Local Invariant Feature Detectors: A Survey
Local Invariant Feature Detectors: A Survey

Tinne Tuytelaars
Department of Electrical Engineering
Katholieke Universiteit Leuven
B-3001 Leuven, Belgium
Tinne.Tuytelaars@esat.kuleuven.be

Krystian Mikolajczyk
School of Electronics and Physical Sciences
University of Surrey
Guildford, UK
K.Mikolajczyk@surrey.ac.uk

now
the essence of knowledge
Boston – Delft
Editorial Scope

Foundations and Trends® in Computer Graphics and Vision will publish survey and tutorial articles in the following topics:

- Rendering: Lighting models; Forward rendering; Inverse rendering; Image-based rendering; Non-photorealistic rendering; Graphics hardware; Visibility computation
- Shape: Surface reconstruction; Range imaging; Geometric modelling; Parameterization;
- Mesh simplification
- Animation: Motion capture and processing; Physics-based modelling; Character animation
- Sensors and sensing
- Image restoration and enhancement
- Segmentation and grouping
- Feature detection and selection
- Color processing
- Texture analysis and synthesis
- Illumination and reflectance modeling
- Shape Representation
- Tracking
- Calibration
- Structure from motion
- Motion estimation and registration
- Stereo matching and reconstruction
- 3D reconstruction and image-based modeling
- Learning and statistical methods
- Appearance-based matching
- Object and scene recognition
- Face detection and recognition
- Activity and gesture recognition
- Image and Video Retrieval
- Video analysis and event recognition
- Medical Image Analysis
- Robot Localization and Navigation

Information for Librarians
Foundations and Trends® in Computer Graphics and Vision, 2007, Volume 3, 4 issues. ISSN paper version 1572-2740. ISSN online version 1572-2759. Also available as a combined paper and online subscription.
Local Invariant Feature Detectors: A Survey

Tinne Tuytelaars\textsuperscript{1} and Krystian Mikolajczyk\textsuperscript{2}

\textsuperscript{1} Department of Electrical Engineering, Katholieke Universiteit Leuven, Kasteelpark Arenberg 10, B-3001 Leuven, Belgium, Tinne.Tuytelaars@esat.kuleuven.be

\textsuperscript{2} School of Electronics and Physical Sciences, University of Surrey, Guildford, Surrey, GU2 7XH, UK, K.Mikolajczyk@surrey.ac.uk

Abstract

In this survey, we give an overview of invariant interest point detectors, how they evolved over time, how they work, and what their respective strengths and weaknesses are. We begin with defining the properties of the ideal local feature detector. This is followed by an overview of the literature over the past four decades organized in different categories of feature extraction methods. We then provide a more detailed analysis of a selection of methods which had a particularly significant impact on the research field. We conclude with a summary and promising future research directions.
Preface: The Local Features Paradigm

Interest points have become increasingly popular over the last few years. Today, they are the preferred strategy for solving a wide variety of problems, from wide baseline matching and the recognition of specific objects to the recognition of object classes. Additionally, similar ideas have been applied to texture recognition, scene classification, robot navigation, visual data mining, and symmetry detection, to name just a few application domains.

Yet, in spite of their recent success and gain in popularity, local features can hardly be called novel. In fact, they have been around since the late 1970s — even though different terminology was used at the time and the level of invariance was less than what we typically work with today. Indeed, the term “interest points” has been introduced by Moravec back in 1979 [155], and there exists a long tradition in corner, blob, and edgel detectors — all of which fall under the general category of “local features.”

Interest points were popular in the past mainly because of their efficiency, information content, and invariance. However, the recent upraise of local feature based approaches is not so much due to the locality of the features nor to their increased levels of invariance. We claim it
is rather caused by a shift in paradigm on how to use such features. Previously, the stress was on accurate (even subpixel) localization and search for correspondences, and on the gain in efficiency by considering only a carefully chosen subset of pixels. These arguments still hold today. Yet on top of that, the emphasis moved toward representing the image content in a robust and flexible way, with image understanding the primordial goal.

