)) - g(\mathbf{p}_i^2, f_{\mathbf{W}}(\mathbf{p}_i^2))||_2^2, \tag{2.6}$$

where $\mathbf{p}_i$ is a pair of points in two different images of the same 3D point in the scene, $f_{\mathbf{W}}$ is the orientation assigned by the regressor tunned with the parameters $\mathbf{W}$ and $g(\mathbf{p}, \theta)$ is the descriptor (used as a black-box) of the local region rotated by θ. Basically, the loss function is the square Euclidean distance of the descriptions of rotated version of the patches where the orientation is inferred by the network. A remarkable advantage of this approach is the flexibility as it is possible to use the CNN as robust orientation estimator and reach rotation invariance with others descriptors.

In order to learn feature detectors without supervision, the detection task can be defined as a regression problem. In [94], the authors introduce the covariance constraint which allows to detect repeatable keypoints under drastic imaging changes in the images.

Instead of using annotated samples as training data, the covariance itself is used as the objective function. In a simplified case for pure translation and corner points, the covariance constraint is defined as $\psi(T\mathbf{x}) = T + \psi(\mathbf{x})$, where $\mathbf{x}$ is an image patch, $T \in \mathbb{R}^2$ is the translation vector and $\psi \colon \mathbf{x} \mapsto \mathbf{f}$ is the function to be learn by the regressor to map the patches to a feature point ($\mathbf{f} \in \mathbb{R}^2$), i.e. the detector itself. The constrain can be generalized to more complex transformations (similarities and affinities) and feature shapes, from 2D points to rotated circles or rotated ellipses. Finally, the objective function to be minimized is presented in Eq. 2.7 where $\mathbf{x}_i$ and T_i are pairs of image patches and transformations in the training data and the optimization is with respect to the parameters of ψ implemented with a deep CNN.

$$\min_{\psi} \frac{1}{n} \sum_{i=1}^{n} ||\psi(T_i\mathbf{x}_i) - \psi(\mathbf{x}_i) - T_i||^2 \tag{2.7}$$

Once ψ is learned, it is convolved at all image locations, because of overlapping local regions, the feature location are voted by bilinear interpolation in a 2D grid and a non-maxima suppression is applied.

An extension of the previous approaches for learning feature detectors is the *Learned Invariant Feature Transform* (LIFT) [206], where the feature detector is learned in supervised manner using a labeled dataset of image patches extracted after 3D reconstruction using VisualSFM [201] that operates with SIFT features. The pipeline also comprehends a orientation estimation and description stage as part of the learning task. The objective function is defined as the hinge embedding loss of the Euclidean distance between the description vector. Moreover, in the same manner of concatenating differentiable losses, a full pipeline including the feature correspondences computation (for wide-baseline stereo matching) is proposed in [207], where the input data are pairs of 2D points in correspondence.

Many CNN-based local region detectors are trained using pre-annotated images or patches, however as an effort to build automatically a large dataset of interest point locations with pseudo-ground truth annotations is presented in [43]. A fully convolutional CNN keypoint detector is pretrained with an annotated corners dataset of synthetic shapes in the images and used to extract point locations of real images. The images are warped multiple times by homographies in order to provide more perspectives and scales from the same scene, sharing the concept of the data augmentation [147] and similar to the view synthesis for image matching [134]. The approach is known as *homographic adaptation* and works as a self-supervised training of the detector boosting its performance with larger repeatability, giving place to the detector called *Superpoint*.

The first fully unsupervised training approach for transformation invariant keypoint detector is presented by the Quad-network approach [166]. The regressor (a deep CNN) learns a pixel ranking functions that must be invariant under a set of transformation classes. The network is trained with quadruplets of points, i.e. two pair of corresponding points since the objective is defined as a margin loss between two negative samples (non-matching pairs) generated from the input quadruple. Finally, good feature points are located in the extrema locations of the ranking function computed by a non-extrema suppression. The detector is trained with ground truth correspondences in RGB images, fully unsupervised correspondences computation and employing RGBD images obtained 3D scanners. Depth images are employed in [145] to reach unsupervised learning pipeline for the same task.

2.2 Keypoint descriptors

The problem of description can be summarized as the task of computing an identifier for an entire image or a local region of it. The identifier is a high dimensional vector that describes the region to be recognizable even after a set of transformations (intensity and/or geometric changes, even different imaging devices). The usefulness of the descriptor depends on the number invariants it has, i.e. illumination, rotation, scale, similarity, affine invariance. In this section we present approaches for describing local regions (descriptors) that are popular in the task of stereo matching since it is the core setup of our object detection criterion. The description of a region can be defined as the function δ shown in Eq. 2.8 which domain is the squared (2-dimensional) image patch to be described and the codomain is the D-dimensional descriptor.

$$\delta : X \rightarrow Y, \text{where } X \in \mathbb{R}^{N\times N}, Y \in \mathbb{R}^{D} \tag{2.8}$$

In the literature is possible to find a wide range of techniques for description [88, 95] depending on the feature to be captured, i.e. color, texture, shape, etc. Nevertheless, we focused on the most robust descriptors to changes in the extrinsic and intrinsic parameters of the camera. Notice that after the CNNs have been shown to be very powerful regressors, the got a lot of attention in the description field leading to a wave of approaches based on deep learning, therefore we present in independent sections the hand-crafted and learned approaches.

2.2.1 Hand-crafted descriptors

The gold standard descriptor was introduced by Lowe in [106] as part of the framework (detection and description) called SIFT. The feature detection approach of SIFT is described in Section 2.1.1. The description approach aims to achieve robustness to lighting variations and small positional shifts by encoding the image information in a localized set of gradient orientation histograms. Some variations of this descriptors have been proposed. Like RootSIFT [149] which normalizes the standard SIFT vector Y as follows: L1 normalize Y, apply element-wise square root X to give X' then X' is L2 normalized which allows us to use the Hellinger kernel to compute the distance between descriptors.
Some authors propose to reduce the high dimensionality of SIFT (128-D) using popular techniques like PCA for the descriptor called PCA-SIFT [203]. Its description procedure can be divided into two steps: projection matrix generating and descriptor establishing. It makes a new vector of lower dimensionality than a standard one with the least correlated dimensions in the feature space. Another approach to enrich the SIFT descriptor is by the concatenation of another descriptor with different features, for example GSIFT [137] concatenate a 64-D weighted global texture descriptor to add context information to the original vector. In [136] the authors address the specific task of comparing images related by an affine transformation with ASIFT. The principle is similar to the view synthesis for image matching [134] which uses the affine camera model to create synthetic views from the original image. ASIFT detects keypoints and describes them from all affine images.

A fast alternative for SIFT is its approximation presented in SURF [18] which as well is provided with detection and description components. SURF reaches the rotation invariance by computing its own gradient orientation histograms with approximations of the Hessian-Laplace responses by using 2D box filters also known as Haar wavelets. The descriptor is computed in a SIFT-like fashion using the Haar wavelets feature maps in a 4×4 grid surrounding the center of the keypoint.

Another descriptor inspired by SIFT is GLOH (Gradient Location and Orientation Histogram) [120]. It considers more spatial regions (bins) for computing the histograms of gradients, i.e. it changes the location grid and the number of bins. The proposed grid has 17 location and 16 orientation bins that gives a vectorized histogram of 272-dimensions. In order to match the same dimensionality of the standard SIFT, PCA is applied over the intermediate representation.

The DAISY [185] is a descriptor that is related to SIFT and GLOH. It is proposed as a more efficient way to compute the 3-D gradient orientation histograms. Each bin contains a weighted sum of the norms of the image gradients around its center, where the weights roughly depend on the distance to the bin center since the contribution of each pixel is spread over a $2 \times 2 \times 2$ neighboring bins to avoid boundary effects. The weighted sums of gradient norms is replaced by convolutions of the original image with several oriented derivatives of Gaussian filters which provides the same invariance with faster computation. The descriptor is finally built with the weighted sum of gradient norms in a radial pattern similar to GLOH but applying Gaussian kernels instead of triangular ones. This descriptor has a better performance in dense image matching.

Some vision applications like SLAM [141, 199, 150] work with local features to recognize images of places previously visited, a mobile robot takes such images during a traversal. The identification requires fast algorithms for detection and description, thus a popular solution is to use binary descriptors which are particular beneficial for speed. BRIEF (Binary Robust Independent Elementary Features) [29] is a recent feature descriptor that performs simple binary tests between pixels in a smoothed image patch. Its performance in recognition and matching is similar to SURF but it is computed in $1/16$ of the time. However, the baseline implementation of BRIEF is not invariant to rotation and scale changes. ORB [164] framework (presented as a learned approach in Sec. 2.2.2) addresses the similarity invariance to outperform FAST.

Inside the field of binary descriptors and following the strategy of binary comparison of pairs of pixels, FREAK (Fast REtinA Keypoint) [2] is inspired by the human vision system. In comparison with BRIEF in which pairs of pixels are randomly selected, FREAK uses the retinal sampling grid which is circular and has higher density of points near the center of the region. Its robustness and speed are higher than SURF and BRISK, and under some conditions, comparable to SIFT, however, it is not faster than BRIEF.

Finally, it is worth to include the BRISK (Binary Robust Invariant Scalable Keypoints) [97] framework, that provides detector, descriptor and matcher. It is built on top of FAST [162] applying scale-space analysis in combination with the assembly of a bit-string descriptor from intensity comparisons. The pattern of pixel locations for the binary tests is similar to the one used in DAISY while the main difference is that the Gaussian kernel for smoothing the patch has a standard deviation proportional to the distance of the pixel to the center. Two sets of binary comparisons are considered based on the distance between locations. The keypoint orientation is computed with the farther set and the binary vector is constructed with the closer set, as a result we have a descriptor of 64 dimensions equivalent to 516 tests.

None of the previous approaches requires training data for choosing the test or features to include in the final vector, however, there is a popular field of machine learning that investigate the description step as an optimization problem, using from simple regressors to deep CNNs.

2.2.2 Learned descriptors

A common practice for different learned descriptors is to use a dataset of annotated image patches such as Brown dataset [26] and HPatches [11], for supervised learning.

As a supervised method we find the descriptor of ORB [164] framework. It consists in BRIEF with orientation normalization. The orientation is computed by the intensity centroid of the local region where the keypoint is located. ORB framework describes keypoints detected by FAST. The robustness of baseline BRIEF descriptor is improved by selecting binary comparisons via a greedy algorithm, that requires a training dataset of N keypoints. All possible binary comparisons inside the patch are applied to the training data. Each comparison has a score computed as the distance from the mean value across the training set to 0.5. The later enforces high variance in the feature space. The descriptor is constructed with a subset of comparisons with the highest scores and only the least correlated ones are included in final vector.

In the supervised approaches, annotated datasets are used. For the same 3D point in the scene there are multiple normalized image patches of its projections in different images taken with different cameras and poses. Some authors propose to use discriminative projections to construct binary descriptors, e.g. in D-Brief (Discriminative BRIEF) descriptor. Each projection represents a dimension in the descriptor which is binarized by thresholding. Both, the projections and the thresholds are learned with a two steps approach by LDE [26] and sparsity estimation [10].

A more general definition of binary strings inspired by AdaBoost [51] is proposed in BinBoost [188]. The learning approach improves the robustness and compactness of the descriptor. Each bit of the string is computed with a boosted binary hash function, and efficient optimization is performed in such a way that the different hash functions complement each other enforcing a compact representation of keypoints. Weak classifiers (learners) are functions which parameters are a rectangular bounding boxes and a gradient direction. The intuition behind weak classifier is to measure the relative number of positions inside the bounding box where the direction of the gradient agrees with the one indicated as input. The selection of the classifiers in the hash function is driven in a supervised fashion, enforcing equal hash function outputs in pair of patches correctly matched and different outputs for mismatched ones.

In the field of learned descriptors there is a seminal work [200] where the authors introduce a dataset of match/non-match image patches extracted from a Structure-from-Motion system (SfM) [179]. The authors created the dataset detecting DoG keypoints in 1000 images over 3 datasets: Yosemite, Liberty, and Notre Dame. They used dense surface models obtained via stereo matching to establish correspondences between real interest points. In order to get the ground truth labelling of the keypoints they used provided depth maps to transfer a local dense sampling of points that surrounds the keypoint into a second image and then use least squares to estimate the expected position, scale and orientation of the projected interest point. In this same work [200], DAISY test pattern is evaluated for multiple steerable filters of gradient-based features extractor to find the best configurations that minimize the matching error.

With the Deep Learning wave in the field of machine learning [42, 182, 174, 181], the researchers paid attention to approaches intended to learn descriptors based on CNNs playing with different architectures, objective functions, training strategies, regularizations, etc. In the sake of simplicity, we focus in a few representative approaches for learning descriptors using deep architectures. In [211] the addressed problem is to learn a general similarity function for comparing image patches without extracting manually-designed features from them as a prior step. The features are learned as a part of the back-propagation pipeline using a well known architecture types, i.e. 2-channels [58], siamese [34] and pseudo-siamese [68]. For all chosen architectures, the input data is a pair of squared images patches of the same size and the objective functions has a regularization term and minimization of the classification error term, i.e. the product of the network output and the label of the patch pair (positive-matching pair, negative-non-matching pair). A similar approach is proposed for the MatchNet [202] descriptor which uses a siamese fashion where each branch follows the AlexNet [42] model. The network is trained minimizing the cross-entropy loss function.

We review a CNN-based descriptor HardNet [128] that we used in some of our experiments for matching (see Section 3.3). It uses the architecture of L2Net [184] network. Instead of optimizing 3-term loss function, it mimics the matching strategy of SIFT where a 1st nearest neighbor match is confirmed by thresholding the distance ratio of the 1st and 2nd nearest neighbor [106]. The cost function is defined as a triplet margin loss for a negative mining approach. HardNet descriptor represent the state-of-the-art for stereo matching task together with an affine normalization approach introduced in [135].

A local image descriptor called *DEep Local Features* (DELF) is also based on CNNs, however, it is trained only with image-level annotations on a landmark image dataset. The authors propose an attention mechanism to select semantically useful local features for image retrieval. DELF framework can be used for image retrieval replacing other keypoint detector and descriptor. Experiments show DELF increases the accuracy in feature matching and geometric verification.

2.3 Image matching

This section is focused on the stereo matching task and it presents different approaches reported in the literature that are relevant to this thesis since image matching is the core of our object detection approach.

The image matching problem assumes that local regions are covariant between viewpoint changes. In some cases, the transformation underlying between the two views is not a pure translation or composed only by translation and rotation. The change of the camera pose may involves a scale and affine transformation over the shape of regions in the images.

The problem can be seen as a search problem, where the goal is to find a common part of a scene captured in two (or more) images with different camera poses. The geometric relation between views, represented by a linear transformation, is estimated with certain degree of confidence depending on the size of the common region and the visual information contained in it e.g. shapes, geometric structures, texture, etc. The geometric transformation allows to convert the coordinate system of the first image to the coordinate frame of the second one.

The overview of the image matching task is split into two main branches: The stereo matching setup and the wide baseline stereo matching. Both branches are exemplified in the following sections with methods found in the literature.

2.3.1 Stereo matching

For stereo matching [167], the rows of both images are aligned with the projection of the baseline of the Epipolar geometry setup. Hence, local shapes can be approximated by a pure translational model. Fig. 2.3 shows images taken with a stereo camera, viewpoint changes at local region can be modeled as Euclidean isometries. One approach to build feature correspondences in this setup is based on image patch correlation. The size of the image patch (window) affects the accuracy of the dissimilarity estimate [170]. The size of the window can be adjusted driven by the

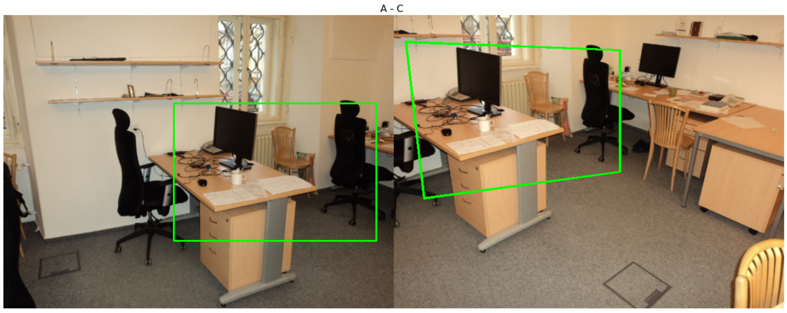
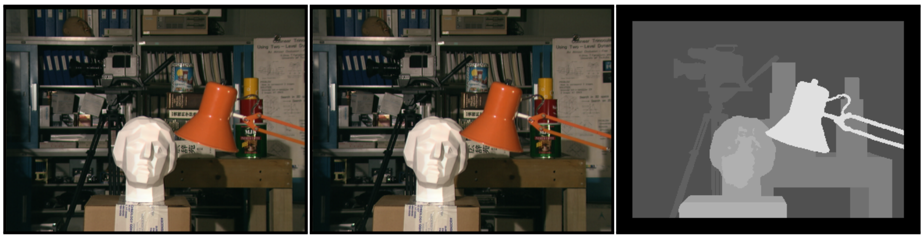


Figure 2.2: Example of two images captured with the same camera in different viewpoints. The image regions depicting a common part of the scene is indicated by the **green** bounding boxes. The shape of the left box is covariant to the transformation underlying in the viewpoint change.



(a) (b) (b)

Figure 2.3: Example of an image stereo matching setup. Images belong to the Tsukuba sequence of the Middlebury stereo dataset [167]. The left image (a) is matched with the right image (b) and the ground truth image (c) has as intensity values the normalized Euclidean distance between corresponding points, also called disparity map.

disparity estimate uncertainty of pixels inside the window [77]. The estimation approach fits a parametric statistical models with sum of squared differences (SSD) between patch intensities. Finally, the task is addressed as a minimization problem where the objective involves the pixel uncertainty as a function of the window size.

The accuracy of the disparity map estimation is compromised by multiple factors, noise related to imaging devices, the lack of texture in the objects, discontinuities in the disparity map, and the occlusions. Graphical models [90], like Markov Random Fields (MRF) [80], have been used in this task. Such models are trained with algorithms based on Belief Propagation (BP) to address the nuisance factors. In [74], the authors propose an architecture with three coupled MRFs [74]. The first one enforces the smoothness of the depth field, the second models the discontinuities with a spatial line process, and the third models the occlusions with a binary process. After eliminating the line and binary processes by introducing robust functions, the Maximum A posteriori Probability (MAP) estimate is performed by Bayesian BP.

The best results of this approach are obtained incorporating hard constrains in the object boundaries via an object segmentation-based algorithm [183]. In this kind of energy minimization tasks, the color information can be introduced to improve accuracy of depth estimate of the non-occluded pixels [204]. The window size used to compute the disparity maps is adjusted with respect to the RGB correlation between pixel and its neighborhood. Such approach shows higher accuracy on both smooth regions and discontinuities in the disparity maps.

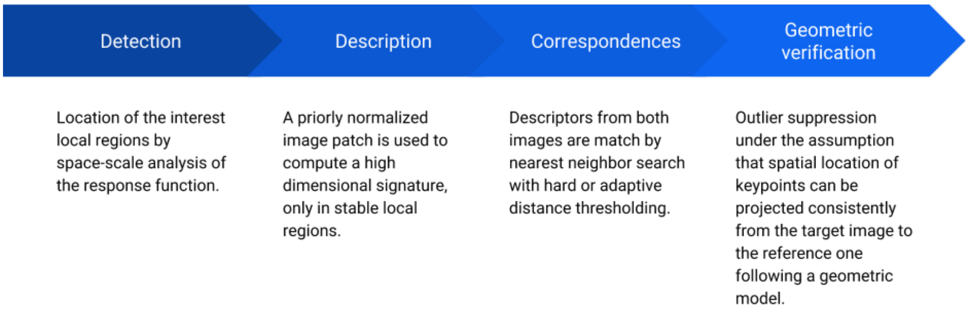
The high correlation of the color in local regions of the image is associated with the structures in the scene. This information can be used to create larger connected components (segments) in the image, applying color segmentation, for example. In [99], the disparity field is estimated in the segment domain and the problem is defined as an energy minimization. Finally, depth map is approximated by the Graph Cut algorithm [24].
The BP-based method is improved by using the graph cuts algorithm jointly with locally shared labels. Since the optimization is performed with sub-modular movements, the optimal labelling at each min-cut is guaranteed, and it allows to initialize the labelling proposals with a randomized search. As a result, the approach finds smooth disparity fields in planar localities and gets the highest sub-pixel accuracy.

The principle of cooperative optimizations [52] can be applied on top of the segmented image, by mean-shift in the color space, to estimate the depth field. The estimation is performed by an initial local optimization step and followed by a global optimizations step. The depth consistency in adjacent regions in the map is enforced by minimizing the energy functional ($E_i(x)$) by a local optimization method[66] of the cost function

$$\Psi_i(x) = \min\left((1-\lambda_i)E_i(x) + \lambda_i \sum_{j\neq i} w_{ij}E_j(x)\right) \text{ for } i,j = 1, ..., N, \qquad (2.9)$$

for the i-th region, where the j-th regions are adjacent to the i-th one, $\lambda_i \in [0,1]$, $w_{ij} \in [0,1]$ are the corresponding weights.
A similar approach is proposed in [21], where scene is assumed to be composed by a set of smooth surfaces approximated by B-splines. The optimization task of a pixel-wise MRF is performed with the fusion move approach [93].

A CNN-based stereo matching approach is presented in [194]. The architecture is a siamese network that compares image patches with a learned similarity measure implemented with a concatenation layer followed by a set of fully connected layers [194]. In order to speed up the training process, the authors of [109] replace the concatenation layer and the fully connected layers with single product layer.



Location of the interest local regions by space-scale analysis of the response function.

A priorly normalized image patch is used to compute a high dimensional signature, only in stable local regions.

Descriptors from both images are match by nearest neighbor search with hard or adaptive distance thresholding.

Outlier suppression under the assumption that spatial location of keypoints can be projected consistently from the target image to the reference one following a geometric model.

Figure 2.4: Generic pipeline for the wide baseline stereo matching task.

2.3.2 Wide baseline stereo matching

The so called *wide baseline stereo matching* involves different camera setups compared to the *stereo matching* (described in Sec. 2.3.1). In Epipolar geometry [61], the line defined between two camera centers is larger (wider), i.e. the distance between the viewpoints or positions where the cameras where located, is significant larger compared to the stereo configuration. The image planes of the two cameras are not coplanar and the Y-axis of the image coordinate frames are not collinear. In addition, the images are expected to be zoomed in or out, projective transformed as an effect of a perspective change in the camera poses. As described in Sec. 2.1.1, under this camera configuration the local regions in the images are assumed to be deformed co-variantly by affinity or similarity transformations. The sliding window approaches are computational expensive and inefficient for this scenario then the goal is establishing dense correspondences between keypoints located independently in both images.

A generic pipeline for wide baseline matching was introduced by Lowe [107] starting from the search of image locations and scales of the interest points, the patch normalization for description with the corresponding invariants and building correspondences by nearest neighbor search in the descriptor space (see Fig. 2.4). The keypoints are detected finding the extrema of the DoG function that can be computed efficiently subtracting nearby scales in the image pyramid. The precision improvement of the point location in space and scale is done through a 3D quadratic function fitting locally at the point location in order to interpolate the response function [27]. DoG local maxima laying in edges are dropped by thresholding the ratio between the largest magnitude eigenvalue and the smallest one of the Hessian matrix ($\mathbf{H}$) computed at the corresponding location and scale of the keypoint under testing. The ratio of principal curvatures is presented in Eq. 2.10, where $\alpha = r\beta$ is the eigenvalues ratio and α, β are the largest and smallest eigenvalue, respectively. This feature detector is commonly known as SIFT detector.

$$\frac{\text{Tr}(\mathbf{H})^2}{\text{Det}(\mathbf{H})} < \frac{(r+1)^2}{r} \tag{2.10}$$

Before the descriptor computation and as part of the patch normalization process, there is an additional step required to reach rotation invariance, that is the estimation of the dominant orientation of the gradient. The mode of the histogram of gradient orientation is computed with the samples taken from the region around the keypoint. The number of bins is fixed to 36 covering the 360 degree range of possible orientation and, in case of multi-modes or that others peaks in the histogram with frequencies higher than 80% of the mode peak, multiple rotated instances of the same keypoint are included in the feature set.

Description step is performed with the so-called SIFT descriptor [107] and it is widely used in multiple computer vision tasks, not bounded to image matching. The descriptor of a keypoint is a 128 dimensional signature consisting in a local histogram of gradient orientation that allows for small misalignments without hurting the matching performance. The SIFT descriptor is extracted from the normalized patch (transformed into a canonical coordinate frame) which is spatially tessellated with a 4×4 lattice rising each of them a histogram of gradient orientation. The samples are quantized into 8 bins with interpolation where the bins contain the weighted sum of the gradient magnitudes. This popular descriptor can be interpreted as a 3D histogram where 2 dimensions are the XY location of the pixel and the third one is the image gradient orientation, an illustration of the SIFT construction is shown in Fig. 2.5. The recognition task, presented in [107], proceeds by matching individual features to a database of features from known objects using a fast nearest-neighbor algorithm. The geometric verification is performed by a Hough transform to identify clusters belonging to a single object, and a least-squares solution for consistent pose parameters estimation.

In [111], the pipeline for robust wide-baseline matching is capable to match more challenging viewpoints. MSER detector is used as the interest region extractor and the feature shapes are approximated as ellipses since detections are affine-covariant regions. After image patch normalization, instead of SIFT that is scale and rotation invariant, the descriptor is computed with rotational invariants by estimating complex moments in the color patch [126] leading to the affine invariance. The tentative correspondences are computed by correlation and outlier suppression by RANSAC over Epipolar geometry in a two-step fashion. The first epipolar geometry estimate uses the gravity center of the MSERs, then the subset of consistent tentatives is pruned again by a correlation thresholding of affine and rotation normalized regions using the correspondence of covariance matrices and epipolar lines. Afterwards, a *finer* Epipolar geometry is estimated by RANSAC with the new correspondence set, resulting in a higher precision of the geometric model estimate.

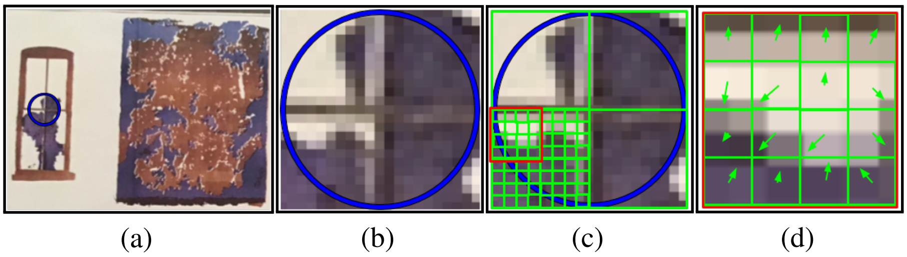
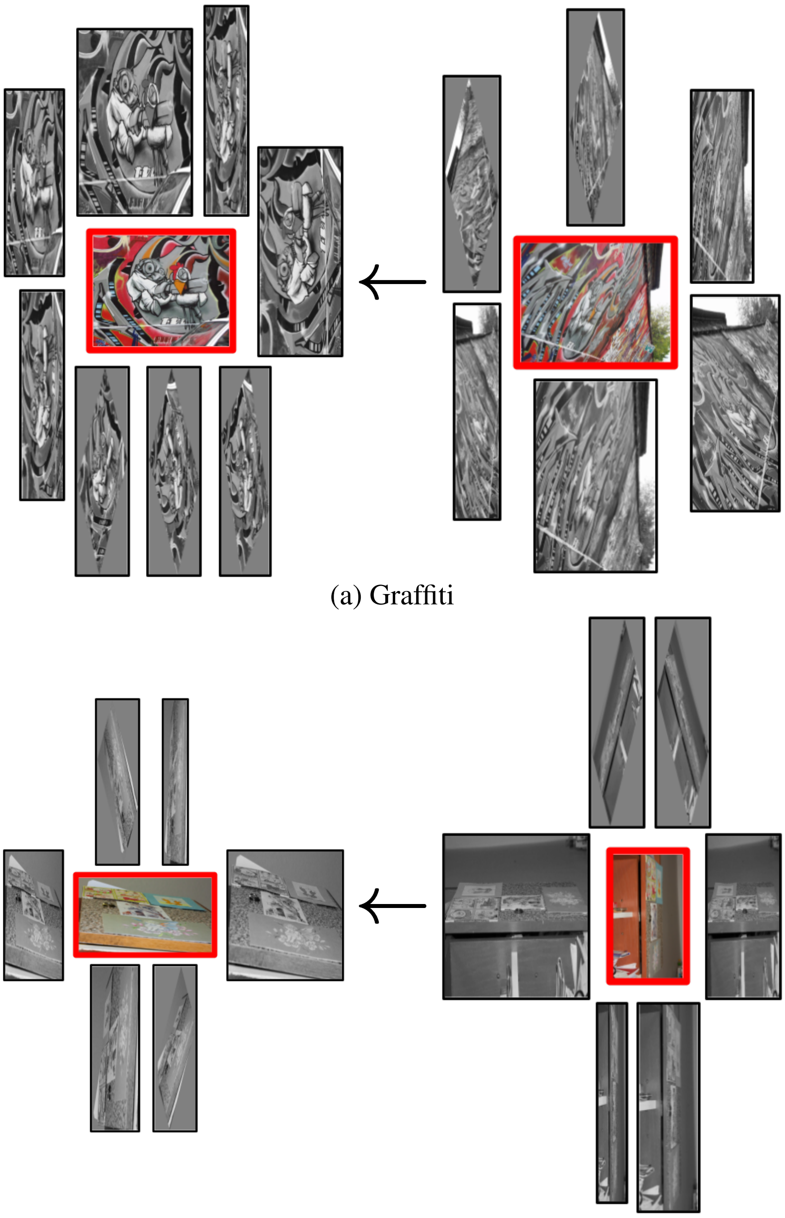


Figure 2.5: Illustration of the SIFT descriptor computation. A keypoint indicated with a **blue** is shown in (a) and (b) is a zoom-in of the point. The 4×4 tesselation (in **green**) used to compute the 16 local histograms of gradient orientations is shown in (c) together with a 4×4 pixel region to compute a single histogram (**red**). The zoom-in of the histogram with the gradient orientation (arrows) that are quantized to build the descriptor.

Instead of estimating the geometric relation of the two views by RANSAC with the full set of tentative correspondences, some authors proposes to do it with agglomeration. In [151], the set of tentatives is split into small clusters of spatial close points in order to compute local homographies that reproject features from one image to the other with high accuracy, then the correspondence set is growth by using RANSAC algorithm iteratively taking as initialization the local homographies. The tentative matches that are consistent in each iteration are aggregated to the next RANSAC run. The union of consistent correspondences confirmed by expanding all local homographies become the final set of inlier matches. A similar approach with tighter constrains over local regions and the global epipolar relation between images and introducing a deformation model (caused by under rotation, scale, intensity and moderate affine transformations by means of a systematic protocol proposed in [119], where it empirically shows very good performance.

The correspondence keypoint pairs are established by nearest neighbor (NN) search with a presorting step in order to efficiently search for similar descriptors by Euclidean distance. Descriptors are stored in a kD-tree structure to relieve the cursed of dimensionality. Correspondences are filtered by the first to second NN distance ratio. Distance ratio higher than 0.8 are confirmed to be in distortion) is presented in [46].

