Object Recognition Using Invariant Pixel-Set

Diploma Thesis

Proposed by

Lars Kietzer

Supervisor: Dr. habil. Josef Pauli

Christian-Albrechts-Universität zu Kiel
Institut für Informatik und Praktische Mathematik
Preusserstr. 1-9   24105 Kiel   Germany

February 20, 2002

Abstract

This work presents an object recognition method, constructed and inspired from existing approaches. A probabilistic model for recognition has been developed based on local descriptors and their spatial constellations. Because of their saliency and repeatability, image corners are chosen as very suitable features. Corners have a high information content, and the local characteristics of their nearest surroundings are described by differential grey-value invariants. The computed descriptors are invariant against Cartesian coordinate transformation as well as change of resolution via scale-space extension. Furthermore, the intensity profiles between adjacent corners are estimated as semi-local surface description of the objects. Considering the intensity profiles and the spatial coherence of the corners insures the correct model-matching on a more global level and enables an additional pose estimation of the object. Different invariants and constraints are tested and compared. The method enables a good adaptation of the model building process due to the task of recognition.

Contents

Chapter 1

Introduction

1.1 Motivation

Industrial automation has reached a high level during the last century. Computers are now used to control entire production processes. During such a process, a variety of different sensor types supply the control unit continuously with input data about every measurable value of the product. The control unit on the other hand has a protocol of routines to follow as reaction to deviation in the measures. One of the most sophisticated and versatile sensors is the digital camera. Human process and product control is highly based on the visual sense. Errors in production, the product class or other features are mostly recognized visually. The counterpart in a computer control system is computer vision. Its task is the realization of automatic image analysis and scene interpretation to substitute the human supervision. The range of application is widely spread: from medical image analysis, over object recognition and pose estimation on conveyer belts to the visual navigation of mobile robots. Every case is based on image processing, the extraction and recognition of known features in the image. The results lead to further conclusion of the image contents. A robot, for example, can be used to estimate a map of the environment it is navigating through. Or the movements of a robot arm to grab an object are controlled via visual pose-estimation. This can be used in assembly lines or sorting facilities.

This thesis is addressed to the task of object recognition and is the approach of an effective combination of different existing techniques.

1.2 Object Recognition

Object recognition in computer vision has produced a large variation of different approaches and techniques. Recognition is generally based on the comparison of object images as input data with a database of stored object models from known objects.

The methods are different in the kind of input data and the a priori knowledge about it, different in the model building process [1], the feature extraction and comparison and the final estimation of the corresponding model. A general solution for every recognition task has not been developed yet.

Recognition is especially difficult if the objects have a complex 3D figure. Imaging the object is a 3D-to-2D-projection. Information is lost by this transformation and the appearance variation of the same object is possibly large.

Parallels to Human Vision

To recognize an object the computer must have a database containing some kind of model of each object. A model could for example be a virtual object replication. Virtual means, that it is computed in a 3D-graphic design program, a *CAD*-program, which is the attempt to simulate every natural way of appearance of the object. This approach would conform our association of an object model. This is, however, not the only possible object representation.

Still, it is impossible to "clone" the human vision system with computers. Our learning and recognition ability is far superior to machines and it is not as simple to copy. It is based on years of visual experience, during which we learn to name and distinguish object classes (like lamps or chairs) and their different ways of appearance in our visual world. Humans are able to classify objects, they have not even seen before, by recognizing their function. The task of simulating the human brain, however, is the main objective of another research area, named artificial intelligence.

As said, the reproduction of the human brain functions is far from being realized. It is especially impossible to achieve and implement the same amount of a priori knowledge in a computer vision system and the precise way of biological recognition technique has not even been examined in detail yet. Since the technical resources are limited, the success of the recognition system is reliant on its efficiency: the extraction and optimization of purely necessary data.

[1] a model is either learned and trained automatically from images or it is precalculated, human designed

However, visual learning is a well-developed component of the human vision system and some of the natural techniques can be assumed by computers. To visually memorize an object, a human being would rotate the object and study it appearance from different directions. Psycholophysical findings of Tarr and Pinker [19] indicate, that objects are represented in the mind by a set of two-dimensional views rather than a single three-dimensional model.

Thus, instead of storing a complex virtual 3D-model of the object, a series of appearance images could be used for comparison.

In contrast to the human vision it is possible to adapt the system build-up to a specific recognition task. This is much easier than building a single computer vision system to solve all kinds of problems. As said, automatic recognition fortunately does not have to depend on the global virtual modeling of an object, but rather on the representation of salient features. The efficiency of the individual system thereby depends on the compression of information to a sufficient amount by eliminating redundancy.

Prerequisites

Basically the following questions should be answered: What can and must be seen in the image and what kind of information is necessary and possible to be extracted?

In particular the automatic model training has attracted the researchers attention. By that, the time-consuming designing of various complex models can be left completely to the machines.

The difficulties and inadequacies in computer vision, however, appear on a much more primitive level: Because of pixelization and quantization effects it is already difficult to estimate whether or not a pixel is part of the object or its environment. Furthermore the same object may appear in very different ways because of illumination changes, which again cause reflection and shadows on the surface.

Without reference points or exact stereo pictures it is almost impossible, to estimate the specific shape of an object from its 2D-images, let alone automatically calculating an exact CAD-model. Instead it is practical to draw the attention towards the reduction of extracted information to salient features, which are sufficient to recognize the object correctly.

1.3 Existing Recognition Methods

There are several ways of recognition, varying in the method of object modeling, feature comparison and estimating technique that finally leads to the correct answer.

Some existing approaches use geometric object features, others rely on the luminance signature only.

Geometric Object Features

The former try to extract 3D properties such as corners, lines or surface blobs to create a graph-representation of the object structure or they use existing 3D-shape models to match their features to the object images.

This requires *image understanding*. Image structures have to be recognized and classified as image primitives. Then their relationships has to be determined. Graph methods have been developed to compare model structures with image structures. Pauli [10] has, for example, developed a vision system to verify solder joints on integrated circuits. A solder joint is described by the wire, the contact point and the end of the wire, sticking out at the other side of the solder joint. These three primitives are related to each other by additional features. The extracted image structures are converted to a comparable feature representation. The relationships of the primitives are represented in a graph. This graph structure is now comparable to the model structure. If the solder point is broken, the relationship of the primitives deviates from the model structure.

The drawback of these methods is the limitation on rather simple geometric objects and the sensitiveness towards local errors. The objects can only be compared as whole constructions, since the features are globally related. That means, if some features are miss-detected, the complete structure is not comparable anymore.

In the case of more complex, natural objects the data size and need of calculation power grows rather huge. Especially, if the system should be able to take account of the complex interaction of object shape, surface reflectance, illumination and observers viewpoint. As said above, this is a costly and unnecessary information overflow. Beyond that, the more features have to be compared and differentiated, the more errors may have influence on the final result.

Appearance Features

Other approaches are appearance based. They do not depend on what has to be seen in the image, they rather use what really *is* seen. The pixel values inside the object are only used to built an appearance profile, no features of the shape are extracted. Primary experiments tried to reduce the collection of images showing every possible appearance of the same object to an illumination invariant intensity histogram [17]. But this was inadequate in the case of many objects and a large variety in their appearance. As mentioned above, the appearance of an object depends on its shape, pose in the scene, surface reflectance and the illumination conditions. Pose and illumination vary from scene to scene, while the shape and surface consistence are intrinsic properties and constant for a rigid object. This means, that a compact object model, parameterized by pose and illumination should be a sufficient representation.

The realization of this idea is the transformation of the object images into the *eigenspace*. Turk and Pentland have been some of the pioneers in this matter. In their work [21] from 1991 they introduced a face recognition system based on the eigenspace. They projected face images onto a low-dimensional feature space that spans the significant variations among known face images.

The eigenspace has a very high dimensionality, exactly the number of pixels in the image. Each image is transformed into a vector, describing a single point in this space, depending on the intensity values of each pixel. The eigenvectors of the covariance matrix of this vector set now characterize the variation between face images. Still each individual face can be exactly represented in terms of a linear combination of the complete set of eigenvectors. Reducing this set to the eigenvectors, that have the m highest eigenvalues (corresponding to most variance within the set), leads to an efficient compression of the eigenspace into the "face-space". Still, an approximated representation of every individual face is preserved, since features of minor variance do not contain valuable information. Recognition is now established by projecting a query image into this face-space and classifying the face by comparing its position in face-space with the database of known faces.

Murase and Nayar transferred this approach into general object recognition [9]. It was one of the first techniques for automatically learning comparable models of three-dimensional objects from their appearance in two-dimensional images. They took pictures from the object from several orientations and illumination directions. The obtained image set is normalized in brightness and scale to achieve invariance to image magnification and illumination intensity. This image set is transformed into the *parametric* eigenspace to obtain a set of points. Like in face space only a few eigenvectors are generally sufficient to capture the significant appearance characteristics of the object. After the learning phase every object is represented in *object space* by the *manifold* of these characteristics. A query object image is then similarly normalized and transformed into the eigenspace. It is recognized, determining the closest manifold point.

All together, these approaches are holistic. That means, the image is computed as a whole and the features are global. Therefore these methods have difficulties in dealing with partial occlusion or extraneous properties. The objects need to be extracted from any environment to be normalized and must not be occluded at all.

For a more extensive overview of appearance based methods the reader should refer to Nicola Duffy [1].

1.4 This Approach

This study is based on the approaches of Schmid [15] and Matas [8]. Both methods are appearance based and semi-local. Image structures are used as salient features for the object appearance.

Schmid extracts differential invariants at interest points as a set of individual local descriptors. The spatial constellation of the descriptors is restricted by semi-local constraints.

Matas uses normalized intensity profiles between adjacent interest points as surface descriptors. Both methods are based on interest points extracted from the image by a corner detector.

These methods are discussed in detail in the next chapter.

Motivation for this Approach

The idea of this study is to combine both methods to improve their individual results and robustness. The advantage of the interest point descriptors is their individuality. Opposite to a global model comparison it is possible to recognize the object even if it corresponds only partially. The method is robust against single miss-matches, as long as the majority of descriptors correspond to the correct model.

The limitation on local features also prevent an explosive increase of the costs of calculation. Not the entire image is used to built a model. Instead, selected points of interest act as representatives for the entire object. This compressed information represents a handy and sufficient object description.

Because of their saliency, corners are very suitable features. They are easy to recognize in several images and can be located exactly. Corners have a high information content, and the local characteristics of their nearest surroundings can be described by differential grey-value invariants[2], introduced by Florack et al. [3, 7]. It is important to extract features, which are invariant against some of the parameters, which have influence on the object appearance. That way the degree of freedom in the appearance space is reduced and less descriptors are necessary.

---

[2]ensuring invariance under the group of invertible grey-value transformations (see section 2.3)

Beyond that, semi-local constraints like the spatial constellation [15] or intensity profiles between these points [8] further ensure the correct recognition.

The method has been optimized for the test images of the *COIL-20* database [11], a publicly available database of 1440 images. See figure 1.1 for images of object examples.

Figure 1.1: The 20 objects of the *COIL-20* database [11]

Structure and Implementation

During the training phase a database is built up, containing a data record of every model descriptor. Thereto a trainingsset of images is processed. Interest points are detected and local descriptors extracted at their position.

In the recognition phase the image of a query object is processed by the same algorithm to extract the object descriptors. Then they are compared to the model descriptors in the database. To be robust against descriptor variations the recognition task is based on a probability estimation. The descriptor comparison is content with similarity and does not require equality. A voting algorithm counts every similarity. The highest number of votes nominates the corresponding model, so single mismatches should not falsify the result. A sufficient number of correct votes makes sure that the object is recognized correctly. Thus, even if concordant features are missed or unexpected correspondences are detected, the model with the highest correspondence is chosen.

That means robustness against occlusion and the ability to recognize the object even in front of a very noisy background. Thereby it does no matter, whether or not the object can be distinguished from the background.

Unfortunately, objects are difficult to describe equally. Objects with homogeneous surfaces and a rotund shape do not provide many possible interest points for the descriptors, while textured, cornered objects have plenty. To adjust this unequal feature distribution, the simple voting method is replaced by a probabilistic model.

The so far absolute local object description is extended by the consideration of the more global coherence of the local descriptors. Spatial constraints between the interest points, which must be fulfilled by the descriptors of the query image as well as of the corresponding model. This eliminates arbitrary correspondences of the wrong descriptors.

Beyond that, the intensity profiles, suggested by Matas, offer a semi-local characterization of the object between the interest points. The local descriptors can be used as a presorting for the profiles, which again assure the correct correspondences.

The statistical character of the recognition method simultaneously takes into account disturbing effects of noise or other divergence. This approach combines a fast model acquisition with a good error tolerance.

1.5 The Structure of this Thesis

First, both methods are introduced and discussed. Then their implementation is explained. Necessary and useful modifications and extensions are presented. The recognition abilities of the individual techniques as well as of a combined implementation are tested. The results are discussed and an improvement of the individual performances is achieved.

Chapter 2 presents the model acquisition in detail. First of all the corner detection is explained, then the local descriptor extraction and the global coherence of these descriptors are illustrated.

Chapter 3 starts with the set-up of the model database. A fast indexing technique is presented to accelerate the descriptor comparison. The detection of correspondences between particular, similar descriptors of the query object image is exposed, followed by the contemplation of their semi-local coherence to the evaluation of the concordant model.

Experimental results are presented in chapter 4. The efficiency of the different components of the model and the recognition algorithms are examined. Furthermore, the robustness of the recognition, that means, the invariance of the model descriptors against all kinds of modification of the image data, is tested.

Finally the thesis concludes with a summary of the recognition method. Its efficiency and usability is discussed and possible extensions are proposed.

Chapter 2

Model Acquisition

The following object model is a further development of the approach of Schmid [15] and extended by some ideas of Matas [8].

To recognize an object from an arbitrary view point the computer must have some kind of object model. This model must contain information that describe every possible way of appearance of the object. In the same way, the information must be comprehensive enough to distinguish between different objects.

So, the model of one object is achieved from a variety of model images. These model images function as an average sample of every appearance. The extracted features of every model image contributes to the entire object model.

To develop a fast and effective system, thus to keep computing time low, the amount of necessary and comparable information has to be lowest possible. That means that the system must be able to handle the automatic extraction of the image features with the highest information density. The request to the model is that these features are invariant to projective transformation (affine and orthogonal transformation, like translation, rotation, etc.) and are salient enough to distinguish one object from all the others.