The introduction of powerful local descriptors by Lowe [126] had a significant impact on the popularity of local features. Interest points combined with local descriptors started to be used as a black box providing reliable and repeatable measurements from images for a wide range of applications. The vision community soon realized the local descriptors computed on the interest points can capture the essence of a scene without the need for a semantic-level segmentation. Separating the different foreground objects from the background is a very hard problem indeed — a problem that probably cannot be solved in a generic way using low-level features only. Assuming you have such a segmentation available prior to the actual image interpretation thus results in a chicken-and-egg problem. However, representing the image as a set of overlapping local regions, this problem can be circumvented, as it yields an implicit segmentation: since the features are local, some of them cover part of the foreground object(s) and can be considered as relevant, while others fall on the background or on object boundaries and can be considered as irrelevant. It is the task of the subsequent higher-level processing steps to filter out the relevant information or at least to be robust to the (sometimes high) percentage of outliers. This new way of looking at local features has opened up a whole new range of applications and has brought us a step closer to cognitive-level image understanding.

This survey focuses on the feature detectors only, with the emphasis on local features well suited for image understanding applications. Local descriptors will be discussed in another survey.
## Contents

1 Introduction 1

1.1 What are Local Features? 1
1.2 Why Local Features? 2
1.3 A Few Notes on Terminology 3
1.4 Properties of the Ideal Local Feature 5
1.5 Global versus Local Features 8
1.6 Overview of this Survey 11

2 Local Features in the Literature 13

2.1 Introduction 13
2.2 Contour Curvature Based Methods 14
2.3 Intensity Based Methods 18
2.4 Biologically Plausible Methods 23
2.5 Color-based Methods 27
2.6 Model-based Methods 27
2.7 Toward Viewpoint Invariant Methods 29
2.8 Segmentation-based Methods 34
2.9 Machine Learning-based Methods 35
2.10 Evaluations 37

3 Corner Detectors 41

3.1 Introduction 41
3.2 Harris Detector 42
## Contents

3.3 SUSAN Detector 45  
3.4 Harris-Laplace/Affine 47  
3.5 Edge-based Regions 51  
3.6 Discussion 54  

4 Blob Detectors 57  
4.1 Hessian Detector 57  
4.2 Hessian-Laplace/Affine 58  
4.3 Salient Regions 61  
4.4 Discussion 62  

5 Region Detectors 65  
5.1 Intensity-based Regions 65  
5.2 Maximally Stable Extremal Regions 66  
5.3 Segmentation-based Methods (Superpixels) 69  
5.4 Discussion 71  

6 Efficient Implementations 73  
6.1 Difference-of-Gaussians 73  
6.2 SURF: Speeded Up Robust Features 75  
6.3 FAST: Features from Accelerated Segment Test 77  
6.4 Discussion 79  

7 Discussion and Conclusion 81  
7.1 How to Select Your Feature Detector? 81  
7.2 Summary on the Detectors 83  
7.3 Future Work 86  
7.4 Conclusions 88  

Acknowledgments 91  

References 93
In this section, we discuss the very nature of local (invariant) features. What do we mean with this term? What is the advantage of using local features? What can we do with them? What would the ideal local feature look like? These are some of the questions we attempt to answer.

1.1 What are Local Features?

A local feature is an image pattern which differs from its immediate neighborhood. It is usually associated with a change of an image property or several properties simultaneously, although it is not necessarily localized exactly on this change. The image properties commonly considered are intensity, color, and texture. Figure 1.1 shows some examples of local features in a contour image (left) as well as in a grayvalue image (right). Local features can be points, but also edgels or small image patches. Typically, some measurements are taken from a region centered on a local feature and converted into descriptors. The descriptors can then be used for various applications.
1.2 Why Local Features?

As discussed shortly in the preface, local (invariant) features are a powerful tool, that has been applied successfully in a wide range of systems and applications.

In the following, we distinguish three broad categories of feature detectors based on their possible usage. It is not exhaustive or the only way of categorizing the detectors but it emphasizes different properties required by the usage scenarios. *First*, one might be interested in a specific type of local features, as they may have a specific semantic interpretation in the limited context of a certain application. For instance, edges detected in aerial images often correspond to roads; blob detection can be used to identify impurities in some inspection task; etc. These were the first applications for which local feature detectors have been proposed. *Second*, one might be interested in local features since they provide a limited set of well localized and individually identifiable anchor points. What the features actually represent is not really relevant, as long as their location can be determined accurately and in a stable manner over time. This is for instance the situation in most matching or tracking applications, and especially for camera calibration or 3D reconstruction. Other application domains include pose
estimation, image alignment or mosaicing. A typical example here are the features used in the KLT tracker \cite{228}. \textit{Finally}, a set of local features can be used as a robust image representation, that allows to recognize objects or scenes without the need for segmentation. Here again, it does not really matter what the features actually represent. They do not even have to be localized precisely, since the goal is not to match them on an individual basis, but rather to analyze their statistics. This way of exploiting local features was first reported in the seminal work of \cite{213} and \cite{210} and soon became very popular, especially in the context of object recognition (both for specific objects as well as for category-level recognition). Other application domains include scene classification, texture analysis, image retrieval, and video mining.