Some approaches extend the capability of the SIFT method to reach affine invariance in the local feature extraction. By generating sample views of the initial images, which simulate changes in the camera orientation (latitude θ and longitude ϕ angles), the similarity invariant SIFT can treat the two additional parameters to effectively cover the six parameters of the affine transformation. The ASIFT (Affine-SIFT) [136] transforms each image into a set of affine distortions which simulates all possible camera position changes by a fix set of quantized values for the θ and ϕ parameter setting. In Fig. 2.6 two pair of matched images are shown with the synthetic views computed for each initial image. The matching algorithm



(a) Graffiti

(b) Fox

Figure 2.6: Two examples of view synthesis for reliable geometry estimation. The pair (a) is part of *OxAff* [119] and (b) belongs to *EVD*[134]. The original images are **red** framed and the surranding ones are their warped versions. The orientation of the matching task is idicated by ←, i.e. the reference images is in the left side and the target image is on the right. Views were synthetized by MODS[131].

proposed in the baseline version of SIFT is applied to all image pairs created from the synthetic views and the initial images and the correctly matched features are normalized into the initial image coordinate frame. In [134], improvements over the view synthesis approach are presented withing the Matching On Demand with view Synthesis algorithm (MODS). Instead of generating all the simulated views corresponding to a fix set of quantized values of camera motion model, the views are generated progressively *on-demand* until a reliable estimation of the geometric model relating the image pair is obtained. In the same manner, the complexity in the feature detection step is control by aggregating more computational expensive detectors.

The first attempt to generalized the wide baseline stereo problem is reported in [133], as a two-view image matching problem where two or more of the image formation and acquisition properties significantly change. Types of changes considered are the illumination, geometry, sensor and appearance. Jointly, a dataset known as WxBS is provided with pairs of images where one or more type of changes are present.

Making reference to the last stage of the stereo matching task, the geometric verification of the tentative correspondences, i.e. outlier suppression, the *gold standard* method is RANSAC [50] for robust estimation due to its good performance against high outlier ratio. One of the main disadvantages is the long time to reach convergence but some approaches like PROSAC [38], LO-RANSAC [36], MLESAC [186] speed up the estimation. A unified framework of tools for robust estimation is called USAC [154]. A novel approach of an end-to-end trainable architecture is presented is introduced in [207]. The input of the architecture is the set of tentative correspondences and the intrinsic parameter of the camera. The network is intended to learn the labelling of the correspondences (as inlier or outliers) and using them for recovering the relative pose of the two views, jointly. For the optimization of the parameters, a hybrid loss function is proposed over the individual correspondences and the fundamental matrix estimation. State-of-the-art performance is achieved but is important to notice that RANSAC is applied as the last step over the set of inlier correspondences outputted by the network.

2.4 Relevant shot detection

The problem of detecting and classifying video shots into interesting ones for the user or not have been addressed from multiple perspective, the most representative ones reported in the literature are described in this section.

From our knowledge the first approach for assessing the content of the video shared through the Web is reported in [44], where the goal is to construct automatically a database of video shots labeled by the action captured in them. First, videos are retrieved by means of the web API querying 100 kinds of actions. In this step the videos are ranked scoring the co-occurrence of the tags among all the retrieved videos [205]. As a preprocessing step, the videos are segmented into shots and for each shot a set of features of multiple types are extracted to describe the corresponding video segment. The spatio-temporal features [143] are defined as triplets of SURF points that are moving along the shot, i.e. a visual object tracker is applied to classify the points as steady or moving. Local appearance and motion features are extracted from each triplet leading to a 256-D descriptor. The second type of feature is the global motion in a frame, Lucas-Kanade [108] tracking features are extracted at fix 8 pixel rectangular grid and a global histogram of motion magnitude and direction is built, leading to an additional 15-D descriptor. Finally, the appearance of the shot is described by Gabor texture features computed locally with respect to fix grained frame tessellation, giving as a results 400 24-D vectors from a single frame. A single descriptor is composed from the earlier described ones and they are vector quantized in order to obtain a bag of feature representation of the shot. The final assignment of the shots into the action kinds is addressed by the *VisualRank* [75] algorithm which requires a distance matrix where ,in this approach, the similarity metric is histogram (shot descriptor) intersection. This unsupervised method for ranking the shots achieves a 49% mean precision at rank 100, which significantly behind the 80% obtained by the supervised approach with MKL feature fusion proposed by [143].

In a follow-up work [44], the same authors proposed a slightly different approach that actually increases the precision in 3% with respect to the baseline method. After video downloading, decomposition into shots and feature extraction, the *Ordering Points To Identify the Clustering Structure*(OPTICS) [8] hierarchy clustering method is applied to vectors computed from a single action query. Inside each cluster the outliers are filtered out by scoring the descriptors by the isolation with respect to its surroundings [32] samples.

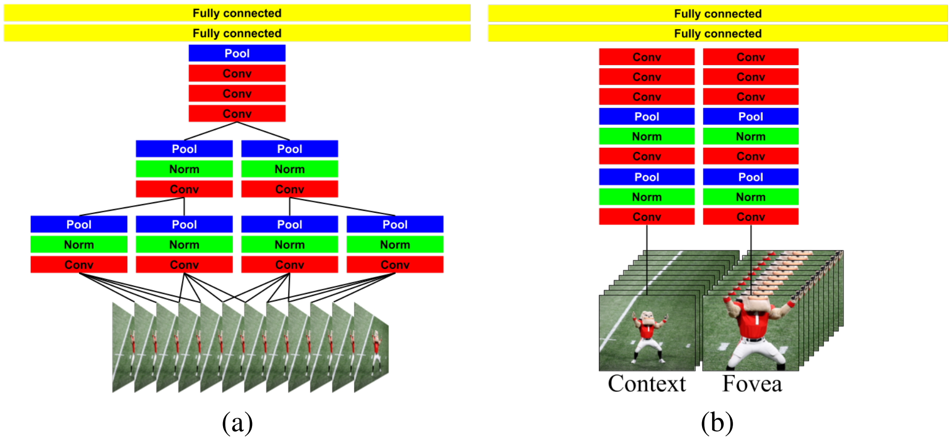


Figure 2.7: (a)Time information fusion in CNN architecture called *Slow fusion* which input is a subsequence of a video clip and (b) the multi-resolution architecture change called *fovea* and *context* streams to speed up the training of the networks.

In recent publications deep network architectures are used for modeling the time dependency between video frames for large-scale video classification. In [78], the spatiotemporal features are extracted with a novel architecture called *slow fusion* which takes an interval of consecutive frames, store them in a single 4-D tensor as input and convolve it with $11 \times 11 \times 3 \times T$ filters, where $T = 4$ in the first convolutional layer and $T = 2$ in the second and third layer, the following layer are similar to ImageNet [87] model. A simplified illustration of the slow fusion topology is shown in Fig. 2.7(a).

Additionally, in the same work there is a major contribution regarding the speed of the training, a multi-resolution architecture is proposed to accelerate the training by reducing the size of the input images without compromising the performance. The architecture is called *Fovea and context streams* and it is shown in Fig. 2.7(b), there are two separate streams of processing over two spatial resolutions. Both streams are feed with images of half of the size in the original architectures, the context one received a downsampled frame at half of the original spatial resolution and the fovea one receives the central region at the original resolution, assuming that the object of interest appears are most of the time well centered in frame. The experiments shows that the mixed-resolution architectures is an effective way to accelerate the network without compromising the accuracy and the temporal features encoded in the models with 4-D convolutions performs better than the single-frame models. The 4-D convolutions used to learn spatiotemporal features are also known as *temporal convolutions* and open the question "how deep we should go in time to improve the performance of such models?". In the sake of getting an answer to this question, a systematic evaluation is presented in [190] with an architecture that has 5 convolutional layer where the 4-D filters are of size $3 \times 3 \times 3 \times K$, where K is the number of input channels, i.e. $K = 3$ for RGB input and $K = 2$ for XY-flow input computed by Brox optical flow method [28]. The input subsequence of T frames is reshaped into K 3D stacks of the same channel each. Notice that the $4th$ dimension of the kernels runs over the channels, in contrast with the *slow fusion* model where it runs over the time. The time window is evaluated for $T \in \{20, 40, 60, 80, 100\}$ giving place to the so called *long-term temporal convolutions*. The experiments shows the classification accuracy monotonically increase with larger T and the best results are obtained by combining (average) the classification scores from the spatial and temporal networks.

Another popular way to encode the temporal relation between samples is the use of *Long Short-term Memory* (LSTM) networks that are employed in *Natural Language Processing* (NLP) [54, 33] approaches and time series analysis [100]. The LSTM cell can be considered as an improved architecture of the Recurrent Neural Network (RNN) cell [180, 192], which are hard to train with long-term dynamics (long sequences) due to the problem of vanishing and exploding gradients [64]. LSTM is provided with memory units that allow the network to learn when to forget previous hidden states and when to update them given new information. A popular LSTM architecture, that has shown good performance for representing specific type of programs [212], is shown in Fig. 2.8 where $\sigma(x) = (1 + e^{-x})^{-1}$ is the sigmoid

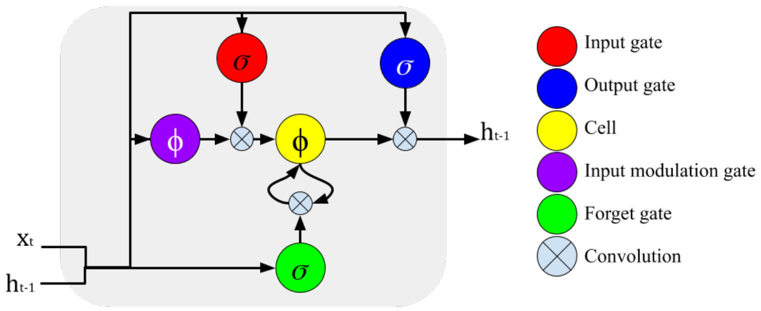


Figure 2.8: Diagram of the LSTM cell architecture proposed in [45]

non-linearity, $\phi(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}} = 2\sigma(2x) - 1$ is the hyperbolic tangent non-linearity, the output h_t is updated given inputs x_t and h_{t-1}.

On the side of activity recognition and video description, LSTM has been incorporated to learn and encode the temporal features in conjunction with the powerful representation of the visual information in the video frames with CNN, giving place to the Long-term Recurrent Convolutional Networks (LRCN) proposed in [45]. The frames are transform to a fixed-length vector representation that correspond to the activations in some layer of a minor variant of *AlexNet* [86]. The vectors are fed into a stack of LSTM cells to run the sequential learning. Finally, the inference consist of estimating at each time-step a prediction distribution applying a *softmax* function over the sequential model (LSTM) outputs. LRCN is a class of spatially and temporally deep architecture that outperforms previous approaches that encode temporal information with visual information [73].

The usage of CNN architectures with independent streams for video classification is another approach that has been studied recently. Basically, there is one stream trained for spatial information and other for the temporal information, independently, and the inference output is fused in testing time, exclusively. In [173], both streams are implemented with 2D CNNs (ImageNet [42]), the input of the spatial stream is the RGB video frames stacked channel-wise and the temporal stream is fed with dense optical flow, extracted with *Brox* [28] approach, that represent the temporal component of the video. A softmax layer is added at the end of both streams and they are combined by late fusion, i.e. averaging the outputs or training an SVM classifier with the stacked L2-normalized scores as features, see Fig. 2.9. The previous approach is improved in [30], substituting the 2D CNNs by 3D ones and suppressing the high computational cost of the optical flow computation by the motion vector representations that are precomputed and stored in the compressed video and can be extracted with the video codec efficiently. In the case of the spatial stream, the authors propose to use 4-D RGB inputs where the 4th dimension is the time step. The overhead of the 3D convolutions and the high number of parameters of the 4D architectures, the frames are not decoded selectively driven by the motion activity embedded in the codec for decompression.

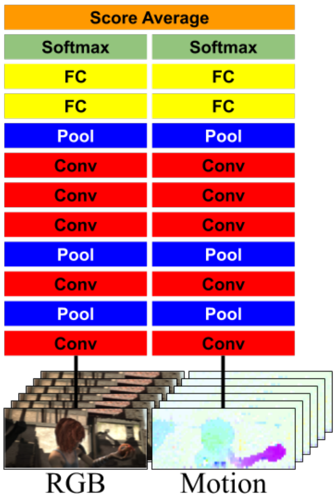


Figure 2.9: Two-stream architecture for video classification [173, 30]. The two streams are trained independently and the maximum scores of the softmax outputs are average to combine the knowledge encoded in both pipelines.

One of the main goals in this thesis is the automatic detection of interesting parts of videos and fast access to them without exhaustive passing through the full sequence, our approach is described in Chapter 4. In the state-of-the-art, the capability to accesses the interesting parts (subsequences) of a video with respect to the visual content has been addressed as an activity detection task. In [89], the problem is defined as Markov decision process where the length of the jump between frames (fast-forwarding) is learned by a supervised Q-learning method [195], which is widely used in reinforcement learning tasks in movement planning [20].

2.5 Additional work related to Saddle detector

The Saddle detector is a contribution of this doctoral research project that emerged as an independent component of the pipeline for relevant shot detection based on visual content of the videos. As a result, Saddle is used as an alternative for the local feature detection step of the pipeline meanwhile the rest is preserved in the standard setup, increasing the frame rate and improving the detection accuracy. The experiments of Section 3.3 show that its robustness and speed make Saddle a strong competitor against slightly faster detectors on natural images. On recent publications, the image domain where Saddle is used is different from the datasets and tasks included in [7]. In [156] the authors proposed a derivative version of our detector known as D-Saddle which is intended to extract keypoints from *Fundus* (retinal) [5] images as part of a registration approach. D-Saddle diverges from the standard Saddle in the scale-space pyramid which is constructed in the SIFT-like fashion with 4 octaves of 1 level each. The levels are DoG responses approximated by subtracting two blurred images with different scales. Finally, keypoints are detected by the standard Saddle and described by HoG [41]. The experiments show that Saddle handles well the low-contrasted images and it fires with high density along the blood vessels, even in the peripheral retina, which is desirable for robust image stitching. A follow up from the previous work can be found in [31] where D-Saddle is tested for matching ability with different descriptors again for medical images.

Besides the results obtained with medical images, a recently published comparison [67] of keypoints detectors tested in a novel dataset called *ApolloScape* revealed that Saddle has the state-of-the-art performance regarding repeatability. The sequences included in the dataset come from images and videos acquired by cameras mounted in automobiles/cars navigating along traffic streets from different cities, since it is intended to develop vision-based navigation approaches. The ApolloScape dataset consists in N continuous sequences (single shot without interruptions) recorded in different routes also known as traversals. The experiments shows that the repeatability of Saddle is significantly higher in most of the traversals. As a result, Saddle turns out to be the most suitable feature extractor for the vision system of autonomous vehicles among recently detectors that use deep CNN features and still being extracted in a fraction of the time.

Chapter 3

The *Saddle* feature detector

One of the main contributions presented in this thesis is our detector called Saddle, a novel similarity-covariant feature detector that extracts points whose neighborhoods, when treated as a 3D intensity surface, have a saddle-like intensity profile. The opportunity window that motivate this proposal is the trade-off between accuracy in detection and speed for the whole pipeline involved in the object detection on videos. Our experiments shows that the use of fast approximation of corners as feature detector hurts significantly the precision and recall of the relevant assessment while highly precise affine covariant feature detectors are too slow for the real-time constrains but provide the best recognition performance. As a consequence we designed robust and fast detector. The saddle condition is verified efficiently by intensity comparisons on two concentric rings that must have exactly two dark-to-bright and two bright-to-dark transitions satisfying certain geometric constraints.

This chapter is focused on the Saddle detector, its properties and experiments to test it performance in a wide range of computer vision tasks. We present a brief overview of the proposed detector compared to previous works and some use cases in Section 3.1. We introduce and justify the design of the patterns tested by the detector in the Section 3.2. In Section 3.3 we present an extensive set of experiments to show the performance in different scenarios as well as experiments that support the design decisions on the detector and its parameter setting. Finally, a shallow discussion of the findings on our detector in Section 3.4.

3.1 Overview

The detector extracts points whose neighborhoods, when treated as a 3D intensity surface, have concave and convex profiles in a pair of directions close to be orthogonal, see Fig. 3.1; in a continuous setting the points would have a negative determinant of the Hessian matrix. The saddle condition is approximately verified on two concentric approximately circular rings which must have exactly two dark-to-bright and two bright-to-dark transitions satisfying certain geometric constraints, see Fig. 3.2.

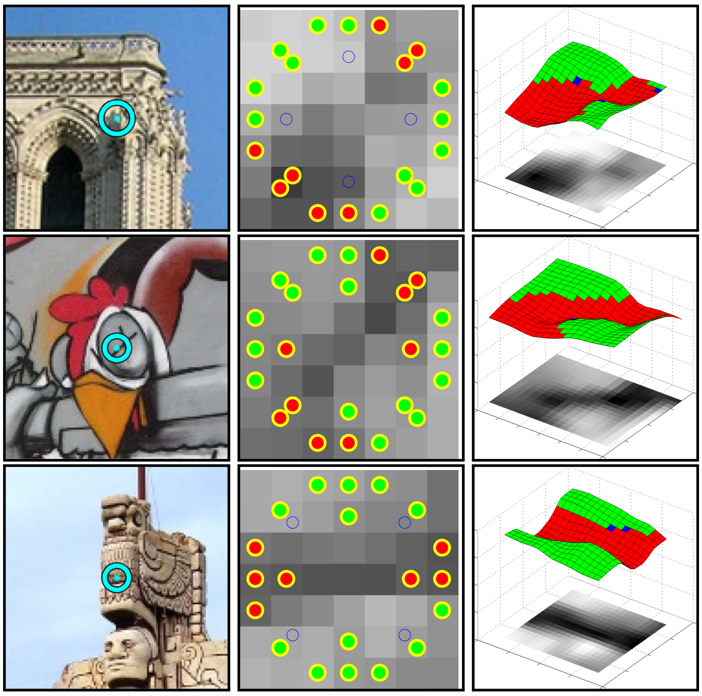


Figure 3.1: Saddle feature examples (left column). Corresponding image patches with accepted arrangements of dark (marked red), bright (green) and intermediate (blue) pixel intensities (central column). Pixel intensities around Saddle points visualized as a 3D surface (right column).

Experiments show that such points exist with high density in a broad class of images, are repeatably detectable, distinctive and are accurately localized. The Saddle points are stable with respect to scale and thus a coarse pyramid is sufficient for their detection, saving time and memory. Saddle is faster than SURF, a popular choice of detector when fast response is required, but slower than ORB. Overall, the Saddle detector provides an attractive combination of properties sufficient to have impact even in the mature area of local feature detectors. Saddle-like interest points (among others) were tested previously in a methodology for scale-selection and image matching in [104].

Saddle falls into the class of detectors that are defined in terms of intensity

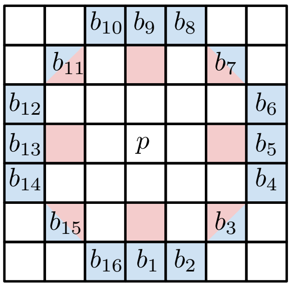


Figure 3.2: The 8 pixel positions marked red form the *inner ring* and the 16 positions b_j marked blue form the *outer ring*. Positions shared by both rings are bicolored.

level comparisons, together with BRISK [97], FAST [161], its similarity-covariant extension ORB [164], and its precursors like SUSAN [178] and the Trajkovic-Hedley detector [187]. With the exception of BRISK, the intensity-comparison based detector aim at corner-like features and can be interpreted as a fast approximation of the Harris interest point[1] detector [60]. Saddle is novel as it uses intensity comparisons for detection of different local structures, related to Hessian rather than the Harris detector.

Despite recent success of the deep learning based methods, local features methods are still state-of-art in, in particular, robotic applications like navigation [84] and place recognition in changing environments [165]. The very recent local feature competition [12] have shown that while learned descriptors significantly outperform handcrafted ones, the opposite is true for the local feature detectors. The top-performing method is based on DoG keypoints. The primary version of this paper has been published in [6].

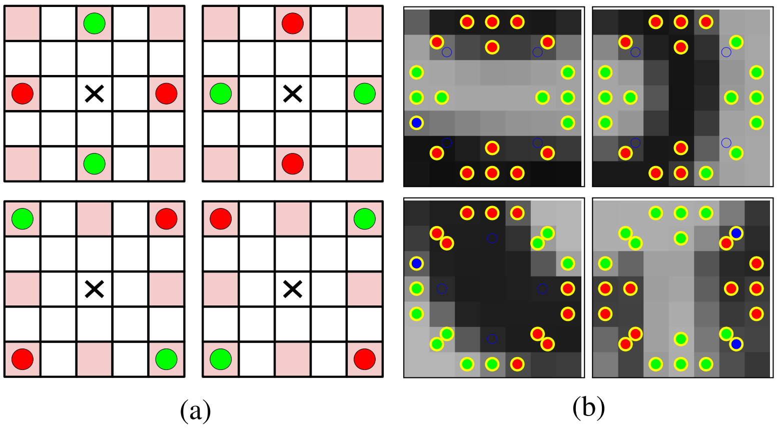


Figure 3.3: (a) The fast test for an alternating-pattern on the inner ring required for Saddle. In each of the four patterns, green dots depict pixels with intensity strictly brighter than the intensity of pixels marked red. The location is eliminated if none of the patterns is observed. (b) Examples of accepted patterns.

A recently proposed evaluation framework [81] for keypoint detector on the ApolloScape dataset [67] presents a comparison between novel deep-learning based

[1] In fact, the ORB final interest point selection is a function of the Harris response computed on points that pass a preliminary test.

Algorithm 1 Saddle feature detection

```
Input: Image $I$, $\epsilon$
Output: Set $F$ of Saddle keypoints
  for pyramid level $I_n$ do
      for every pixel $\mathbf{p}$ in $I_n$ do
          if INNER($\mathbf{p}$) then
              Compute $\rho_{\mathbf{p}}$
              if OUTER($\mathbf{p}$, $\rho_{\mathbf{p}}$, $\epsilon$) then
                  Compute response $R(\mathbf{p})$
      Non-Maxima Supression
      Coordinate Refinement
  return
```

interest point detectors (LIFT, TILDE, Superpoint and LF-Net) and hand-crafted keypoints detectors (FAST, ORB, DoG, AGAST, AKASE, BRISK, Saddle) for *repeatability*. The Saddle detector has the best average repeatability for all evaluated traversals (0.177), achieving the highest repeatability in 6 out of 9 traversals. In the remaining 3 traversals, it is closely behind the best performing detectors. The paper concludes that Saddle is the best choice for real-live applications of the autonomous driving type.

3.2 The Saddle Interest Point Detector

The algorithmic structure of the Saddle keypoint detector is simple. Covariance with similarity transformation is achieved by localizing the keypoints in a scale-space pyramid [101]. At every level of the pyramid, the Saddle points are extracted in three steps. First, a fast alternating-pattern test is performed on the inner ring, see Figs. 3.2 and 3.3. This test eliminates about 80–85% of the candidate points. If a point passes the first test, an alternating pattern test on the outer ring is carried out. Finally, points that pass both tests enter the post-processing stage, which includes non-maxima suppression and response strength selection. The algorithm is summarized in Alg. 1.

3.2.1 Alternating-pattern on the inner ring

The first test is designed to be very fast and to reject majority of points. For a given position $\mathbf{p}$ in the image, the test operates with the pixels located at the pink squares shown in Fig 3.2. In the test, two pairs of orthogonal directions are considered, one shaped as a $+$ sign and the other shaped as a $\times$ symbol. The test is passed if both points on the inner ring in one direction are strictly brighter than both points in the orthogonal direction. The four cases for passing the test are depicted in Fig. 3.3 (a). Note that either of the $+$ and $\times$ shapes can pass the test, or both.

From the intensity values of the pixels of the inner ring satisfying the alternating-pattern test, either four or eight pixels, depending whether one or both patterns passed the test, central intensity value $\rho_{\mathbf{p}}$ is estimated at pixel $\mathbf{p}$. As a robust estimate, the median of the intensity values is used. The computation of the median is implemented efficiently using the implicit sorting algorithm of the 4 pixels (per orientation) on the inner ring in order to test the alternating pattern. Under the assumption that the swapping inner pattern is fulfilled, one of the two opposite pair of pixel positions are smaller than the other and vice versa. The median value is computed efficiently as the intensity average of the highest value of the darker pixels and the lowest of the brightest pixels.

3.2.2 Alternating-pattern on the outer ring

The second test considers the 16 pixels that approximate a circle of radius 3 around the central point. The outer ring is depicted in Fig. 3.2 in light blue. Let the pixels on the outer ring be denoted as $B = \{b_j \mid j = 1 \ldots 16\}$. Each of the pixels in B is labeled by one of three labels $\{d, s, l\}$ that stand for *darker*, *similar* and *lighter* respectively. The labels are determined by the pixel intensity I_{b_j}, the central intensity at the saddle point ρ, and the method parameter offset ε as follows

$$L_{b_j} = \begin{cases} \bullet\, d, & I_{b_j} < \rho - \varepsilon \\ \bullet\, s, & \rho - \varepsilon \leq I_{b_j} \leq \rho + \varepsilon \\ \bullet\, l, & I_{b_j} > \rho + \varepsilon \end{cases} \tag{3.1}$$

The color of the dots in (3.1) corresponds to the color of the dots in the outer ring in Figs. 3.1 and 3.3 (b).

The test is passed if the outer ring contains exactly two consecutive arcs of each label l and d, the arcs are of length 2 to 8 pixels and are alternating – the l arcs are separated by d arcs. To eliminate instability caused by ρ-crossing between l and d arcs, up to two pixels can be labeled s at each boundary between l and d arcs. Labels s are pixels with intensity in ε-neighborhood of ρ, where ε is a parameter of the detector.

The test may seem complex, but in fact it is a regular grammar expression, which is equivalent to a finite-state automaton and can be implemented very efficiently.

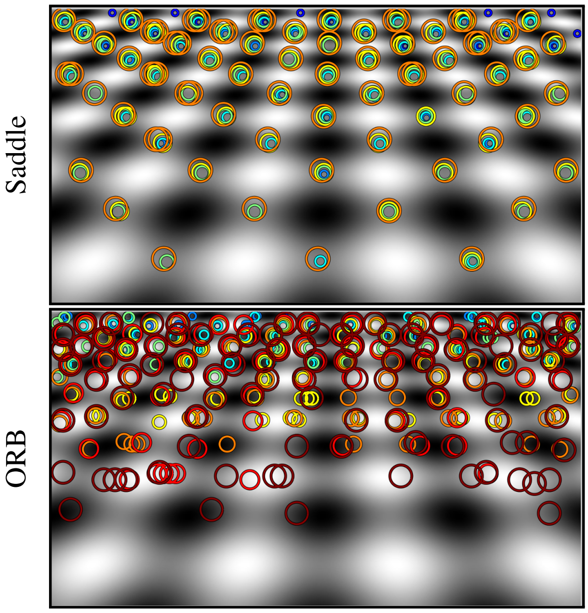


Figure 3.4: Detection on a 2D sinusoidal pattern under a perspective transformation. Saddle and ORB detections are shown as circles of the outer ring size.

Both, the inner and outer rings can be unwrapped to become a sequence of discrete elements (symbols) and be fed into a finite-state machine (FSM), known as the *acceptor* [57] that after receiving the last element of the sequence, it outputs the binary hidden state as accepted or rejected. The set of all possible accepted by the sequences belong to the regular language defined by the FSM.

3.2.3 Post-processing

Each point $\mathbf{p}$ that passed the alternating pattern test for both the inner and outer ring is assigned a response strength

$$R(\mathbf{p}) = \sum_{b_j \in B(\mathbf{p})} |\rho_{\mathbf{p}} - b_j|.$$

The value of the response strength is used in the non-maxima suppression step and to limit the number of responses if required.

The non-maxima suppression is only performed within one level of the pyramid, features at different scales do not interact as the scale pyramid is relatively coarse. This is similar to non-maxima suppression of ORB. For the non-maxima suppression, a 3×3 neighborhood of point $\mathbf{p}$ is considered.

As a final post-processing step, position refinement of points that passed the

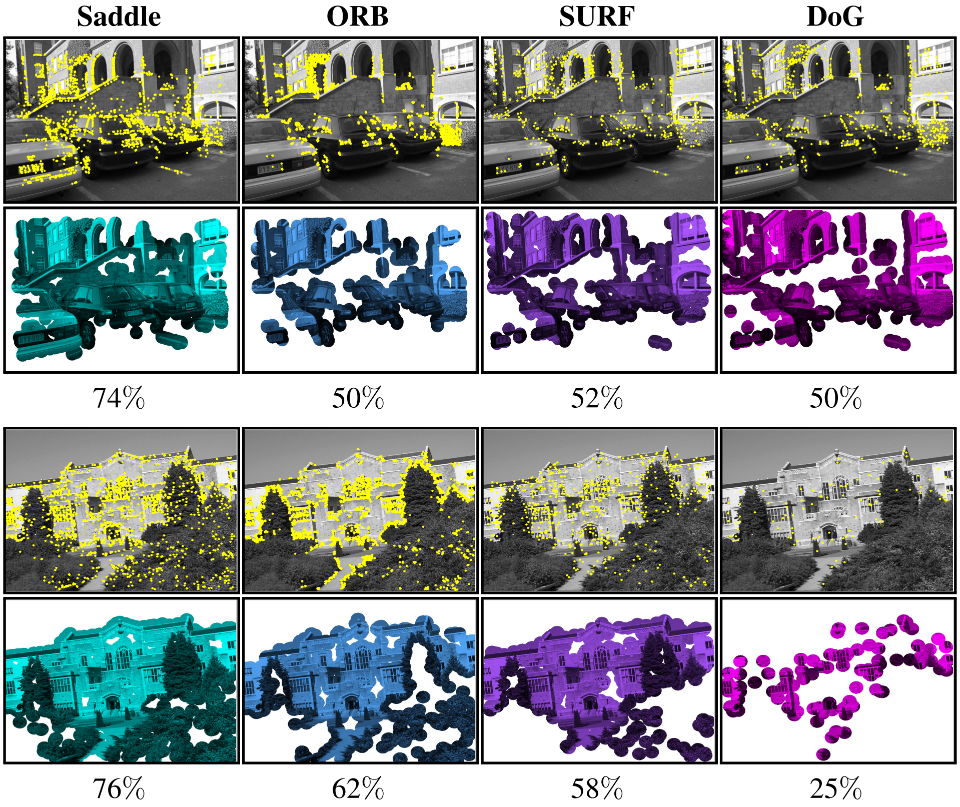


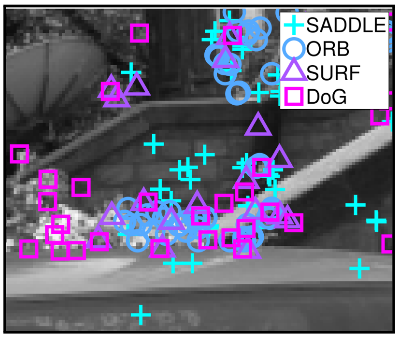
Figure 3.5: Coverage by ground-truth validated feature matches on selected image pairs from the Oxford dataset[122, 120]. In the rows with gray-scale images, the positions of the feature centers are marked with yellow dots. The covered area is computed as a union of circles with a 25 pixel radius centered on the matches. The areas are visualized in the rows of images with masks colored consistently with the detectors. Each column corresponds to the detector writen on the top and the percentage of the image area covered by the mask is writen at the bottom.

non-maxima suppression state takes place. A precise localization of the detected keypoint $\mathbf{p}$ within the pyramid level is estimated with sub-pixel precision. The x and y coordinates of $\mathbf{p}$ are computed as a weighted average of coordinates over a 3×3 neighborhood, where the weights are the response strengths R of each pixel in the neighborhood. Response of pixels that do not pass the alternating pattern tests is set to 0. The feature orientation is defined by the vector from the feature center to the intensity centroid [158], computed within the image patch of 31×31 pixels in the corresponding scale.