According to Schmid [15] such invariant image features of high information content are hidden in corners or interest points. In a grey-value image a corner is a local, single point, at which the signal changes two-dimensionally. These interest points can be located exactly and are easy to match.

A sufficient description of the interest points is needed to distinguish or correspondingly to recognize them in different views. For sure, the grey-value of an interest point or its position is not enough to build an adequate model. Furthermore, the model should not contain any position-bound information, instead it should be content on information that can be extracted from the local neighborhood of the interest points.

Local grey-value invariants, primarily developed by Florack et al. [2, 3, 7], are such suitable descriptors. They are invariant under the group of Cartesian transformation. They generate a compact object model, reduced to salient information, that accomplishes the requirements towards the object recognition system.

To be insensitive to miss matches these local descriptors are affiliated by some semi-local or more global constraints. These constraints consider the spatial constellation of the interest points or the characteristics of the connecting line in between them. The latter constraint is taken from the work of Matas [8].

Section 2.1 discusses the features of interest points or corners. Different corner detectors are presented and their usefulness is estimated. The Harris corner detector points out to be most suitable for this task. The algorithm of the detection is then described in detail in section 2.2, followed by an elaborate presentation of *differential invariants* in section 2.3. Both subjects are discussed explicitly to understand the various tests in chapter 4. In section 2.3.4 a brief discussion of the influence of *scale* on image structures is presented. The calculated invariants are summarized to a specific model descriptor for each interest point in section 2.4. Finally, the chapter concludes with the extension of the model with semi-local coherence in section 2.5.

2.1 Interest Points

To detect the interest points a respective corner detector is used. There exists a wide variety of these detectors. Because good matching property is one of the basic demands, the detector must have a high repeatability rate especially under rotation. And because of the different possible appearances of the model, it has to be robust concerning projective transformation.

In [16] an elaborate examination of interest point or corner detectors is presented.

Corner detectors can be divided into contour based methods, signal based methods and methods based on template fitting.

According to the results of [16], the latter methods are not applicable in a general 3D-context. These detectors try to fit the image signal to a parametric model of a specific type of interest point, but there is no need for a special kind of corner model. So the template fitting methods are useless for this approach.

Contour based methods are either searching for maximal curvature or inflexion points along the contour. These methods are costly and unfortunately the repeatability under rotation is poor because the three steps of contour extraction, line detection and the calculation of the intersection points of the lines introduce cumulative errors.

At least the signal based methods are most suitable for this approach. They are fastest and most robust because they are computing the location of corners directly from the signal. A detailed comparison of different signal based detectors under varying conditions in [16] has finally shown, that the Harris corner detector [4] is the most repeatable and robust detector.

2.2 Harris Corner Detector

First of all, the characterization of a corner, that differs this point from its local neighborhood has to be established. To classify a pixel Harris [4] considers the average change of intensity inside a local window by shifting the window a small amount in various directions from the origin. The possible result of this difference estimation can be divided into three cases:

1. If the window lies on a local flat, then all shifts will result in a small amount only.
2. If the window straddles an edge, a shift along the edge will result in a small change, while a shift perpendicular to the edge will produce a large change.
3. Only if the windowed image patch is a corner or an isolated point, every shift will result in a large change.

Thus, a corner is detected if the minimum change produced by any of the shifts is large.

A mathematical specification of the above is given by:

$$E_{x,y} = \sum_{u,v=-1}^{1} w_{u,v} [I_{x+u,y+v} - I_{u,v}]^2 \tag{2.1}$$

Denoting the image intensity by I, the change E is produced by a shift (x,y) inside the considered window $w_{u,v}$.

To avoid an anisotropic response the method is modified by using a Gaussian circular window instead of a discrete set of shifts.

An image example to visualize the manner of operation of the Harris corner detector is presented at the end of this section.

2.2.1 Gaussian Window

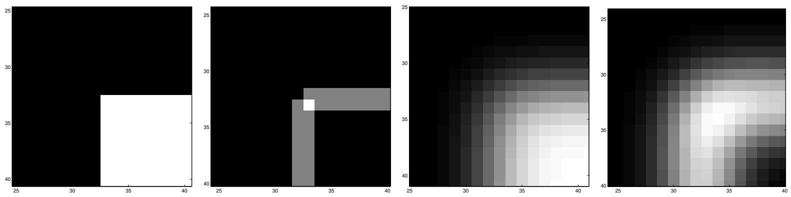
The two-dimensional Gaussian window or *kernel* is given by

$$w_{u,v} = \frac{e^{-(u^2+v^2)}}{2\sigma^2}. \tag{2.2}$$

The circular structure of the two-dimensional Gaussian takes account of the euclidic distance of each window pixel to the center pixel. Compared to the Gaussian the response of a binary and rectangular window is very noisy caused by the distance error between diagonal shifts and shifts along the pixels[1]. Besides, the smoothing characteristics of the Gaussian filter mask is a favorable side effect, since it does not eliminate accurate corners but decreases the appearance of false matches due to quantization errors.

The extent of the blurring effect is controlled by the variance σ of the Gaussian kernel. σ presents simultaneously the size of the Gaussian window, thus its range of effect on the adjacent pixels.

This range affects also the exact localization of the detected corners. This is illustrated in figure 2.1. The corner is defined as the local maximum of the two-dimensional differentiation. Here, its the maximum of the sum of the two orthogonal edge differentiations. This maximum is greatest exactly at the L-junction of two perpendicular lines, if σ is close to zero, whereas it is localized more inward the object if σ is large. The individual gradient value of the edges wanes at the L-junction due to the smoothing. Thus, the maximum of the sum is now located more inward the object away from the uttermost object pixel.



(a) (b) (c) (d)

Figure 2.1: Difference in corner localization caused by blurring: (a) example image, (b) shows the estimated corner point, the maximum value of $E_{x,y}$. (c) is the blurred image of (a). $\max(E_{x,y})$ is now detected more inward the white square in (d).

[1] The length of a diagonal shift is $\sqrt{2}$ compared to a perpendicular shift of length 1 (pixel).

On the other hand, because σ affects the size of the Gaussian kernel, it allows the kernel to fit the resolution of the considered image structure. By that, σ is a measure for the *inner scale* of the image. The inner scale denotes the least extend of interest image structures. *Scale space* is discussed later in detail in 2.3.4.

In the case of corner detection σ is fixed to a sufficient scale for smoothing. It has to be constant to guarantee an uniform localization of the corner detections in the original image.

2.2.2 Gradients

Instead of considering the difference between vicinal pixel weighted by the Gaussian function, it is much more effective to convolve the image with a differential Gaussian kernel:

$$\begin{aligned} X &= I \otimes \frac{\partial w}{\partial x} \\ Y &= I \otimes \frac{\partial w}{\partial y} \end{aligned}$$

X and Y represent the perpendicular gradients, respectively in x- and y-direction.

(2.1) can be reformulated to:

$$E_{x,y} = \sum_{u,v=-1}^{1} w_{u,v}[xX + yY + O(x^2, y^2)]^2 \tag{2.3}$$

But all the shifts $O(x^2, y^2)$ in between the aligned gradients have to be considered, too. Instead, it is easier to use tensor representation (Jähne [6]). Then it is possible to adjust the respective coordinate system to the direction of the gradient. By that, the factor $O(x^2, y^2)$ is negligible.

2.2.3 Tensor Representation

With

$$\begin{aligned} J_{xx} &= X^2 \otimes w \\ J_{yy} &= Y^2 \otimes w \\ J_{xy} &= (XY) \otimes w \end{aligned}$$

Equation (2.3) can easily be transformed into tensor representation:

$$E_{x,y} = J_{xx}x^2 + 2J_{xy}xy + J_{yy}y^2 \tag{2.4}$$

or concisely:

$$Ex, y = (x, y)\boldsymbol{J}(x, y)^T \tag{2.5}$$

where

$$\boldsymbol{J} = \begin{pmatrix} J_{xx} & J_{xy} \\ J_{xy} & J_{yy} \end{pmatrix}. \tag{2.6}$$

The structure tensor is an adequate first-order representation of the local neighborhood. Due to symmetry it can be brought into a diagonal form by rotating the coordinate system. Then, equation (2.6) is reduced to:

$$\begin{pmatrix} J_{x'x'} & 0 \\ 0 & J_{y'y'} \end{pmatrix} = \begin{pmatrix} \lambda_1 & 0 \\ 0 & \lambda_2 \end{pmatrix} \tag{2.7}$$

Eigenvalue decomposition leads to the estimation of $J_{x'x'}$ and $J_{y'y'}$.

2.2.4 Eigenvalues

It is assumed that λ_1 and λ_2 are the eigenvalues of $\boldsymbol{J}$:

$$\boldsymbol{J}\boldsymbol{v} = \lambda\boldsymbol{v} \tag{2.8}$$

A corresponding eigenvector $\boldsymbol{v}$ is not rotated by multiplication with $\boldsymbol{J}$, but only scaled by the eigenvalue. So, the coordinate system of the diagonal structure tensor in (2.7) is spanned by the eigenvectors.

Furthermore the eigenvector to the maximum eigenvalue gives the orientation of the local neighborhood, that means, it is tangent to the gradient vector at the origin.

The eigenvalues can be classified in the same way as the result of the shifted window above:

$\lambda_1 = \lambda_2 = 0$ Both eigenvalues are zero. This means that there is no specific gradient, so the local neighborhood has constant values. It is a homogeneous, intensity-flat region.

$\lambda_1 > 0, \lambda_2 = 0$ One eigenvalue is zero. The intensity values do not change in direction of the corresponding eigenvector. The neighborhood has a ideal orientation orthogonally to that eigenvector, this indicates an edge.

$\lambda_1 > 0, \lambda_2 > 0$ Both eigenvalues are unequal zero, which indicates a gray value change in all directions. Only a corner or single point shows this characteristic.

According to Harris [4] the task of corner detection only depend on the eigenvalues as a gradient indicator and not on the specific gradient orientation; there is no need to calculate the corresponding eigenvector. From (2.8) follows:

$$det(\boldsymbol{J} - \lambda \mathbf{I}) = 0 \tag{2.9}$$

It can easily be solved:

$$det \begin{pmatrix} J_{xx} - \lambda & J_{xy} \\ J_{xy} & J_{yy} - \lambda \end{pmatrix} =$$

$$(J_{xx} - \lambda)(J_{yy} - \lambda) - J_{xy}^2 =$$

$$\lambda^2 - (J_{xx} + J_{yy})\lambda + J_{xx}J_{yy} - J_{xy}^2 = 0$$

using the *pq*-formula:

$$\lambda_{1,2} = \frac{J_{xx} + J_{yy}}{2} \pm \sqrt{\frac{(J_{xx} + J_{yy})^2}{4} - (J_{xx}J_{yy} - J_{xy}^2)} \tag{2.10}$$

But the values $\lambda_{1,2}$ do not have to be evaluated individually.

2.2.5 Corner Response

According to Harris [4] it is possible to combine the eigenvalues into a single corner response function R with a response corresponding to the classification given above:

$$R = \lambda_1 \lambda_2 - k(\lambda_1 + \lambda_2)^2 \tag{2.11}$$

The constant k makes it possible to adjust the response sensibility of R:

$\lambda_1 = \lambda_2 = 0$ Both eigenvalues are zero. This means R is zero $\Rightarrow$ homogeneous region.

$\lambda_1 > 0, \lambda_2 = 0$ One eigenvalue is zero. Now R is less zero, because the first term of 2.11 disappears $\Rightarrow$ edge region.

$\lambda_1 > 0, \lambda_2 > 0$ Both eigenvalues are unequal zero. That means, R is greater zero, because the second term will always be greater than the first one, as long as $\lambda_1 \approx \lambda_2$ and $k < 1/4$. $\Rightarrow$ corner region.

To prove the latter statement (2.11) is transformed to:

$$R = (1 - 2k)\lambda_1\lambda_2 - k(\lambda_1^2 + \lambda_2^2) \tag{2.12}$$

In the case of a singular point or a homogeneous region λ_1 is equal λ_2 and (2.12) simplifies to:

$$R = (1 - 4k) * \lambda^2 \tag{2.13}$$

Now it is obvious, that the constant k is simply an experimentally determined value to adjust the classification of homogeneous regions, edges and corners. Increasing k decreases the corner regions, detected in the image.

An advantage of using R is the avoidance of the explicit eigenvalue decomposition of $\boldsymbol{J}$. Equation (2.10) leads to:

$$\lambda_1 + \lambda_2 = J_{xx} + J_{yy} \tag{2.14}$$

$$\lambda_1 * \lambda_2 = J_{xx}J_{yy} - J_{xy}^2 \tag{2.15}$$

(2.14) and (2.15) correspond to the trace and determinant of $\boldsymbol{J}$, respectively. So, R can be much easier determined than in equation (2.11):

$$R = Det - kTr^2 \tag{2.16}$$

2.2.6 Peak Search

R represents a contour image of I. It indicates homogeneous, edge and corner regions. Because corner regions are characterized by higher values than homogeneous or edge regions, the data can be easily reduced to the corner regions by simply using a threshold.