Clearly, each of the above three categories imposes its own constraints, and a good feature for one application may be useless in the context of a different problem. These categories can be considered when searching for suitable feature detectors for an application at hand. In this survey, we mainly focus on the second and especially the third application scenario.

Finally, it is worth noting that the importance of local features has also been demonstrated in the context of object recognition by the human visual system \cite{20}. More precisely, experiments have shown that removing the corners from images impedes human recognition, while removing most of the straight edge information does not. This is illustrated in Figure 1.1.

1.3 A Few Notes on Terminology

Before we discuss feature detectors in more detail, let us explain some terminology commonly used in the literature.

1.3.1 Detector or Extractor?

Traditionally, the term \textit{detector} has been used to refer to the tool that extracts the features from the image, e.g., a corner, blob or edge detector. However, this only makes sense if it is \textit{a priori} clear what the corners, blobs or edges in the image are, so one can speak of “false detections” or “missed detections.” This only holds in the first usage.
scenario mentioned earlier, not for the last two, where extractor would probably be semantically more correct. Still, the term detector is widely used. We therefore also stick to this terminology.

1.3.2 Invariant or Covariant?

A similar discussion holds for the use of “invariant” or “covariant.” A function is invariant under a certain family of transformations if its value does not change when a transformation from this family is applied to its argument. A function is covariant when it commutes with the transformation, i.e., applying the transformation to the argument of the function has the same effect as applying the transformation to the output of the function. A few examples may help to explain the difference. The area of a 2D surface is invariant under 2D rotations, since rotating a 2D surface does not make it any smaller or bigger. But the orientation of the major axis of inertia of the surface is covariant under the same family of transformations, since rotating a 2D surface will affect the orientation of its major axis in exactly the same way. Based on these definitions, it is clear that the so-called local scale and/or affine invariant features are in fact only covariant. The descriptors derived from them, on the other hand, are usually invariant, due to a normalization step. Since the term local invariant feature is so widely used, we nevertheless use “invariant” in this survey.

1.3.3 Rotation Invariant or Isotropic?

A function is isotropic at a particular point if it behaves the same in all directions. This is a term that applies to, e.g., textures, and should not be confused with rotational invariance.

1.3.4 Interest Point, Region or Local Feature?

In a way, the ideal local feature would be a point as defined in geometry: having a location in space but no spatial extent. In practice however, images are discrete with the smallest spatial unit being a pixel and discretization effects playing an important role. To localize features in images, a local neighborhood of pixels needs to be analyzed, giving
1.4 Properties of the Ideal Local Feature

all local features some implicit spatial extent. For some applications (e.g., camera calibration or 3D reconstruction) this spatial extent is completely ignored in further processing, and only the location derived from the feature extraction process is used (with the location sometimes determined up to sub-pixel accuracy). In those cases, one typically uses the term *interest point*.

However, in most applications those features also need to be described, such that they can be identified and matched, and this again calls for a local neighborhood of pixels. Often, this neighborhood is taken equal to the neighborhood used to localize the feature, but this need not be the case. In this context, one typically uses the term *region* instead of interest point. However, beware: when a local neighborhood of pixels is used to describe an interest point, the feature extraction process has to determine not only the location of the interest point, but also the size and possibly the shape of this local neighborhood. Especially in case of geometric deformations, this significantly complicates the process, as the size and shape have to be determined in an invariant (covariant) way.

In this survey, we prefer the use of the term *local feature*, which can be either points, regions or even edge segments.

1.4 Properties of the Ideal Local Feature

Local features typically have a spatial extent, i.e., the local neighborhood of pixels mentioned above. In contrast to classical segmentation, this can be any subset of an image. The region boundaries do not have to correspond to changes in image appearance such as color or texture. Also, multiple regions may overlap, and “uninteresting” parts of the image such as homogeneous areas can remain uncovered.

Ideally, one would like such local features to correspond to semantically meaningful object parts. In practice, however, this is unfeasible, as this would require high-level interpretation of the scene content, which is not available at this early stage. Instead, detectors select local features directly based on the underlying intensity patterns.
Good features should have the following properties:

- **Repeatability**: Given two images of the same object or scene, taken under different viewing conditions, a high percentage of the features detected on the scene part visible in both images should be found in both images.