Examples of local regions that fire Saddle detector are presented in Fig. 3.7. The image pyramid used has 8 levels and the decimation factor is 1.3. The 4 features in the same row were chosen randomly along all detections that belong to the same level. Saddle tests are applied on a decimation pyramid of L levels (each image independently) which is computed resizing the input image as follows:

$$[u_i, v_i] = [u, v] * \sigma^{-i}$$

Figure 3.6: Positions of matched interest regions detected with Saddle, ORB, SURF and DoG showing the detection complementarity.



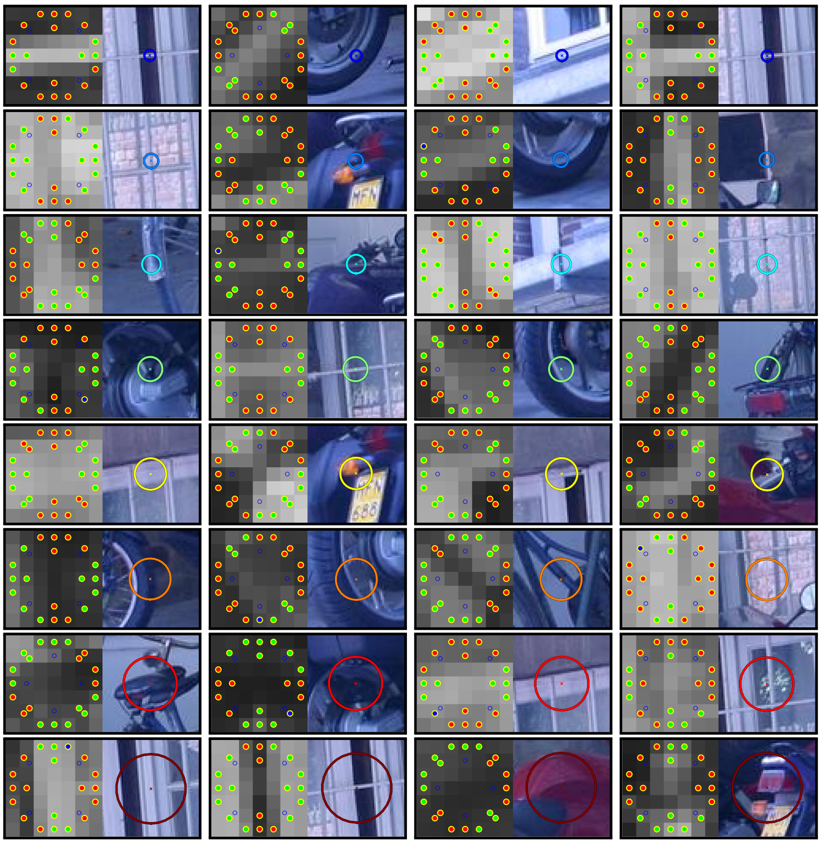


Figure 3.7: Saddle detections on a natural image. The left-side of each sub-image shows the inner and outer rings overlapped with the actual image patch observed by the detector, and the right-side shows the position and scale of the keypoint and its neighboring region on the original resolution. The color code of the circular geometries from the 1st to the 8th level of the pyramid are **blue**, **light blue**, **cyan**, **green**, **yellow**, **orange**, **red** and **brown** respectively.

where, u_i, v_i are the size of the i-th level, u, v are the size of the input image and σ is the scaling factor [164]. Since the Saddle detector is very dense for small values of ε (1 in our experiments), the number of features is bounded on each level of the pyramid decimation.

$$|\mathcal{F}_i| = |\mathcal{F}| * \left(\frac{1-\sigma^{-1}}{1-\sigma^{-L}}\right) * \sigma^{-i}$$

where $|\mathcal{F}_i|$ is the size of the feature set detected on the i-th level and $|\mathcal{F}|$ is the size of the union set of all levels [164]. In most of the natural images tested in our experiments, the number of detections on each level is higher than $|\mathcal{F}_i|$, hence the set of points must be bounded by ranking them with respect to their responses, i.e. for all points detected in i-th level only the $|\mathcal{F}_i|$ points with highest R are taken. The ranking step is efficiently implemented with the *quick-sort* algorithm that partially sorts the list of points where the top-$|\mathcal{F}_i|$ points have greater or equal response than the $|\mathcal{F}_i|$-th point. The response function can be interpreted as the contrast present in the center of the Saddle point. It is defined as the absolute difference of the two pixels that are nearest to the median of intensities corresponding to the position of the inner ring, i.e. the two pixels used to compute the median itself.

3.3 Experiments with Saddle detector

In this section, we experimentally evaluate the properties of the proposed Saddle detector. The performance is compared with a number of commonly used feature detectors on standard evaluation benchmarks. A more detailed description of the experiments regarding our proposed detector and the whole stereo matcher, see the paper [7] which is published as part of this research project.

3.3.1 Synthetic images

We first compare the properties of the Saddle and ORB detectors with three experiments on synthetically generated images.

First, features are detected on a chessboard pattern with progressively increasing blur, see Fig. 3.8. Saddle point detection is expected in the central strips, ORB detection on the corners on the right edge and potentially near the saddle points. Saddle features are repeatedly detected at all blur levels and are well located at the intersection of the pattern edges. ORB features are missing at higher blur levels and their position is less stable.

A phenomenon common to corner feature points – shifting from the corner for higher scales and blur levels is also visible. Note that since the scaling factor between pyramid levels of Saddle is 1.3 while for ORB it is 1.2, Saddle is run on a 6 level pyramid and ORB with 8 to achieve a similar range of scales.

Second, a standard synthetic test image introduced by Lindenberg and used in scale-space literature [101] is used, see Fig. 3.4. The Saddle points are output at locations corresponding to saddle points across all scales in the perspectively distorted $f(\xi, \eta) = \sin(\xi)\sin(\eta)$ pattern. Since there are no corners in the image, ORB detections are far from regular and are absent near the bottom edge. Fig. 3.6 shows the detector complementarity, i.e. Saddle fires on regions where other detectors have none detections. For the experiments in synthetic data, Saddle has the minimum arc length equal to 2 pixels, the maximum equal to 8 pixels, ε equal to 1 and the image pyramid with factor equal to 1.3 with 6 levels. On the other hand, the ORB setting has contrast threshold equal to 20 (preserved fix for later experiments) and the pyramid is built with a decimation factor equal to 1.2 with 8 levels.

As the last experiment with synthetic data, the behavior of the compared detectors is shown in two geometric patterns employed for testing the accuracy of corners detectors, likewise the SUSAN [178] and SFOP [53] detectors. For consistency, the parameters of the detectors are set identically and in order to avoid an overpopulated plot L is equal to 4, σ is equal to 1.3, ε is 21 and $|\mathcal{F}|$ is set to 200. The Fig. 3.9 presents the detections of ORB (top row) and Saddle (bottom row). Notice that the number of keypoints in the image of a Siemens star with regular beams (left column) reach the threshold L. The detectors fire in multiple levels due to the high contrast and sharped edges of the patterns, however Saddle fires consistently in the center of the image where the spatial frequency is higher and the beams at certain resolution look like intensity saddle points, thus the detections lie farther from the star center with the increasing scale. The image with gray scale polygons (right column) allows Saddle to fires rarely since there are no shapes similar to saddle points, nevertheless ORB overfires in the same image, even along the edges which is not convenient for stable tracking points.

3.3.2 Coverage of interest regions

Saddle and ORB detections are compared with respect to their spatial distribution on a set of 27 images with medium level of noise from Oxford-Affine [120] and HPatches [210]. The evenness of a spatial distribution of points in the scene that a given detector can locate with high repeatability is a desirable feature of the detector itself. Therefore, we propose to measure this property with two metrics of the normalized area in the image covered by the keypoints. The metrics are defined as follows, a mask is computed driven by the location of the detected keypoints and its area is normalized by the image size, thus for a completely covered image the metrics gives 1 and 0 for an empty feature set. However, the two metrics differ in the

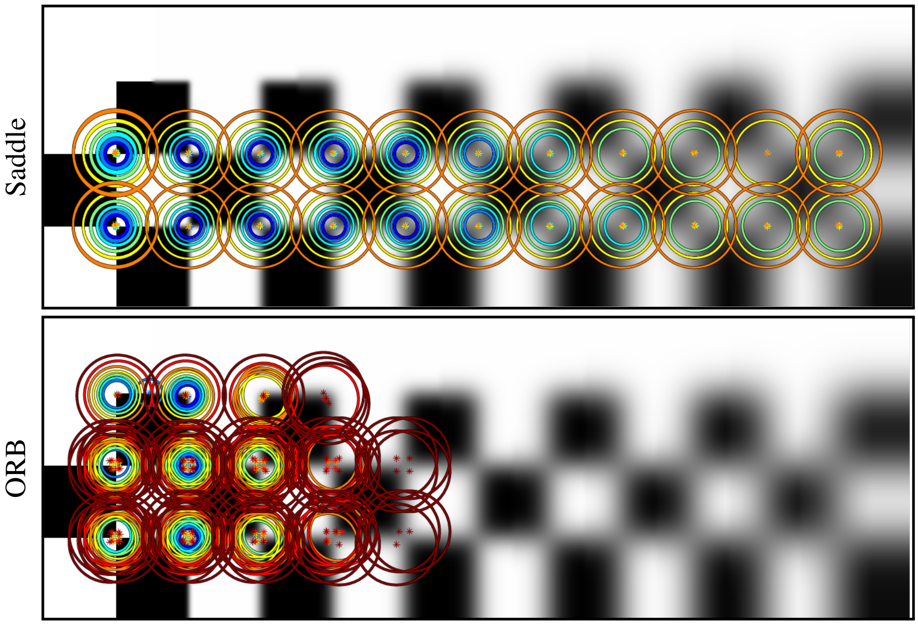


Figure 3.8: Detection on a progressively blurred chessboard pattern. Circle color reflects feature scale, its size shows the extent of the description region.

way the mask is computed. For the first metric, a circle with center in the keypoint position [70] and 25 pixel radius is drawn in the mask for all keypoints with fixed radius. Finally, the mask is computed as the union of circles. The second metric is inspired by [125] where it identifies the most commonly photographed parts of a building for image retrieval. In this case, the mask is computed considering the size of the keypoints, i.e. the scale. The radius of the circles is proportional to the measurement regions [146] (local regions used for description), again, the mask is the union of the given circles.

The heatmaps are defined as the accumulative number of circles drawn in the mask of the second metric, as examples, 6 images are presented in Fig. 3.10, in all images, Saddle covers a larger area than ORB, covering relevant areas for image registration or 3D reconstruction, i.e. the facades of the buildings. The experiment shows that Saddle fires at wider areas along different scenes. Fig. 3.11 shows the coverage obtained by running both detectors on the image set, fixing the decimation factor to 1.3, and Saddle gets a higher coverage on 100% of the images under the fixed radius circles approach with a mean absolute difference of 6.1%. The measurement region driven approach shows that Saddle has a larger coverage in 78% of the images with a mean absolute difference of 5.2% when Saddle has larger coverage and 1.6%, otherwise. Since most of the feature detectors can be tune to over-fire, we introduce an experiment where 4 detectors with equivalent setups and bounding the number of keypoints to $1K$ for the sake of consistency. In order to not biased the experiment the keypoints are selected with respect to the corresponding response, i.e. we take the top $1K$ strongest points. In addition the coverage is computed increasing the size of the feature set. The idea behind is to show the contribution in coverage meanwhile

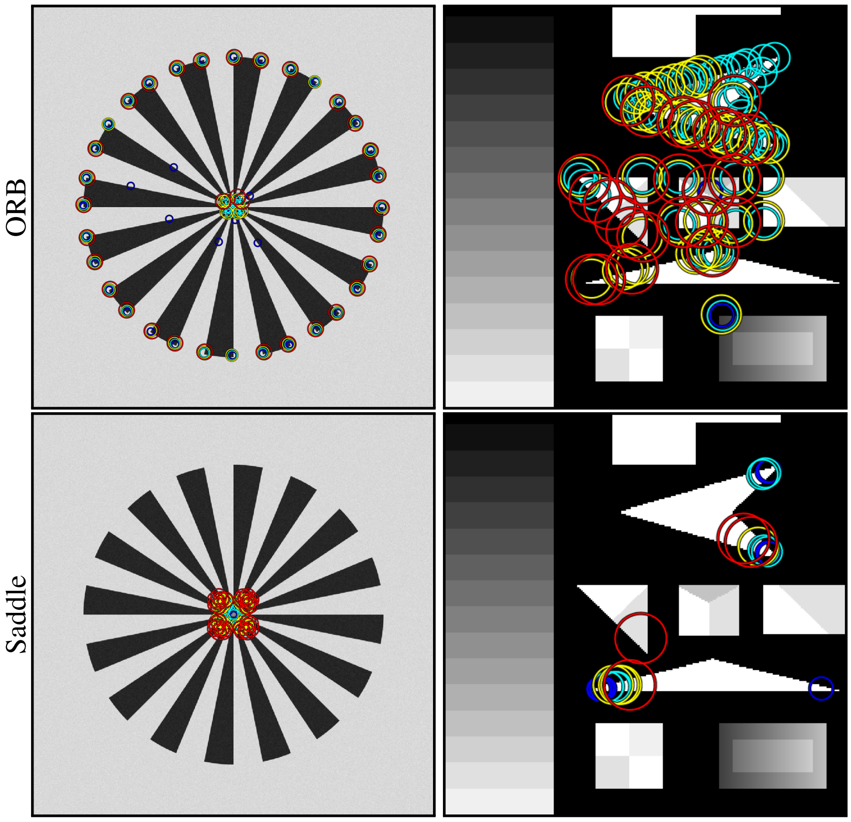


Figure 3.9: Saddle and ORB features detected in synthetic images designed for accurate location of corner detectors. The left column shows the geometric pattern proposed in SFOP keypoint detector[53] and right column shows the pattern proposed by the SUSAN corner detector[178]. For the sake of clarity, the image pyramid have 4 levels, the maximum number of features is fixed to 200 and the decimation factor is 1.3, for both detectors. The ε value is equal in both detectors. Notice that, in some cases that maximum number of features is not reached.

points with weaker response are added progressively. Fig. 3.12 shows that Saddle covers a larger area for a fixed set of features in comparison with other detectors.

3.3.3 Coverage of matched regions

In some task, such as structure from motion, good coverage of the image by matched point is crucial for the stability of the geometric models and consequently for the reliability of the 3D reconstruction [70]. Note that the coverage is a complementary criterion to the number of matched features, which is addressed in Section 3.3.5. A high number of clustered matches may lead to poor geometry estimation and to incomplete 3D reconstruction.

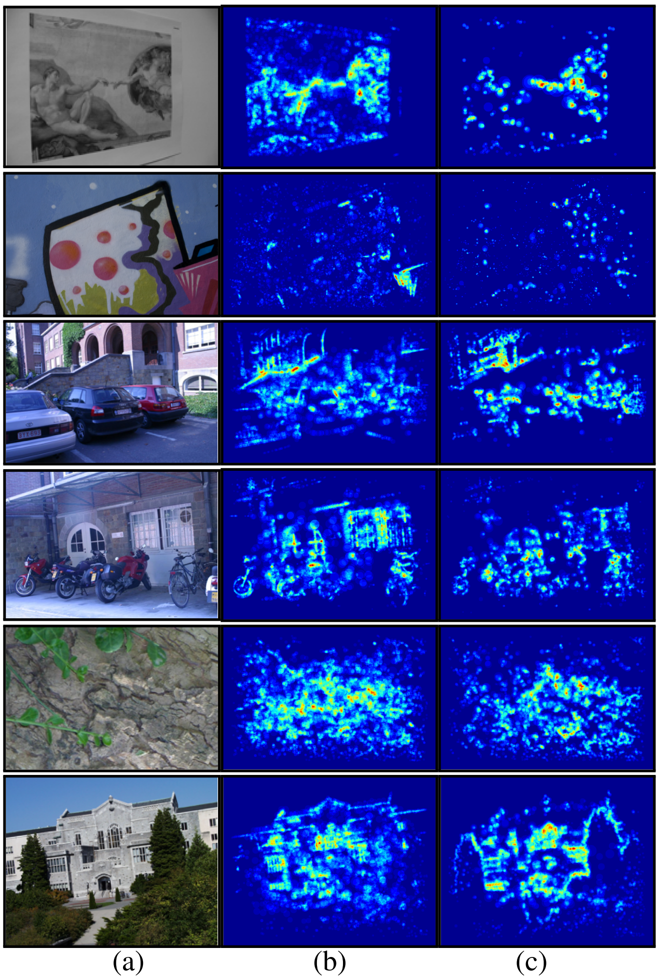


Figure 3.10: Heatmaps for Saddle and ORB detections. Column (a) shows the original images, columns (b) and (c) show heatmaps for Saddle and ORB, respectively. Heatmap represents the number of interest regions current pixel belongs to. The heat pseudo-color indicates the final countings along the feature sets.

To compare the coverage of different feature detectors, we adopt the measure proposed in [70]. An image coverage mask is generated from matched features. Every tentative correspondence geometrically consistent with the ground truth homography adds a disk of a fixed radius (of 25 pixels) into the mask at the location of the feature point. The disk size does not change with the scale of the feature. The matching coverage is then measured as a fraction of the image covered by the coverage mask.

Extensive experiments show that the proposed Saddle detector outperforms all other compared detectors: ORB, SURF and DoG. The covered areas are shown in Fig. 3.5. The superior coverage of the Saddle detector is visible on Fig. 3.13.

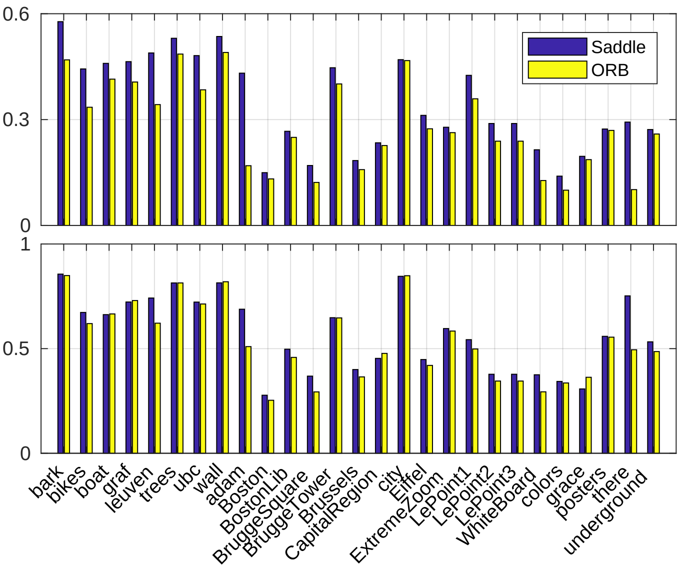


Figure 3.11: Coverage of keypoint locations in 27 images taken from OxAff[120, 122], EF[215], GDB[79] and SymB[62] datasets. The normalized area of the image covered by union (overlap) of fixed radius circles centered on the feature positions is shown in the top row, and the coverage computed by thresholding to 1 the number of features whose measure regions lie on each pixel is shown in the bottom row. Note that Saddle covers larger area in 100% of the images (upper row) and also it is higher in 78% for the second metric (bottom row).

3.3.4 Saddle position accuracy

The accuracy of Saddle was assessed on the Oxford-Affine dataset. The cumulative distributions of reprojection errors with respect to the ground truth homography of the *OxfAff* dataset is presented in [6] where Saddle is compared against its competitors. Saddle marginally outperforms ORB and DoG performance is superior in most cases. In fact, the reprojection error ϵ is defined as

$$\epsilon(\mathbf{p}, \mathbf{p}') = ||\mathbf{p} - \mathbf{H}\mathbf{p}'||, \tag{3.2}$$

where $\mathbf{p}, \mathbf{p}' \in \mathbb{R}^2$ are the XY position of a pair of points in correspondence from the reference and target images, respectively. The matrix $\mathbf{H} \in \mathbb{R}^{3\times3}$ is the ground truth homography that transforms the coordinate system of the target image into the reference one. A feature match is considered as an inlier if $\epsilon \leq 5$ pixels.

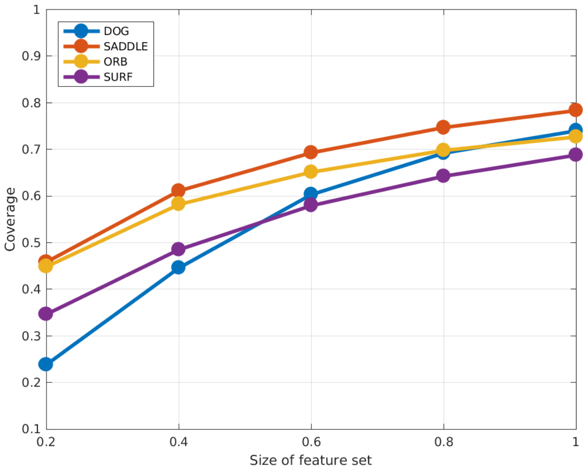


Figure 3.12: Average coverage of detected keypoints along all images of the *OxAff* dataset. The y-axis is the normalized coverage and the x-axis is the percentage of the complete keypoint set. Only $1K$ points are taken from each detector and points are selected by ranking them with respect to the response.

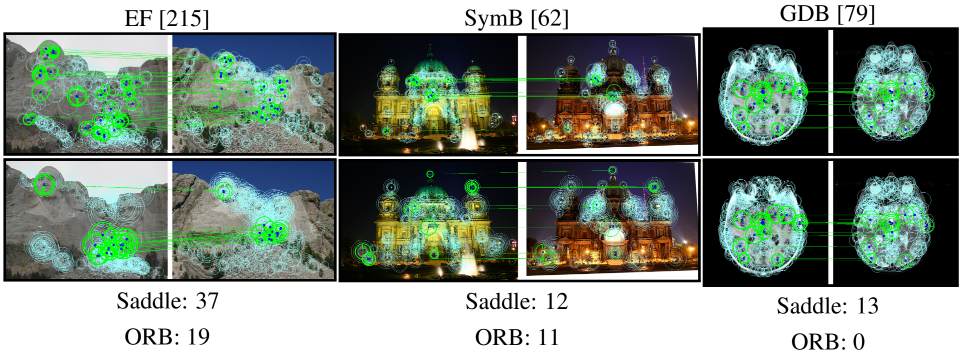


Figure 3.13: Detected and matched keypoints for Saddle (top) and ORB (bottom). The inliers count is given for both detectors for each image. Note that Saddle points are spread more evenly making the homography estimation more stable.

3.3.5 Matching ability

In this section we follow the detector evaluation protocol from [130]. We apply it to a restricted number of detectors – those that are direct competitors of Saddle: ORB [164], Hessian [123] (extracting similar keypoints) and SURF [18] (also known as FastHessian). The evaluation of the matching performance is done over 5 challenging dataset with natural images where the image pairs (cameras) have wide-baseline due to different type of nuisance. Details of the datasets is presented in the Tab 3.1.

We focus on getting a reliable answer to the match/no-match question for challenging image pairs. Performance is therefore measured by the number of successfully matched pairs, i.e. those with at least 15 inliers found. The average number of inliers provides a finer indicator of the performance.

Results are presented in two tables. Table 3.2 shows the results for a setup that focuses on matching speed and thus uses the fast [29] and FREAK [2] descriptors (OpenCV implementation). Saddle works better with FREAK, while ORB results are much better with BRIEF. Saddle covers larger area and on broad class of images (e.g. see Figure 3.13), but needs different descriptor than BRIEF, possible optimized for description of saddle points, i.e. selecting the binary questions similar to those of BRIEF but training with Saddle rather than FAST features.

In the experiment Saddle is run with a combination of RootSIFT [9] and Half-RootSIFT [79] as descriptors (see Tab. 3.3). This combination was claimed in recent benchmark [130] as best performing along broad range of datasets and it is suitable for evaluation of the matching potential of the feature detectors. The MODS-ORB and MODS-Saddle are added as state-of art matchers in their original setup, where ORB is replaced by Saddle. Most time is taken by description and matching.

Note that one could use both Saddle and ORB detectors and benefit both from their speed and their complementarity (last rows in Tab. 3.2).

Matching performance is compared on two additional datasets are used: A perspective change of planar scenes dataset following the same protocol as [122, 120] proposed in [39] and a dataset proposed by Lebeda in [92] which consist of 16 image pairs geometrical related by homography.

Short name	Description	Proposed by	#images	Nuisanse type
OxAff	Affine Covariant Regions	Mikolajczyk, 2013	8x6	Geom., blur, illum.
EF	Edge Foci Interest Points	Zitnick and Ramnath, 2011	8x6	geom., blur, illum.
GDB	Multi-modal and non-linear intensity	Kelman, 2007	22x2	illum., sensor
SymB	Local Symmetry Features	Hauagge and Snavely, 2012	46x2	appearance
HP	HPatches benchmark	Balntas, 2017	116x6	geom., illum.

Table 3.1: Datasets used in the evaluation of the detectors in the wide-baseline stereo task: OxAff[120, 122], EF[215], GDB[79], SymB[62] and HP[11]. The parameter variance between image pairs is described in the *nuisanse type* column.

3.3.6 Strategies for tentative matches

As an additional contribution of this work, we propose to match binary descriptors with the strategy called First Geometric Inconsistent Nearest Neighbor (1GINN) using Hamming distance. The 1GINN strategy was proposed in [132] to match floating point descriptors with Euclidean distance. The experiments in [132] show

Detector	# Features	Scale factor	Descriptor	EF			Oxford			SymB			GDB			HPi			HPv		
				#	time [s]	inl.	#	time [s]	inl.	#	time [s]	inl.	#	time [s]	inl.	#	time [s]	inl.	#	time [s]	inl.
ORB	500	1.2	rBRIEF	7	0.1	33	34	0.1	127	6	0.1	65	6	0.1	53	236	0.3	95	264	0.3	101
Saddle	500	1.3	rBRIEF	16	0.2	40	34	0.2	133	12	0.3	46	10	0.4	52	248	0.5	109	267	0.6	95
ORB	500	1.2	FREAK	9	0.1	32	35	0.1	100	6	0.2	54	5	0.2	49	219	0.4	74	267	0.4	97
Saddle	500	1.3	FREAK	9	0.2	27	31	0.2	96	12	0.2	31	6	0.3	44	212	0.5	79	256	0.7	86
ORB	1000	1.2	rBRIEF	20	0.1	39	37	0.1	240	22	0.2	50	9	0.2	84	269	0.3	173	280	0.4	195
Saddle	1000	1.3	rBRIEF	20	0.3	71	36	0.2	269	17	0.3	71	8	0.5	104	255	0.6	207	277	0.7	190
Saddle+ORB	500+500	1.3/1.2	rBRIEF	23	0.5	50	36	0.6	259	16	0.7	64	9	0.9	88	264	0.7	193	277	0.9	193
ORB	1000	1.2	FREAK	9	0.2	54	34	0.2	207	10	0.2	72	6	0.2	99	234	0.4	146	268	0.5	198
Saddle	1000	1.3	FREAK	11	0.2	57	34	0.3	193	11	0.3	63	4	0.4	112	225	0.6	144	265	0.7	177
Saddle+ORB	500+500	1.3/1.2	FREAK	12	0.2	63	34	0.3	196	11	0.3	67	7	0.4	92	230	0.7	142	263	0.8	180

Table 3.2: Saddle evaluation with fast BRIEF and FREAK descriptors. The subcolumns are: the number of successfully matched image pairs (left), average running time (all stages: read image-detect-describe-match-RANSAC), average number of inliers in matched pairs (right). Darker cell background indicates better results.

Detector	# Features	Descriptor	EF			Oxford			SymB			GDB			HPi			HPv		
			#	time [s]	inl.	#	time [s]	inl.	#	time [s]	inl.	#	time [s]	inl.	#	time [s]	inl.	#	time [s]	inl.
ORB	500	SIFT	6	0.4	32	33	0.6	116	8	0.5	46	5	0.5	60	226	0.9	80	240	1.2	98
Saddle	500	SIFT	11	0.9	34	33	0.6	107	11	1.1	42	4	0.8	57	229	1.2	84	244	1.6	93
ORB	1000	SIFT	15	0.7	34	35	1.1	229	16	0.9	59	7	0.9	9	255	1.5	156	261	2.0	183
Saddle	1000	SIFT	15	0.7	44	34	1.1	226	16	1.0	59	7	1.5	71	255	2.2	154	258	3.3	176
Saddle+ORB	500+500	SIFT	11	1.1	48	34	1.5	221	15	1.3	59	8	1.5	69	252	1.5	143	260	2.0	174
MODS-ORB	n/a	mix	33	0.6	34	40	0.2	148	44	2.6	37	18	2.2	73	285	0.2	97	295	0.2	123
MODS-Saddle	n/a	mix	33	0.7	36	40	0.3	143	43	2.5	34	20	1.7	69	285	0.6	102	295	0.6	108

Table 3.3: Saddle evaluation with a combination of RootSIFT and HalfRootSIFT descriptors. The subcolumns are the same as in Table. NMS stands for spatial non-maximum supression, indicating its application. In MODS-S, ORB was replaced by Saddle+FREAK, other parameters kept original. Darker cell background indicates better results.

that the 1st to 2nd descriptor distance ratio degrades its performance when multiple observations of the same feature are present. The matching approach with view synthesis [130] is the responsible for creating multiple and similar descriptors of the same point of interest thus the 1GINN compares the first closest descriptor distance with the distance to the descriptor that is geometrically inconsistent with the first one. The descriptors in one image are geometrically inconsistent if the Euclidean distance between centers of the regions is $\geq N$ pixels, in our experiments we set N equal to 5.

In the literature we can find that binary descriptors are matched [29, 164, 97, 2], as an standard practice, with the strategy called First Mutually Nearest Neighbor (1MNN) with hard thresholding of Hamming distance[140, 188]. Our proposed matching strategy avoids dropping tentative matches by the distance ratio of multiple instances of the same region, and speed is not hurt because the Hamming distance is computed very efficiently on CPU architectures. Even that in our matching task there is no view synthesis involved, the keypoint instances are replicated because the non-maximum-suppression is not performed across scales (3-Dimensionally). As a consequence, the same point in the scene fires the detector in more than one level of the pyramid thus in the final feature set there are very similar descriptors corresponding to the same spatial location.

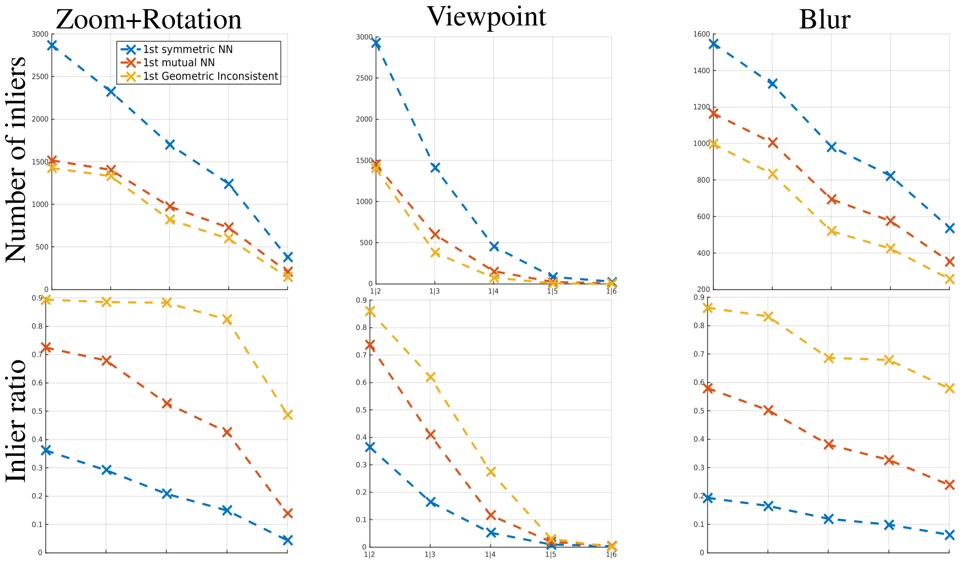


Figure 3.14: Matching strategies for rBRIEF descriptor: Number of correct matches (left column), the inlier ratio (right column) over two sequences where the imaging nuisances are the change on zooming and rotation (upper row) and the viewpoint (bottom row). Strategies: 1st symmetric (1SNN), 1st mutual (1MNN) and 1st geometric inconsistent (1GINN) nearest neighbor.