Finally, to minimize the regions to their isolated peaks, a 3×3 maximum filter is used. This order statistic filter replaces the center pixel with the

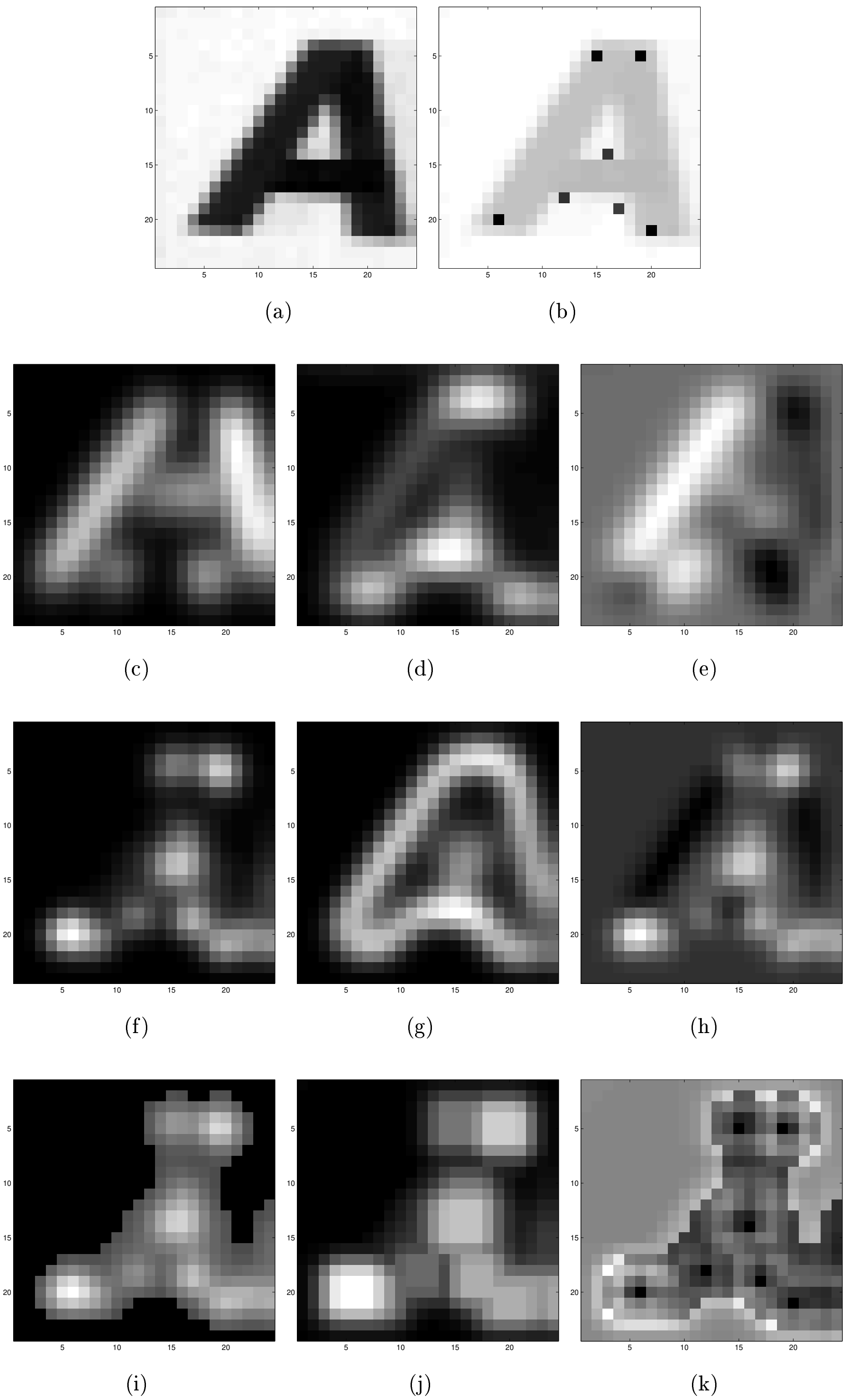


Figure 2.2: A clipping from an image example. (a) is the original image, (b) shows the finally estimated corner points, $(c\text{-}e)$ the Tensor elements J_{xx}, J_{yy} and J_{xy}, (f) the determinant and (g) the trace of $\boldsymbol{J}$, (h) the corner response R, (i) shows R reduced to the corner regions, (j) the maximum order filter response of (i) and (k) the difference of the last two images.

greatest value of all nine window pixels. Because the corner location is exactly the local maximum, the difference between this filtered image and the preprocessed one is only zero at the peak positions. Thus easily to be found by subtracting the two images from each other.

These points are the final result of the Harris corner detector.

Adjustment of the Detection

The quality and quantity of the corner detection depends on the two parameters σ and k. σ allows an optimal adaptation of the detection due to the resolution and quality of the image data. It determines further the exact localization of the corners. k influences the final amount of detections, whether or not an interest point is salient enough. Different values for σ and k are tested in chapter 4.

Interest Point Identification

As said, this detector is rotational invariant and has a high repeatability. So the probability of detecting the same corner in two different images of the same object is very high as long as the corner is not occluded. The problem is rather to identify the same corner in both images, if its location has changed. To match these points of interest it is not enough to compare their intensity values.

Object corners imaged from different points of view propose a comprehensive variety of appearance. The light conditions influence the intensity values. Because of the 3D-shapes of the objects, not only the location and orientation of the corners change, they are also influenced by all kinds of projective transformation. That means the nearest surroundings of a corner may vary, too. Beyond that, scale changes, due to alteration of the distance between object and camera, have to be considered. So, how can interest points be characterized and compared?

It is not possible to achieve one single description, invariant against all kinds of appearance (especially against 3D-transformation of the object). It is, however, possible to reduce the degrees of freedom of appearance. First of all, a local description of a corner is needed, so far possible invariant against planar 2D-transformation.

2.3 Differential Invariants

Local image structure has been carefully studied by Florack and Koenderink, et al. [2, 3, 7]. Using Cartesian tensor calculus and scale-space theory they have developed a coordinate-independent robust formalism to describe local image properties.

Appropriate for the description of the local neighborhood at a single base point (interest point) is a complete and irreducible set of differential invariants. An integration filter with an effective extent proportional to the inner scale of the neighborhood is needed. As said, this inner scale determines the extent of the interest structure. It is a degree of freedom in the image interpretation and depends on the resolution. If one e.g. looks at an image area from a distance, it may look homogeneous, while a closer look reveals a sharp texture. So, depending on the scale a differential operator can interpret this area as homogeneous as well as highly diversified. To be insensitive to scale change a multi-scale representation is proposed. The complete scale-space is generated by the Gaussian kernel. Gaussian derivatives are the perfect filters for this task, because the width σ of the Gaussian corresponds to the inner scale. Furthermore the blurring effect adjusts the pixel values, so no spurious detail is created during the filtering.

The derivatives are the principal elements used to construct invariant local image descriptors. Invariants are image properties which are insensitive to variations of appearance parameters.

The purpose of Florack's method is to extract these properties. The mathematical way is to define a transformation group that effects exactly these parameters. Generally, this requires a priori knowledge about redundancy of image data. In this case, invariance is only required under the group of *Cartesian transformations* (translation and orthogonal transformation, like rotation or reflexion), which preserves the complete local grey-level structure. *Cartesian tensors*, invariant under the Cartesian transformation group, are used to express intrinsic local image properties.

The tensors are built from Gaussian derivatives, describing the interest image structure.

2.3.1 Gaussian Derivatives

With the Gaussian derivatives a complete family of local neighborhood operators can be constructed. This construction of differential derivatives up to any order N is called *local jet* of order N. The common Gaussian appears as the zeroth order member of the entire Gaussian family of scaled differential operators.

A linear derivative is created by formally applying the appropriate conjugate differential operator to the smooth Gaussian kernel. The resulting derivative kernel is then convolved with the image data.

Definition 1 *A complete hierarchically ordered family of Gaussian derivatives up to order N in D dimensions is given by the following set of convolution filters:*

$$\left\{G_{i_1 \ldots i_n}(\mathbf{x};\sigma) = \frac{\partial^n}{\partial x_{i_1} \ldots \partial x_{i_n}} G(\mathbf{x};\sigma) \mid (\mathbf{x};\sigma) \in \mathbb{R}^D \times \mathbb{R}^+, n \in \mathbb{Z}_0^+\right\}, \qquad (2.17)$$

with

$$G(\mathbf{x};\sigma) = \frac{1}{\sqrt{2\pi\sigma^2}^D} \exp\left(-\frac{1}{2}\frac{x^2}{\sigma^2}\right), \qquad (2.18)$$

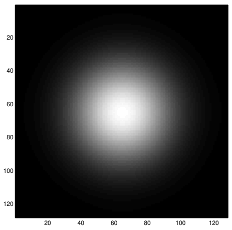
the zeroth order Gaussian. The general notion for spatial coordinates is $\mathbf{x} = (x_1, \ldots, x_D)$

The Gaussian derivatives are symmetric with respect to the interchange of any pair of indices. So the number of filters minimally required (but also sufficient) for a full n-th order description of local image structure on a given level of scale is

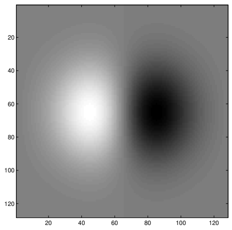
$$m_{n;D} = \begin{pmatrix} n+D-1 \\ D-1 \end{pmatrix} \qquad (2.19)$$

That means $m_n = n + 1$ in the case of two-dimensional images.

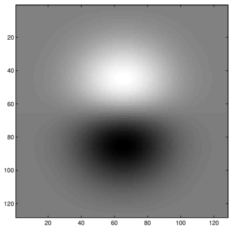
The complete family of Gaussian derivative kernels from zeroth to third order is shown in Figure 2.3. The third order is sufficient for this approach. Higher order kernels do not improve the accuracy of the descriptors, moreover they increase the time of calculation substantial. (The number of derivatives in (2.19) increases about 50%, if the fourth order is considered too).



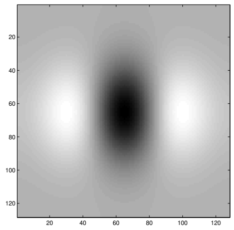
(a) $G(\mathbf{x};\sigma)$



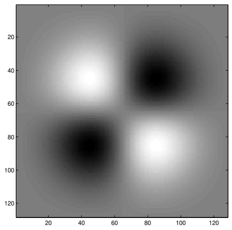
(b) $G_x(\mathbf{x};\sigma)$



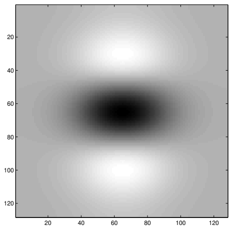
(c) $G_y(\mathbf{x};\sigma)$



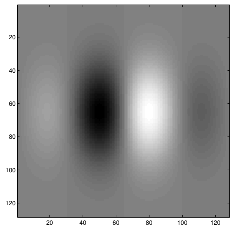
(d) $G_{xx}(\mathbf{x};\sigma)$



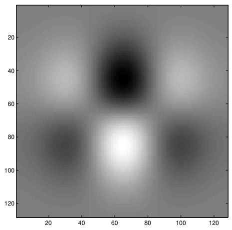
(e) $G_{xy}(\mathbf{x};\sigma)$



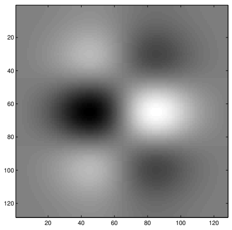
(f) $G_{yy}(\mathbf{x};\sigma)$



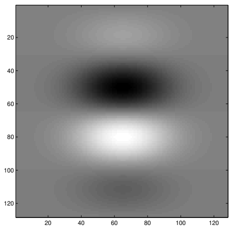
(g) $G_{xxx}(\mathbf{x};\sigma)$



(h) $G_{xxy}(\mathbf{x};\sigma)$



(i) $G_{xyy}(\mathbf{x};\sigma)$



(j) $G_{yyy}(\mathbf{x};\sigma)$

Figure 2.3: The complete family of Gaussian derivative kernels from zeroth (top image) to third order (bottom row).

2.3.2 The Local N-Jet

Now the elementary building blocks for constructing Cartesian invariants are provided by the local N-jets of a given image. A local jet is defined as the equivalence class of functions that have the same local structure up to order N. That means, all images in a given local jet are locally indistinguishable when only the Gaussian derivatives are used for the information extraction.

Definition 2 (Local N-jet) *Let σ be a given scale and $i_k \in \{x_1, \dots, x_D\}$. $L_{i_1 \dots i_n}(\mathbf{x}, \sigma)$ is the convolution of Image $I : \mathbb{R}^D \to \mathbb{R}$ with the Gaussian derivatives $G_{i_1 \dots i_n}(\mathbf{x}, \sigma)$. The local jet of order N at point $\mathbf{x} = \{x_1, \dots, x_D\}$ is then defined by:*

$$J^N[I](\mathbf{x}, \sigma) = \{L_{i_1 \dots i_n}(\mathbf{x}, \sigma) \,|\, (\mathbf{x}, \sigma) \in I \times \mathbb{R}^+; n = 0, \dots, N\} \tag{2.20}$$

This includes every differential Gaussian kernel from order zero to N. The complete Gaussian family $G_{i_1 \dots i_n}(\mathbf{x}, \sigma)$ is specified in (2.17).

The jets are useful to characterize local image structure. They can be parted into categories of geometrical interpretation:

The zeroth order jet is the local (smoothed) illuminance, the first order represents the gradient. The second order jet detects *line orientation* and the third order *line curvature*, due to [7].

Assuming the N-jet components are obtained, they can be joined to produce invariants. Thereby the geometrical interpretation is reduced to a quantity value; information about orientation is lost. That means, Cartesian invariants are scalar quantities (0-tensors), that have coordinate independent values, associated with a single base point in the image and a given inner scale σ.

The rules for joining these elementary building blocks to obtain invariants are the following, according to Florack [2]:

Form any tensor product by multiplying tensor members of the N-jet, supplemented with the 2 constant tensors δ_{ij} and $\epsilon_{i_1 \dots i_D}$, the Kronecker and the Levi-Civita tensor, respectively. Then perform a full contraction on its free indices.

Contraction means, summing over the total number of dimensions, defined by the Einstein summation convention, which is explained later. The effect of the full contraction is rotation invariants. The result are polynomial invariants. Even though the number of degrees of freedom is finite, the number of possible combinations to create polynomial N-jet invariants is uncountable huge. Arbitrary invariants can now be obtained as functions of these polynomials. Obviously almost all of them are functions of a finite number of basic invariants. That is why the index notation for a given invariant is generally not unique.

A finite, minimal number of invariants can be singled out, which completely characterize the N-jet and in terms of which all other invariants can be expressed. This is proved by David Hilbert. The complete proof would go beyond the extent of this thesis, so the reader is kindly referred to the work of Hilbert [5].

As said, the purpose of using tensors is the coordinate independence of their indices. The index notion can easily be transferred into any particular coordinate system. To prevent any misunderstanding, the local coordinates w and v for a given base point in a 2D-image replace the static x, y-coordinates. This is a convenient way to impose invariance under the Cartesian transformation, without actually establishing that particular coordinate system by transformation. Considering differential invariants, indices are used for the spatial coordinate in which the differentiation is taken.