- **Distinctiveness/informativeness**: The intensity patterns underlying the detected features should show a lot of variation, such that features can be distinguished and matched.

- **Locality**: The features should be local, so as to reduce the probability of occlusion and to allow simple model approximations of the geometric and photometric deformations between two images taken under different viewing conditions (e.g., based on a local planarity assumption).

- **Quantity**: The number of detected features should be sufficiently large, such that a reasonable number of features are detected even on small objects. However, the optimal number of features depends on the application. Ideally, the number of detected features should be adaptable over a large range by a simple and intuitive threshold. The density of features should reflect the information content of the image to provide a compact image representation.

- **Accuracy**: The detected features should be accurately localized, both in image location, as with respect to scale and possibly shape.

- **Efficiency**: Preferably, the detection of features in a new image should allow for time-critical applications.

Repeatability, arguably the most important property of all, can be achieved in two different ways: either by invariance or by robustness.

- **Invariance**: When large deformations are to be expected, the preferred approach is to model these mathematically if possible, and then develop methods for feature detection that are unaffected by these mathematical transformations.

- **Robustness**: In case of relatively small deformations, it often suffices to make feature detection methods less sensitive to
such deformations, i.e., the accuracy of the detection may decrease, but not drastically so. Typical deformations that are tackled using robustness are image noise, discretization effects, compression artifacts, blur, etc. Also geometric and photometric deviations from the mathematical model used to obtain invariance are often overcome by including more robustness.

1.4.1 Discussion

Clearly, the importance of these different properties depends on the actual application and settings, and compromises need to be made.

Repeatability is required in all application scenarios and it directly depends on the other properties like invariance, robustness, quantity etc. Depending on the application increasing or decreasing them may result in higher repeatability.

Distinctiveness and locality are competing properties and cannot be fulfilled simultaneously: the more local a feature, the less information is available in the underlying intensity pattern and the harder it becomes to match it correctly, especially in database applications where there are many candidate features to match to. On the other hand, in case of planar objects and/or purely rotating cameras (e.g., in image mosaicing applications), images are related by a global homography, and there are no problems with occlusions or depth discontinuities. Under these conditions, the size of the local features can be increased without problems, resulting in a higher distinctiveness.

Similarly, an increased level of invariance typically leads to a reduced distinctiveness, as some of the image measurements are used to lift the degrees of freedom of the transformation. A similar rule holds for robustness versus distinctiveness, as typically some information is disregarded (considered as noise) in order to achieve robustness. As a result, it is important to have a clear idea on the required level of invariance or robustness for a given application. It is hard to achieve high invariance and robustness at the same time and invariance, which is not adapted to the application, may have a negative impact on the results.
Introduction

Accuracy is especially important in wide baseline matching, registration, and structure from motion applications, where precise correspondences are needed to, e.g., estimate the epipolar geometry or to calibrate the camera setup.

Quantity is particularly useful in some class-level object or scene recognition methods, where it is vital to densely cover the object of interest. On the other hand, a high number of features has in most cases a negative impact on the computation time and it should be kept within limits. Also robustness is essential for object class recognition, as it is impossible to model the intra-class variations mathematically, so full invariance is impossible. For these applications, an accurate localization is less important. The effect of inaccurate localization of a feature detector can be countered, up to some point, by having an extra robust descriptor, which yields a feature vector that is not affected by small localization errors.

1.5 Global versus Local Features

Local invariant features not only allow to find correspondences in spite of large changes in viewing conditions, occlusions, and image clutter (wide baseline matching), but also yield an interesting description of the image content for image retrieval and object or scene recognition tasks (both for specific objects as well as categories). To put this into context, we briefly summarize some alternative strategies to compute image representations including global features, image segments, and exhaustive and random sampling of features.

1.5.1 Global Features

In the field of image retrieval, many global features have been proposed to describe the image content, with color histograms and variations thereof as a typical example [237]. This approach works surprisingly well, at least for images with distinctive colors, as long as it is the overall composition of the image as a whole that the user is interested in, rather than the foreground object. Indeed, global features cannot distinguish foreground from background, and mix information from both parts together.
Global features have also been used for object recognition, resulting in the first appearance-based approaches to tackle this challenging problem. Turk and Pentland [245] and later Murase and Nayar [160] proposed to compute a principal component analysis of a set of model images and to use the projections onto the first few principal components as descriptors. Compared to the purely geometry-based approaches tried before, the results of the novel appearance-based approach were striking. A whole new range of natural objects could suddenly be recognized. However, being based on a global description, image clutter and occlusions again form a major problem, limiting the usefulness of the system to cases with clean backgrounds or where the object can be segmented out, e.g., relying on motion information.