The matching performance of Saddle features described by rBRIEF is tested on sequences of [122], where the imaging changes are zooming, rotation and perspective. Notice that rBRIEF [164] is an improved approach of BRIEF [29] and it is used inside of ORB. rBRIEF uses a greedy algorithm for searching the set of the least uncorrelated and with means nearest to 0.5 binary tests among all possible ones inside a 21×21 image patch. The learned binary descriptor has a significant improvement in the variance and correlation over the baseline implementation.

In addition to the 1MNN and 1GINN matching strategies, the First Symmetric Nearest Neighbor (1SNN) strategy is included in the comparison. 1MNN and 1SNN are distantance threshold based and 1GINN is ratio based. The results are shown on Fig. 3.14. Notice that the number of correct matches is significantly higher for 1SNN, which is expected since by construction the tentative correspondences are computed from the union of the left-to-right set and right-to-left set of tentatives, while 1MNN is the intersection (more strict rule) between sets and 1GI consists on left-to-right set only. However, the inlier ratio is significantly higher for 1GINN which leads to a faster convergence of RANSAC without sacrificing speed on descriptor dissimilarity computation. Results on blaring sequences are consistent and similar to the previous ones. Notice that 1MNN and 1SNN are both computed by hard thresholding of the Hamming distance, showing that the standard matching of binary descriptors is outperformed by the 1GINN.

Min. arc	Max. arc	ε	Num. levels	Decimation factor	EF			GDB			Oxford			SymB			HPv			HPi			Solved pairs
					#	time $[s]$	inl.	#	time $[s]$	inl.	#	time $[s]$	inl.	#	time $[s]$	inl.	#	time $[s]$	inl.	#	time $[s]$	inl.	%
2	6	1	7	1.3	14	0.3	37	11	0.5	44	36	0.3	238	17	0.4	26	284	0.4	180	264	0.3	188	87.80
2	7	1	8	1.3	16	0.3	37	10	0.5	42	37	0.3	243	18	0.4	28	276	0.4	179	266	0.3	186	87.38
2	6	1	9	1.3	17	0.4	38	9	0.5	40	37	0.3	238	19	0.4	29	280	0.4	175	261	0.4	193	87.38
2	5	5	7	1.3	16	0.3	35	12	0.4	42	37	0.3	226	16	0.4	25	283	0.4	167	259	0.3	176	87.38
2	7	5	8	1.3	16	0.3	38	10	0.5	41	37	0.3	242	19	0.4	28	281	0.4	178	260	0.3	183	87.38
2	6	5	8	1.4	17	0.3	35	12	0.4	44	37	0.3	251	20	0.4	28	279	0.4	163	258	0.3	184	87.38
3	7	1	7	1.3	19	0.3	41	9	0.5	40	38	0.3	233	18	0.4	27	278	0.4	166	260	0.3	179	87.24
2	8	3	7	1.3	15	0.3	37	10	0.5	43	37	0.3	243	18	0.4	28	279	0.4	181	263	0.3	186	87.24
2	5	3	8	1.3	17	0.3	38	13	0.5	44	36	0.3	225	16	0.4	25	280	0.4	166	260	0.3	177	87.24
2	6	5	7	1.2	17	0.3	38	12	0.5	45	37	0.3	227	14	0.4	25	279	0.4	175	262	0.3	174	87.10

Table 3.4: Performance of multiple setups for the Saddle detector. A setup is defined by the minimum and maximum arc lengths of the significant darker/brigther outer ring pixels, the significant constrast threshold (ε), the number of levels in the decimation pyramid and the decimation factor.

3.3.7 Saddle and Hessian intersection

Saddle is intended to fire in the positions where the negative determinant of the Hessian matrix is a local maximum, thus parameter settings are fixed in order to fulfill the inner and outer patterns in the same positions in space and scale of a Hessian pyramid. In order to find the Saddle's setup with the best matching performance, the 713 image pairs included in the six dataset presented in Tab. 3.1 were matched with MODS [132] using Saddle as detector and rBRIEF as descriptor. A setup involves: Minimum (m) and maximum (M) arc length of the significantly darker/brighter pixels in the outer ring, the intensity threshold *epsilon* (ε), the number of levels in the decimation pyramid (L) and the decimation/scale factor (σ), then each image pair with more than 15 correct correspondences is regarded as a solved problem. The setups were ranked by the number of solved problem across all datasets and the top 10 best setups are shown in Tab. 3.4 with average matching time and number of inlier correspondences on 4 datasets.

In order to investigate whether there is a correlation between the number of solved problems and the precision of Saddle locations with respect to the local minima of the Hessian response for a given parameter setup, the Saddle-Hessian intersection metric is proposed in a *precision recall* fashion as follows

$$recall = \frac{\#\ caught\ hessians}{\#\ hessians} \quad \text{and} \quad precision = \frac{\#\ caught\ hessians}{\#\ Saddles},$$

where a Hessian feature is regarded as *caught* if its measurement region overlaps (intersection over union) with the closest Saddle region, in euclidean distance to the feature centers, is larger than 40% likewise the overlapped regions shown in Fig. 3.19. The metric is computed on the second image of each pair, i.e. the image

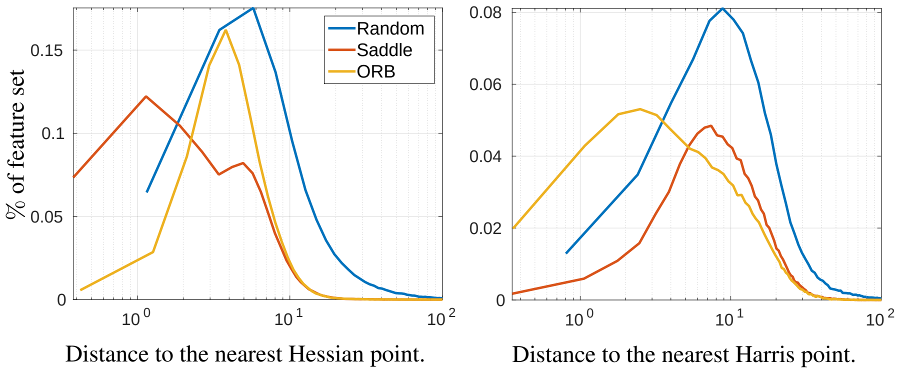


Figure 3.15: Normalized histograms of euclidean distances from Random, ORB and Saddle points to the nearest reference feature points, top – Hessian, bottom – Harris. Distances are in pixels.

matched against the reference one. Finally, the mean *F1 score* (defined in Eq. 3.3) is computed across all image pairs.

$$\mathit{F1\ score} = 2 \cdot \frac{precision \cdot recall}{precision + recall} \tag{3.3}$$

Since Saddle interest regions are supposed to be posed on intensity saddle points in the image and ORB regions on corners, their spatial distribution on the image are expected to be consistent with Hessian response minima and Harris response maxima, respectively. The spatial distributions of Saddle and ORB are represented by the histograms of Euclidean distances from the feature center to the nearest Hessian (with negative determinant) and Harris feature centers, which are shown on Fig. 3.15 with the addition of the distribution of distances to random points. Note that as expected, the hand-crafted feature detectors fires on the corresponding kind of interest points, i.e. saddles and corners.

3.3.8 Repeatability

In this section we evaluate the repeatability and the number of correspondences with the benchmark of [123], where the interest regions detected on the reference image are reprojected on the target images with the ground truth homography that geometrically relates the image pair. For each pair, the set of correspondences is formed up with the feature pairs with a normalized overlap larger than 60% and the repeatability is defined as the number of correspondences normalized by the cardinality of the smallest feature set.

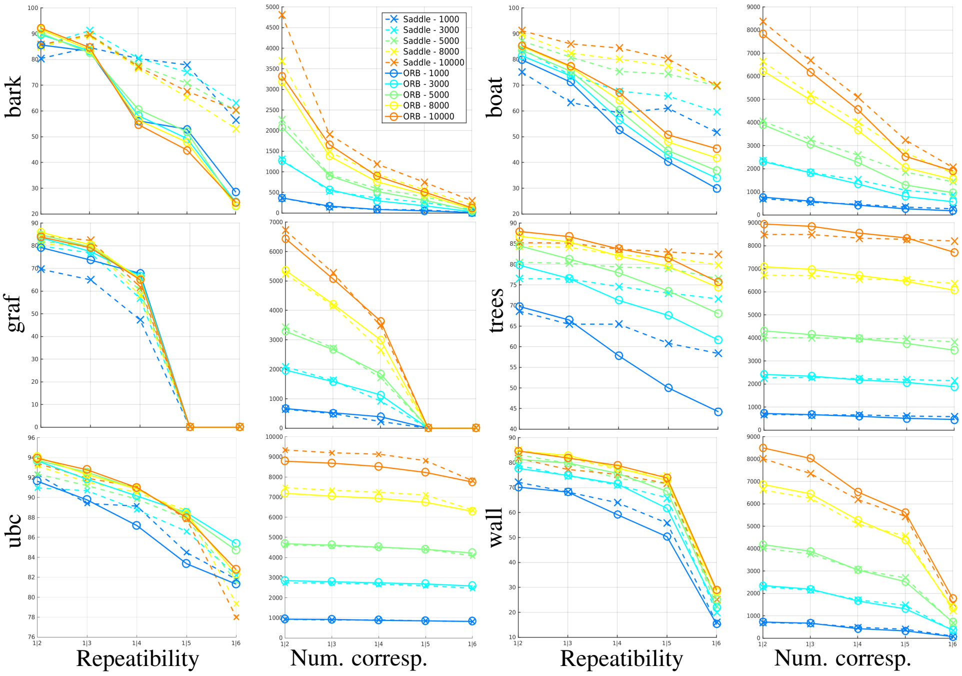


Figure 3.16: Repeatability and correspondences on the *OxAff* dataset. Features from j-th view are reprojected to the reference image (i) with the ground truth homography that relates them. Image pairs $i|j$ are indicated on x-axis.

The *OxAff* dataset is used for evaluation and the number of features is bounded for each experiment, following the criterion described on Sec. 3.2.3, to test the matching power with respect to the size of the feature set. The results for 6 sequences are shown in Fig. 3.16, where the x-axis indicates the image pair $1|j$ for the j-th target image.

The experiments show that Saddle outperforms ORB in the sequences where the nuisance is *zoom+rotation* (bark and boat) for both repeatability and correspondences. For changes on *blurring* (trees) ORB is slightly better on the easiest pairs of the sequence and Saddle performs better for the hardest pairs. *JPEG* compression (UBC) and reprojection affect the correspondences of both detectors almost identically but it hurts the repeatability of Saddle more. Finally, the changes in *viewpoint* (graffiti and wall) affects both to the same degree building correspondences, Saddle has better repeatability on scenes that are similar to wall and worst repeatability when it is similar to graffiti. Note that for changes in viewpoint the differences in performance are almost despicable.

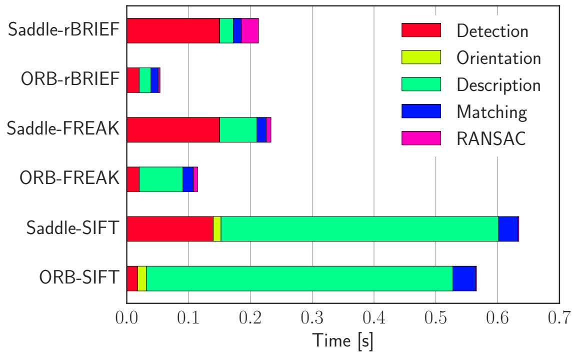


Figure 3.17: Average run-time for ORB and Saddle on the Oxford-Affine dataset (average image size is $\approx$ 900 $\times$ 600 and average number of features is $\approx$1000).

3.3.9 Speed

The time breakdown for Saddle and ORB image matching on the Oxford-Affine dataset is shown in Fig 3.17. Saddle is about four times slower than ORB in the detection part. However, we have not utilized SSE instructions in the Saddle tests. The results show that both Saddle and ORB are faster than the FREAK descriptor, but significantly slower than BRIEF. The slowest RANSAC step is observed when Saddle is described by BRIEF, the time increase because RANSAC is reaching the maximum number of iterations when it fails to match an image pair. For each of the two cross inner tests of Saddle, 75% of the points require 2 comparisons, 5% require 3 and 20% require 4 comparisons. 15% of the points are accepted after 4 comparisons. Percentages refer to the total number of neighborhoods tested.

3.3.10 Photo-tourism stereo matching

In this section we present results of large-scale detector evaluation recently proposed in [12]. The dataset consists of 11 sets, 100 images each. Up to 8K keypoints and descriptors are extracted in each image, then the exhaustive matching is applied. The obtained correspondences are then fed into RANSAC which estimates essential matrix between each pair of images. Then the obtained essential matrices are compared to ground truth matrices. Unlike other benchmarks, this one measures not the single property of the detector – like repeatability or coverage, but performance for the end task - recovering camera pose.

Because of the wide baseline, the binary descriptors are performing poorly in the benchmark, so we used near state-of-the-art HardNet descriptor [129] for all the detectors.

All the images are upright, so orientation estimation procedure was turned off for all the setups. Thus, the results show the pure detector performance.

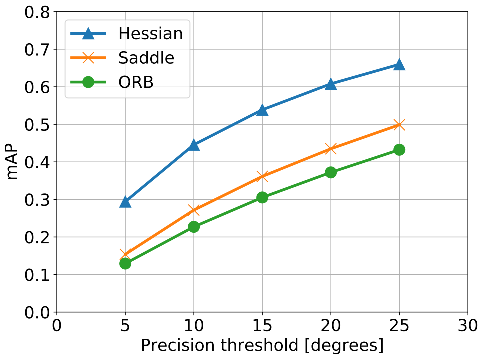


Figure 3.18: Detectors comparison on recovering camera pose difference on Phototourism Challenge [12]. Mean average precision at different precision thresholds is reported.

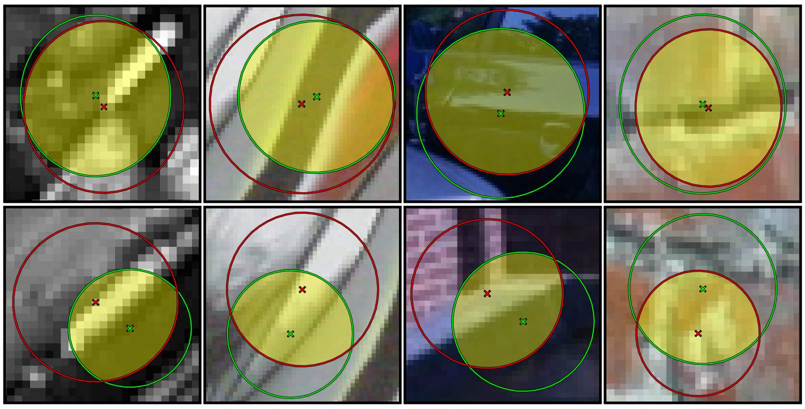


Figure 3.19: Examples of Hessian (green) and Saddle (red) regions overlap. The circles represent the regions used for description and the colored area (yellow) shows their intersection. The two feature pairs presented in each column were selected as the maximum and minimum normalized overlaps, whether exists, found in the same image.

We report the results for the Saddle, ORB and Hessian detectors in Figure 3.18. Both Saddle and ORB are worse than Hessian, but Saddle outperforms ORB (FAST) for all precision thresholds.

3.4 Discussion

Experiments show that the Saddle features are general, evenly spread and appearing in high density in a range of images. The Saddle detector is among the fastest proposed. In comparison with detector with similar speed, the Saddle features show superior matching performance on number of challenging datasets. Compared to recently proposed deep-learning based interest point detectors and popular hand-crafted keypoint detectors, evaluated for repeatability in the *ApolloScape* dataset [67], the Saddle detectors shows the best performance in most of the street-level view sequences also knowns as traversals.

After the exhaustive evaluation of Saddle, we conclude that its performance is suitable to be integrated in the relevant shot detection pipeline for video re-ranking. Saddle is a fast and reliable feature detector that enable super real-time performance and the usage of binary descriptor brings the advantage of fast similarity metric, like Hamming distance, and compact models. Finally, Saddle is part of the local region of interest detection component among other detectors. The performance of the pipeline selecting Saddle is presented in Sec. 4.9.2. The following chapter describes the full pipeline for video re-ranking.

Chapter 4

Relevant shot detection

In this chapter we focused on the main problem addressed in the dissertation, the full pipeline for scoring videos by relevance with respect to a object of interest. The query, at the very first stage, is defined as a string that contains textual information about the object and later is enriched with visual information of related images after multiple filters to suppress outliers. The video list returned by the server is re-ranked after scoring the each video with the final objective of having at the top of the list the videos with longer exposure of the query in the screen. A suitable use case of this approach is to establish a link between two important source of information, Wikipedia and YouTube. The first one provided by textual information and images embedded in the documents/pages and the second one a massive database of videos, currently without a mechanism capable of relating their contents.

In the Section 4.1 we present an outline of all components that are part our proposed approach, the description of the workflow and the context that support the need of our fast feature detector presented in details in Chapter 3. We also present in Section 4.2 the novel dataset used as the source of images to describe the visual appearance of the target object and how we propose to index the dataset for textual retrieval. In the Section 4.3 we discuss the approach for indexing the image database in a different fashion, the goal is to build a search engine for image retrieval using a state-of-the-art global descriptor based on deep features. The three algorithms for modeling the query based on local features is present in the Section 4.4. We discuss briefly in the Section 4.5 the usage of the API provided by the video sharing web page to query the database and the implementation of the video acquisition component. The same section describes the representation of the videos in order to perform fast search on them. The algorithm to analyze a video frame in order to test the presence of the query in the captured scene is presented in the Section 4.6. In the Section 4.7 we introduce our novel shot boundary detector which performs a primal task in video applications, it segments the video sequences into shots. Additionally, in the Section 4.8 we discuss our interface embedded into a web page that access pipeline and is provided with tools for fast access to relevant shots. Finally, in the Section 4.9 we present the experiments related to the evaluation of the full pipeline.

4.1 Overview

In our proposed method, we are not interested in indexing a large and fixed corpus of videos, since it is not realistic to preprocess all videos on the Internet, at least not for us. Instead, we relay on text-based search capabilities provided by video sharing websites, for example, YouTube. A short-list of videos is obtained by querying the search engines of the website, however, the relevancy of such videos is noisy because the search engine does not check the content of the videos.
We propose to perform an efficient visual content-based verification on the fly, in order to re-rank the initial short-list. This document focuses on the object model building from a set of images and on efficient on-line detection of the object in videos. The method is summarized in Fig. 1.1.

The first version of the approach was introduced in [4], a work where we propose a pipeline for retrieving videos from a website by full-text search of the user input string and, additionally, the user must provide a set of images of the object of interest in order to sort the list of videos with respect to their visual content. The pipeline simplifies the user interaction asking only for the input string for searching images and videos simultaneously. In this thesis we present contributions built on top of the previous work. The images are suggested automatically to the user by means of a full-text search engine implemented for this task. The visual assessment uses images provided from a novel dataset built from documents of a community-driven encyclopedia and semantic data from a community maintained ontology.

Due to the textual search of images is noisy by its own discriminative power limitations, we propose to suppress inlier images by means of an image retrieval engine based on state-of-the-art deep features. The model building stage in the pipeline is improved with the support of multiple local image feature detectors and descriptors to perform the image registration, feature reprojection and clustering. Since the object detection method lies on stereo matching, we propose a novel shot boundary detector that avoid the exhaustive frame comparison selecting frames with a line search method followed by a binary search, and the similarity function applies the wide baseline stereo matching, enabling the precomputation of the local features required for the posterior stages. Another contribution is the addition of frame-by-frame manual annotation of the 10 landmarks dataset introduced in [4], in order to quantify the performance of the object detector method with precision-recall curves. Finally, a graphical interface for a WEB demo is provided as part of the pipeline. The interface enables the user to access specific shots where the query object is depicted by means of time stamps.

4.2 Image database for modeling

The collection of images called *Landmarkdb*, which is used to build the visual model of the object of interest, comprehends images downloaded from *Wikipedia* pages. Only pages of objects regarded as landmarks are considered to be processed, i.e. download the images included in the page, captions and additional information that describe the object and the content of the page. The classification is performed with an algorithm that identifies landmarks reliably. In order to consider an object to be a landmark, [72] introduced the following definition:

Definition 4.1. A landmark in Wikipedia, among other datasets, must satisfy the following conditions:

- To have a title.
- To have GPS coordinates.
- To have an immutable location.
- To be well-bordered.

A Wikipedia article has a table in the top-right corner of the page called *info-box* which contains structured information. The info-box has a template name, common to all info-boxes of of similar topic. On top of these template names, a hierarchy is built using *DBpedia* as a method for template categorization. The target is to build a semantic interpretation layer on top of Wikipedia articles exploiting data stored in the info-boxes. DBpedia sorts the task out using a collaboratively edited mappings from the info-box templates to their fields and from article classes to their properties. Additionally, DBpedia contains a community maintained *ontology* which is a set of relations among article categories and their descriptions, creating a knowledge base capable to be processed by computer.

All the objects registered in the ontologies pass through the landmark classification giving a set of 357K landmarks are identified among all Wikipedia and Wiki-data pages in 390 languages. The images included in the corresponding article are downloaded to conform a dataset of ≈ 1.1M images. Some textual information is related to the images, i.e. their filenames and captions. At this stage, it is presumed that actual name of the landmark is contained in at least one of those fields. Such information is indexed in a textual database and loaded in a search engine that allows fast query of the string which is input data of the retrieval pipeline. Not all the images in *Landmarkdb* are reachable by the text-based search because the caption is not available in all Wikipedia images or the text doesn't describe the captured scene properly. In addition, the filenames are very noisy and they do not necessary contain the related landmark name. Examples of textual queries and the retrieved images by the search engine are shown in Figure 4.1.

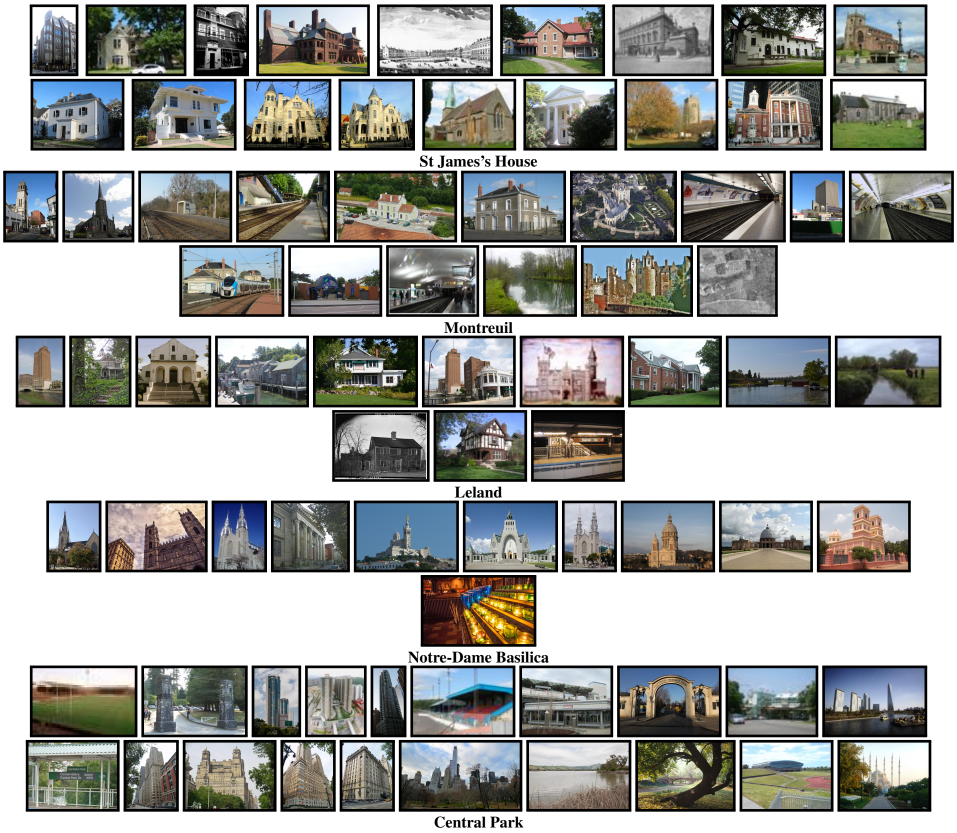


Figure 4.1: The *Landmarkdb* is queried with the string shown at the bottom of each row. A subset of the images embedded with Wikipedia article are presented above.

The retrieved images by the text search engine are very noisy due to the ambiguity of the input string, the lack of precision and subjectivity of the description and captions. Note in Fig. 4.1 that images do not show the same landmark under the same query and neither of them are related with the same object/place. The accuracy in the retrieval is increased including visual information in the search, i.e. an image retrieval approach is implemented for this purpose and it is described in the following section.

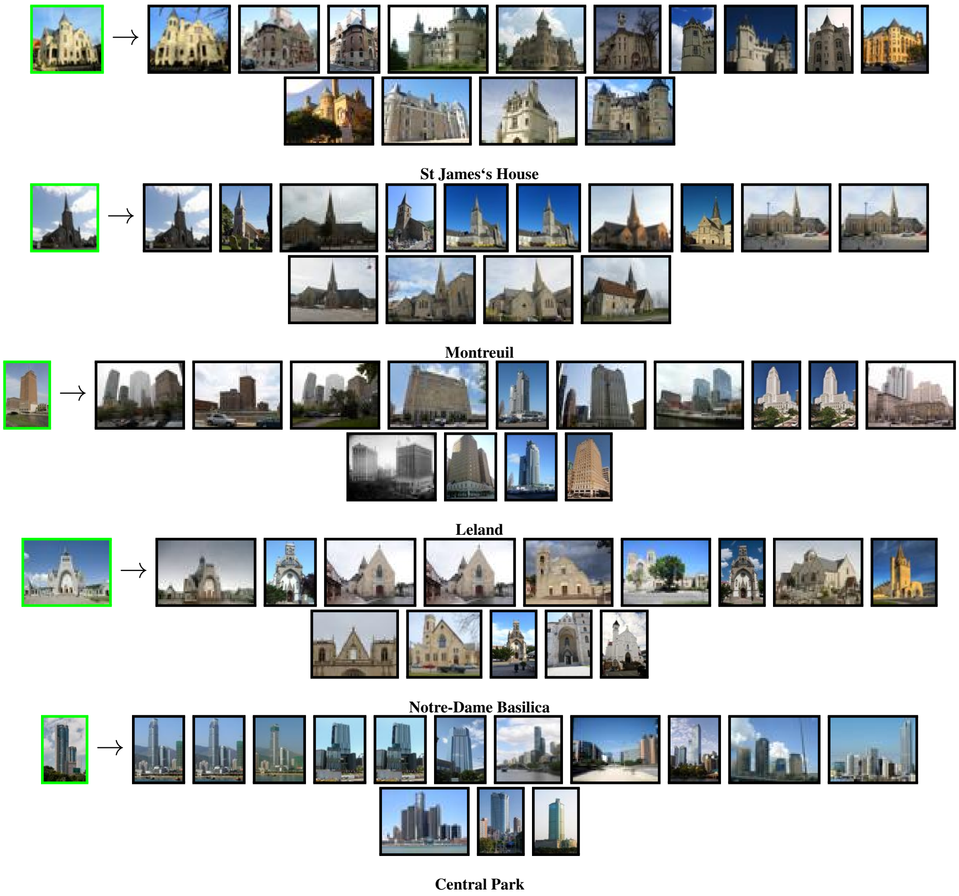
4.3 Image retrieval engine

Once the initial subset of images is retrieved by full text search with the input string as query, we require an outlier suppression step before constructing the model with such images. In this context, an outlier is an image that does not contain the query, the interesting object for the user is not observable in the image. We propose a semi-automatic process with two parts. First, the tentative relevant images are displayed and the user is supposed to select one of them in the manner of manually annotation. In the object modeling step, the selected image is the reference frame where features of additional views are reprojected.

The second step involves an image retrieval engine which by construction introduces visual information into the query. Basically, the image selected by the user is the query of the content-based search. The result is a larger set of similar images that are used to build the model and therefore search for it in the videos.

The database of the retrieval engine is the full *Landmarkdb* dataset (1.1M images) indexed with the state of the art CNN global descriptor proposed in [153]. In our pipeline, the network is used for inference only with the off-the-shelf model provided by the authors. Some details of our setup are: 1) The images fed into the architecture are resized to 1024 pixels along the largest dimension, 2) the image description is done with multi-scale and global representation, 3) the images used to train the model were selected automatically by the Structured-from-Motion 3D reconstruction system introduced in [168, 152], 4) as part of the non-linearities of the regressor, the *Generalized Mean* (GeM) is applied over the scales in the pooling layers of the ResNet101 [63] architecture. ResNet101 is the baseline model chosen and fine-tuned with hard-negative mining, and finally, 5) the fine-tuned GeM vectors are post-processed for whitening and dimensionality reduction using linear discriminant projections learned as Mikolajczyk and Matas proposed in [116], leaving global descriptors of 2048 dimensions.

Fig. 4.2 shows 5 queries of the CNN image retrieval engine. The images with green frame at the left side of the figure are the image queries. Basically, the queries are the visual definition of the object of interest introduced by the user. At the right side, we present the top 14 most similar images scored by a dot vector product of the descriptors. However, even that the images are visually similar they do not always contain the same object therefore and additional outlier suppression step is required and it is included as the object modeling step that is described in details in the following section.



St James's House

Montreuil

Leland

Notre-Dame Basilica

Central Park

Figure 4.2: Image retrieval results. The user selects the left image as the query (**green**) and the short list of relevant images in the *Landmarkdb* based on the visual content are shown ranked from left to right. The landmark are consistent with Figure 4.1.

4.4 Query object modeling

In this section, the process of the object model construction is described. Our model is a collection of local interest regions detected in the images that presumable contain the object of interest. Rather than working with a single image as query, we use a set of relevant images retrieved by the previous pipeline stage, i.e. the image retrieval engine with deep CNN global descriptors. With multiple views of the same object we get a larger and more diverse feature set as representation. We assume some parts of the object are not represented with a single view of the object, i.e. counting with only one camera pose to capture the object appearance can lead to incomplete coverage caused by temporal occlusion caused by moving objects or partial occlusion due to the camera pose itself. Additionally, the image quality can be compromised by noise in the pixel intensities, illumination changes, etc. These factors reverberate the descriptive power of the query, therefore, we propose to lessen the performance hurting effects by adding more images to the model with different perspectives of the object.