Without loss of generality, the local coordinate system of w and v is rotated in such a way that it fulfill $L_v = 0$ at the given base point. This is always admissible and realizable through a suitable Cartesian transformation of an arbitrary Cartesian frame. At the origin the v-axis is then tangent to the isophote, the constant gray-level curve, and the w-axis is tangent to the gradient, respectively. The following invariant differential operators can be defined with this (v,w)-system:

$$\partial_v = \frac{1}{\sqrt{L_k L_k}} \epsilon_{ij} L_j \partial_i$$

$$\partial_w = \frac{1}{\sqrt{L_k L_k}} \delta_{ij} L_j \partial_i$$

or explicit:

$$\begin{pmatrix} v \\ w \end{pmatrix} = \frac{1}{\sqrt{L_x^2 + L_y^2}} \begin{pmatrix} L_y & -L_x \\ L_x & L_y \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix}$$

Applying these directional differential operators to an image in every conceivable way up to order N will yield a complete set of invariants[2]:

$$\mathcal{G} = \frac{\partial^{m+n} L}{\partial v^m \partial w^n}$$

Tensors can be combined to form a geometrical description of the image structure. Via full contraction of a tensor product, it is reduced to a characteristic, Cartesian invariant, scalar value of the considered structure.

[2]in which the indices (m, n) run over all possible orders up to any desired jet order, except of $L_v = 0$ as defined

2.3.3 Complete Set of Differential Invariants

The complete set of differential invariants, limited to third order, is given in table 2.1 in *manifest tensorial notion*: indices are used for the coordinate to which the differentiation is taken. For simplification, here and in all that follows, the *Einstein summation convention* is used. All latin indices run from 1 to D and the components are summed up over all possible values.

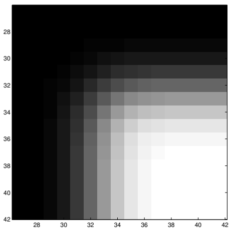
To be sure about the summation convention, have a look at this example:

$$\begin{aligned} L_i L_{ij} L_j &= L_x L_{xx} L_x + L_x L_{xy} L_y + L_y L_{yx} L_x + L_y L_{yy} L_y \\ &= L_x^2 L_{xx} + 2 L_x L_{xy} L_y + L_y^2 L_{yy} \end{aligned}$$

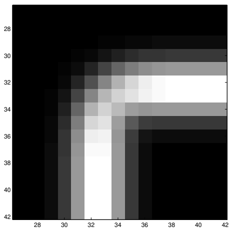
Order	(v,w)-system	Manifest invariants
0	L	L
1	$L_w L_w$	$L_i L_i$
2	$L_{vv} L_w^2$	$L_{ii} L_j L_j - L_{ij} L_i L_j$
	$L_{vw} L_w^2$	$-\epsilon_{ij} L_{jk} L_i L_k$
	$L_{ww} L_w^2$	$L_i L_{ij} L_j$
3	$L_{vvv} L_w^3$	$\epsilon_{ij}(L_{jkl} L_i L_k L_l - L_{jkk} L_i L_l L_l)$
	$L_{vvw} L_w^3$	$L_{iij} L_j L_k L_k - L_{ijk} L_i L_j L_k$
	$L_{vww} L_w^3$	$-\epsilon_{ij} L_{jkl} L_i L_k L_l$
	$L_{www} L_w^3$	$L_{ijk} L_i L_j L_k$

Table 2.1: Complete set of differential invariants, limited to third order

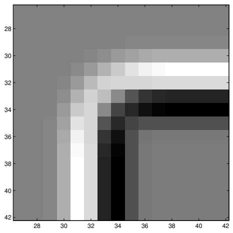
Figure 2.4 shows the corresponding filter responses of a corner region. The scalar differential invariants of this specific corner are respectively the intensity values of the precise corner pixel of each image.



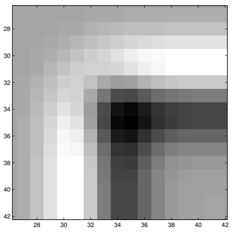
(a) L



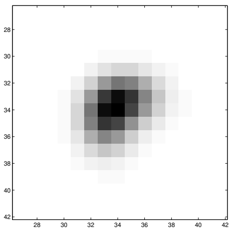
(b) $L_i L_i$



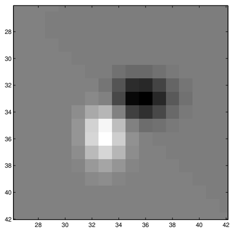
(c) $L_{ii} L_j L_j - L_{ij} L_i L_j$



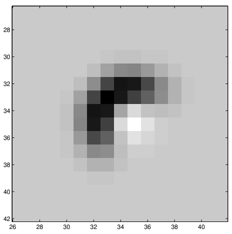
(d) $-\epsilon_{ij} L_{jk} L_i L_k$



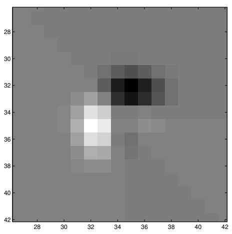
(e) $L_i L_{ij} L_j$



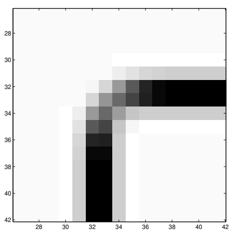
(f) $\epsilon_{ij}(L_{jkl} L_i L_k L_l - L_{jkk} L_i L_l L_l)$



(g) $L_{iij} L_j L_k L_k - L_{ikj} L_i L_j L_k$



(h) $\epsilon_{ij} L_{jkl} L_i L_k L_l$



(i) $L_{ijk} L_i L_j L_k$

Figure 2.4: The 9 descriptors of a perfect corner.

The geometrical meaning of some of the two-dimensional invariants is well known and presented in table 2.2.

Name	Cartesian	(v,w)-system
Intensity	L	L
Gradient-magnitude	$L_x^2 + L_y^2$	L_w^2
Laplacian	$L_{xx} + L_{yy}$	$L_{vv} + L_{ww}$
Isophote curvature	$\frac{2L_xL_yL_{xy} - L_y^2L_{yy} - L_y^2L_{xx}}{(L_x^2+L_y^2)^3/2}$	$-\frac{L_{vv}}{L_w}$

Table 2.2: Some examples of two-dimensional invariants, taken from [20]

For sure, higher order invariant properties exist, but it becomes increasingly more complex to assign a proper geometrical meaning or name to the expression.

To give a compact, local description of the image structure at the interest points, the set is stacked in a 9 dimensional vector, named $\mathcal{V}$. It is given in equation (2.21), again in tensorial notation (Einstein summation convention).

$$\mathcal{V}[1..9] = \begin{bmatrix} L \\ L_iL_i \\ L_{ii}L_jL_j - L_{ij}L_iL_j \\ -\epsilon_{ij}L_{jk}L_iL_k \\ L_iL_{ij}L_j \\ \epsilon_{ij}{}^3(L_{jkl}L_iL_kL_l - L_{jkk}L_iL_lL_l) \\ L_{iij}L_jL_kL_k - L_{ijk}L_iL_jL_k \\ -\epsilon_{ij}L_{jkl}L_iL_kL_l \\ L_{ijk}L_iL_jL_k \end{bmatrix} \tag{2.21}$$

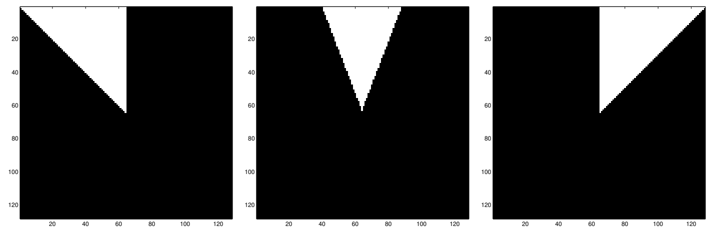
with $i, j, k, l \in \{x, y\}$.

$\mathcal{V}$ presents a full, invariant description of an interest point detected in the image. Hence, $\mathcal{V}$ is called a *descriptor*.

[3] 2D antisymmetric Epsilon tensor: ϵ_{ab}

$$\epsilon_{ij} = -\epsilon_{ji} = 1$$

$$\epsilon_{ii} = -\epsilon_{jj} = 0$$



(a) (b) (c)

Figure 2.5: Images of a rotated corner.

Table 2.3 illustrates the (approximate) rotation invariance of the descriptor $\mathcal{V}$, extracted from the corner in figure 2.5.

geometrical meaning	pic. (a)	pic. (b)	pic. (c)
luminance	0.4811954	0.5172395	0.4811954
gradient magnitude	0.0143137	0.0101232	0.0143137
	-0.0012717	-0.0009949	-0.0012717
	0.0001570	0.0000000	-0.0001570
	0.0082149	0.0097882	0.0082149
	-0.0001276	-0.0001151	-0.0001276
	-0.0977720	-0.1096523	-0.0977720
	-0.0000073	0.0000000	0.0000073
	-0.0000084	0.0000000	0.0000084
est. x (exactly)	63(64)	64(64)	66(65)
est. y (exactly)	62(64)	62(65)	62(64)

Table 2.3: The complete set of differential invariants computed from the corner in figure 2.5. The last two rows present the estimated pixel-location of the corner; the exact value is given in parentheses. The deviation is caused by blurring as discussed above.

At this stage, a local structure description, invariant against Cartesian coordinate transformation has been achieved. But the attributes of the descriptor are highly depending on the filter range (the scale). If the resolution of the image structure is extended or reduced, some of the descriptor attributes might change dramatically and the structure can not be recognized anymore.

2.3.4 Scale Space

Objects can be represented in different *scales*. Image structure and scale space have been intensively studied by Florack [2]. Scale limitations are posed by the acquisition device, i.e. the grid size and the field of view. The smallest scale of interest will be called the *inner scale*. Object recognition is based on local image features, and for sure, these features depend on resolution. Imagine, that the scale indicates the amount of pixels, which represent the same object view in images. It is changed by variation of either resolution of the image (the image size) or distance between camera and object during the picture shots (the size of the object inside the image).

Since the interesting structures are corners near pixel-scale, an under-sampling problem appears. To prevent under-sampling, the inner scale must be sufficiently large. The variable width of the structure operator, based on the Gaussian kernel, accounts for the spatial scaling degree of freedom. With this parameter σ, the kernel size can be adapted to the size of the considered structure or inner scale. Unfortunately it is not possible to estimate this adjustment automatically. Either an optimal operator width has to be fixed or, in the case of serious scale changes, "scale-invariance" can be achieved by evaluating multiple operators at quantisized scales. Otherwise, the descriptors of image structures at different scales can not be compared.

The *distance*, i.e. the pixel coordinates in images of different levels of inner scale can be re-parameterized to their scale:

$$\boldsymbol{u} = \frac{\boldsymbol{x}}{\sigma\sqrt{2}},$$

according to

$$w_{\boldsymbol{u}} = \frac{e^{-\boldsymbol{u}^2}}{2\sigma^2}.$$

For the grey-level function L of an Image I a scale change of σ can be described by a simple change of variables, $L(\boldsymbol{x}) = G(\boldsymbol{u})$. The effect on the m-th derivatives of L is

$$\partial_m L(\boldsymbol{x}) = \left(\frac{1}{2\sigma^2}\right)^{m/2} \partial_m G(\boldsymbol{u}).$$

In this notation, distances are now expressed in the same unit steps, i.e. in steps of the inner scale. With the prefactor of σ, m, the structure derivatives are comparable, too.

That way, local descriptors extracted from images of different scales can be adapted, so that corresponding descriptors are recognized.

2.4 Model Descriptors

Together, the descriptors of every interest point form a signature of this object image. But this is not characteristically enough to recognize every object appearance correctly. The theoretical invariance of these descriptors is practical limited. As mentioned before, the loss of information during the projection exceeds the redundancy of information around an interest point.

For example, the vertex of a pyramid may be an interest point. In a side view you can maximal see two sides of the pyramid, while in a view from above the vertex, every side of the pyramid can be seen. This means a huge variation in the surrounding of the vertex and thereby of the descriptor.

This example shows the impossibility of calculating a single corner description, truly invariant against all kind of 3D-transformation.

This problem is solved by calculating repeatedly descriptors of the same interest point from different views. To cover every possible appearance of one interest point, the model consists of a "framework" of sample descriptors. Now, the invariance of the descriptors within a smaller deviation is sufficiently to ensure, that the same interest point can be matched in every possible view in between these sample views.

Practically this is done by choosing every n-th picture of an object image series [4], detecting all the interest points and calculating their descriptors. To be sure, that the points are not accidently detected, they have to be matched in the previous as well as the following picture, too.

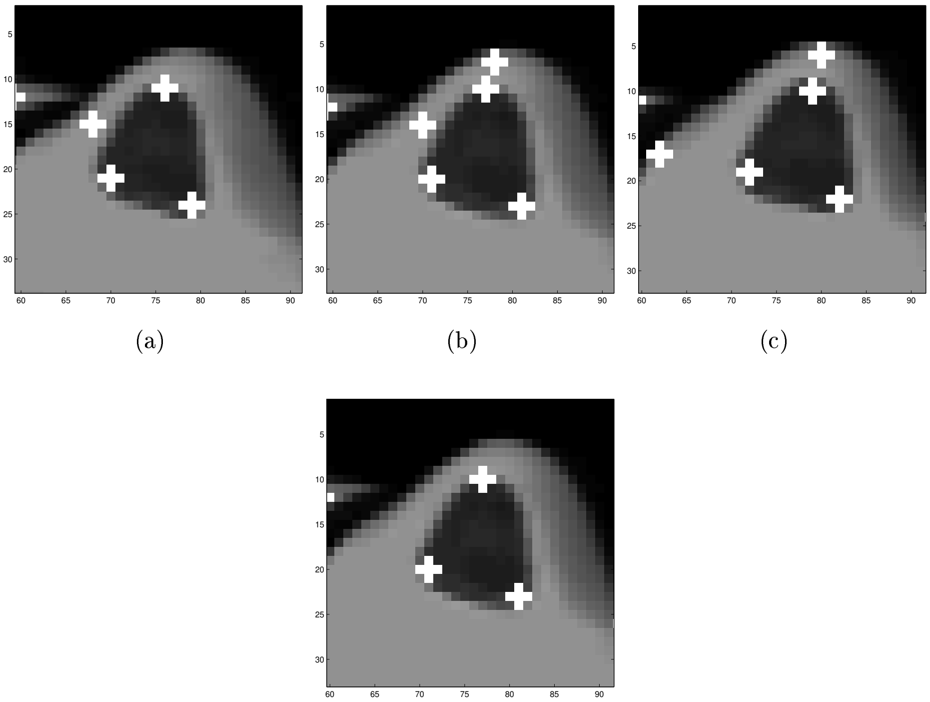
Have a look at figure 2.6 for an example. (b) is the reference image. Now, in the nearest surroundings of a detected interest point of (b) a corresponding descriptor is sought after in the preceding (a) and following images (c). The matching criterion is, apart from the position adjacency, a similarity threshold for the descriptors. Similarity is estimated by a distance calculation[5]. If there are several candidates in the respective region, then the most similar one is chosen. If it is detected in all three images in a row, this interest point has demonstrated repeatability and is therefore considered to be significant (image (d) in figure 2.6).