1.5.2 Image Segments

An approach to overcome the limitations of the global features is to segment the image in a limited number of regions or segments, with each such region corresponding to a single object or part thereof. The best known example of this approach is the blobworld system, proposed in [31], which segments the image based on color and texture, then searches a database for images with similar “image blobs.” An example based on texture segmentation is the wide baseline matching work described in [208].

However, this raises a chicken-and-egg problem as image segmentation is a very challenging task in itself, which in general requires a high-level understanding of the image content. For generic objects, color and texture cues are insufficient to obtain meaningful segmentations.

1.5.3 Sampled Features

A way to deal with the problems encountered with global features or image segmentations, is to exhaustively sample different subparts of the image at each location and scale. For each such image subpart, global features can then be computed. This approach is also referred to as a sliding window based approach. It has been especially popular in the context of face detection, but has also been applied for the
recognition of specific objects or particular object classes such as pedestrians or cars.

By focusing on subparts of the image, these methods are able to find similarities between the queries and the models in spite of changing backgrounds, and even if the object covers only a small percentage of the total image area. On the downside, they still do not manage to cope with partial occlusions, and the allowed shape variability is smaller than what is feasible with a local features based approach. However, by far the biggest drawback is the inefficiency of this approach. Each and every subpart of the image must be analyzed, resulting in thousands or even millions of features per image. This requires extremely efficient methods which significantly limits the scope of possible applications.

To overcome the complexity problems more sparse fixed grid sampling of image patches was used (e.g., [30, 62, 246, 257]). It is however difficult to achieve invariance to geometric deformations for such features. The approach can tolerate some deformations due to dense sampling over possible locations, scales, poses etc. 00, but the individual features are not invariant. An example of such approach are multi-scale interest points. As a result, they cannot be used when the goal is to find precise correspondences between images. However, for some applications such as scene classification or texture recognition, they may well be sufficient. In [62], better results are reported with a fixed grid of patches than with patches centered on interest points, in the context of scene classification work. This can be explained by the dense coverage, as well as the fact that homogeneous areas (e.g., sky) are also taken into account in the fixed grid approach which makes the representation more complete. This dense coverage is also exploited in [66], where a fixed grid of patches was used on top of a set of local invariant features in the context of specific object recognition, where the latter supply an initial set of correspondences, which then guide the construction of correspondences for the former.

In a similar vein, rather than using a fixed grid of patches, a random sampling of image patches can also be used (e.g., [97, 132, 169]). This gives a larger flexibility in the number of patches, the range of scales or shapes, and their spatial distribution. Good scene recognition results are shown in [132] based on random image patches. As in the case of
fixed grid sampling, this can be explained by the dense coverage which ignores the localization properties of features. Random patches are in fact a subset of the dense patches, and are used mostly to reduce the complexity. Their repeatability is poor hence they work better as an addition to the regular features rather than as a stand alone method.

Finally, to overcome the complexity problems while still providing a large number of features with better than random localization [140, 146] proposed to sample features uniformly from edges. This proved useful for dealing with wiry objects well represented by edges and curves.

1.6 Overview of this Survey

This survey article consists of two parts. First, in Section 2, we review local invariant feature detectors in the literature, from the early days in computer vision up to the most recent evolutions. Next, we describe a few selected, representative methods in more detail. We have structured the methods in a relatively intuitive manner, based on the type of feature extracted in the image. Doing so, we distinguish between corner detectors (Section 3), blob detectors (Section 4), and region detectors (Section 5). Additionally, we added a section on various detectors that have been designed in a computationally efficient manner (Section 6). With this structure, we hope the reader can easily find the type of detector most useful for his/her application. We conclude the survey with a qualitative comparison of the different methods and a discussion of future work (Section 7).

To the novice reader, who is not very familiar with local invariant feature detectors yet, we advice to skip Section 2 at first. This section has been added mainly for the more advanced reader, to give further insight in how this field evolved and what were the most important trends and to add pointers to earlier work.
References


References


References


References


References


Full text available at: http://dx.doi.org/10.1561/0600000017
References


References


Full text available at: http://dx.doi.org/10.1561/0600000017
References


References

References


References

References


References


References


References