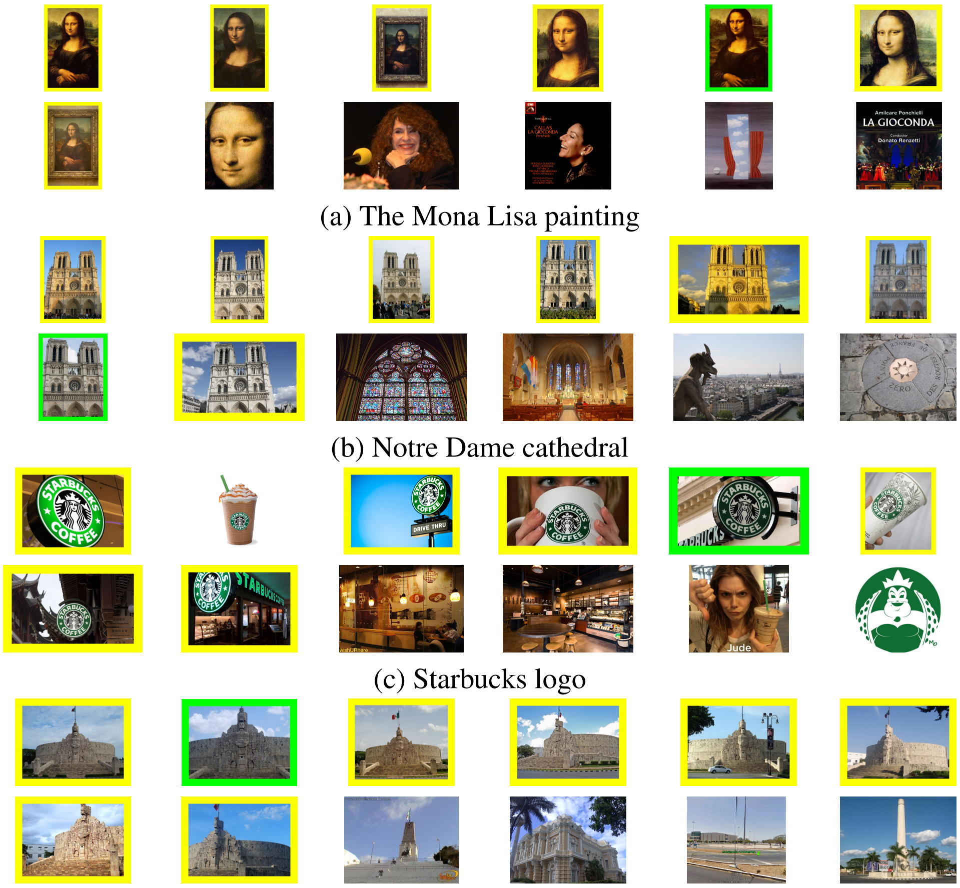
From now on, the set of similar images retrieved from content-based search (described in Sec 4.3) is known as the *pool of images*. As an introduction to the object modeling strategies, we describe the processing workflow with the pool of images.

The pool of images is transformed to a high-dimensional vectorial representation of their local regions. For this task, we opt for multiple covariant region detector, from similarity to affine covariant features. Later on, each geometry is represented by a signature invariant to some image transformations. The robustness of the signature depends on the description algorithm used. The decision made about which detector and descriptor to use in the pipeline affects the overall performance of the system.

The modeling strategies lie on the selection of the reference frame for enriching the feature set. The spatial location of the geometries detected along the pool of images must be transformed into a canonical frame that it comes out to be the coordinate system of one of the images, i.e. the *reference image*. In Section 4.3 is described that the reference image is manually selected by the user, however, this process can be done without supervision by the *Iconoid shift* algorithm [196]. Iconoid shift finds the most central/iconic views of single objects or buildings in large, unstructured image collections.
Each image in the pool works as a seed for the iconoid shifting process once. Image are scored by the number of times it is selected as the *mode*, until the shifting converges. Finally, the reference image is the one with the highest score.

Unrelated images are filtered out from the pool selecting only the top K images inside the mode support. Images are scored by the Homography Overlap Distance (HOD) defined in [196]. The experiments with Iconoid Shifting reveal that it is very expensive for small pools (10 images) and it scales badly which is prohibited due to our time constrains. We decided to not include this approach in our final pipeline but we report the proof of concept of the approach for selecting the reference image unsupervisedly. Fig. 4.3 (a)-(d) shows four pools of images, the green rectangles indicates the reference images and the yellow rectangles correspond to the images included in the mode support.

The image representation with local descriptors has a trade off between the descriptive power related to the number of keypoints and the computational cost to match/detect the model. Under some conditions, if the number of keypoints is high and well distributed in the image, then it is more likely to match two images with different viewpoints of the same scene. However, the number of the nearest neighbor (NN) searches required for computing the tentative correspondences also grows with the size of the feature set, making the representation prohibited for real-time



(a) The Mona Lisa painting

(b) Notre Dame cathedral

(c) Starbucks logo

(d) Monumento a la Bandera

Figure 4.3: (a)-(d)Four examples of *Pool of images*. Reference images in green rectangles and support sets in yellow rectangles.

applications. Besides the significant acceleration obtained by approximated NN search [138], the selection of the keypoints is a important task in order to keep the size of the model reasonable small without sacrificing performance.

The main motivation of our object modeling strategies is to increase the coverage of the local features across the object of interest without adding irrelevant parts, i.e. discard keypoints laying on regions that are not part of the query, like moving objects. Our approach is two folded, first, we aggregate contextual information to the query, adding surrounding regions that are consistent in different images in order to recognize the original query. Second, in common image regions among multiple views, the keypoints are highly overlap or possibly repeated. In consequence, the descriptors of these regions are very close in the feature space leading to lose performance with matching strategies based on distance ratios [106]. Our proposed approach addresses these issues.

The modeling strategy is up to selection of feature detectors (Harris, Hessian, etc.)

and descriptors (SIFT, BRIEF, etc.). The selection leads to performance changes, i.e. the precision and recall in retrieval varies with respect to this selection. Experiments for comparing the performance of different setups are presented in Sec. 4.7.3.

The main input of the model construction is the *pool of images* ($\mathcal{P}$) and is defined as

$$\mathcal{P} = \{\mathbf{p}_j\}, \quad \forall j = 1, \ldots, P, \quad \mathbf{p}_j \in \mathbb{R}^{M \times N}, \tag{4.1}$$

where P is the the pool size (number of images), $\mathbf{p}_j$ is a single image of size $M \times N$. The video, after being decoded, is a set of images defined as,

$$\mathcal{F} = \{\mathbf{f}_i\}, \quad \forall i = 1, \ldots, F, \quad \mathbf{f}_i \in \mathbb{R}^{M' \times N'}, \tag{4.2}$$

where F is the number of frames in the sequence and $\mathbf{f}_i$ is an image of size $M' \times N'$. In order to speed up the processing time, we do not search the object exhaustively across all the frames, the highest sampling frequency is constrained by the codec, since we test every *I-frame* (Intra-coded picture) [198] that it is a complete image with least amount of artifacts caused by the interpolation and compression. As preprocessing step of both model construction and object detection, the images are represented by the set of descriptors computed from the local regions of interest found in them. The specific feature detector and the descriptor used in the representation are parameters to be set in the system.

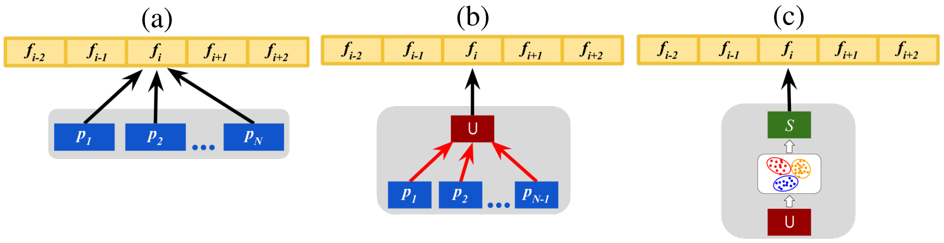


Figure 4.4: The three types of models for our object detection approach: (a) *Imagewise*, (b) *Union* and (c) *Salient* model. The step of the stereo matching is indicated with a **black** arrow and the reprojection of features with a **red** arrow. The direction of the arrows indicates the transformation from the target to the reference image. The images $\mathbf{f}_i$ (represented by the yellow squares) belong to the video clip, the images $\mathbf{p}_j$ (represented by the **blue** squares) are inside the pool used to build the models. In (b) and (c) the *Union* model is represented by the **dark red** square and the salient model by the **green** square.

4.4.1 The *Imagewise* model

The *Imagewise* model is the baseline of our object detection approaches where the elements of $\mathcal{P}$ are treated independently. P pairs of images are defined between the frame under test ($\mathbf{f}_i$) and each member of the pool ($\mathbf{p}_j$), then the pairs are fed into our wide-baseline stereo matcher. The number of correct matches is defined as the relevance score of the given frame. The maximum score among the P image pairs is preserve and assigned to the i-th frame. The Fig. 4.4(a) contains a diagram of this model where the yellow squares represent the frames of the video, the blue ones are the images in the pool and the black arrows indicate the stereo matching task. In addition, the orientation of the arrow indicates the direction of the matching, i.e. from target to reference image. It worth noticing that the Imagewise model is the sets of geometries (and descriptors) without transforming the coordinate frames into a common one.

4.4.2 The *Union* model

The second approach to build the object model in our proposal is called the *Union* model. Assuming that the geometries and descriptors of the images in the pool are computed, the approach requires one image to be the reference one, i.e. the coordinate frame of this image is used as the reference. The remaining subset of the pool are considered as targets to be matched against the reference image. As a result of the matching process, we obtain the homography/affinity that relates each pair of views. The points in correspondence are the input of RANSAC in such a way that the direction of the transformation is from the target to the reference image. Next, the geometries of the keypoints in the target images are reprojected to the reference image, this is done by transforming the coordinate systems of the target images applying the linear transformation that geometrically relates both images. Basically, the feature set of the reference image is enlarged by aggregation of the features detected in the target images under the same coordinate frame as a union set of keypoints. The intuition behind this model is to enrich the detection of the reference image with features extracted across other images, increasing the density and diversity of regions of the image. The advantage of this model is the possibility of storing the descriptors in an efficient data structure (k-D trees, hierarchical clustering, LSH, etc.) and compute fast approximated nearest neighbor search to compute distances between descriptors and run RANSAC only once for the whole model. Once all the descriptors are inserted in the index (tree), the later can be stored and reused for multiple videos as far as the pool does not change.

The Fig. 4.4(b) presents a simplified diagram of the union model where the blue squares are the images in the pool and the yellow squares are the frame sequence of the video. The reprojection of features is represented by the red arrows which shows that the points are transformed from the pool to the reference image giving place to the union set of features indicated with the dark red square with the $\cup$ symbol inside. Finally the union set and the features extracted from the current frame of the clip are fed into the matcher to compute the number of geometrically consistent matched feature pairs. Given number corresponds, consistently to the Imagewise model) to the relevance score. The last approach to model the query requires the union model as input and it is described in the next section.

A visualization of the progressive aggregation of images into the model is presented in the Fig. 4.5. The intensity (color) of the pixels are multiplied by a mask that contains the shapes (ellipses in the case of affine covariant features) of the keypoints detected on the image. The mask is not binary, each pixel contains the number of geometries that lie on its position then regions with high density of keypoints are brighter. At the bottom of the columns is written the number of images included in the model. It worth noticing that contextual information is added to the model when more images are in the pool, this can be observed in the surrounding regions of the reference image.

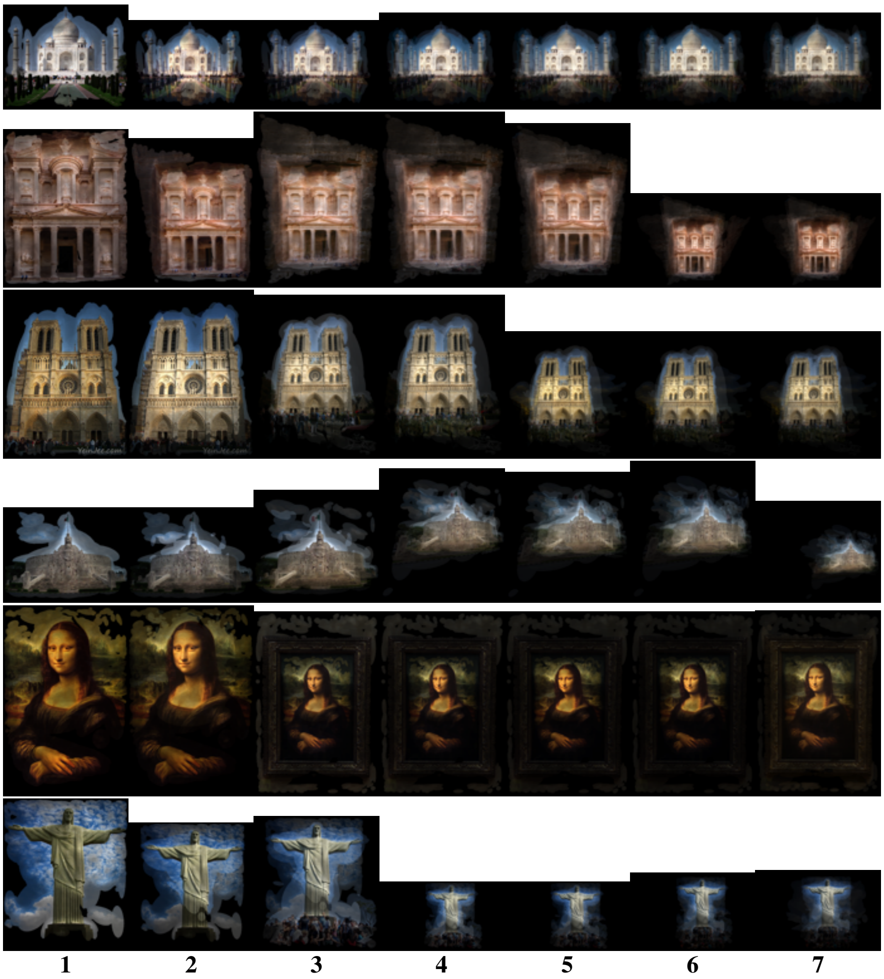


Figure 4.5: Visualization of object models constructed under the *UNION* strategy. The number of images inside the model is indicated at the bottom of each column. Starting from the reference (column **1**), the images in the pool are aggreagated progressively. The shape of the features masks the pixels to merge their color in the registered image. The addition of context information is observed from left to right.

4.4.3 The *Salient* model

The third approach for modeling the query object to be search in video is called *Salient* model. The input is the *Union* and algorithm is intended to reduces the size of the feature set without hurting the performance of the matching or detection step. The approach consists in modeling the density of the union set in the feature space and keep features with high support, i.e. create a feature set of points located at high density regions in the feature space. The density is approximated by the clustering algorithm called *DBSCAN* [48] which is unsupervised and does not require to set the number of cluster a priorly. The main reason to not used the standard k-means algorithm is the lack of prior knowledge about the number of different distinctive regions that the scene may contain, then we decided to do it data driven. In this context, two keypoints are similar not only if the descriptor distance between them is small, additionally, the points must lie close to each other in the XY space in order to not introduce noise in the computation of the centroids. The descriptor used in the clustering is a weighted concatenation of SIFT descriptor and XY position of the feature center and, from now on, we refer to it as *SIFT-XY* descriptor. Basically, every keypoint in the Union model is represented with a 130-D SIFT-XY descriptor, where the position in the image coordinate frame are normalized by a constant scaling factor ω. Experimentally, we fixed $\omega = 0.003$ to get our best results.

The results of the clustering is the representation of object, i.e. the Salient model. The *singletons* are clusters with only one member/sample and correspond to keypoints that are isolated in both image and feature space. Our intuition is that singletons do not belong to the query since they are unstable across different views so they shouldn't be sought in the video frames. On the other hand, *clusters* with cardinality larger than one are subsets of confirmed features, in the sense that regions look similar and they are geometrically consistent after reprojection. We named *salient clusters* to the clusters with more than one sample and these keypoints must be found in the video frame since we assume they belong to the object of interest.

Fig. 4.4(c) presents a diagram of the *salient* model computation where the union model as the dark red square is fed into the clustering algorithm (white box with an example of 3 clusters in a 2D space) and the final computation of the centroids (*salient* features) is represented by green square marked with the letter *S*. Notice that this model preserves the singletons since they provide information about features that are very likely to find near the object but they are confuser samples. The way we take advantage of the singletons is in the matching strategy described in Sec. 4.4.6.

A visualization of the clustering result is presented in Fig. 4.6 showing the three largest salient clusters found in pools of different objects. The images are taken from the dataset introduced in Sec. 4.9.1. Additionally, Fig 4.7 shows some examples of singletons found in the same objects. The patches are normalized following the

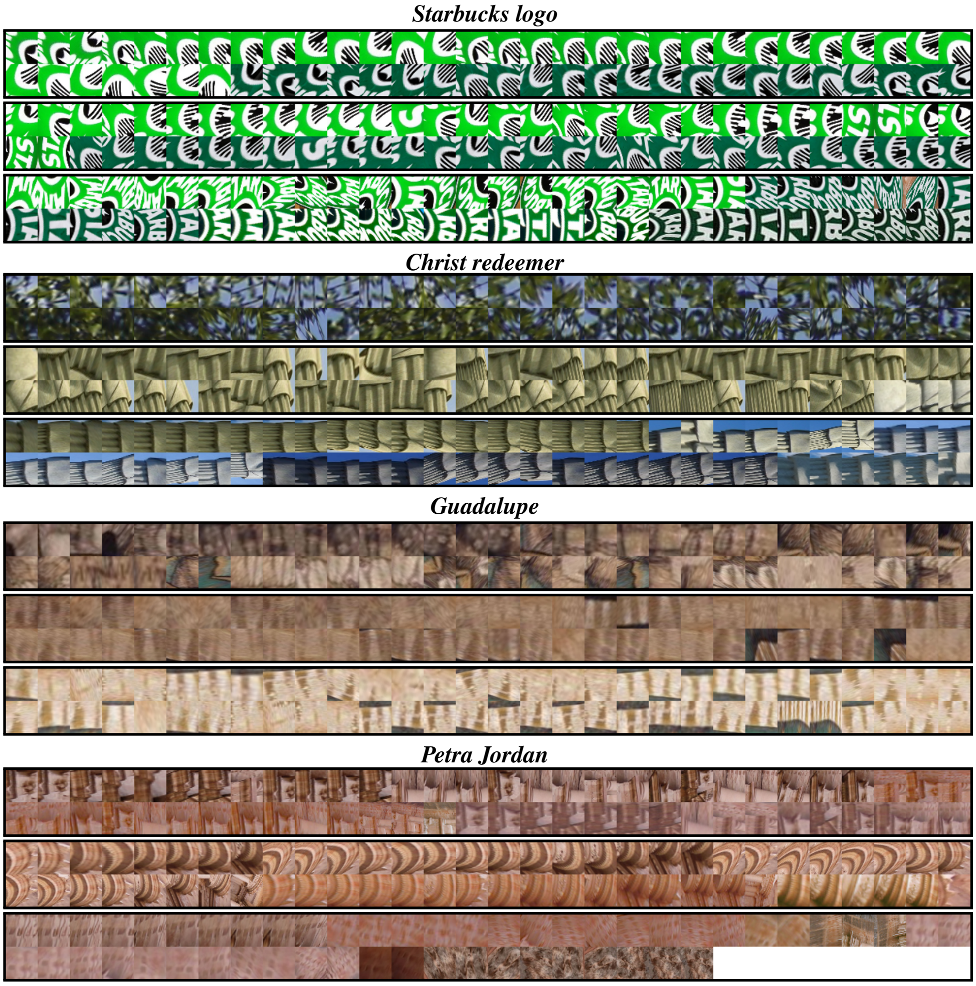


Figure 4.6: Normalized patches of the features that DBSCAN assigns to clusters with more than one member. One cluster gives place to one *salient* feature, since the descriptors inside the cluster are combined to compute the new representative (centroid) of the cluster. The black frames enclose all the patches inside the same cluster. Three clusters per landmark are shown below the landmark name.

geometry of the keypoint, in this case, keypoints are local affine covariant detected by the Hessian Affine detector.

4.4.4 The *Mean-Salient* model

The last modeling approach is the *Mean-Salient* model. We propose it as a compact representation of Salient model, since for each salient cluster we compute a representative member or centroid and then read out the cluster. The centroids are computed as the mean SIFT-XY descriptor of the member in the cluster and we called them

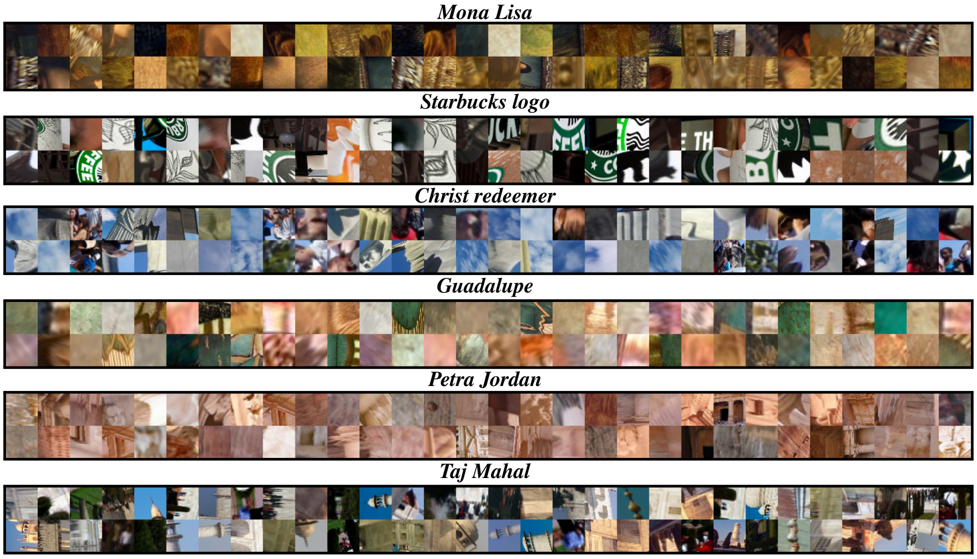


Figure 4.7: Normalized patches of the features that DBSCAN assigns to clusters with only one member a.k.a *singletons*. In the feature space, the descriptors lie to far from larger clusters to be grouped within them. Each column show at the top the name of the query and below 25 patches selected randomly from the full set of singletons of the model.

salient keypoints. All singletons are discarded in this model. A similar approach for computing mid-level features is proposed by Koniusz *et al.* [82]. The full algorithm to compute the set of salient features is summarized in Alg. 2.

4.4.5 Model statistics

The effectiveness of the object representation was tested in the experiments comparing the results using 2 types of representation. The first one was called *Union* which is the set of reprojected features on the reference image with no filtering stage. The second one was the set of salient features (described in Sec. 4.4) and it is called *Salient*.

Tab. 4.1 contains the number of features in the two object representations and the reference image itself. The average size of the *salient* representation is 3% of the whole features detected on the pool of images (union) and 18% of the features detected on the reference image. The significant reduction in the cardinality of the feature sets was reflected in memory allocation and the complexity of matching task. The average time for building a *salient* model was 4.1 sec. for a single image pool. Construction time of Union models (1.27 sec) was obtained subtracting the mean-shift clustering step. The percentages of processing time for each step of the

Algorithm 2 Salient features

```
Input: Pool of images ($P$), reference image ($I_{ref}$)
Output: Set of salient features ($SF$)
  $N \leftarrow |P|$
  // Detect and describe image features
  for $i = 1$ to $N$ do
      $f_i \leftarrow$ hessian_affine_detection($p_i$)
      $d_i \leftarrow$ SIFT_description($f_i$)
  $D = \{d_1, ..., d_N\}$
  // Features in images of the pool without the reference
  $C \leftarrow D \setminus \{d_{ref}\}$
  $c_i \in C, i = 1, ..., N-1$
  // Set of reprojected features (RF)
  $RF \leftarrow \{f_{ref}\}$
  for $j = 1$ to $N-1$ do
      $H_j \leftarrow$ wbs_match($d_{ref}, c_j$)
      $RF \leftarrow \{RF \cup$ reproject_features($H_j, c_j$)$\}$
  $CL \leftarrow$ DBSCAN_clustering($RF$)
  // Salient features are described by average SIFT
  $SF \leftarrow$ average_SIFT($CL, RF$)
      return $SF$
```

model computation are shown in Fig. 4.18.

Query object	Number of features			Ranking quality	
	Ref. image	Union	Salient	Text search	Re-ranked
Taj Mahal	1368	11363	585	**0.78**	0.47
Petra city	3484	28109	1002	0.60	**0.78**
Notre Dame	5981	30611	2962	0.56	**0.60**
Monumento Patria	2764	14758	739	0.47	**0.60**
Mona Lisa	2303	17243	2449	0.47	**0.73**
Christ Reedemer	3771	10965	477	0.51	**0.69**
Coca Cola	834	8466	315	0.51	**0.51**
Starbucks	1408	12345	1017	0.33	**0.56**
Virgin Mary	6594	66675	5589	0.69	**0.73**

Table 4.1: The *number of features* in the object representations is shown: *Ref. image* column for features on the reference images, *Union* column for features detected in the whole pool of images and *Salient* column for selected features only. In addition, Kendall tau rank correlation coefficients between the ground truth video ranked list and both retrieved ranked lists, regarding the *Text search* list and the *Re-ranked* list by relevance assessment, are shown as *Ranking Quality*. Best ranked lists are highlighted with bold font.

4.4.6 Matching strategies with models

The model-to-image matching task requires different strategies depending on the model selected to represent the object. We describe our matching strategies in the following sections.

1GINN for Imagewise, Union and Mean-Salient model

The descriptors extracted from a video frame are matched against a Imagewise, Union or Mean-Salient by the 1GINN strategy proposed by Mishkin *et al.* [134]. The strategy is described in Sec. 3.3.6. 1GINN was originally proposed to match images with view synthesis. A side-effect of the method is the computation of multiple instances of the same 3D point, due to multiple detections in the generated synthetic views. The standard matching strategy with first and second NN distance ratio performs poorly in these conditions (see Sec. 3.3.5). In order to preserve correct correspondences with distance ratio close to 1, 1GINN computes the distance ratio to the nearest descriptor that comes from feature that is sufficiently far away in the image coordinate frame from the tentatively corresponding one, which is known as the first geometric inconsistent.

First Nearest Singleton for Salient model

One of the matching strategy for the Salient model is the *1st nearest Singleton* (1NS). As a result of clustering the SIFT-XY descriptors of the Union model, we obtain the *salient clusters* and the *singleton features*. We assume that singleton features do not belong to the query object even more singletons are counter examples of what we are searching in the frames. On the other hand, we assume salient clusters are stable and repeatable features that belong to the query object. Then, we propose to score the tentative correspondences with the distance ratio of the first nearest salient cluster (the nearest element) over the first nearest singleton. Under this definition the distance ratio can be larger than 1, however, the smaller it is the higher confidence in the correspondence.

1st/2nd Distance Ratio for Salient model

The last matching strategy for the Salient model is similar to the standard one based on distance ratios. We compute tentative correspondences with the 1st/2nd distance ratio (2NN) as proposed in [106]. In order to not discard correct tentative matches due to the high similarity of the elements in the same cluster, the first and the second nearest neighbors must belong to different clusters.

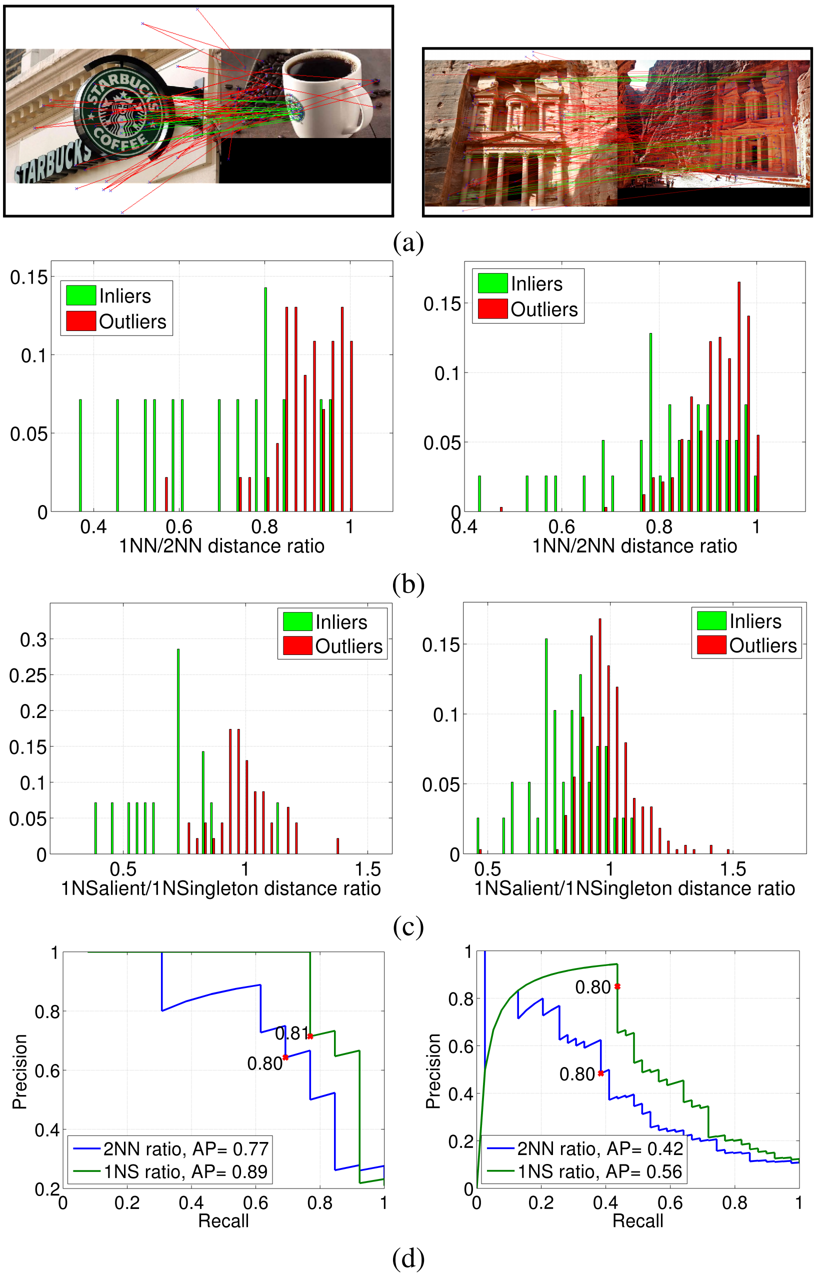
Comparison of matching strategies

In this section we test and compare the performance of the matching strategies described previously. The validation requires a ground truth transformation, in this case an homography, that relates the image coordinate frame of the model with the target image. Since such ground truth homography is not available in this context, we approximate it with manually annotated correspondences between reference and target images followed by LO-RANSAC [92].

In the next step a given matching strategy is applied to compute a set of tentative correspondences. Such correspondences are labeled as inlier or outliers if the reprojection error is larger that 5 pixels under the geometric transformation of the approximated homography. Finally, the performance of the matching strategy is measure by the number of inlier matches rejected and the number of outlier matches accepted by the thresholding criterion of the distance ratio, i.e. a correspondence is rejected if the distance ratio is lower than a threshold. Consistently to [106] experimentally we found out that threshold 0.8 in average gives the best results for all strategies.

Two examples of matching by *1st nearest neighbor* (1NN) are shown in row (a) of Figure 4.8, the inlier matches found by RANSAC are indicated with green lines and the outliers with red lines. Then, 2NN and 1NS distance ratios are computed for all tentatives correspondences in order to generate histograms for inlier and outlier matches.
Fig. 4.8 presents the histograms of distance ratios in rows (b) and (c). Finally, using the approximated homography as ground truth, we compute the Precision-Recall curves with respect to the distance ratio threshold (see Figure 4.8(d)).

In both examples, the histograms show narrower and less overlapped discrete distributions for the 1NS, which leads to a better tentative correspondences quality changing the threshold. For threshold 0.8, as proposed in [106], both precision and recall were higher for 1NS. Moreover, 1NS had an Average Precision (AP) of 0.89 in the first example and 0.56 in the second one, higher than 0.77 and 0.42, respectively, for the 2NN matching strategy.



(a)

(b)

(c)

(d)

Figure 4.8: Row (a) Pairs of images matches by 1NN and RANSAC, green lines are geometrical consistent tentative correspondences (inliers) and red lines are incorrect ones (outliers). For both inlier and outlier matches, the 2NN distance ratio was computed and (b) shows the normalized histograms. Row (c) shows the normalized histograms of the 1NS distance ratio. The Precision-Recall curves of both distance ratio definitions is shown in row (d), changing the threshold for dropping correspondences. Operating point for threshold 0.8 is indicated by red markers, and the Average-Precision is written in the plot legend.