Table 2.4 contents the complete nine-dimensional descriptor $\mathcal{V}$ of the lower right corner of the cat's ear in the three pictures of figure 2.6.

The disparity value between the first and the second descriptor in the table is 0.1910 and respectively 0.2897 between the second and third. This indicates a good correspondence, since these values are below the experimentally estimated threshold of 0.5 (see section 4.1).

[4] This approach is based on image data in sequence form. That means, a continuous sequence of views, 360 degree around the object, is required.

[5] see section 3.2 for an accurate definition



(a) (b) (c)

(d)

Figure 2.6: This is an example for the interest point selection. (a)-(c) shows every detected interest point in a specific image clip. (b) is the reference image, (a) the previous and (c) the next image in a sequence. (d) contains only the remaining points of (b), used for the model. Only these points could be matched in every picture as a required indication of their significance (The considered clip shows the left ear of a cat, object no.4 of the *COIL-20* database).

pic. (a)	pic. (b)	pic. (c)	*MEDIAN*
4.0098e-01	4.4030e-01	4.2443e-01	4.2191e-01
3.6233e-02	3.7640e-02	3.0715e-02	3.4863e-02
4.0576e-03	3.6204e-03	3.0472e-03	3.5751e-03
2.4667e-02	2.5565e-02	2.6404e-02	2.5545e-02
2.4670e-02	2.6025e-02	2.6576e-02	2.5757e-02
-2.7966e-04	-1.7323e-04	-4.2994e-05	-1.6529e-04
1.2111e-04	2.2307e-05	8.4916e-05	7.6112e-05
6.7606e-05	2.0823e-05	8.2918e-05	5.7116e-05
-3.0068e-04	-3.9581e-04	-9.5131e-05	-2.6387e-04

Table 2.4: The complete set of differential invariants computed from the lower right corner of the three pictures in figure 2.6 and the average values for the median descriptor.

The median of the three interest point descriptors forms the model descriptor, which is finally stored in the model database.

2.5 Semi-Local Coherence

So far, the model strictly consist of almost punctual selected features. Correspondences are based on similarity between sole pairs of descriptors. Correlations between the points are completely disregarded.

2.5.1 Spatial Constraints

The spatial constraints are suggested by Schmid [15]. If a correspondence occurs between Q and model image M_m, it is requested, that at least 50% of the n nearest neighbors of M_m have also correspondences to the query image *and* accomplish the demand to their constellation: their angle must conform approximately (see Fig 2.7).

It is supposed, that a modest 3D-transformation of the object in the real world only slightly affects the image of the 2D-projection. So, as long as the test image shows a view of the object that is within this range of one of the images used for training, the shape configuration should remain similar. Under this assumption the geometrical coherence of the related interest-point configuration represents a further constraint, then. Basic range measures like the distance or angle between the points, as proposed by Schmid [15], are strong restrictions to the descriptor correspondence.

The utilization of these constraints is discussed in section 3.4 and modifications suggested.

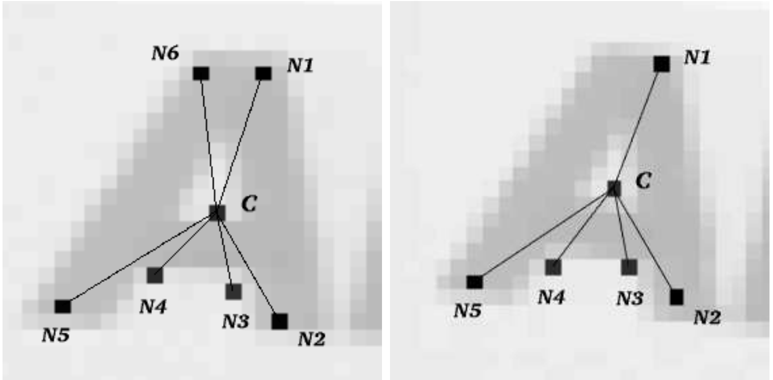


Figure 2.7: After a 10-degrees (spatial) rotation of the object, 6 of the 7 interest points, found in the first image clip, are detected in the second, too.

2.5.2 Intensity Profiles

Another approach is presented by Matas [8]. Instead of considering the intensity values of the interest points, he uses the extension line in between them as an invariant descriptor. Research into projective invariants has shown that perspective projection preserves convex hulls [18]. A line segment can be considered as the simplest convex hull of a pair of points. So the intensity profile of the line segment between two interest points represents again a more global invariant descriptor of the object.

Opposite to the corner descriptor the intensity profile characterizes the object surface and returns a coherent object description.

The calculation of the intensity lines have not been explained any further by Matas, so the following technique is self-contained developed.

Strategy

To make the profiles comparable, the line segment is normalized to a length of l. Because of pixelisation, the straight extension line will mostly run in between quantisized pixel values. So the profile has to be calculated as a line of sub-pixels. A sub-pixel lays in between the standard pixel coordinates (Figure 2.8). Its value is the sum of the four surrounding pixel values, weighted according to their covered part.

Assuming that we have obtained two interest points g_1, g_2 at the coordinates $\boldsymbol{x}_1 = (x_1, y_1)$ and $\boldsymbol{x}_2 = (x_2, y_2)$, respectively. To compute the intensity profile $\mathcal{P} = \{p_0, \ldots, p_l\}$ of $(l+1)$ grey-level values in between (and inclusive) these points, the following procedure is proposed:

- $p_0 = g_1$ and $p_l = g_2$ is trivial.

- $p_i, \forall\, i \in \{1, \dots, l-1\}$:
  - calculate the coordinates of the sub-pixel $\boldsymbol{x}_{sub}(i)$,

$$\boldsymbol{x}_{sub}(i) = \boldsymbol{x}_1 + \frac{i}{l}(\boldsymbol{x}_2 - \boldsymbol{x}_1)$$

  - estimate the negative deviation $d\boldsymbol{x}(i)$ of the sub-pixel from the according lower right pixel,

$$d\boldsymbol{x}(i) = \boldsymbol{x}_{sub}(i) - \lceil \boldsymbol{x}_{sub}(i) \rceil$$

  - and finally determine the intensity value p_i of the sub-pixel:

$$p_i = \sum_{u=0}^{1}\sum_{v=0}^{1} \Big|(dx(i)+u)(dy(i)+v)\Big| g\Big(\lceil x_{sub}(i)\rceil - 1 + u, \lceil y_{sub}(i)\rceil - 1 + v\Big)$$

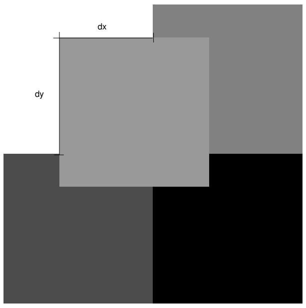


Figure 2.8: A sub-pixel

It makes sense to restrict the number of profiles. Like the spatial constellation, only the intensity line between every interest point and the n nearest neighbors are considered. Short profiles does not change so much during a slight 3D-transformation and combinatorial explosion along with the number of interest points is avoided.

The values of the profile are normalized over the length to be invariant against intensity changes.

Like the differential descriptors these intensity profiles form a comparable signature of an object image and can be stored in a database in the same way.

The model components, like the local descriptors, the spatial constraints or the intensity profiles can either be used as (more or less) stand-alone applications or contribute to the entire recognition. In the latter case, they are integrated as successive modules and ensure the former results. Then the respective constraints and thresholds can be unfasten to achieve a larger error tolerance.

The realization of the model construction and the feature comparison for the object recognition is illustrated in the next chapter.

Chapter 3

Object-Model Comparison and Assignment

To recognize a query object an object description corresponding to the model description have to be derived. Thereto the same algorithms as for the model building process are used to extract the interest points and to calculate the corresponding differential invariants. The new acquired object descriptors are then compared to every model descriptor stored in the database of object models. To enable a fast access method to the requested entries, a multi-dimensional indexing structure [22, 12] is presented in section 3.1. Due to small deviations of the invariant values, *similar* and not identical descriptors are requested. According to Schmid [15], similarity is limited to a distance threshold between two descriptors. The distance estimation is defined in section 3.2. Every model descriptor in the distance-range of one of the query object descriptors gets a vote for the corresponding model.

The voting method [15] has a sufficient statistical tolerance, which should be capable of occasional miss-matches or lack of correspondences. But it depends on an equal distribution of descriptors. Objects can be very different from each other, considering their quality of description. Because the description is completely depending on the preceding corner detection; squared or textured objects are more salient than rotund or homogeneous.

The simple voting method is inadequate taking account of this saliency and it is therefor replaced by a probabilistic model (inspired by [14]). This is discussed in detail in section 3.3.

To further decrease the probability of false matches, semi-local constraints [8, 15] are finally admitted to the feature comparison in section 3.4.

3.1 The Database

As mentioned above, the model descriptors are kept in a database. The voting algorithm looks for every model descriptor, which is similar to the one of the query image.

Without indexing the complexity of the voting algorithm is about $l \times N$, where l is the number of descriptors of the query image and N the total number of model descriptors in the database. To accelerate the comparison an efficient data structure and search technique is required. Tree structures have been developed to solve different search operations.

Due to the statistical character of the voting, the request toward this data structure is not to find exact matches, but to make a range-query work effectively. That leads to a grouping of similar descriptors. Because the data consists of N-dim-vectors, similarity is required in multi-dimensions.

k-d-Tree

The so called k-d tree [12] is based on such a multi-dimensional indexing technique (k-d means k dimensions, see 2-d-example below). It is the generalization of a binary sort tree with a k-dimensional sort key. Each knot of the tree contains a complete data record and two pointers to the left and right son. The values of the vector attributes are alternating used for discrimination. The resulting structure is static and probably not well balanced. Deleting an entry is therefor very difficult; the complete training must be repeated instead.

The following is a little two-dimensional example of this tree structure, taken from a lecture of Vornberger [22].

2-d-Tree Example

Figure 3.1 shows an arbitrary (x,y)-distribution.

The data is now entered into a 2-d-tree. The entries are classified alterant considering the x-values at the root and odd levels and the y-values at even levels. The rules for every knot with key $[x/y]$ are:

	left son	right son
odd level	all keys $\leq x$	all keys $> x$
even level	all keys $\leq y$	all keys $> y$

This leads to the binary tree structure of figure 3.2 in which the branches divide the data field into more and more refined rectangles (the lines in figure 3.1).

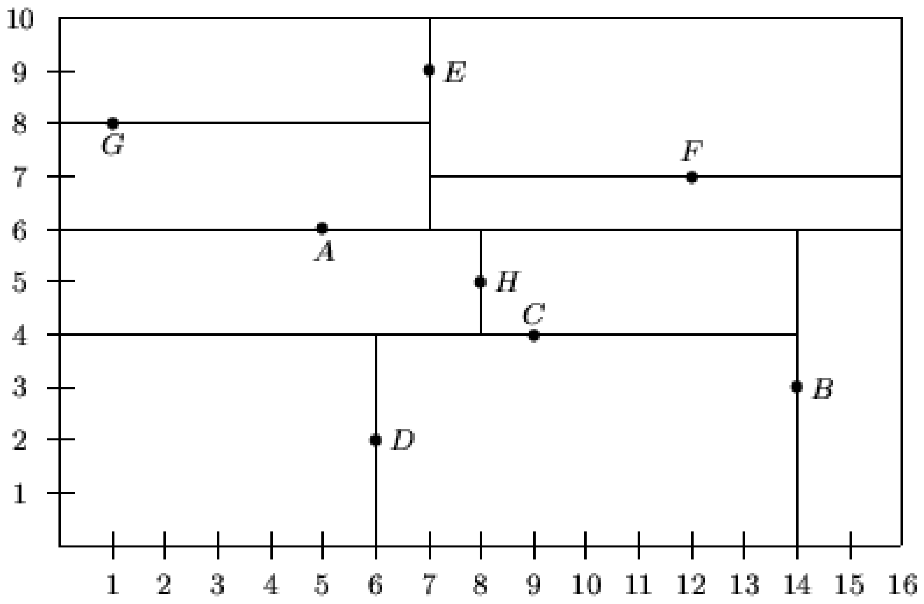


Figure 3.1: An arbitrary two-dimensional data distribution. Inserting the points into the 2d-tree leads to a more and more fine segmentation of the space, represented by the lines. The figure is taken from [22].

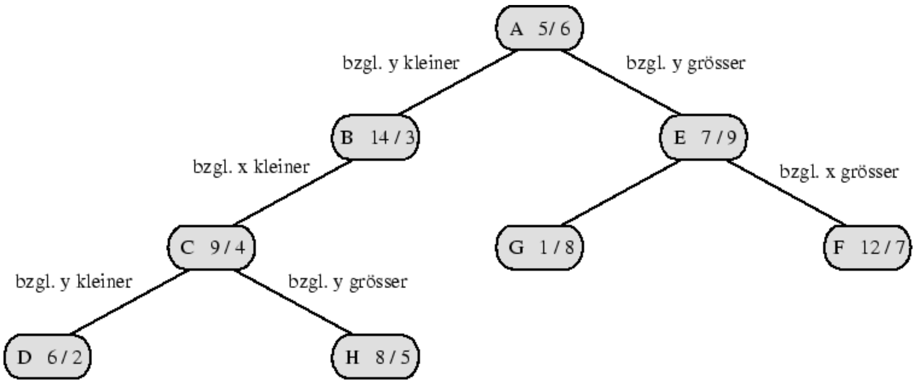


Figure 3.2: The 2-d-tree for the data given in figure 3.1. The knot values in the left branches are always lower than the values in the knot point above in the according dimension. Respectively are the values in the right branches higher. The figure is taken from [22].