4.5 Video acquisition and representation

The video re-ranking pipeline relies on a third party video server with a search engine working independently, since we do not attempt to build our own indexed video database. Even though the approach for video relevance assessment and re-ranking is up to the video database infrastructure, we use YouTube service for sharing videos publicly. The motivation of our selection is the API provided by the developers of the server, such API is flexible and provide us with all capabilities to shape our queries.

Specifically, the API is called *YouTube Data API* [55]. Such interface allows to add features available in the website. Some capabilities are uploading videos, managing playlists and subscriptions, update channel settings, and among others. Our proposed pipeline uses the API to search for videos matching specific search terms, topics, locations, publication dates, etc. The main method used for our concern is *search.list* since it queries the video database with a string as input, presumably, the name of the object of interest. Additionally, the method supports searching for playlists and channels, however, processing such data types are out of the scope of this thesis.

The procedure for retrieving a short list relevant videos with respect to the query string is:

1. Query YouTube database with the input string. The server responds with a list of hashes (*videoIDs*) of the relevant videos.
2. Download the videos by hash with the library called *Youtube-dl* [157].
3. Get the indexes of the keyframes inserted by the encoder using *FFMPEG* library [49].
4. Under the setup for the relevance assessment pipeline, detect and describe the keypoint from the keyframes only.

The set of videos collected from the text-based retrieval with the YouTube search engine are represented by a subset of keyframes (Intra-coded frames or I-frames) related to the CODEC used for compressing and packing the video. Local covariant features are detected and described on every selected keyframe. This stage avoids the wide-baseline stereo matching over all frames of the video. The object model is matched against up to 1% of the total number of frames. For shot boundary detection, we applied a simple detector [23], that thresholds the sum of pixel-wise absolute differences. To reduce the number of selected keyframes, we dropped keyframes close to the shot boundary, as these are typically corrupted by the shot transition. Finally, after the feature extraction step of the video frames, the encoded video is discarded to avoid memory consumption since the set of features (geometries and descriptor) are the representation required in the next components.

The CODEC of the video usually do not include a keyframe if the content from one frame to the next one does not change significantly, therefore, detecting local features in keyframes only save processing and memory consumption without increasing recall of the system. Additionally, analyzing similar (too close in time) frame causes overfiring of our system which is reflected in multiple time markers (see Sec. 4.8) in the same shot. Depending on the CODEC, contiguous keyframes are inserted thus down-sampling the video by keyframes is inefficient. In our pipeline the videos are represented with a larger distance between keyframes indicated by a scale factor, for instance, describe every the nearest keyframe every 100 frames.
A clever strategy is based on shot segmentation by sampling the keyframe located at the middle of every shot. In this thesis, we present an approach for detecting shot boundaries based on stereo matching and it is introduced in details in Sec. 4.7.2.

4.6 Object detection in video frames

A shot was regarded as relevant if the object or landmark appears on at least one of its selected frames. The object recognition was addressed as a *Wide-Baseline Stereo Matching* problem, as proposed in e.g. [112]. To efficiently detect the nearest neighbor SIFT descriptors, approximate nearest neighbor search was used [138]. Global geometric model and supporting tentative correspondences were robustly estimated using LO-RANSAC [36]. The geometric model of homography or affine transformation were compared.

The relevance of the video with respect to the object model was given by the number of relevant frames that appear in the video.

4.7 Shot segmentation by wide-baseline stereo

In many problems like video representation, retrieval and scene segmentation a fundamental preprocessing step is the shot segmentation. Shots are defined as a subsets of adjacent frames captured with the same camera continuously without time interruptions[15]. In this thesis we introduce the *Stereo Shot Detector* (SSD), a novel shot detection approach that computes the features required for the later object recognition step in our pipeline. The goal is to locate pairs of frames that contain a transition between them along the video sequence, named *transition boundaries* (TB). The boundary between two consecutive shots is known as a transition and, commonly, we can find them as sharp or gradual transitions, including complex animations.

4.7.1 Related work

A similar approach to our proposal is [65] and it works as follows, this method used SIFT to find image correspondence, and applied a fixed threshold to the number of matched points of neighboring video frames to find the transitions. Our method differs from this method in several aspects. First, our method does not rely on a fixed threshold of the number of matched points; the threshold applied in our method is varied with the local maxima and minima of the number of matches, which can handle the variations of transitions better. Second, we do not match neighboring frames or frames apart from a fixed period only, but also match nonadjacent frames inferred by shot-change interval estimation, which can further increase the detection accuracy. Third, our method can find both the shot boundaries and the transition intervals of shots.

4.7.2 SSD algorithm

The SSD algorithm addresses the problem of finding TBs by alternating between two main steps. First, a loose TB is found by taking an anchor and a target frame which indexes are i_{anchor} and i_{target} respectively. The latter is conditional moved forward in a manner of line search with constant step length (α) and positive step direction (δ) since the target frame is moving forward in every iterations. A line search method seeks for the position of a local minimum i^* of a function $f(i)$ iteratively as follows,

$$i_{t+1} = i_t + \delta_t, \tag{4.3}$$

$$\delta_{t+1} = \alpha \cdot \delta_t, \tag{4.4}$$

where, i_t is the index of the (target) frame to be compare with the anchor one. Notice that α in Eq. 4.3 is constant, this is due to in our approach there is not such a function f to be minimized enabling this parameter to be tuned during the search. Experimentally we found that 1.4 gives the best results. The condition to update the position of the target frame is the output of a binary similarity function $S(I_i, I_j) \in \{0, 1\}$ where $I_i, I_j \in \mathbb{R}^{M \times N}$ are frames of the video sequence.

$$S(I_i, I_j) = \begin{cases} 1 & \text{if } \textsc{StereoMatch}(I_i, I_j) \geq L, \\ 0 & \text{otherwise,} \end{cases} \tag{4.5}$$

where, $\textsc{StereoMatch}(\cdot, \cdot)$ is a function that register the two input images by wide-baseline stereo matching and its codomain is the number of local features matched correctly (*inliers*) with respect to a geometric model, the *homography* in our approach, and L is the threshold for the minimum number of inlier matches needed in a pair to regard them as similar images. The target frame index is moved forward if the current anchor and target frames are similar, otherwise this step finishes assuming the putative transition is located between the anchor and the latest target frames.

In the second step, a binary search is performed to get tighter TB or discard the tentative transition. The video interval defined between the initial anchor and target frames is split recursively. The recursion depth is constrained by the frame pair similarity, i.e. if $S(I_i, I_j) = 1$ then the recursion halts discarding the transition, otherwise continues until a minimum interval length is reached returning the indexes of the verified TB. Finally, the summarized method is presented in the Alg. 3 and Fig. 4.9 shows a visualization of the line (above arrows) and binary search (below arrows) to detect a gradual transition.

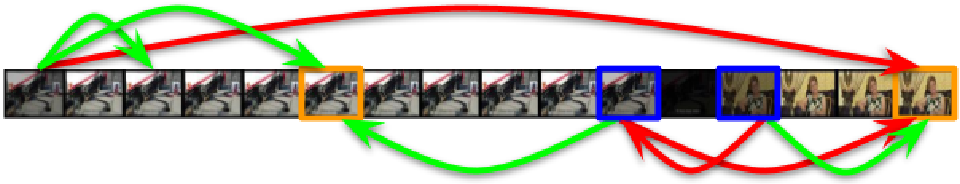


Figure 4.9: Linear and binary search of the SSD algorithm. Arrows on top indicate the forward step for finding the widest frame interval (**orange** squares) where the potential transition is located. Arrows below indicate the binary search to find tighter transition boundaries(**blue** squares). The correctly matched images are indicated with **green** arrows, otherwise in **red**.

4.7.3 Evaluation of SSD

The evaluations of the SSD shot boundary detector is performed on two publicly available datasets. The RAI dataset is a collection of ten challenging broadcasting videos from the Rai Scuola video archive [14], mainly documentaries and talk shows. The BBC Planet Earth (BBCPE) dataset [13] contains ground truth shots and scene annotation for each of the 11 episodes of the BBC Planet Earth educational TV Series. Each shot and scene has been manually annotated and verified by a set of

Algorithm 3 Stereo Shot Detector

```
Input: Video $V$, initial position $\delta_{init}$, step size $\alpha$, spliting threshold $\mu$.
Output: Set $T$ of transitions
  $T \leftarrow \{\}$
  $i_{anchor} \leftarrow 1$
  while not end-of-video do
      $\delta \leftarrow \delta_{init}$
      $i_{target} \leftarrow i_{anchor} + \delta$
      while $S(V(i_{anchor}), V(i_{target}))$ do
          $\delta \leftarrow \alpha \cdot \delta$
          $i_{tested} \leftarrow i_{target}$
          $i_{target} \leftarrow i_{anchor} + \delta$
      $i_{anchor} \leftarrow t_{end}$
      $T_{part} \leftarrow \textsc{BinarySearch}(i_{tested}, i_{target})$
      $T \leftarrow \{T \cup T_{part}\}$
  return $T$
  procedure BinarySearch($t_{begin}, t_{end}$)
      if $t_{end} - t_{begin} > \mu$ then
          $t_{mid} \leftarrow \lfloor 0.5 \cdot (t_{end} - t_{begin}) \rceil$
          if $S(V(t_{begin}), V(t_{mid}))$ then
              $T_{left} \leftarrow \{\}$
          else
              $T_{left} \leftarrow \textsc{BinarySearch}(t_{begin}, t_{mid})$
          if $S(V(t_{mid}), V(t_{end}))$ then
              $T_{right} \leftarrow \{\}$
          else
              $T_{right} \leftarrow \textsc{BinarySearch}(t_{mid}, t_{end})$
          $T \leftarrow \{T_{left} \cup T_{right}\}$
      else
          $T \leftarrow \{(t_{begin}, t_{end})\}$
  return $T$
```

human experts. The datasets contains the indexes of the frames where the transitions between one shot to the next one are. In addition, the transitions are labeled as hard, smooth and gradient.

We compare the performance of SSD against a standard approach called the ImageLab Shot Detector (ILSD) [15] which is presented as part of a complete pipeline for story detection [16]. The SSD and ILSD are compared in the Table 4.3 and 4.2, respectively. The results shows that our method does not outperform its competitor in none of the datasets. However, the performance on the RAI dataset is comparable. The analysis of the failure cases shows that the performance is hurt by the fact that some of the shot boundaries are abrupt camera pose changes keeping the same scene. In such cases, the shot boundary is not detected because SSD gets a high number of correctly matched features. Others failure cases are related to static images superposed to the videos, i.e. brand logos or static stylized text. The static content is matched across all shots losing all shot boundaries. Both factors increments the number of false negatives in the evaluation of the SSD. Fixing the failure cases is part of our future work.

Video	ILSD [15]	SSD
V_1	**0.97**	0.89
V_2	**0.97**	0.86
V_3	**0.95**	0.85
V_4	0.78	**0.87**
V_5	0.38	**0.62**
V_6	**0.96**	0.59
V_7	**0.94**	0.66
V_8	**0.94**	0.86
V_9	0.76	**0.86**
V_{10}	0.77	**0.79**
Average	0.84	0.79

Table 4.2: Shot detection performance in the RAI dataset [15] with 10 broadcast videos. SSD algorithm (our approach) outperforms ILSD in 4 cases significantly, even so, in average it performs worse. Results are presented in *F-score* and bold font indicates better performance.

4.8 WEB GUI

The interaction of the user with the system is performed through a graphical interface, which allows to query the multiple databases for retrieving images and video. Figure 4.10 shows the three main sections in the interface: The visualizer where the videos are played (highlighted in a red frame), the list of videos sorted by their visual relevance (blue frame) from the sharing web-site and, finally, the list of time stamps sorted chronologically to the position in the video corresponding to the beginning of the scene where the object was detected (orange frame).

The example shown in Figure 4.10 belongs to the search results of the textual query *Pisa tower*. The list of relevant shots includes the frame index and time (hh:mm:ss) where the marker is located, see Figure 4.11 for an example searching for *Notre Dame cathedral*.

Video	ILSD [15]	SSD
From Pole to Pole	**0.92**	0.62
Mountains	**0.89**	0.76
Ice Worlds	**0.91**	0.63
Great Plains	**0.91**	0.61
Jungles	**0.90**	0.71
Seasonal Forests	**0.88**	0.73
Fresh Water	**0.92**	0.58
Ocean Deep	**0.70**	0.45
Shallow Seas	**0.93**	0.45
Caves	**0.77**	0.63
Deserts	**0.89**	0.75
Average	0.88	0.63

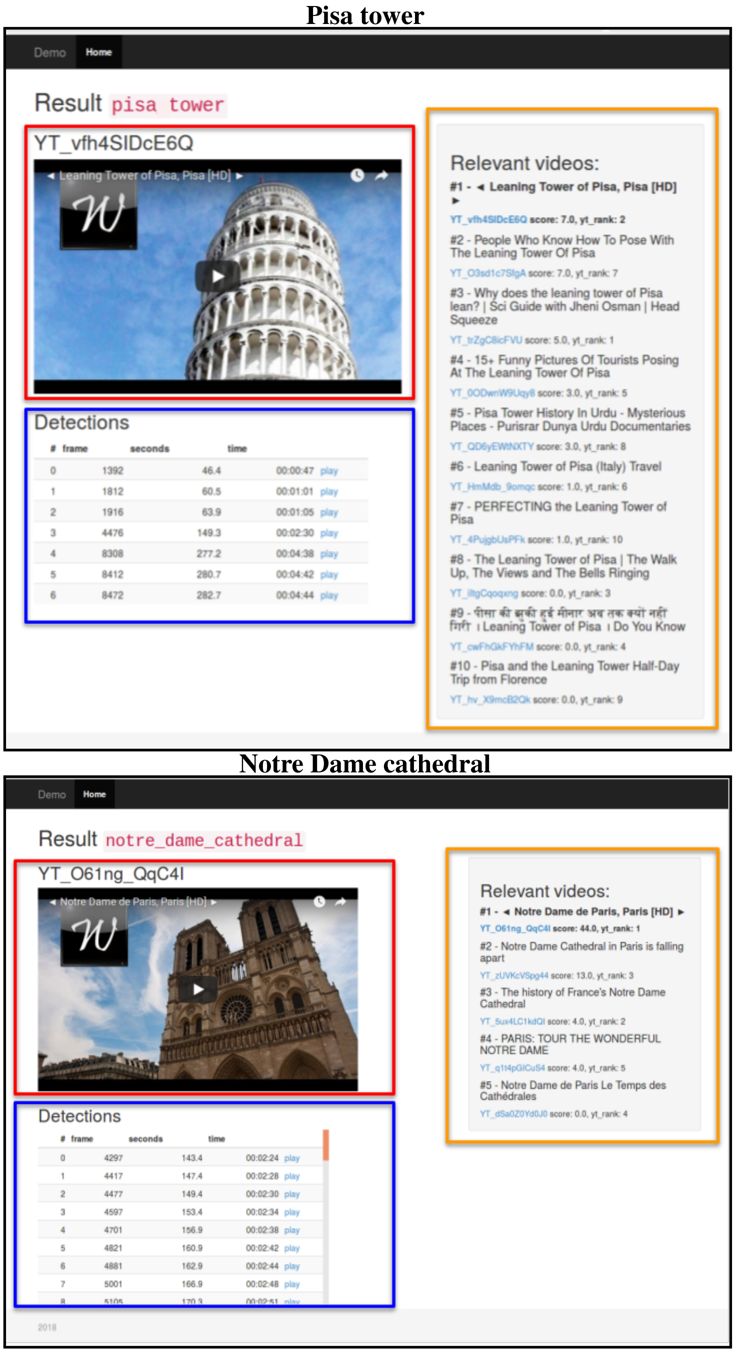
Table 4.3: Shot detection performance in the BBCPE dataset [13] with 11 videos of an educational TV series. Results are presented in *F-score* and bold font indicates better performance.

4.9 Evaluation of Relevance Assessment and Re-ranking

The evaluation of the video re-ranking by visual assessment is addressed in two parts. First, the performance of the object detection in videos by stereo matching is tested to compare the modeling strategies introduced in Sec. 4.4 and their matching capabilities under the strategies described in Sec. 4.4.6. Second, the performance of the video re-ranking is tested as well, since the final order of the video list assigned by the re-ranking approach with respect to the visual relevance with the query must be tested. Priorly to our work, there were no benchmarks or dataset with ground truth annotations of videos and image suitable to evaluate our setup publicly available.

4.9.1 *Specific Object Search dataset*

The task of detecting a specific object in video sequences requires that the user provides a definition of the query, i.e. the name and images of the object, thus a model can be constructed from that data to do the search. In order to measure the performance of the outcomes of the system, we require a benchmark or labeled dataset. For similar tasks, one can find different dataset, for instance, for video classification [1], human activity recognition [115], sport recognition [142], shot boundary detection [176], among other. To the extent of our knowledge, so far there is no dataset for searching specific object on videos available in the literature. Hence, we decided to present a new dataset suitable for this task.



Demo
Home
◄ Leaning Tower of Pisa, Pisa [HD] ►
Demo
Home
◄ Notre Dame de Paris, Paris [HD] ►

Pisa tower

Result pisa tower

YT_vfh4SIDcE6Q

Detections

#	frame	seconds	time	
0	1392	46.4	00:00:47	play
1	1812	60.5	00:01:01	play
2	1916	63.9	00:01:05	play
3	4476	149.3	00:02:30	play
4	8308	277.2	00:04:38	play
5	8412	280.7	00:04:42	play
6	8472	282.7	00:04:44	play

Relevant videos:

#1 - ◄ Leaning Tower of Pisa, Pisa [HD] ►

YT_vfh4SIDcE6Q **score: 7.0, yt_rank: 2**

#2 - People Who Know How To Pose With The Leaning Tower Of Pisa

YT_O3sd1c7SfgA score: 7.0, yt_rank: 7

#3 - Why does the leaning tower of Pisa lean? | Sci Guide with Jheni Osman | Head Squeeze

YT_trZgC8icFVU score: 5.0, yt_rank: 1

#4 - 15+ Funny Pictures Of Tourists Posing At The Leaning Tower Of Pisa

YT_0ODwnW9Uqy8 score: 3.0, yt_rank: 5

#5 - Pisa Tower History In Urdu - Mysterious Places - Purisrar Dunya Urdu Documentaries

YT_QD6yEWtNXTY score: 3.0, yt_rank: 8

#6 - Leaning Tower of Pisa (Italy) Travel

YT_HmMdb_9omqc score: 1.0, yt_rank: 6

#7 - PERFECTING the Leaning Tower of Pisa

YT_4PujgbUsPFk score: 1.0, yt_rank: 10

#8 - The Leaning Tower of Pisa | The Walk Up, The Views and The Bells Ringing

YT_iltgCqoqxng score: 0.0, yt_rank: 3

#9 - पीसा की झुकी हुई मीनार अब तक क्यों नहीं गिरि । Leaning Tower of Pisa । Do You Know

YT_cwFhGkFYhFM score: 0.0, yt_rank: 4

#10 - Pisa and the Leaning Tower Half-Day Trip from Florence

YT_hv_X9mcB2Qk score: 0.0, yt_rank: 9

Notre Dame cathedral

Result notre_dame_cathedral

YT_O61ng_QqC4l

Detections

#	frame	seconds	time	
0	4297	143.4	00:02:24	play
1	4417	147.4	00:02:28	play
2	4477	149.4	00:02:30	play
3	4597	153.4	00:02:34	play
4	4701	156.9	00:02:38	play
5	4821	160.9	00:02:42	play
6	4881	162.9	00:02:44	play
7	5001	166.9	00:02:48	play
8	5105	170.3	00:02:51	play

Relevant videos:

#1 - ◄ Notre Dame de Paris, Paris [HD] ►

YT_O61ng_QqC4l **score: 44.0, yt_rank: 1**

#2 - Notre Dame Cathedral in Paris is falling apart

YT_zUVKcVSpg44 score: 13.0, yt_rank: 3

#3 - The history of France's Notre Dame Cathedral

YT_5ux4LC1kdQI score: 4.0, yt_rank: 2

#4 - PARIS: TOUR THE WONDERFUL NOTRE DAME

YT_q1t4pGICuS4 score: 4.0, yt_rank: 5

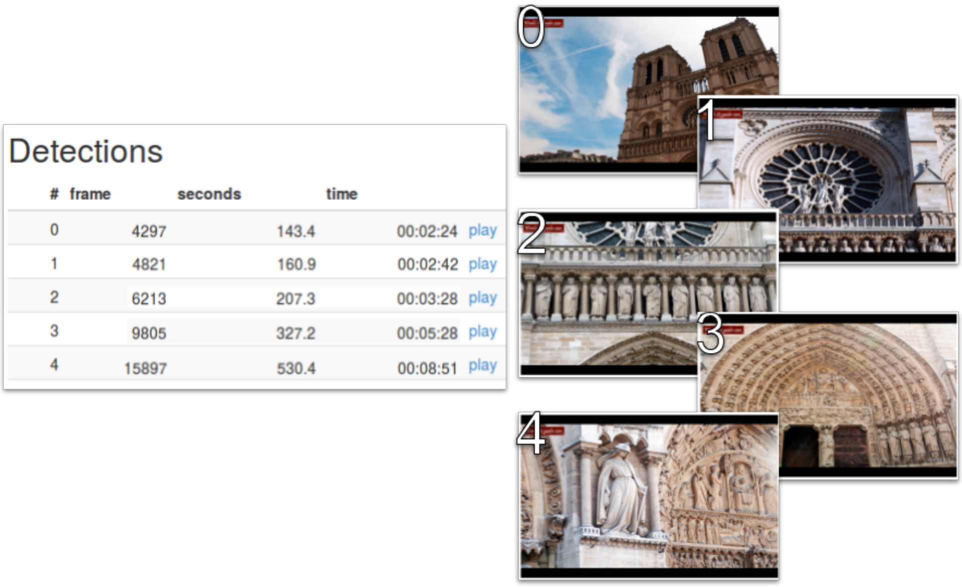
#5 - Notre Dame de Paris Le Temps des Cathédrales

YT_dSa0Z0Yd0J0 score: 0.0, yt_rank: 4

2018

Figure 4.10: Screenshots of the user interface. The pipeline allows the user to navigate through the video list (**red** rectangle) sorted by relevance with respect to the visual interest, visualize the videos in the player ((**orange** rectangle)) and pick specific shots depicting the query (**blue** rectangle).

The evaluation of the detection pipeline requires test data with multi-media information. We introduce a labeled dataset named *Specific Object Search* (SOS) dataset for detection on videos as a contribution of this work. The collection and



Detections

#	frame	seconds	time	
0	4297	143.4	00:02:24	play
1	4821	160.9	00:02:42	play
2	6213	207.3	00:03:28	play
3	9805	327.2	00:05:28	play
4	15897	530.4	00:08:51	play

Figure 4.11: Example of shot selection for the query *Notre Dame cathedral* in one element of the ranked list of potential relevant videos. On the left side, the table with the five shots where the object of interest appears with their time stamps. On the right, the first frame of the shot with the index of the place in the table.

construction of the dataset is conformed by 100 videos and 70 images retrieved under 10 queries:

- ***Petra city*** in Jordan
- ***Notre Dame cathedral*** in France
- ***Taj Mahal palace*** in India
- ***The Mona Lisa*** painting in France
- ***Monumento a la Patria*** in Mexico
- ***Christ The Redeemer*** in Brazil
- ***Coca-Cola*** logo
- ***Lola perfume*** container
- ***Starbucks*** logo
- ***Virgin Mary*** painting in Mexico

For each query, 10 videos are retrieved from YouTube and 7 images from *Google images* by means of textual search only. The video sets has 7 videos with the corresponding object captured in at least one shot (positives) and the remaining three videos are confusers (negatives). The image sets contain positive images only since confusers are supposed to be provided as dependency of the application. In Fig. 4.12, the images of the SOS dataset are presented. The videos are provided with a ground truth labelling. The schema to classify the frames manually follows the classes introduced in the *Oxford 5K* dataset [148] for image retrieval. There are four labels: *NO-OBJECT* for images that do not contain the query, *GOOD* for a nice and clear picture of the object, *OK* when more than 25% of the object is clearly visible and *JUNK* when less than 25% of the object is visible, or there are very high levels of occlusion or distortion.

The relevant shot detection algorithm was applied to a dataset of images and videos collected from 10 different queries: *Petra city* in Jordan, *Notre Dame cathedral* in France, *Taj Mahal palace* in India, *The Mona Lisa* painting in France, *The Merida's Monumento a la Patria* in Mexico, *Christ The Redeemer* in Brazil, the Coca-Cola logo, the Lola perfume container, Starbucks logo and Virgin Mary painting.

The image pools contain 7 images (top 7 images in the mode support ranked by the HOD) per query object with a fixed width of 640 pixels and keeping the aspect ratio. All images were stored in JPEG format. The number of local affine covariant features detected on the images are presented on Tab. 4.1.

The video set contains 100 videos downloaded from You Tube. Every object of interest has 10 videos, 7 of them actually depict the object and 3 of them works as confusers (videos were retrieved by querying You-Tube with the same text search but the object never appears on scene).

The videos have an average duration of 3 minutes, the frame rate was fixed 25 fps, the size of the frames is 640×480 pixels. All videos were stored with the codec H.264, which inserts a keyframe (Intra-coded picture) every 60 frames. Notice that only keyframes were processed.

Finally, the SOS dataset is provided with ground truth relevance order of the videos for each object. The frame-by-frame annotation is used to score the videos with the number of *GOOD* and *OK* frames, hence the ground truth is the sorted list with respect to this score. Only the 10 videos of a specific query are included in a ground truth list, hence we end up with 10 lists. The confuser videos are not sorted since the permutation of them have no influence in the performance metric.

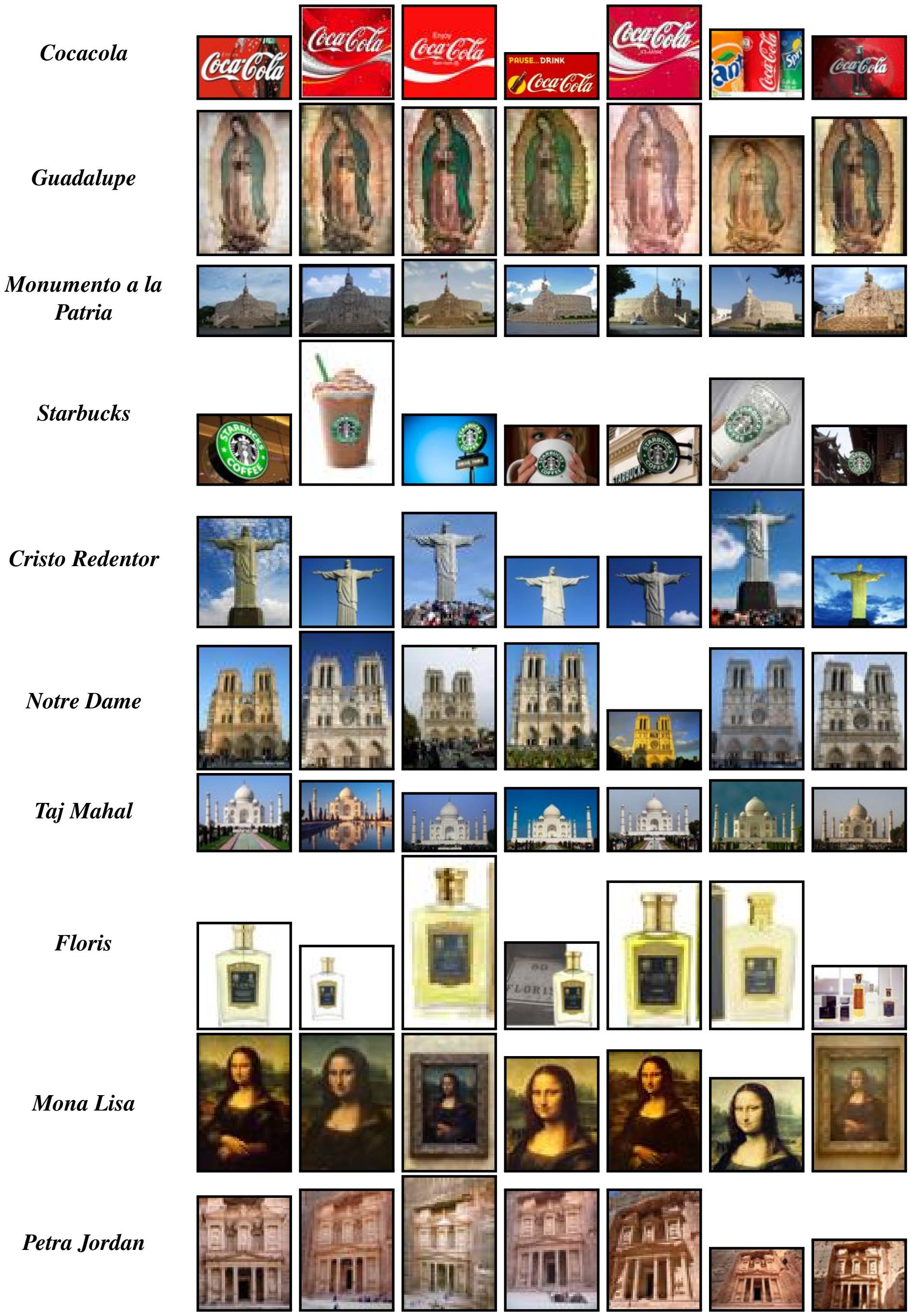


Figure 4.12: Images of the Specific Object Search (SOS) dataset. Each row corresponds to the textual query presented at the left side column. The images were retrieved from *Google images* searching for the queries.

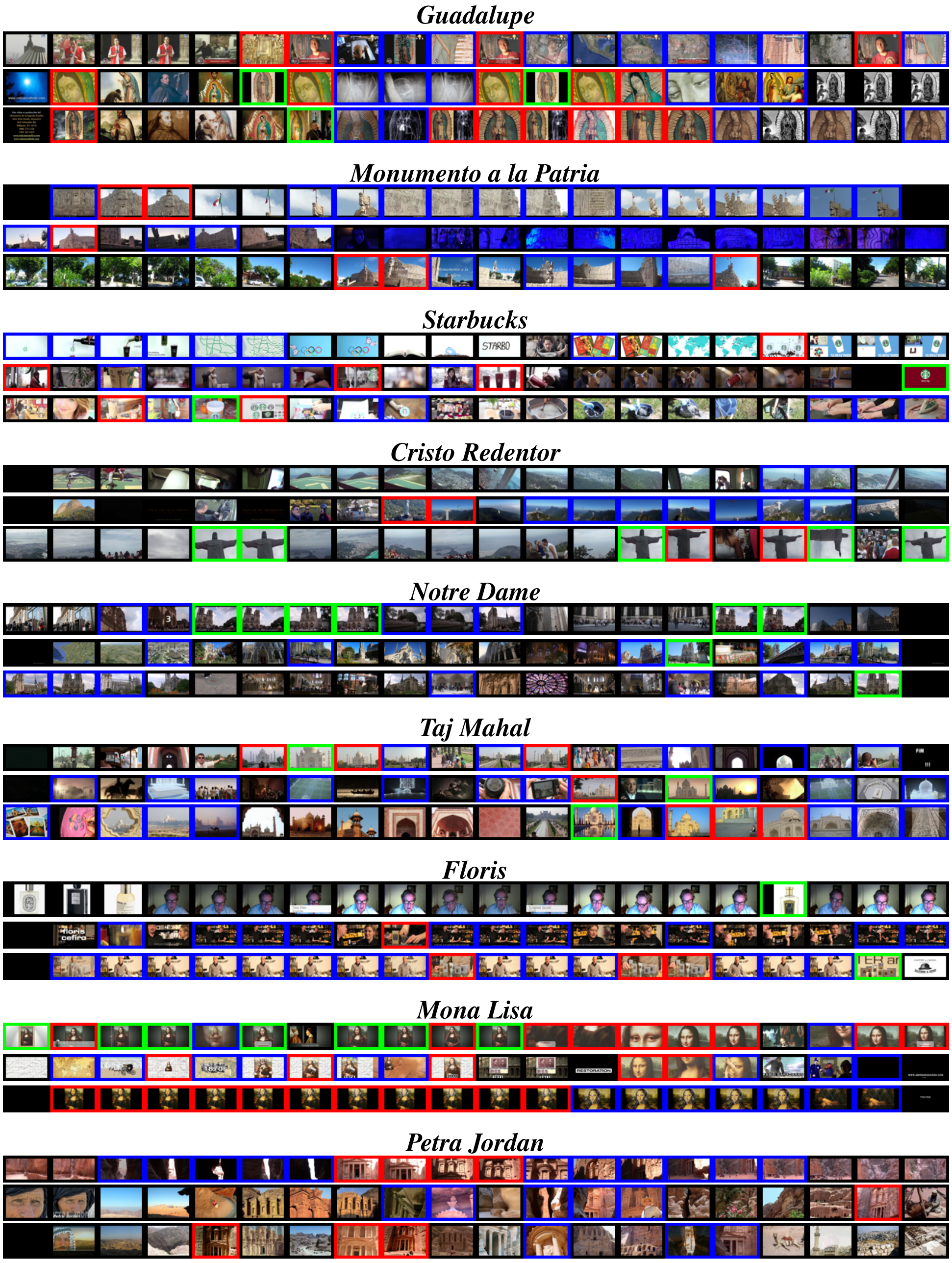


Figure 4.13: Videos of the Specific Object Search (SOS) dataset. Three video sequences for each query are sampled homogeneously. Along the sequences only 20 frames equally separated are shown row-wise. The videos were retrieved from *YouTube* by text searching for the queries. The labelling of the frames is indicated by the colored borders, the *GOOD* frames are **green**, the *OK* frames are **red**, the *JUNK* frames are **blue** and the *NO-OBJECT* frames are **black**.