General Notation of the k-d-tree

Insert: A vector entry is inserted into the data structure by comparing its first attribute to the root entry. If it is higher than the value of the root entry, the new entry is referred to the right branch, otherwise to the left. On each following level the attribute of the next dimension is used for discrimination. The depth of the structure can grow beyond the number of dimensions. In this case, the discrimination continues with the first attribute again, until finally the son pointer is empty: the final place of the new entry in the tree structure is found.

The advantage of this structure is not so much the fast detection of an exact match but rather of several similar entries inside a search rectangle. Generally spoken, that means for the k-d-tree a k-dimensional search cube. Entries similar to a reference vector can be found easily and fast by a *range-query* search.

Range-query search: A k-dimensional search cube is calculated by extending every component of the reference vector with a threshold range. At the tree knots, where the discrimination line crosses the search cube in the respective dimension, test the similarity or distance of the entire entry to the reference vector and descent in both sons, otherwise descent in the appropriate son.

In this approach the model descriptors, extracted during the training, are inserted into this tree structure. Descriptor comparison is then performed by a range-query search of every model descriptor, similar to one of the query image.

3.2 Comparison of the Descriptors

The comparison of the descriptors can be carried out in a manifold way, because of their multi-dimensionality. There are several techniques of distance estimation and possibilities to consider statistical errors and variance. This is necessary, though the features extracted from the images are differential invariants. But they are based on projected 2-dimensional data and thus they simply cannot be invariant against all kind of preceding 3D-transformation of this data. Detection errors of the interest point locations cause further deviation of the values.

The descriptor is an N-dimensional vector. Each attribute is another differential invariant, extracted from the interest points. The similarity of two descriptors is estimated anti-proportional to the distance between them. There are several requests toward the distance estimation:

- The distance is assembled equally of the individual distance of every attribute.
- This means that the attributes must be normalized to have equal weight for comparison, regardless of their magnitude. (The relative distance is important - not the total!)
- Only values of the same attributes are compared.

The Distance Estimation Function

The distance between two n-dimensional points a and b can be measured using different metrics.

Minkowski metric sums up the distance of every single dimension.

$$\left\{ \sum_{i=1}^{n} |(a_i - b_i)^p| \right\}^{1/p}$$

The constant p shifts the weight of the individual distances for the total distance. For $p \to \infty$ is $d_{\boldsymbol{a},\boldsymbol{b}} = (d_{a_i,b_i})_{max}$. For the special case of $p = 1$, the Minkowski metric gives the *City Block metric*, and for $p = 2$, it gives the *Euclidean distance*. The latter is the most common used distance estimation. It provides the direct distance through space with every vector attribute weighted equally.

In the case of statistical distribution, however, a comparable distance estimation is needed, which takes account of difference in the variance and magnitude[1] of the particular descriptor attributes.

To accomplish the equality request, the magnitude has to be normalized and the individual distance threshold must adapt the variance of the according attribute.

A general weighting of the attributes is not possible because of their big variation. An invariant attribute of the same interest point varies seldomly over more than one decade (in the range of "invariance"). But considering all the interest points, the value varies over a far larger range.

Mahalanobis distance is proposed in [13].

$$d_{\boldsymbol{a},\boldsymbol{b}} = \sqrt{(\boldsymbol{b} - \boldsymbol{a})^T \Lambda^{-1} (\boldsymbol{b} - \boldsymbol{a})}$$

[1] Magnitude means the size of the scalar values

It takes account of the different magnitude as well as the covariance matrix of the components. But in order to obtain accurate results for the distance, a representative covariance matrix is important to have. The matrix can be estimated statistically by tracking interest points in image sequences. To have an accurate and representative covariance matrix it is required, that the number of sample vectors of the same interest point exceeds the number of dimensions. This means in the normal case of nine descriptor attributes, that at least ten images for each interest point are needed. So, to restrict the variance of the attributes a very large dataset of sample images is required for the training.

The purpose of this approach, however, is also to examine and (if possible) to increase the efficiency of the recognition system. This means, for example, to decrease the necessary amount of training-data to build an accurate model.

A New Distance Function

Instead of regarding to existing distance estimation methods an individual distance estimator is proposed here. It takes the magnitude into account and is robust against single "elopers".

The distance between a model descriptor $\mathcal{V}_M$ and a query object descriptor $\mathcal{V}_Q$ is given by:

$$\text{dist} = \frac{1}{9} \sum_{i=1}^{9} \left| \log \left(\frac{\mathcal{V}_{M,i}}{\mathcal{V}_{Q,i}} \right) \right| \tag{3.1}$$

The division of the vector components in (3.1) results in a deviation value which is independent of the magnitude (10/100 is the same as 0.01/0.1). This is very important, since the magnitude of the differential invariants is not a quality measure but only reflecting the local frequency. So, removing the magnitude difference between the different vector components results in an equal weighting of the components.

The next step is to calculate the absolute logarithm of the division result. The logarithm of a division is a handier value. Considering the absolute value has two advantages:

- The division result is the same, no matter which way you divide the two numbers:

$$\left|\log(\frac{100}{10})\right| = 1 = |-1| = \left|\log(\frac{10}{100})\right|$$

- If the division is negative, the logarithm results in a complex number and the absolute value is higher than the according real number. That means a larger distance according to the contrary algebraic signs of the values.

$$|\log(-1)| = |0 + 1.36i| = 1.36 > 0 = |\log(1)|$$

Tracking identical interest points in several images show, that the values of the differential invariants deviate, but generally not the sign of the values. Thus, a contrary sign results in a greater descriptor distance than, for example, an alteration of one decade (comply with 1).

Finally all (positive) section results are summed up to the total distance value, normalized over the number of dimensions.

- If the result is zero, the two vectors are completely identical.
- If the result is one, the average deviation of every component is about a decade.
- If a single component has a large deviation, but the rest is almost identical, this can be presumed as an "eloper" and the final result will still be small.

A threshold between zero and one is now used to estimate, whether or not two descriptors are similar. If their distance is beneath this threshold, there exists a descriptor correspondence and the according model gets a vote.

Have a look at figure 2.6 and the belonging table in section 2.4 for a concrete example.

3.3 Descriptor Correspondence

As mentioned above, a number of error sources can falsify the recognition result. Quantization effects and illumination changes are namely the reason for possible sequence errors:

- The repeatability of the corner detector is good, comparatively the best, but not perfect.
- There might be miss-matches of corners during the training.
- The differential invariants are only invariant to some extent. Deviation of the values can cause correspondence errors.

So, a given feature might vote for several models. More specific: for an extracted interest point descriptor of the query image, there might exist correspondences to several descriptors of different models.

Due to error tolerance, recognition should be based on a probabilistic model. The voting method is already a statistical majority decision, but experiments have shown, that this is only sufficient for 50% of the images.

Unfortunately, simple voting does not take account of the distribution of interest points or the saliency of descriptors. In images of objects with an accurate shape or texture a lot of interest points are detected (like object 5 and 9 of the *COIL-20* database, see fig. 1.1). There against, it is difficult to find corners in rotund objects with homogeneous surfaces (like object 2 or 12, respectively).

It is particularly possible to affect the number of detected interest points via the threshold k in (2.11) on page 18, but this affects concurrently the quality of the corner detection. k is proportional to the robustness or anti-proportional to the sensibility of the detection. Increasing the sensibility leads to arbitrary detections of unimportant points. These miss-detections can not be used for matching. Because of their coincidental occurrence, they are not repeated in other images.

Due to this fact, some model images are described by over eighty descriptors, while others have less than ten. Clearly, objects with a large amount of descriptors get probably more votes than the others. Since the recognition is based on probability, the statistical distribution has to be assimilated.

A probabilistic model is presented, which takes account of the different quantities. The model is inspired by Schmid [14], but the strategy and implementation are self-contained developed and are very distant modifications from [14].

The complete presentation of the probabilistic model of Schmid would go beyond the extent of this thesis, so the interested reader should refer to [14]. The results obtained with the following implementation are conform with the results of Schmid and are less calculation-intense.

Probabilistic Model

According to the notation of [14], a query image is in the following denoted by Q and the set of model images by $\mathcal{M} = \{M_m\}$ with $\hat{m}$ being the number of model images. Each image I is characterized by a set of descriptors $\mathcal{D} = \{D_d(I)\}$ with $\hat{d}$ being the number of descriptors for image I.

The complete set of descriptor correspondence for Q is denoted by $\mathcal{C}(D_i(Q)) = \{C_c(D_i(Q))\}$ with C_c being the c-th correspondence of descriptor $D_i(Q)$. A correspondence $C_c(D_i(Q)) = D_j(M_m)$ between Q and the model image M_m

occurs, if the distance between the pair of descriptors $dist(D_i(Q), D_j(M_m))$ is below a threshold.

The complete set of correspondences between Q and a specific model image M_m is denoted by $\mathcal{C}_m(\mathcal{D}_m(Q)) = \{C_{c_m}(D_{c_m}(Q))\}$ with $c = \{1, \ldots, \hat{c}_m\}$ and $\hat{c}_m$ being the number of correspondences.

The simple voting method for a model image is extended by the following considerations:

The probability of a model correspondence P_C is expressed by the ratio of the quantity $\hat{c}_m$ of correspondences $\mathcal{C}_m(\mathcal{D}_m(Q))$ between Q and M_m over the sum of their distances $dist(D_i(Q), D_j(M_m))$. Every individual descriptor correspondence of $\mathcal{C}_m$ is the more probable the less the distance is between the two descriptors or the more similar they are. The median value of these single probabilities gives the entire probability of the model correspondence. The larger this value, the more probable the entire conformity is between the query image and model M_m:

$$P_C = \frac{\hat{c}_m}{\sum_{c=1}^{\hat{c}_m} dist(D_{i_c}(Q), D_{j_c}(M_m))} \tag{3.2}$$

The distribution of correspondences P_D is considered by the ratio of the number of correspondences $\hat{c}_m$ for a model M_m over the total amount $\hat{d}_m$ of its descriptors $\mathcal{D}(M_m)$:

$$P_D = \frac{\hat{c}_m}{\hat{d}_m} \tag{3.3}$$

This is a more reasonable indicator than the number of correspondences alone. It considers the unequal distribution mentioned above.

It is now possible to combine the individual probabilities in a simple multiplication with the voting distribution:

$$P = \hat{c}_m \cdot P_C \cdot P_D = \frac{(\hat{c}_m)^3}{\hat{d}_m \sum_{c=1}^{\hat{c}_m} dist(D_{i_c}(Q), D_{j_c}(M_m))} \tag{3.4}$$

Figure 3.3 shows an example of the different probability distributions. A query image is compared to the complete database of 360 model images. The diagrams show the probability values of the correspondence between the query image and each model image. The object is already recognized correctly by the voting algorithm (the model image with the highest number of votes in (a)). There is also a peak in (b) at the same position, which indicates, that the descriptor correspondences between this model and the query object are closest. This is only a slight deflection, but it indicates also the correct model. The ratio of votes over the number of descriptors of each model in (c) indicates the model with the highest rate of correspondence. Again, the correct model is chosen. The multiplication of the individual probabilities ensures the correct result and increases the rate of recognition further. This product is represented in (d).

Still, the recognition is strictly based on local descriptors. This probability model is able to compensate unequal distribution of both quantity and quality of the object description.

Furthermore, it is possible to decrease the number of arbitrary, incorrect descriptor correspondences by considering the spatial relation of the interest points.

3.4 Semi-Local Coherence

The so far local object description can be extended with a more global coherence: The spatial constraints and the intensity profiles, both described in section 2.5.

This semi-local or more global contemplation of the object images leads to an useful and powerful expansion of the model.

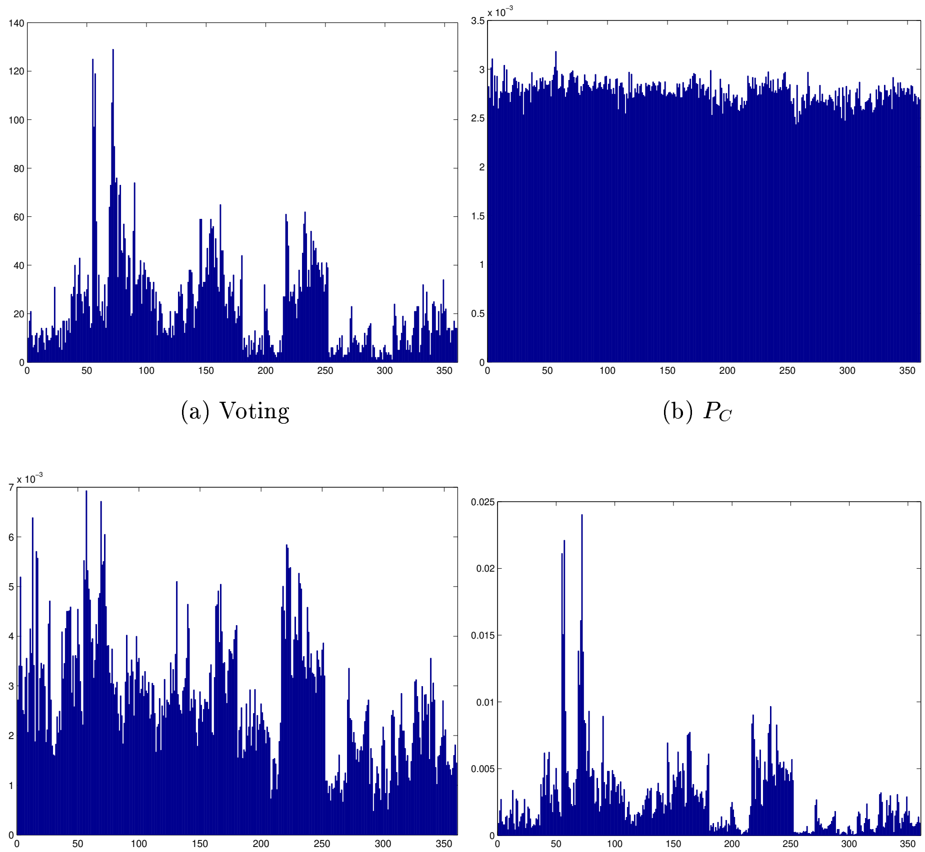
Spatial Constraints

Considering the spatial constellation of the interest points offers a good verification of the individual descriptor correspondence.