4.9.2 Experiments

The experiments for testing the performance of the approach use the SOS dataset, 10 queries or objects of interest with 10 related videos each. The evaluation results for the two main tasks are presented in the following sections.

Detection of SOS queries

The four modeling strategies (see Sec. 4.4) are compared in the task of detecting the object of interest in the videos in the configuration of wide-baseline stereo matching. All models are computed with 7 images (full set) of the dataset for a corresponding query, and we present the results over the 10 queries.

The pipeline is configured with four feature detectors of different geometries, i.e. the affine covariant Hessian detector (HessAff), ORB, Saddle and SURF detections. Regarding the descriptors, we experiment with rBRIEF, SURF and SIFT. Finally, the geometric verification step performed by LO-RANSAC approximates affinity (AF) and Homography (HG) as the geometric models.

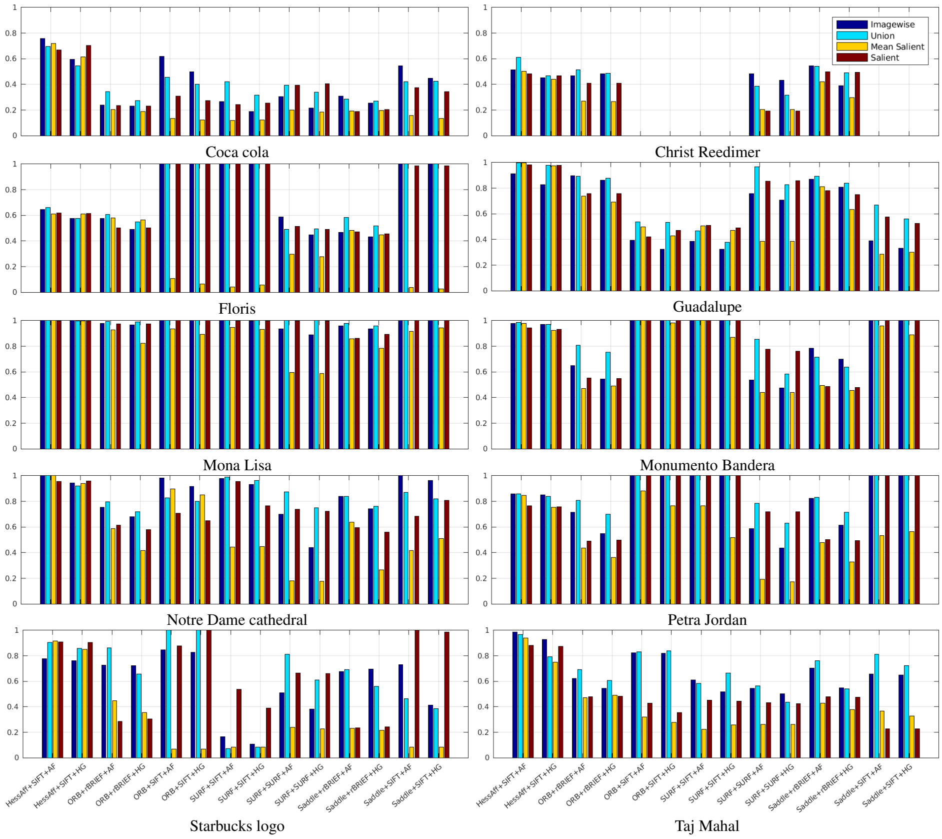


Figure 4.14: Performance comparison of the four model approaches on the 10 landmarks of the SOS dataset. The Y-axis is the Mean Average-Precision and the X-axis indicates the settings of the stereo matcher, i.e. feature detector+ descriptor + RANSAC model, where AF holds for Affinity and HG for Homography.

We compare the performance of the modeling approaches with all possible setups (parameter settings) in Fig. 4.14 that shows the mean average precision (mAP) of the detector under a specific setting <detector> + <descriptor> + <geometric model>. We group the bar plots query-wise since we want to evidence the kind of scenes where the setups perform the best or, in the other hand, the poorest performance for an specific object.

Complementary, Fig. 4.15 shows the mAP for each modeling strategy in average across all queries, in order to simplify the comparison of the setups and models. In most of the cases, Hessian Affine detector performs the best which is expectable based on the stronger normalization of the images patches before descriptions and the multiple filters to keep very distinguishable keypoints with a high precision in the scale selection. However, Hessian Affine is a very costly detector that gives place to the second best performance detector Saddle in average.

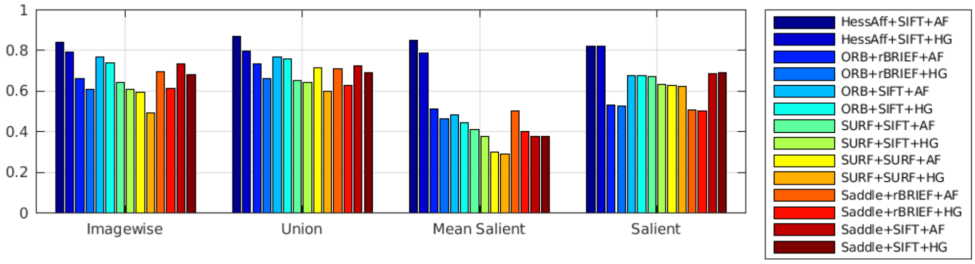
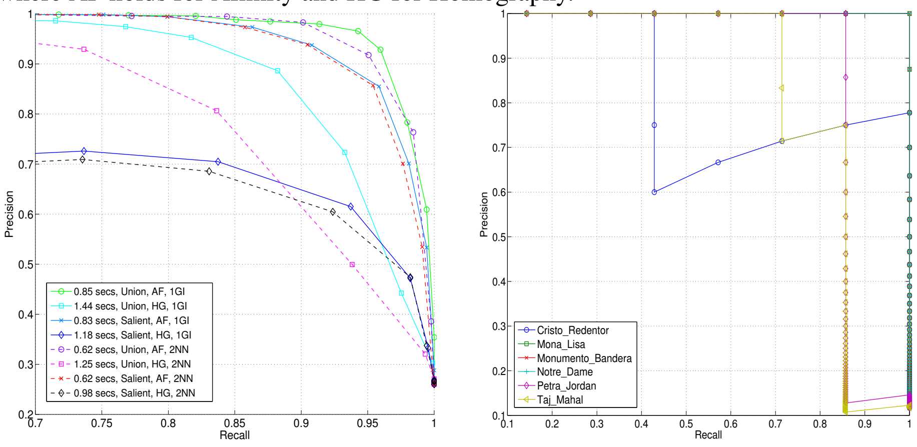


Figure 4.15: Performance comparison of the four model approaches on the SOS dataset. The Y-axis is the Mean Average-Precision and the X-axis indicates the modelling approach used as query, i.e. feature detector + descriptor + RANSAC model, where AF holds for Affinity and HG for Homography.



(a)

(b)

Figure 4.16: (a) Precision-Recall curves on SOS dataset for Union and Salient model. The mean processing time per frame is shown in the legend. (b) Recall/Precision curves for the "Salient, AF, 1GI" method applied to the 10 landmarks.

Supplementary, we present a specific comparison between Union and Salient models. In this experiment we focus in the strategies for tentative correspondences described in Sec. 4.4.6, i.e. 2NN and 1GI. For the sake of simplicity, the local features are detected with HessAffine only, and for consistency with the previous experiment, the geometric verification uses HG [36] and AF [35] as models.

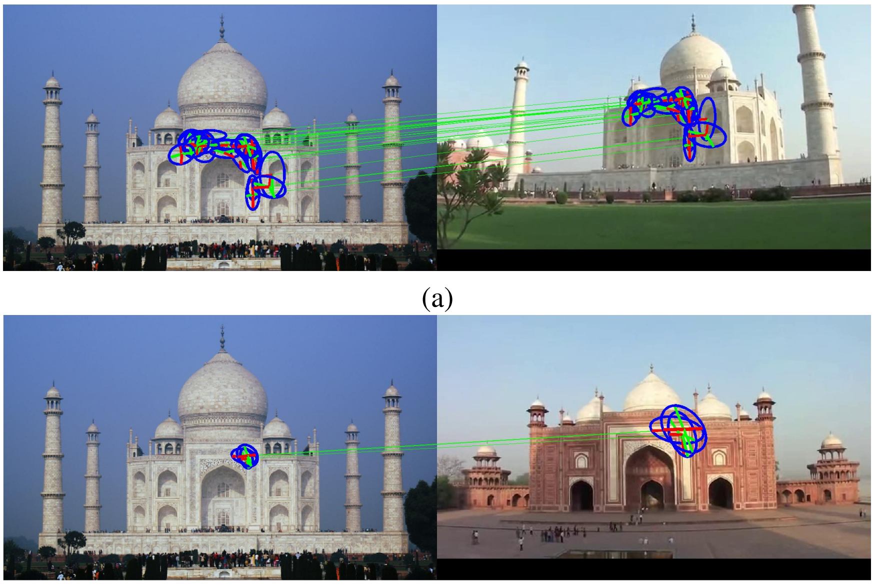
Fig. 4.16 (a) shows precision-recall curves obtained with the 8 setups. It is worth noting that verifying the geometric consistency of feature matches with AF model gives better performance above the HG, due to low inlier ratio HG is unstable and converges incorrect solutions that drop all correct matches. The experiment also revels that Union model performs slightly better compared to Salient model with higher precision and recall. Finally, 1GI gives no significant improvement to Salient model as a consequence of removing very similar features and multiple instances of the same point by the modeling strategy.

Fig. 4.17 presents two examples of the matching results on the *Taj Mahal palace* landmark, using HessAffine, SIFT and AF. The Union model (visualized in the left side with the reference image) is matched with a single frame belonging to a relevant video. In Fig. 4.17(a) the model is matched against a positive sample (frame containing the query and labeled as *GOOD*) shown in the right side of the figure, and Fig. 4.17(b) shows the model matched with a negative sample (the query does not appear in the frame and labeled as *NO-OBJECT*). The visualizations contain correct matches (green lines) and the geometries (blue ellipses) of the keypoints involved.

The pipeline finds 39 correct feature matches with 0.4 inlier ratio in the positive sample and 6 inliers with 0.11 inlier ratio in the negative sample, even though there are no correct matches between the model and the frame. The threshold related to the number of inliers to classifying a frame as positive sample is set to 12, therefore the decision is correct in both examples.

Fixing the parameter of the pipeline to Union model, 1GI and AF, we compute the precision-recall curve with training data composed by 6 out of 7 relevant videos from the 10 objects of SOS dataset. We observe that precision is 0.94 and recall 0.96 for an inlier threshold equal to 12. The validation dataset consist of 10 relevant videos, one per landmark, taken out from the training set. The evaluation with inlier threshold equal to 12 gets average recall and precision of 0.88 and 0.94, respectively, with a average processing time per frame of 0.49 sec. The absolute difference between training and testing is 0.05 and 0.02 for precision and recall, respectively.

In practice, the selection of the inlier threshold is highly dependent on the application, for instance, the classification of videos into relevant or irrelevant requires at least true positive and the number of false positives does not affect the result. Then, for this task the pipeline can be tune for a recall higher than 0.95 with a low precision of 0.3 since with only one frame classified correctly, the whole video is classified as relevant, which is correct up to a score of relevance.



(a)

(b)

Figure 4.17: Examples of *Union* object model matched with a video frame. (a) Positive frame with the query object appearing in the frame and (b) a negative samples where the object is not captured in the frame. In both cases, the reference image of the model is at the left side and the target frame at the right side. The correct features matches are indicated with **green** lines, the geometries of the keypoints are drawn with **blue** ellipses and **red** and blue lines inside the ellipses correspond to the X and Y axis of the canonical frame of the features.

Speed

In the experiment introduced in Sec. 4.9.2, the setup with Union model, AF geometric model, and 1GI matching strategy gets the best performance with respect to precision and recall, however, such setup gets the second fastest mean processing time per frame. The fastest processing time is obtained with *Salient* model, AF and 2NN, with a lower precision compared to the fastest setup with an absolute difference of 8%. The mean processing time per frame with the last setup is 0.83 sec, see Fig. 4.18 for time ratios per processing stage. The processing time for 1GI and 2NN were not significantly different because of the previous SIFT-XY filtering that suppresses multiple instances of the same feature which hurt the 2NN matching strategy.

4.9.3 Video Re-ranking evaluation

The SOS dataset contains the sorted lists of videos, one per query. The videos are sorted by relevance following the criterion described in Sec. 4.9.1. The order of the videos is the ground truth to validate the re-ranking step of our pipeline. In

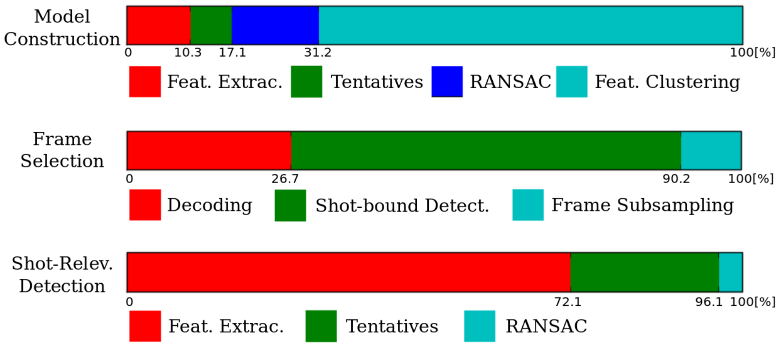


Figure 4.18: The fraction of the time spent in the main steps of the relevance-shot detection (percentages): building the object representation (top), the frame selection (middle) and the detection task (bottom).

order to compare the performance of the re-ranking approach with the text-based search ranked lists, we propose to use the *Kendall tau rank correlation coefficients*. Basically, higher correlation with the ground truth list (order) the better performance. The re-ranked list by our method are higher correlated with the ground truth in 9 of 10 SOS objects, see Table 4.1 for more details.

In order to test the re-ranking step in a more challenging testing set, we increase the number of confusers. In the following experiment, the testing dataset contains all videos (relevant ones and confusers) of 6 queries, i.e. 60 videos in total. In validation, relevant videos of different objects rather than the current query are considered confusers, since none of them contain other SOS objects.

The precision-recall curves for the 6 queries with the *Salient* model are shown in Fig. 4.16 (b). The curves show that for recall 0.95, the average precision is 0.67 and mAP is 0.92 across all queries. For the *Union* model for the same recall value, the average precision is 0.64 with mAP of 0.9.

In our experiments, the approach have a poor performance with the query *Christ Redeemer*, due to the fact that geometries and descriptors of the keypoints detectable in the statue vary with illumination changes. The statue is almost textureless and the observable shapes are artifacts of the shades and shadows projected over the surface. In addition, the relevant videos for this specific query were recorded in very challenging view points, i.e. from long distance, blurring caused by camera vibrations, or strong perspective changes with respect to the images used to model the query.

Considering the video representation (see Sec. 4.5) and the full length of the videos, our average processing time is equivalent to 454.5 fps, the ratios of the time spent on each step of the pipeline is presented in 4.18.

Setting a side the model computation, the shot segmentation, and the video subsampling, the relevance shot detection process the videos with a frame rate of 208 fps. The stereo matching of the images in the pool, as part of the object modeling step, is the most expensive step in time and computation resources. However, modeling the object has constant cost with respect to the length of the videos and number of videos retrieved by text-based search.

Finally, our experiments show that there is a trade-off between the detection accuracy and speed depending on the pipeline settings. The most accurate detection is reached by the HessAffine detector with the slowest performance. On the other hand, Saddle gets the highest precision and recall for the fastest processing time.

Chapter 5

Conclusions

In this chapter we discuss the conclusions about the two main approaches presented in this thesis. In the Section 5.1 we present the discussion is focused on the video re-ranking by the relevant assessment approach (see Chapter 4). In addition, the Section 5.2 presents our conclusions of our Saddle detector. Finally, in Section 5.3 we discuss the future steps in the different research topics presented in the dissertation.

5.1 Relevance assessment

In the thesis, we have considered the following problem. Given a set of images that includes images of an object of interest and possibly outliers and a pool of videos, re-rank the videos by relevance to the object of interest. Further, the videos are augmented with a list of shots depicting the object of interest. The proposed approach first builds a visual model of the object of interest based on local image features. The relevant shot detection builds on wide baseline stereo matching. Shot relevance is defined as the recording time spent capturing the object of interest reflected in the number of frames depicting it. A number of algorithmic options have been experimentally evaluated. The experiments were carried out on a set of 100 videos collected querying You-Tube with 10 different text phrases.

The best performing method builds the model as a union of features from all example images and constructed the tentative correspondences using the 1^{st} geometrically inconsistent rule. Averaged over the 10 landmarks, mAP is 0.92 querying the object model based on salient features that turns out to outperforms the union model by 2%

on mAP. The implementation runs faster than real-time at 208 fps.

5.2 Saddle

Experiments show that the Saddle features are general, evenly spread ad appearing in high density in a range of images. The Saddle detector is among the fastest proposed. In comparison with detector with similar speed, the Saddle features show superior matching performance on number of challenging datasets, and it shows to have better coverage with respect to ORB on detection. Our features matched correctly are wider spread on the image compared to ORB after matching under the same conditions. Furthermore, a recently proposed stability evaluation of interest point detectors [67] on a street-level view dataset shows that Saddle is the best performing detector. After reprojection, Saddle overlaps the negative determinant Hessian features on natural images.

Under some conditions the repeatability and the number of (descriptor-independent) correspondences is better for Saddle compared to ORB. In order to find the best setup for Saddle detector, we run a experiment aimed to find a correlation between the number of solved problems (image pairs correctly matched) and F1 score. The matching strategy known as first geometric inconsistent with Hamming distance improves significantly the inlier ratio compared to the standard hard thresholding.

5.3 Future work

In this section we discuss some of the future directions for further work in the topic.

The experiments for testing the location accuracy of keypoint detectors, presented in Section 3.3, show that Saddle is outperformed by ORB with a lower mean reprojection error of inlier matches. The main reason is that Saddle mostly fires in fringe-like regions (see Fig. 3.7), structures that allow location drifting related to the aperture problem. It is possible to avoid firing Saddle in this kind of regions increasing ε, however the number of keypoints drops very fast with this parameter change. It is worth to research a solution for this trade off without losing speed in detection by adding expensive filtering criteria.

We proposed Saddle as a similarity covariant feature detector that it is inspired by

ORB, and we show that it is suitable for a similar detection/description framework. The approach for computing the orientation of Saddle keypoints is the intensity centroid to be described by rBRIEF. Such orientation estimation is unstable by construction at intensity saddle points but it is convenient for our task because can be computed very fast. The standard histogram of gradient orientation [106] gives a robust estimation at a very high computational cost, even the orientation computed in SURF descriptor is significant slower than rBRIEF. Therefore, a very fast and robust orientation estimation for saddle points is an important topic to research and it is out of the scope of this thesis.

In our experiments for image matching using BRIEF descriptor, there is no retraining of the descriptor for our specific detector. BRIEF is trained on FAST features that fires in high contrast corners which is a different morphology than Saddle detection. Additionally, the image dataset used to collect image patches of FAST detection is PASCAL dataset, which lead us to research on the effects of using more diverse dataset for training and improving the generalization of the descriptor. In consequence, it is worth boosting the descriptive power of BRIEF as one of the fastest binary descriptor for our detector.

In the relevancy assessment for visual video re-raking there is plenty room for improvement, for instance, expanding the pool of images progressively. Once the model is constructed with the pool as input, it is possible to include frames where the query is detected while the video is scanned sequentially. The operation is almost for free since the stereo matching is part of the object detection step, therefore, the estimated geometric relation, between model and frame, is an available outcome used to reproject the features of the additional view into the model. Previous work on adding new views of the target object can be found for visual object tracking [76].

On the other hand, the incremental strategy to enlarge the pool of images with different views of the object leads to a natural extension of our work to model the query as a 3-D model. A Strong inspiration can be found in the work of Lebeda [91], which combines techniques from the fields of visual tracking, SfM and SLAM to model the 3-D structure of the object to address the appearance changes of the objects due to out-of-plane rotations.

In relation to the SSD shot boundary detector proposed in this thesis, there is a potential improvement in the forwarding stage. Specifically, the coefficients of the linear search can be learned while the video scan is performed. The intuition is that certain types of videos has a characteristic behavior, i.e. length and number of shots of a documentary, soap opera, sport video, etc. Statistic models can be computed from the temporal location of the shot boundaries detected while the video is scanned in order to predict the length of the following shots.

Finally, in the technical part of the proposed solution, as we describe in Section 4.5, downloading videos is a mandatory step in the pipeline that represent a bottleneck in the workflow. Currently, YouTube does not grant access to the encoded videos directly. It is desirable to read out of this step having direct access to a video dataset. A possible solution is to incorporate the video re-ranking by visual relevance assessment pipeline as part of the search engine of the video sharing website.

Bibliography

[1] Sami Abu-El-Haija, Nisarg Kothari, Joonseok Lee, Apostol (Paul) Natsev, George Toderici, Balakrishnan Varadarajan, and Sudheendra Vijayanarasimhan. YouTube-8M: A large-scale video classification benchmark. In *arXiv:1609.08675*, 2016.

[2] Alexandre Alahi, Raphaël Ortiz, and Pierre Vandergheynst. FREAK: Fast Retina Keypoint. In *CVPR*, 2012.

[3] Javier Aldana-Iuit. Wide-Baseline Stereo Matching for Object Detection on Videos. In Libor Husník, editor, *Proc. 18th International Student Conference on Electrical Engineering*, Technicka 2, Prague, Czech Republic, May 2014. Czech Technical University in Prague. Paper key ICS13.

[4] Javier Aldana-Iuit, Ondřej Chum, and Jiři Matas. Relevance Assessment for Visual Video Re-ranking. In *ICIAR*, Vilamoura, Algarve, Portugal, October 2014. Springer.

[5] Javier Aldana-Iuit, M. Elena Martinez-Perez, Arturo Espinosa-Romero, and Rufino Diaz-Uribe. Minimizing camera-eye optical aberrations during the 3-D reconstruction of retinal structures. In *Optics, Photonics, and Digital Technologies for Multimedia Applications*, volume 7723, pages 406 – 415. SPIE, 2010.

[6] Javier Aldana-Iuit, Dmytro Mishkin, Ondrej Chum, and Jiri Matas. In the Saddle: Chasing fast and repeatable features. In *ICPR*, pages 675–680, 2016.

[7] Javier Aldana-Iuit, Dmytro Mishkin, Ondřej Chum, and Jiří Matas. Saddle: Fast and repeatable features with good coverage. *Image and Vision Computing*, 2019.

[8] Mihael Ankerst, Markus M. Breunig, Hans-Peter Kriegel, and Jörg Sander. OPTICS: Ordering Points to Identify the Clustering Structure. *SIGMOD Rec.*, 28(2):49–60, June 1999.

[9] R. Arandjelović and A. Zisserman. Three things everyone should know to improve object retrieval. In *CVPR*, 2012.

[10] Francis Bach, Rodolphe Jenatton, Julien Mairal, and Guillaume Obozinski. Optimization with sparsity-inducing penalties. *Found. Trends Mach. Learn.*, 4(1):1–106, January 2012.

[11] Vassileios Balntas, Karel Lenc, Andrea Vedaldi, and Krystian Mikolajczyk. HPatches: A benchmark and evaluation of handcrafted and learned local descriptors. In *CVPR*, pages 3852–3861. IEEE Computer Society, 2017.

[12] Vassileios Balntas, Vincent Lepetit, Johannes Schönberger, Eduard Trulls, and Kwang Moo Yi. CVPR 2019 Workshop. Image Matching: Local Features and Beyond. In *CVPR*, 2019.

[13] Lorenzo Baraldi, Costantino Grana, and Rita Cucchiara. A deep siamese network for scene detection in broadcast videos. In *ACM*, 2015.

[14] Lorenzo Baraldi, Costantino Grana, and Rita Cucchiara. Measuring scene detection performance. In *IbPRIA*, 2015.

[15] Lorenzo Baraldi, Costantino Grana, and Rita Cucchiara. Shot and scene detection via hierarchical clustering for re-using broadcast video. In George Azzopardi and Nicolai Petkov, editors, *Computer Analysis of Images and Patterns*, pages 801–811, Cham, 2015. Springer International Publishing.

[16] Lorenzo Baraldi, Costantino Grana, and Rita Cucchiara. Neuralstory: An interactive multimedia system for video indexing and re-use. In *Proceedings of the 15th International Workshop on Content-Based Multimedia Indexing*, pages 21:1–21:6, New York, NY, USA, 2017. ACM.

[17] Adam Baumberg. Reliable feature matching across widely separated views. In *CVPR*, 2000.

[18] Herbert Bay, Andreas Ess, Tinne Tuytelaars, and Luc Van Gool. Speeded-Up Robust Features (SURF). *CVIU*, 2008.

[19] P. R. Beaudet. Rotationally invariant image operators. In *Proceedings of the 4th International Joint Conference on Pattern Recognition*, pages 579–583, Kyoto, Japan, November 1978.

[20] Neil E Berthier. Learning to reach: a mathematical model. *Developmental psychology*, 32(5):811, 1996.

[21] Michael Bleyer, Carsten Rother, and Pushmeet Kohli. Surface stereo with soft segmentation. In *CVPR*, pages 1570–1577. IEEE Computer Society, 2010.

[22] Liefeng Bo and Cristian Sminchisescu. Efficient match kernel between sets of features for visual recognition. In Y. Bengio, D. Schuurmans, J. D. Lafferty, C. K. I. Williams, and A. Culotta, editors, *NIPS*, pages 135–143. Curran Associates, Inc., 2009.

[23] J. S. Boreczky and L. A. Rowe. Comparison of video shot boundary detection techniques. *Storage and Retrieval for Still Image and Video Databases IV*, pages 170–179, 1996.

[24] Y. Boykov, O. Veksler, and R. Zabih. Fast approximate energy minimization via graph cuts. *PAMI*, 23(11):1222–1239, Nov 2001.

[25] Jane Bromley, Isabelle Guyon, Yann LeCun, Eduard Säckinger, and Roopak Shah. Signature verification using a "siamese" time delay neural network. In J. D. Cowan, G. Tesauro, and J. Alspector, editors, *NIPS*, pages 737–744. Morgan-Kaufmann, 1994.

[26] M. Brown, G. Hua, and S. Winder. Discriminative learning of local image descriptors. *PAMI*, 33(1):43–57, Jan 2011.

[27] M. Brown and D. Lowe. Invariant features from interest point groups. In *BMVC*, pages 23.1–23.10. BMVA Press, 2002. doi:10.5244/C.16.23.

[28] T. Brox, A. Bruhn, N. Papenberg, and J. Weickert. High accuracy optical flow estimation based on a theory for warping. In *ECCV*, volume 3024 of *Lecture Notes in Computer Science*, pages 25–36. Springer, May 2004.

[29] M. Calonder, V. Lepetit, M. Ozuysal, T. Trzcinski, C. Strecha, and P. Fua. BRIEF: Computing a local binary descriptor very fast. *PAMI*, 34(7):1281–1298, July 2012.

[30] Aaron Chadha, Alhabib Abbas, and Yiannis Andreopoulos. Video classification with cnns: Using the codec as a spatio-temporal activity sensor. *IEEE Transactions on Circuits and Systems for Video Technology*, 29:475–485, 2017.

[31] N. H. Chan, K. Hasikin, and N. A. Kadri. Evaluation of feature descriptor on d-saddle keypoint detection in retinal image registration. In *2019 IEEE 15th International Colloquium on Signal Processing Its Applications (CSPA)*, pages 178–181, March 2019.

[32] A. L. M. Chiu and Ada Wai-chee Fu. Enhancements on local outlier detection. In *Seventh International Database Engineering and Applications Symposium, 2003. Proceedings.*, pages 298–307, July 2003.

[33] Do Kook Choe and Eugene Charniak. Parsing as language modeling. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 2331–2336, Austin, Texas, November 2016. Association for Computational Linguistics.

[34] Sumit Chopra, Raia Hadsell, and Yann LeCun. Learning a similarity metric discriminatively, with application to face verification. In *CVPR*, pages 539–546, Washington, DC, USA, 2005. IEEE Computer Society.

[35] O. Chum and J. Matas. Homography estimation from correspondences of local elliptical features. In *Proceedings of the 21st International Conference on Pattern Recognition (ICPR2012)*, pages 3236–3239, Nov 2012.

[36] O. Chum, J. Matas, and Josef Kittler. Locally Optimized RANSAC. In Bernd Michaelis and Gerald Krell, editors, *Pattern Recognition*, volume 2781 of *Lecture Notes in Computer Science*, pages 236–243. Springer Berlin Heidelberg, 2003.

[37] O. Chum, A. Mikulík, M. Perdoch, and J. Matas. Total recall II: Query expansion revisited. In *CVPR*, pages 889–896, June 2011.

[38] Ondrej Chum and Jiri Matas. Matching with PROSAC - Progressive Sample Consensus. In *CVPR*, pages 220–226, Washington, DC, USA, 2005. IEEE Computer Society.

[39] Kai Cordes, Bodo Rosenhahn, and Jörn Ostermann. High-resolution feature evaluation benchmark. In Richard Wilson, Edwin Hancock, Adrian Bors, and William Smith, editors, *Computer Analysis of Images and Patterns*, pages 327–334, Berlin, Heidelberg, 2013. Springer Berlin Heidelberg.

[40] Corinna Cortes and Vladimir Vapnik. Support-Vector Networks. *Machine Learning*, 20(3):273–297, September 1995.

[41] Navneet Dalal and Bill Triggs. Histograms of oriented gradients for human detection. In *CVPR*, pages 886–893, Washington, DC, USA, 2005. IEEE Computer Society.

[42] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei. ImageNet: A Large-Scale Hierarchical Image Database. In *CVPR*, 2009.