To find the n closest neighbors, as suggested by Schmid, appeared to be impractical. Their search costs calculation power and the n closest points may vary between two images.

Experiments showed that the angle deviation between interest points does not depend on the distance, which is self-evident. On the other side, they showed indeed that the neighborhood of the interest points deviates from image to image, which leads to a comparison of improper interest points.

So the local constraints used here, differ from [15]. Instead, the local configuration of all the pairs of corresponding points between the model and the query image are considered . A pair of correspondence is only kept if it also fulfills the spatial constellation with another pair. This means, that the angle and distance between the two points of the query image must confirm with the two corresponding points of the model image. This spatial constraint can be required for two or n pairs of descriptors, it depends on the rigidity.



(a) Voting

(b) P_C

(c) P_D

(d) Probability Model

Figure 3.3: The improvement of the voting method: The x-axis contains the 360 model images, 18 for each of the 20 objects. The test image is of object four and its pose is between the third and fourth model image. This means between 57 and 58 on the x-axis (Object four stretches from 55 to 72.) (a) shows the simple vote distribution. There appears already a peak at the designated position. (The second peak corresponds to the mirrored model pose, since the object is partly symmetric.) (b) and (c) represents the normalized probabilistic distribution of P_C and P_D, respectively. Finally (d) is the normalized product of (a) to (c). The two peaks are now even more outstanding.

Thus, a model gets a final vote if n pairs of descriptor correspondences also correspond in their spatial constellation.

As the results in section 4.1 prove, this semi-local constraint is a powerful tool to increase the recognition rate substantially.

Intensity Profiles

Opposite to the spatial constraints, here, the restriction to adjacent points is essential to avoid vast calculation costs. To limit the amount of comparison furthermore and to avoid comparison of a false neighborhood, the local descriptors are used to select corresponding interest points, between which the intensity profiles can be calculated. If the points are adjacent, as described in section 2.5.2, their exist a comparable profile stored in the model database.

The correspondence of the profiles is estimated by considering the intensity values as vector attributes. Then the euclidic distance between two vectors must fall below a similarity threshold.

In the following chapter the particular recognition methods are elaborately tested and an optimal combination is evaluated.

Chapter 4

Experimental Results

The effect of the particular modules of the entire recognition model are elaborately examined in section 4.1. First, the influence of the diverse thresholds on the recognition result are discussed, followed by the recognition result of the solitary modules. Thereby, the improvement, achieved by the combination of the modules, becomes obvious. Then, the robustness of the complete model is demonstrated against different image distortion, which make a correct recognition difficult. After all, the experimental results are discussed in section 4.2.

Input Data

As mentioned previously, the image data of this work is taken from the *COIL-20* [11], a publicly available database of 72 views of 20 natural objects rotating on a turntable. The size of the 1440 grey-scale images is 128×128 pixel and they show the object each time rotated about 5 degrees (see Fig. 1.1 and 4.1). The objects are illuminated from the camera direction. Background-pixels have been removed and the intensity values are normalized to a maximum value of 1. The images are rescaled, so that the objects fit exactly the image borders in their largest expansion in every appearance. This means, the images contain both scaling, blurring[1] and intensity variance in the object appearance.

For the recognition task one part of the pictures is used to train the program and the other remains as test images.

[1] as effect of the image crop and resize

Figure 4.1: The 72 views of object no. 4 of the COIL-20 database [11]. The object has been rotated about 5 degrees between every snapshot.

Software

All the programs have been developed and tested with *Matlab* (Release 12.1) using the *Matlab image toolbox*.

The program structure is given in appendix A.

4.1 Tests with the *COIL-20* Database

This object recognition system has been adapted primarily to the *COIL-20* database. But it should be capable of other similar recognition tasks, because of its local extracted, appearance based object description. The demand to the training set is an image sequence of contineously changing object poses. Thereby, it must be possible to track interest points from one image to the next.

To obtain best results, first of all, the thresholds of the corner detection and interest point description have to be adapted to the image data to achieve optimal model descriptors:

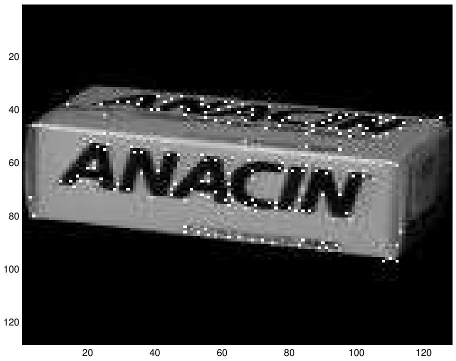
4.1.1 Thresholds of the Algorithms

Adjusting the thresholds appeared to be a real task of engineering. Optimal values exist for every experiments but there exist no *generally* best values. The thresholds highly influence the results and must become experimentally optimized for every different task.

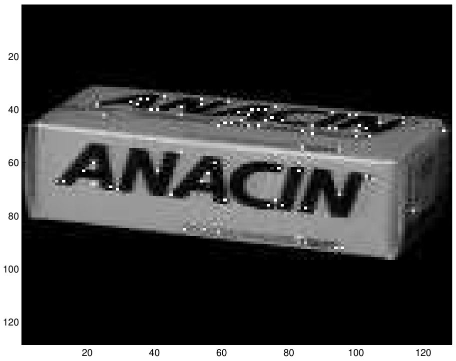
Corner Detection

σ and k in the algorithm of the corner detection must be adjusted to the image quality and to a minimum quantity of detections. σ is also responsible for the resolution of the local corner descriptors (here σ is independent of the corner detection and might have another value). The considered size or inner scale of the image structures depends on the range of the Gaussian differential kernel. This value must be identical in both training and testing to guarantee comparable descriptors. The effects of the variations of these values are demonstrated in figure 4.2.

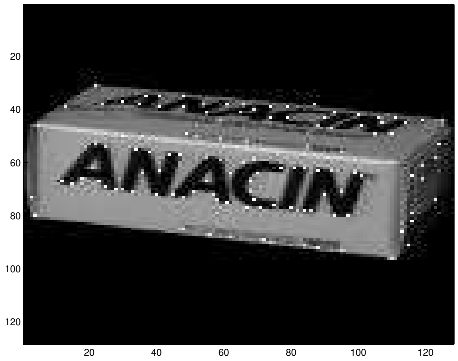
In the same way, σ of the Gaussian derivatives, which build the model descriptors must be adapted to the inner scale, the extent of the image structure.



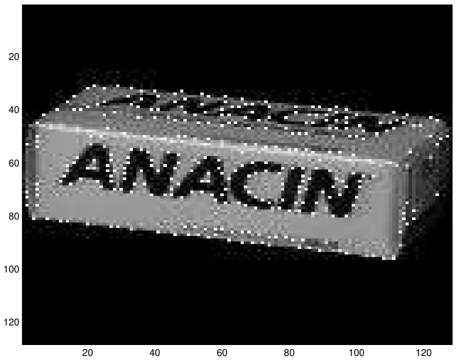
(a) σ too small



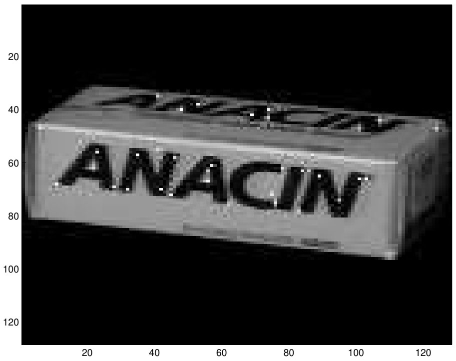
(b) σ too large



(c) optimal



(d) k too small



(e) k too large

Figure 4.2: Different values for σ and k in the corner detection algorithm are presented (The effects of the alteration of these values are discussed in section 2.2). In (a) σ (a measure for the extent of the detector) is so small, that some of the corners are detected twice. In (b) this extent is chosen too large, so that corners are missed and the location of the detections is very inaccurate. (c) presents the optimal result of the (experimentally) optimized values. (d) and (e) shows too loose or too rigid values respectively for k.

Descriptor Correspondence

The similarity constraint for the descriptor comparison depends on the deviation between the model images. It must neither be too rigid, in case of the object appearance varying a lot in the image sequence, nor too loose to avoid false correspondences. During the training it should be a constant to guarantee an even variance. The tests indicate that the similarity constraint during the testing should depend on the number of detected descriptors in the query image. In case of a large number the constraint should be rigid to avoid a confusing number of miss-matches and in case of only a few descriptors it should be loose. This is explained by the fact, that the spatial constraint supplements the similarity constraint very well. It does not matter, if a few descriptors produce a large number of miss-matches, as long as the correct correspondences are detected, too. Then the spatial constraint ensures the correct result. Thus, the threshold for the similarity constraint should be anti-proportional to the number of query image descriptors.

Experimental results indicate, that a distance threshold between 0.3 and 0.6 is most properly. A threshold beneath 0.3 cuts away too many correct correspondences and a threshold beyond 0.6 tolerates too many miss-matches.

Semi-local Constraints

The distance threshold for the spatial constraints depends on the image resolution. But in case of scaling, it should not be used at all. The threshold for both the angle and distance constraint corresponds to the maximal expected deviation of the object pose to the model image.

For the comparison of the intensity profiles the length of the profiles is set to 7 sub-pixels and the values are normalized to be invariant against illumination changes. The euclidic distance between the profiles is estimated and a threshold of 0.01 indicates correspondence.

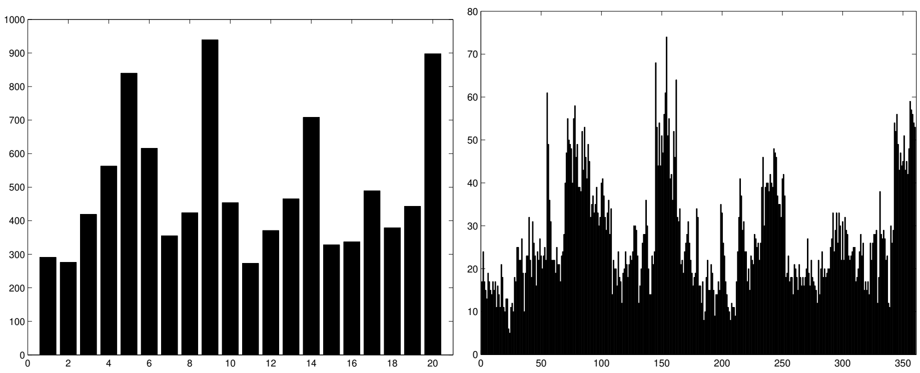
4.1.2 Recognition Rate of the Single Modules

After the optimal adaptation of these thresholds and constraints the success of the single recognition modules is examined. The results are presented in two different kind of diagrams. The one show the recognition rate, the number of correct recognitions per object as a result distribution of a complete set of test images. The second show a vote or probability distribution of a single query image over the set of model images to demonstrate the certainty of a single recognition.

In the following, every image of 20 degree rotation is used for the training. That means, the model database contains the descriptors of 1/4 of the images. Additionally, half of the images are used to separate the salient descriptors from arbitrary detections, as described in section 2.4. They are not used as model images, but only to elect the practical descriptors. Thus, each fourth image of every object image sequence is used for model descriptor extraction, each second image, preceding and following the model image, is used for controlling and the remaining images in between them for the testing. So, every object model is constructed from 18 sample images, considering the entire image database of 72 views per object.

The offset of the first image here and in the following experiments has a negligible influence on the result. For sure, the result distribution varies a little bit, but the total recognition rate does not change evidently.

The model database contains about 10000 descriptors of 360 images. That is an average of 28 descriptors per model image. Actually the number varies from 5 to 80 (see figure 4.3). Due to these large deviation in the descriptor distribution, the recognition based on the voting algorithm easily fails. Since every local descriptor correspondence gives a vote for the related model, the sparsely described models are plainly overreached by arbitrary miss-matches.



(a) model (b) pose

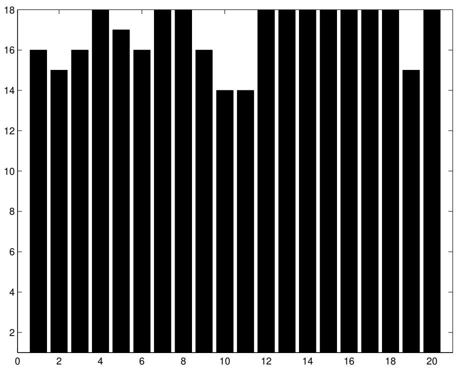
Figure 4.3: The distribution of model descriptors in the database. (a) shows the total amount of descriptors per model, and (b) the number of descriptors stored for every model image (eighteen per model). Thus, for example, the bars of 1-18 in (b) indicates the particular number of descriptors for every model image of object one and so on.

First of all, the accuracy of the differential invariants have been examined to estimate the optimal attributes for the interest point descriptors:

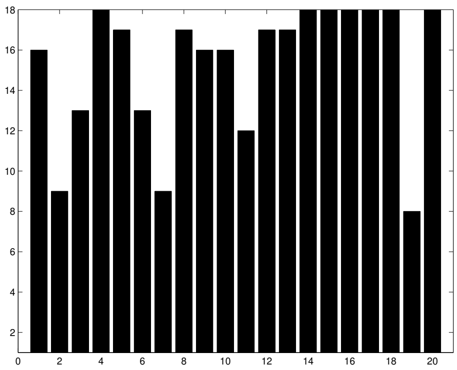
Differential Invariants

The experiments presented here, have been carried out with a different selection of invariants. Using more than third order differentiation does not improve the recognition. The best results have been obtained with the full nine-invariants descriptors (98.1% rate of recognition, restricted to the probabilistic model of local descriptors without any semi-local constraints).

To save calculation time, the number of descriptor attributes can be decreased. But that leads to a decrease of accuracy, too. Using e.g. only the five differential invariants down to second order derates the recognition to 93.6%. Just as the restriction on the four third order invariants (85%, figure 4.4).



(a) 0. to 2. order



(b) 3. order

Figure 4.4: Correct recognition results using different attributes for the local descriptor. (a) shows the result obtained with the five zeroth to second order differential invariants and (b) the results of the four invariants of third order.