[43] Daniel DeTone, Tomasz Malisiewicz, and Andrew Rabinovich. SuperPoint: Self-Supervised Interest Point Detection and Description. *CVPRW*, pages 337–33712, 2018.

[44] Nga Hang Do and Keiji Yanai. Automatic construction of action datasets using web videos with density-based cluster analysis and outlier detection. In *Revised Selected Papers of the 7th Pacific-Rim Symposium on Image and Video Technology - Volume 9431*, PSIVT 2015, pages 160–172, New York, NY, USA, 2016. Springer-Verlag New York, Inc.

[45] J. Donahue, L. A. Hendricks, M. Rohrbach, S. Venugopalan, S. Guadarrama, K. Saenko, and T. Darrell. Long-term recurrent convolutional networks for visual recognition and description. *PAMI*, 39(4):677–691, April 2017.

[46] John Eakins, Margaret Graham, John Eakins, Margaret Graham, and Tom Franklin. Content-based image retrieval. *Library and Information Briefings*, 85:1–15, 1999.

[47] Eric Enge, Stephan Spencer, Jessie Stricchiola, and Rand Fishkin. *The Art of SEO*. O'Reilly Media, Inc., 2nd edition, 2012.

[48] Martin Ester, Hans-Peter Kriegel, Jörg Sander, and Xiaowei Xu. A density-based algorithm for discovering clusters a density-based algorithm for discovering clusters in large spatial databases with noise. In *KDD*, pages 226–231. AAAI Press, 1996.

[49] Fabrice Bellard. Ffmpeg - a complete, cross-platform solution to record, convert and stream audio and video, 2019. [Online; accessed 16.10.2019].

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

[51] Yoav Freund and Robert E. Schapire. A short introduction to boosting. In *In Proceedings of the Sixteenth International Joint Conference on Artificial Intelligence*, pages 1401–1406. Morgan Kaufmann, 1999.

[52] Zeng fu Wang and Zhi gang Zheng. A region based stereo matching algorithm using cooperative optimization. In *CVPR*, pages 887–894, 2008.

[53] W. Förstner, T. Dickscheid, and F. Schindler. Detecting interpretable and accurate scale-invariant keypoints. In *ICCV*, pages 2256–2263, Sep. 2009.

[54] Yoav Goldberg. A primer on neural network models for natural language processing. *JAIR*, 57(1):345–420, September 2016.

[55] Google Developers. YouTube - Data API. `https://developers.google.com/youtube/v3/docs`, 2019. Online, accessed: 15.10.2019.

[56] Petr Gronát, Javier Aldana-Iuit, and Martin Bálek. MaxNet: Neural network architecture for continuous detection of malicious activity. *IEEE Security and Privacy Workshops*, pages 28–35, 2019.

[57] Yuri Gurevich. Sequential abstract-state machines capture sequential algorithms. *ACM Trans. Comput. Logic*, 1(1):77–111, July 2000.

[58] D. Hamester, P. Barros, and S. Wermter. Face expression recognition with a 2-channel convolutional neural network. In *IJCNN*, pages 1–8, July 2015.

[59] Chris Harris and Mike Stephens. A combined corner and edge detector. In *In Proc. of Fourth Alvey Vision Conference*, pages 147–151, 1988.

[60] Chris Harris and Mike Stephens. A combined corner and edge detector. In *AVC*, 1988.

[61] Richard Hartley and Andrew Zisserman. *Multiple View Geometry in Computer Vision*. Cambridge University Press, New York, NY, USA, 2 edition, 2003.

[62] D.C. Hauagge and N. Snavely. Image matching using local symmetry features. In *CVPR*, 2012.

[63] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. *CVPR*, pages 770–778, 2016.

[64] Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural Computation*, 9(8):1735–1780, November 1997.

[65] C. Huang, H. Lee, and C. Chen. Shot change detection via local keypoint matching. *IEEE Transactions on Multimedia*, 10(6):1097–1108, Oct 2008.

[66] Xiaofei Huang. Cooperative optimization for energy minimization in computer vision: A case study of stereo matching. In Carl Edward Rasmussen, Heinrich H. Bülthoff, Bernhard Schölkopf, and Martin A. Giese, editors, *Pattern Recognition*, pages 302–309, Berlin, Heidelberg, 2004. Springer Berlin Heidelberg.

[67] Xinyu Huang, Xinjing Cheng, Qichuan Geng, Binbin Cao, Dingfu Zhou, Peng Wang, Yuanqing Lin, and Ruigang Yang. The apolloscape dataset for autonomous driving. *CoRR*, abs/1803.06184, 2018.

[68] Lloyd H. Hughes, Michael Schmitt, Lichao Mou, Yuanyuan Wang, and Xiao xiang Zhu. Identifying corresponding patches in sar and optical images with a pseudo-siamese cnn. *IEEE Geoscience and Remote Sensing Letters*, 15:784–788, 2018.

[69] Instagram, copyright. Press page Instagram. `http://instagram.com/press/`. Online accessed: 15.08.2014.

[70] Arnold Irschara, Christopher Zach, Jan-Michael Frahm, and Horst Bischof. From structure-from-motion point clouds to fast location recognition. In *CVPR*, 2009.

[71] N. Jacobs, N. Roman, and R. Pless. Consistent temporal variations in many outdoor scenes. In *CVPR*, pages 1–6, June 2007.

[72] Tomáš Jeníček. Canonical views extraction from multimedia databases using non-image information. Master's thesis, Czech Technical University in Prague, Prague. Czech Republic., January 2017.

[73] Shuiwang Ji, Wei Xu, Ming Yang, and Kai Yu. 3-D convolutional neural networks for human action recognition. *PAMI*, 35(1):221–231, 2013.

[74] Jian Sun, Nan-Ning Zheng, and Heung-Yeung Shum. Stereo matching using belief propagation. *PAMI*, 25(7):787–800, July 2003.

[75] Yushi Jing and Shumeet Baluja. Visualrank: Applying pagerank to large-scale image search. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 30:1877–1890, 2008.

[76] Zdenek Kalal, Krystian Mikolajczyk, and Jiri Matas. Tracking-learning-detection. *PAMI*, 34(7):1409–1422, July 2012.

[77] T. Kanade and M. Okutomi. A stereo matching algorithm with an adaptive window: Theory and experiment. *IEEE Trans. Pattern Anal. Mach. Intell.*, 16(9):920–932, September 1994.

[78] A. Karpathy, G. Toderici, S. Shetty, T. Leung, R. Sukthankar, and L. Fei-Fei. Large-scale video classification with convolutional neural networks. In *CVPR*, pages 1725–1732, June 2014.

[79] Avi Kelman, Michal Sofka, and Charles V Stewart. Keypoint descriptors for matching across multiple image modalities and non-linear intensity variations. In *CVPR*, 2007.

[80] R. Kinderman and S.L. Snell. *Markov random fields and their applications*. American mathematical society, 1980.

[81] Jacek Komorowski, Konrad Czarnota, Tomasz Trzcinski, Lukasz Dabala, and Simon Lynen. Interest point detectors stability evaluation on apolloscape dataset. In *ECCV*, 2018.

[82] Piotr Koniusz, Fei Yan, and Krystian Mikolajczyk. Comparison of mid-level feature coding approaches and pooling strategies in visual concept detection. *CVIU*, 117(5):479 – 492, 2013.

[83] Kostas Kostalampros. How to optimize your YouTube videos and channel. `http://www.searchenginejournal.com/how-to-optimize-your-youtube-videos-and-channel/56708/`. Accessed: 13.08.2014.

[84] Tomás Krajník, Pablo de Cristóforis, Keerthy Kusumam, Peer Neubert, and Tom Duckett. Image features for visual teach-and-repeat navigation in changing environments. *Robotics and Autonomous Systems*, 88:127–141, 2017.

[85] Alex Krizhevsky. Learning multiple layers of features from tiny images. Technical report, 2009.

[86] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E. Hinton. ImageNet classification with deep convolutional neural networks. *NIPS*, 25, 01 2012.

[87] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. ImageNet classification with deep convolutional neural networks. In F. Pereira, C. J. C. Burges, L. Bottou, and K. Q. Weinberger, editors, *NIPS*, pages 1097–1105. Curran Associates, Inc., 2012.

[88] Rekhil M Kumar and K Sreekumar. A survey on image feature descriptors. *Int J Comput Sci Inf Technol*, 5:7668–7673, 2014.

[89] Shuyue Lan, Rameswar Panda, Qi Zhu, and Amit K. Roy-Chowdhury. FFNet: Video fast-forwarding via reinforcement learning. *CoRR*, abs/1805.02792, 2018.

[90] Steffen L. Lauritzen. *Graphical Models*. Oxford University Press, 1996.

[91] Karel Lebeda, Simon Hadfield, and Richard Bowden. 2-D or not 2-D: Bridging the gap between tracking and structure from motion. 11 2014.

[92] Karel Lebeda, Ji?í Matas, and Ondrej Chum. Fixing the Locally Optimized RANSAC. In *BMVC*, 2012.

[93] Victor S. Lempitsky, Carsten Rother, and Andrew Blake. Logcut - efficient graph cut optimization for markov random fields. In *ICCV*, pages 1–8. IEEE Computer Society, 2007.

[94] Karel Lenc and Andrea Vedaldi. Learning covariant feature detectors. In *ECCV Workshops (3)*, volume 9915 of *Lecture Notes in Computer Science*, pages 100–117, 2016.

[95] C. Leng, H. Zhang, B. Li, G. Cai, Z. Pei, and L. He. Local feature descriptor for image matching: A survey. *IEEE Access*, 7:6424–6434, 2019.

[96] S. Leutenegger, M. Chli, and R. Y. Siegwart. Brisk: Binary robust invariant scalable keypoints. In *ICCV*, pages 2548–2555, Nov 2011.

[97] Stefan Leutenegger, Margarita Chli, and Roland Y. Siegwart. BRISK: Binary robust invariant scalable keypoints. In *ICCV*, 2011.

[98] Gil Levi and Tal Hassner. LATCH: Learned arrangements of three patch codes. *WACV*, pages 1–9, 2016.

[99] Li Hong and G. Chen. Segment-based stereo matching using graph cuts. In *CVPR*, volume 1, pages I–I, June 2004.

[100] Jessica Lin, Eamonn Keogh, Stefano Lonardi, and Bill Chiu. A symbolic representation of time series, with implications for streaming algorithms. In *DMKD*, pages 2–11, New York, NY, USA, 2003. ACM.

[101] Tony Lindeberg. *Discrete Scale-Space Theory and the Scale-Space Primal Sketch*. PhD thesis, Royal Inst. of Technology, Stockholm, Sweden, 1991.

[102] Tony Lindeberg. *Scale-Space Theory in Computer Vision*. Kluwer Academic Publishers, Norwell, MA, USA, 1994.

[103] Tony Lindeberg. Feature detection with automatic scale selection. *IJCV*, 30(2):79–116, November 1998.

[104] Tony Lindeberg. Image matching using generalized scale-space interest points. *Journal of Mathematical Imaging and Vision*, 52(1):3–36, May 2015.

[105] Tony Lindeberg and Jonas Gårding. Shape-adapted smoothing in estimation of 3-D shape cues from affine deformations of local 2-D brightness structure. *Image Vision Comput.*, 15(6):415–434, 1997.

[106] David G. Lowe. Object recognition from local scale-invariant features. In *ICCV*, pages 1150–1157, 1999.

[107] David G. Lowe. Distinctive image features from scale-invariant keypoints. *Int. J. Comput. Vision*, 60(2):91–110, November 2004.

[108] Bruce D. Lucas and Takeo Kanade. An iterative image registration technique with an application to stereo vision. In *Proceedings of the 7th International Joint Conference on Artificial Intelligence - Volume 2*, IJCAI'81, pages 674–679, San Francisco, CA, USA, 1981. Morgan Kaufmann Publishers Inc.

[109] Wenjie Luo, Alexander G. Schwing, and Raquel Urtasun. Efficient deep learning for stereo matching. *CVPR*, pages 5695–5703, 2016.

[110] Elmar Mair, Gregory D. Hager, Darius Burschka, Michael Suppa, and Gerhard Hirzinger. Adaptive and generic corner detection based on the accelerated segment test. In *ECCV*, ECCV'10, pages 183–196, Berlin, Heidelberg, 2010. Springer-Verlag.

[111] J. Matas, O. Chum, M. Urban, and T. Pajdla. Robust Wide Baseline Stereo from Maximally Stable Extremal Regions. In *BMVC*, pages 384–393, 2002.

[112] Jiri Matas, Stepán Obdrzálek, and Ondrej Chum. Local affine frames for wide-baseline stereo. In *ICPR (4)*, pages 363–366, 2002.

[113] Josh McDermott and Edward H. Adelson. The geometry of the occluding contour and its effect on motion interpretation. *Journal of Vision*, 4(10):9–9, 11 2004.

[114] Franck Michel. How many public photos are uploaded to flickr every day, month, year? `https://www.flickr.com/photos/franckmichel/6855169886/`. Online accessed: 15.08.2014.

[115] Daniela Micucci, Marco Mobilio, and Paolo Napoletano. UniMiB SHAR: A new dataset for human activity recognition using acceleration data from smartphones. *CoRR*, abs/1611.07688, 2016.

[116] K. Mikolajczyk and J. Matas. Improving descriptors for fast tree matching by optimal linear projection. In *ICCV*, pages 1–8, Oct 2007.

[117] K. Mikolajczyk and C. Schmid. Indexing based on scale invariant interest points. In *ICCV*, volume 1, pages 525–531 vol.1, July 2001.

[118] K. Mikolajczyk and C. Schmid. An affine invariant interest point detector. In *ECCV*, ECCV '02, pages 128–142, London, UK, UK, 2002. Springer-Verlag.

[119] K. Mikolajczyk and C. Schmid. A performance evaluation of local descriptors. In *CVPR*, volume 2, pages II–II, June 2003.

[120] K. Mikolajczyk and C. Schmid. A performance evaluation of local descriptors. *PAMI*, 2005.

[121] K. Mikolajczyk, T. Tuytelaars, C. Schmid, A. Zisserman, J. Matas, F. Schaffalitzky, T. Kadir, and L. Van Gool. A comparison of affine region detectors. *Int. J. Comput. Vision*, 65(1-2):43–72, November 2005.

[122] K. Mikolajczyk, T. Tuytelaars, C. Schmid, A. Zisserman, J. Matas, F. Schaffalitzky, T. Kadir, and L. Van Gool. A comparison of affine region detectors. *IJCV*, 65(1):43–72, Nov 2005.

[123] Krystian Mikolajczyk and Cordelia Schmid. Scale & affine invariant interest point detectors. *IJCV*, 60(1):63–86, 2004.

[124] Krystian Mikolajczyk and Cordelia Schmid. Scale &amp; affine invariant interest point detectors. *Int. J. Comput. Vision*, 60(1):63–86, October 2004.

[125] Andrej Mikulík, Filip Radenovic, Ondrej Chum, and Jiri Matas. Efficient image detail mining. In *ACCV*, 2014.

[126] F. Mindru, T. Moons, and L. Van Gool. Recognizing color patterns irrespective of viewpoint and illumination. In *CVPR*, volume 1, pages 368–373 Vol. 1, June 1999.

[127] Anastasiya Mishchuk, Dmytro Mishkin, Filip Radenovic, and Jiri Matas. Working hard to know your neighbor's margins: Local descriptor learning loss. In *NIPS*, 2017.

[128] Anastasiya Mishchuk, Dmytro Mishkin, Filip Radenovic, and Jiri Matas. Working hard to know your neighbor's margins: Local descriptor learning loss. In *NIPS*, 2017.

[129] Anastasiya Mishchuk, Dmytro Mishkin, Filip Radenovic, and Jiri Matas. Working hard to know your neighbor's margins: Local descriptor learning loss. In *Proceedings of NIPS*, December 2017.

[130] D. Mishkin, J. Matas, M. Perdoch, and K. Lenc. WxBS: Wide Baseline Stereo Generalizations. In *BMVC*, 2015.

[131] Dmytro Mishkin, Jiri Matas, and Michal Perdoch. MODS: fast and robust method for two-view matching. *CVIU*, 141:81–93, 2015.

[132] Dmytro Mishkin, Jiri Matas, and Michal Perdoch. Mods: Fast and robust method for two-view matching. *CVIU*, 141:81 – 93, 2015.

[133] Dmytro Mishkin, Jiri Matas, Michal Perdoch, and Karel Lenc. Wxbs: Wide baseline stereo generalizations. In *BMVC*, pages 12.1–12.12. BMVA Press, 2015.

[134] Dmytro Mishkin, Michal Perdoch, and Jiri Matas. Two-view matching with view synthesis revisited. In *IVCNZ*, pages 436–441, 2013.

[135] Dmytro Mishkin, Filip Radenovic, and Jiri Matas. Repeatability Is Not Enough: Learning Discriminative Affine Regions via Discriminability. In *ECCV*, September 2018.

[136] Jean-Michel Morel and Guoshen Yu. ASIFT: A new framework for fully affine invariant image comparison. *SIAM J. Img. Sci.*, 2(2):438–469, April 2009.

[137] E. N. Mortensen, Hongli Deng, and L. Shapiro. A SIFT descriptor with global context. In *CVPR*, volume 1, pages 184–190 vol. 1, June 2005.

[138] Marius Muja and David G. Lowe. Fast approximate nearest neighbors with automatic algorithm configuration. In *VISSAPP*, pages 331–340. INSTICC Press, 2009.

[139] Arun Mukundan, Giorgos Tolias, Andrei Bursuc, Herv'e J'egou, and Ondrej Chum. Understanding and improving kernel local descriptors. *IJCV*, pages 1–15, 2018.

[140] Raúl Mur-Artal, J. M. M. Montiel, and Juan D. Tardós. ORB-SLAM: a versatile and accurate monocular SLAM system. *IEEE Transactions on Robotics*, 31(5):1147–1163, 2015.

[141] Raul Mur-Artal, J. M. M. Montiel, and Juan D. Tardós. Orb-slam: a versatile and accurate monocular slam system. *CoRR*, abs/1502.00956, 2015.

[142] Juan Carlos Niebles, Chih-Wei Chen, and Li Fei-Fei. Modeling temporal structure of decomposable motion segments for activity classification. In *ECCV*, 2010.

[143] Akitsugu Noguchi and Keiji Yanai. A SURF-based spatio-temporal feature for feature-fusion-based action recognition. In Kiriakos N. Kutulakos, editor, *Trends and Topics in Computer Vision*, pages 153–167, Berlin, Heidelberg, 2012. Springer Berlin Heidelberg.

[144] Stepán Obdrzálek and Jiri Matas. Object recognition using local affine frames on distinguished regions. In *BMVC*, pages 1–10, 2002.

[145] Yuki Ono, Eduard Trulls, Pascal Fua, and Kwang Moo Yi. LF-Net: Learning local features from images. In *NIPS*, 2018.

[146] M. Perd'och, O. Chum, and J. Matas. Efficient representation of local geometry for large scale object retrieval. In *CVPR*, pages 9–16, June 2009.

[147] Luis Perez and Jason Wang. The effectiveness of data augmentation in image classification using deep learning. *CoRR*, abs/1712.04621, 2017.

[148] J. Philbin, O. Chum, M. Isard, J. Sivic, and A. Zisserman. Object retrieval with large vocabularies and fast spatial matching. In *CVPR*, 2007.

[149] James Philbin, Michael Isard, Josef Sivic, and Andrew Zisserman. Descriptor learning for efficient retrieval. In *ECCV*, ECCV'10, pages 677–691, Berlin, Heidelberg, 2010. Springer-Verlag.

[150] Pedro Piniés, Lina María Paz, Dorian Gálvez-López, and Juan D Tardós. Ci-graph simultaneous localization and mapping for three-dimensional reconstruction of large and complex environments using a multicamera system. *Journal of Field Robotics*, 27(5):561–586, 2010.

[151] P. Pritchett and A. Zisserman. Wide baseline stereo matching. In *ICCV*, pages 754–760, Jan 1998.

[152] F. Radenovic, J. L. Schönberger, D. Ji, J. Frahm, O. Chum, and J. Matas. From dusk till dawn: Modeling in the dark. In *CVPR*, pages 5488–5496, June 2016.

[153] F. Radenović, G. Tolias, and O. Chum. CNN image retrieval learns from BoW: Unsupervised fine-tuning with hard examples. In *ECCV*, 2016.

[154] Rahul Raguram, Ondrej Chum, Marc Pollefeys, Jiri Matas, and Jan-Michael Frahm. Usac: A universal framework for random sample consensus. *IEEE Trans. Pattern Anal. Mach. Intell.*, pages 2022–2038, 2013.

[155] Roziana Ramli, Mohd Idris, Khairunnisa Hasikin, Noor A. Karim, Ainuddin Wahid, Ismail Ahmedy, Fatimah Ahmedy, Nahrizul Adib Kadri, and Hamzah Arof. Feature-based retinal image registration using d-saddle feature. *Journal of Healthcare Engineering*, 2017:1–15, 10 2017.

[156] Roziana Ramli, Mohd Yamani Idna Idris, Khairunnisa Hasikin, A Karim, Noor Khairiah, Ainuddin Wahid Abdul Wahab, Ismail Ahmedy, Fatimah Ahmedy, Nahrizul Adib Kadri, and Hamzah Arof. Feature-based retinal image registration using d-saddle feature. *Journal of healthcare engineering*, 2017, 2017.

[157] Ricardo Garcia Gonzalez. youtube-dl download videos from YouTube (and more sites), 2019. [Online; accessed 16.10.2019].

[158] Paul L. Rosin. Measuring corner properties. *CVIU*, 73(2):291 – 307, 1999.

[159] E. Rosten and T. Drummond. Fusing points and lines for high performance tracking. In *ICCV*, volume 2, pages 1508–1515 Vol. 2, Oct 2005.

[160] Edward Rosten and Tom Drummond. Machine learning for high-speed corner detection. In Aleš Leonardis, Horst Bischof, and Axel Pinz, editors, *ECCV*, pages 430–443, Berlin, Heidelberg, 2006. Springer Berlin Heidelberg.

[161] Edward Rosten and Tom Drummond. Machine learning for high-speed corner detection. In *ECCV*, pages 430–443, Berlin, Heidelberg, 2006. Springer-Verlag.

[162] Edward Rosten, Reid Porter, and Tom Drummond. Faster and better: A machine learning approach to corner detection. *PAMI*, 32:105–119, 2010.

[163] Ethan Rublee, Vincent Rabaud, Kurt Konolige, and Gary Bradski. ORB: An efficient alternative to SIFT or SURF. In *ICCV*, ICCV '11, pages 2564–2571, Washington, DC, USA, 2011. IEEE Computer Society.

[164] Ethan Rublee, Vincent Rabaud, Kurt Konolige, and Gary Bradski. ORB: An efficient alternative to SIFT or SURF. In *ICCV*, 2011.

[165] Torsten Sattler, Will Maddern, Akihiko Torii, Josef Sivic, Tomás Pajdla, Marc Pollefeys, and Masatoshi Okutomi. Benchmarking 6dof urban visual localization in changing conditions. *CoRR*, abs/1707.09092, 2017.

[166] Nikolay Savinov, Akihito Seki, Lubor Ladicky, Torsten Sattler, and Marc Pollefeys. Quad-Networks: Unsupervised learning to rank for interest point detection. *CVPR*, pages 3929–3937, 2017.

[167] Daniel Scharstein and Richard Szeliski. A taxonomy and evaluation of dense two-frame stereo correspondence algorithms. *IJCV*, 47:7–42, 2002.

[168] J. L. Schönberger, F. Radenović, O. Chum, and J. Frahm. From single image query to detailed 3-D reconstruction. In *CVPR*, pages 5126–5134, June 2015.

[169] Robert Sedgewick and Kevin Wayne. *Algorithms (Fourth edition deluxe)*. Addison-Wesley, 2016.

[170] Jianbo Shi and Carlo Tomasi. Good features to track. Technical report, Cornell University, Ithaca, NY, USA, 1993.

[171] Shuning Wang and Xusheng Sun. Generalization of hinging hyperplanes. *IEEE Transactions on Information Theory*, 51(12):4425–4431, Dec 2005.

[172] E. Simo-Serra, E. Trulls, L. Ferraz, I. Kokkinos, P. Fua, and F. Moreno-Noguer. Discriminative learning of deep convolutional feature point descriptors. In *ICCV*, pages 118–126, Dec 2015.

[173] Karen Simonyan and Andrew Zisserman. Two-stream convolutional networks for action recognition in videos. In Z. Ghahramani, M. Welling, C. Cortes, N. D. Lawrence, and K. Q. Weinberger, editors, *NIPS*, pages 568–576. Curran Associates, Inc., 2014.

[174] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. In *International Conference on Learning Representations*, 2015.

[175] J. Sivic, F. Schaffalitzky, and A. Zisserman. Object level grouping for video shots. In *ECCV*. Springer-Verlag, May 2004.

[176] Alan F. Smeaton, Paul Over, and Aiden R. Doherty. Video shot boundary detection: Seven years of trecvid activity. *CVIU*, 114(4):411 – 418, 2010. Special issue on Image and Video Retrieval Evaluation.

[177] S. M. Smith and J. M. Brady. Susan - a new approach to low level image processing. *IJCV*, 23:45–78, 1995.

[178] Stephen M. Smith and J. Michael Brady. SUSAN: A new approach to low level image processing. *IJCV*, 23(1):45–78, May 1997.

[179] Noah Snavely, Steven M. Seitz, and Richard Szeliski. Photo tourism: Exploring photo collections in 3-D. *ACM Trans. Graph.*, 25(3):835–846, July 2006.

[180] Ilya Sutskever, James Martens, and Geoffrey Hinton. Generating text with recurrent neural networks. In *ICML*, ICML'11, pages 1017–1024, USA, 2011. Omnipress.

[181] Christian Szegedy, Sergey Ioffe, Vincent Vanhoucke, and Alex A. Alemi. Inception-v4, inception-resnet and the impact of residual connections on learning. In *ICLR 2016 Workshop*, 2016.

[182] Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, and Andrew Rabinovich. Going deeper with convolutions. In *CVPR*, 2015.

[183] H. Tao, H. S. Sawhney, and R. Kumar. A global matching framework for stereo computation. In *ICCV*, volume 1, pages 532–539 vol.1, July 2001.

[184] Y. Tian, B. Fan, and F. Wu. L2-Net: Deep learning of discriminative patch descriptor in euclidean space. In *CVPR*, volume 00, pages 6128–6136, July 2017.

[185] Engin Tola, Vincent Lepetit, and Pascal Fua. DAISY: An efficient dense descriptor applied to wide-baseline stereo. *PAMI*, 32(5):815–830, May 2010.

[186] P. H. S. Torr and A. Zisserman. MLESAC: A new robust estimator with application to estimating image geometry. *CVIU*, 78:138–156, 2000.

[187] Miroslav Trajković and Mark Hedley. Fast corner detection. *Image and Vision Computing*, 16(2):75 – 87, 1998.

[188] T. Trzcinski, M. Christoudias, P. Fua, and V. Lepetit. Boosting binary keypoint descriptors. In *CVPR*, pages 2874–2881, June 2013.

[189] T. Trzcinski, M. Christoudias, and V. Lepetit. Learning image descriptors with boosting. *PAMI*, 37(3):597–610, March 2015.

[190] Gül Varol, Ivan Laptev, and Cordelia Schmid. Long-term temporal convolutions for action recognition. *PAMI*, 40(6):1510–1517, 2018.

[191] Yannick Verdie, Kwang Moo Yi, Pascal Fua, and Vincent Lepetit. TILDE: A temporally invariant learned detector. In *CVPR*, pages 5279–5288. IEEE Computer Society, 2015.

[192] Oriol Vinyals, Suman V. Ravuri, and Daniel Povey. Revisiting recurrent neural networks for robust ASR. *ICASSP*, pages 4085–4088, 2012.

[193] P. Viola and M. Jones. Rapid object detection using a boosted cascade of simple features. In *CVPR*, volume 1, pages I–I, Dec 2001.

[194] Jure Žbontar and Yann LeCun. Stereo matching by training a convolutional neural network to compare image patches. *JMLR*, 17(65):1–32, 2016.

[195] Christopher J. C. H. Watkins and Peter Dayan. Q-learning. *Machine Learning*, 8(3):279–292, May 1992.

[196] Tobias Weyand and Bastian Leibe. Discovering favorite views of popular places with iconoid shift. In Dimitris N. Metaxas, Long Quan, Alberto Sanfeliu, and Luc J. Van Gool, editors, *ICCV*, pages 1132–1139. IEEE, 2011.

[197] Wikipedia contributors. History of Wikipedia. `https://en.wikipedia.org/wiki/History_of_Wikipedia`, 2019. Online, accessed: 11.02.2019.

[198] Wikipedia, the free encyclopedia. Video compression picture types. `https://en.wikipedia.org/wiki/Video_compression_picture_types`, 2019. Online, accessed: 06.09.2019.

[199] Brian Williams, Mark Cummins, José Neira, Paul Newman, Ian Reid, and Juan Tardós. A comparison of loop closing techniques in monocular slam. *Robotics and Autonomous Systems*, 57(12):1188 – 1197, 2009. Inside Data Association.

[200] S. Winder, G. Hua, and M. Brown. Picking the best DAISY. In *CVPR*, pages 178–185, June 2009.

[201] C. Wu. Towards linear-time incremental structure from motion. In *3DV*, pages 127–134, June 2013.

[202] Xufeng Han, T. Leung, Y. Jia, R. Sukthankar, and A. C. Berg. MatchNet: Unifying feature and metric learning for patch-based matching. In *CVPR*, pages 3279–3286, June 2015.

[203] Yan Ke and R. Sukthankar. PCA-SIFT: a more distinctive representation for local image descriptors. In *CVPR*, volume 2, pages II–II, June 2004.

[204] Q. Yang, L. Wang, R. Yang, H. Stewénius, and D. Nistér. Stereo matching with color-weighted correlation, hierarchical belief propagation, and occlusion handling. *PAMI*, 31(3):492–504, March 2009.

[205] Qingxiong Yang, Xin Chen, and Gang Wang. Web 2.0 dictionary. In *CIVR*, CIVR '08, pages 591–600, New York, NY, USA, 2008. ACM.

[206] Kwang Moo Yi, Eduard Trulls, Vincent Lepetit, and Pascal Fua. LIFT: Learned Invariant Feature Transform. In *ECCV*, 2016.

[207] Kwang Moo Yi, Eduard Trulls, Yuki Ono, Vincent Lepetit, Mathieu Salzmann, and Pascal Fua. Learning to find good correspondences. *CVPR*, pages 2666–2674, 2018.

[208] Kwang Moo Yi, Yannick Verdie, Pascal Fua, and Vincent Lepetit. Learning to assign orientations to feature points. *CVPR*, pages 107–116, 2016.

[209] YouTube, copyright. Youtube press statistics. `https://www.youtube.com/yt/press/statistics.html`. Online accessed: 15.08.2014.

[210] Guoshen Yu and Jean-Michel Morel. ASIFT: An algorithm for fully affine invariant comparison. *Image Processing On Line*, 1:11–38, 2011.

[211] S. Zagoruyko and N. Komodakis. Learning to compare image patches via convolutional neural networks. In *CVPR*, pages 4353–4361, June 2015.

[212] Wojciech Zaremba and Ilya Sutskever. Learning to execute. *CoRR*, abs/1410.4615, 2014.

[213] Zeeshan Zia, Michael Stark, Bernt Schiele, and Konrad Schindler. Detailed 3-D representations for object recognition and modeling. *PAMI*, 35(11):2608 – 2623, 2013.

[214] Maciej Zieba, Piotr Semberecki, Tarek El-Gaaly, and Tomasz Trzcinski. BinGAN: Learning compact binary descriptors with a regularized GAN. In *NIPS*, 2018.

[215] C. Lawrence Zitnick and Krishnan Ramnath. Edge foci interest points. In *ICCV*, 2011.