The nine-invariants descriptors are used in the following.

Voting Algorithm

As you can see in figure 4.5, the results of the simple voting algorithm in (a) are quite poor. The recognition rate is about 73.3%. Neither the probabilistic model nor semi-local constraints have been considered. Comparison with the descriptor distribution in figure 4.3 makes it obvious, that the objects, described by a lot of model descriptors have the highest recognition rate.

Probabilistic Model

The extension of the voting method with the probabilistic model improves the results. The reasons are discussed in sec 3.3. Figure 4.5 (b) shows the (small) improvement, achieved by considering the ratio of the voting over the total distance of every descriptor correspondences (A recognition rate of 79.4%).

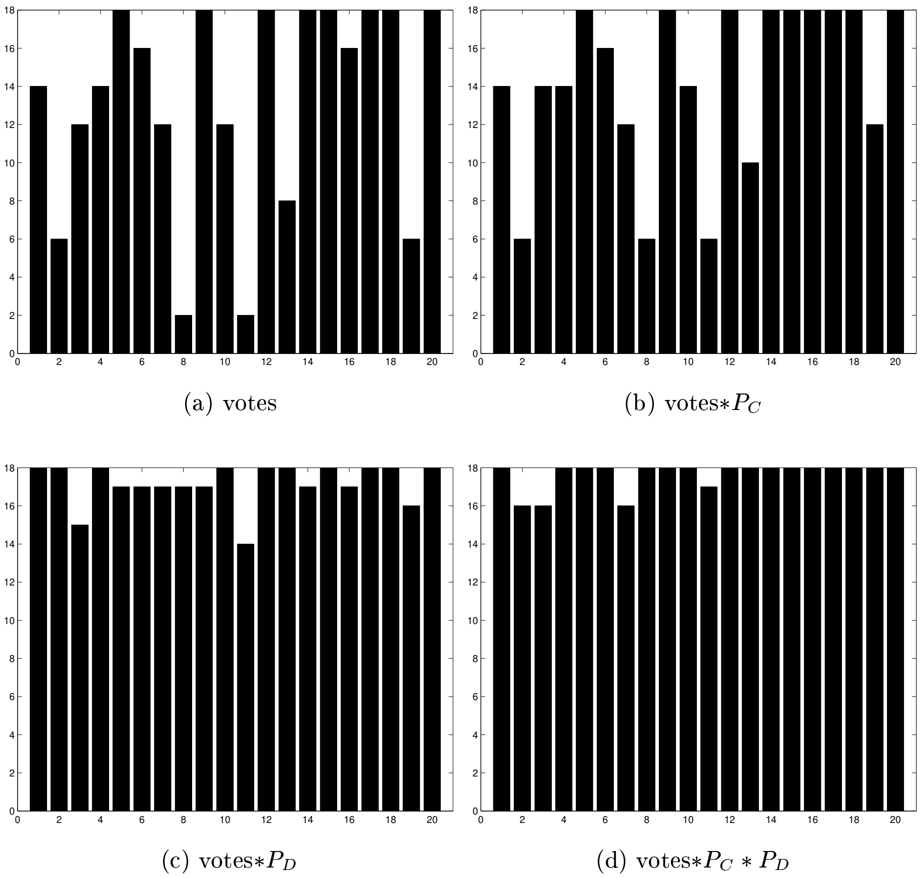


Figure 4.5: The diagrams show the number of correct results achieved by the recognition method restricted on local descriptor correspondence. (a) shows the result of the basic voting algorithm. (b) is the extension of (a) with the distance consideration P_C and (c) the extension of (a) with the descriptor distribution P_D. (d) shows the general improvement, achieved by the entire probabilistic model.

There against the ratio of the voting over the distribution of model descriptors (c) leads to an evident increase of recognition (95.6%). The number of correct results generally exceeds the voting method, but especially for the models, which only are poorly described.

The recognition rate for the entire probabilistic model is 98.1% (d).

Spatial Constraints

The spatial constraints evidently decrease the number of miss-matches. The effects of the different constraints can be seen in figure 4.6. The probability distribution concentrates on the correct model and especially the most similar pose. Now the correct model is easily recognized, but moreover the correct *pose* of the object can be estimated.

Intensity Profiles

Considering the intensity profiles between adjacent interest points is another semi-local image structure description. The results of this method are illustrated in figure 4.7.

Even though the intensity profiles do achieve a recognition result similar to the one of the spatial constraints, attention should be attracted to the fact, that the two methods fail at different object images. Examining the differences between the respective objects indicates, that the spatial constraint more likely fails at objects with a poor descriptor density, while the intensity profiles appear to be a good description of these homogeneous surfaces.

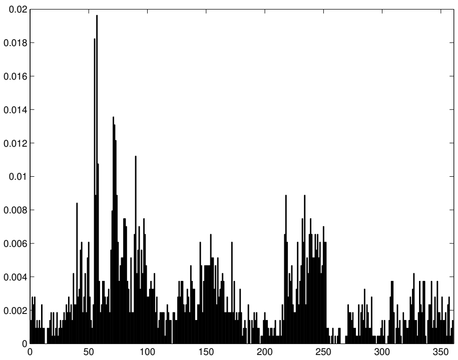
The intensity profiles, on the other hand, are not that suitable to describe very diversified surfaces, since the profiles might differ widely, due to small distance changes of the interest points. In case of a more homogeneous surface, distance changes would not affect the profile values.

This leads to the conclusion, that the intensity profiles and spatial constraints should complete each other. Instead of being implemented successively, they should work parallel. That means, that the two probability distributions, estimated of both methods, are multiplied. The achieved improvement of the results is a complete recognition.

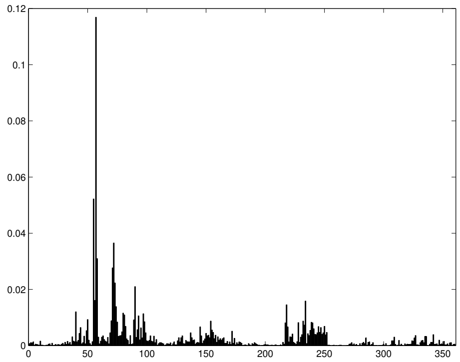
Successful Recognition

A perfect recognition of 100% of the test images is achieved, if every image of 20 degree rotation is used for the training, regardless of the offset. Then, as mentioned above, the model database contains the descriptors of 1/4 of the images. The entire probabilistic model of the local descriptors together with the semi-local constraints and the intensity profiles are used.

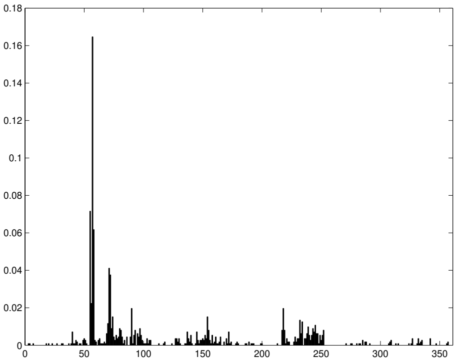
The recognition rate is still 97%, if 1/6 of the images are used for model descriptor and 92%, if 1/9 of the images are used.



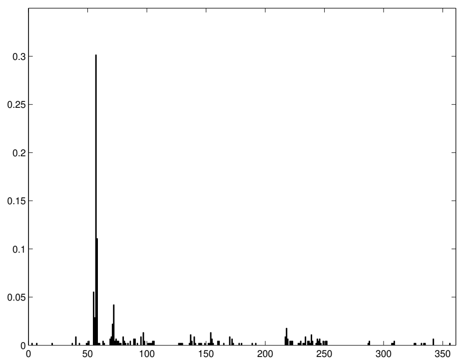
(a) descriptor based model



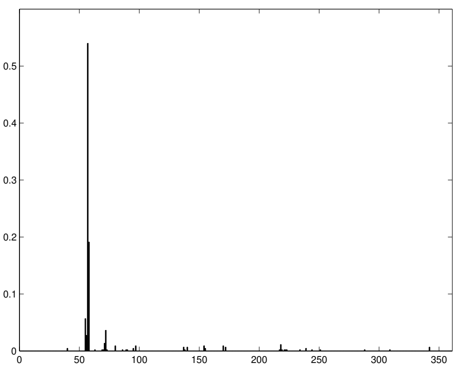
(b) local constraint: distance for n=2



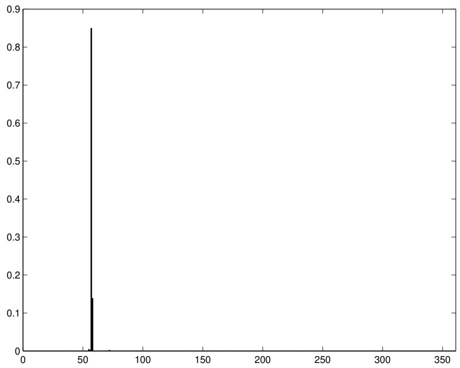
(c) local constraint: radius for n=2



(d) local constraint: radius & distance for n=2

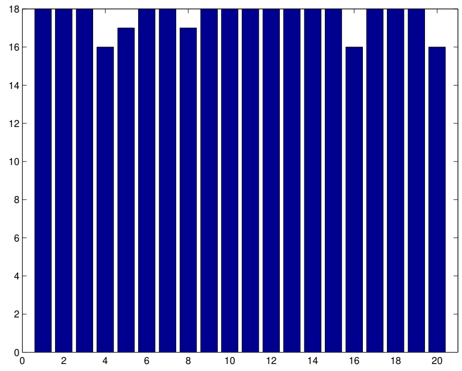


(e) local constraint, radius & distance for n=3



(f) full constraint model

Figure 4.6: The probability distribution of test image pose 11 of object no. 4. (The same image as in figure 3.3). Here, the improvement achieved by the local constraints is illustrated: (a) shows the simple voting model restricted to local descriptors. In (b)-(d) only pairs of correspondences, which fulfill the spatial constraints are considered. In (b) the distance between two points must not vary more than 10 pixels and in (c) the radius between two points must not vary more than 0.2 radian. Both constraints are combined in (d). In (e) these constraints are required for three pairs of correspondences instead of only two. Finally (f) shows the result of the complete probabilistic model of local and spatial constraints.



(a)

Figure 4.7: The recognition result based on the intensity profiles between interest point as image description. The recognition rate is 97.8%

In the latter case, the deviation of the test images to the model images is about 20 degree of spatial rotation.

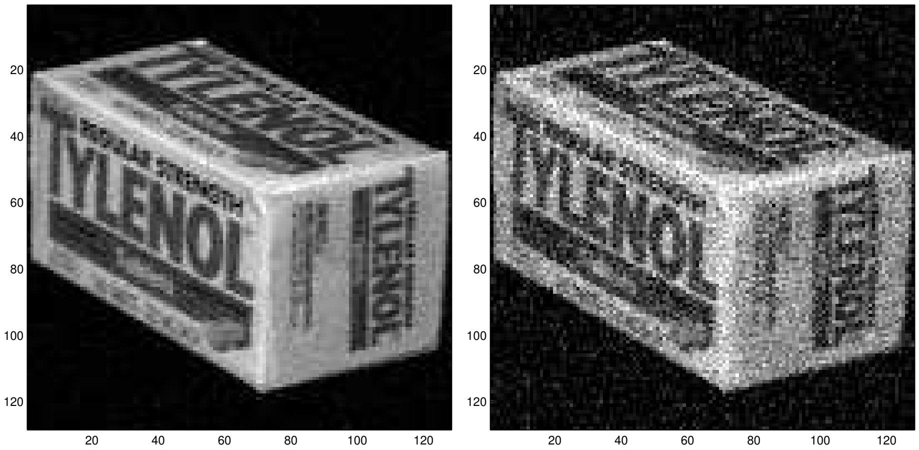
4.1.3 Experiments with Appearance Disturbations

The following experiments have been performed with the entire recognition method. The presented recognition rates are in proportion to the 100% recognition mentioned above.

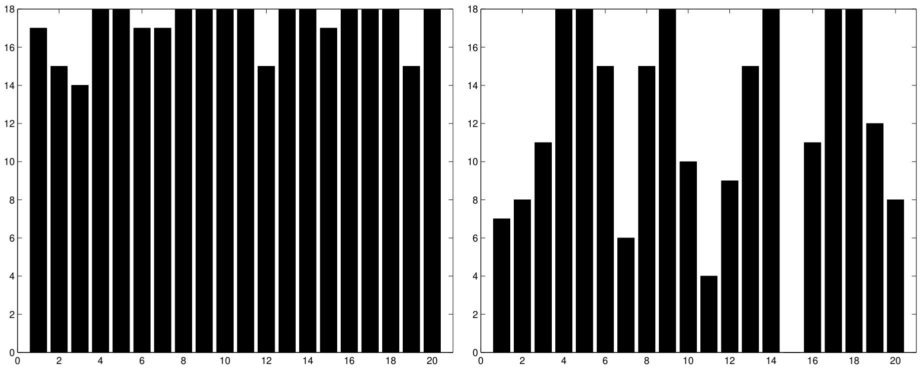
Noise

Gaussian white noise has been added to the test-pictures. The effect on the algorithm can be seen in figure 4.8.

The blurring effect of the Gaussian filter softens the noise and increasing σ counteracts the image perturbation. Unfortunately, σ also effects the scale of the image structure. Thus, modifying σ leads to a deviation in both location and description of the interest points. The minor noise in figure 4.8 (a) has little effect on the recognition, it is nearly compensated by the Gaussian filter. But the noise in (b) is evident and produces a distorting amount of mis-matches. Here, the intensity profiles are useless, since they are completely distorted by the noise. They worsen the results and must be omitted.



(a) Gaussian white noise, $\sigma^2 = 0.001$ (b) Gaussian white noise, $\sigma^2 = 0.01$



(c) recognition result, $\sigma^2 = 0.001$ (d) recognition result, $\sigma^2 = 0.01$

Figure 4.8: The effect of additive noise on the recognition result: (a) Gaussian white noise, zero mean, variance = 0.001 (b) Gaussian white noise, zero mean, variance=0.01. (c) and (d) show the recognition rate, 95.3% and 66.4%, respectively.

Rotation

Both the corner detection as well as the differential invariants should be rotational invariant. But this is not general valid for every arbitrary rotation.

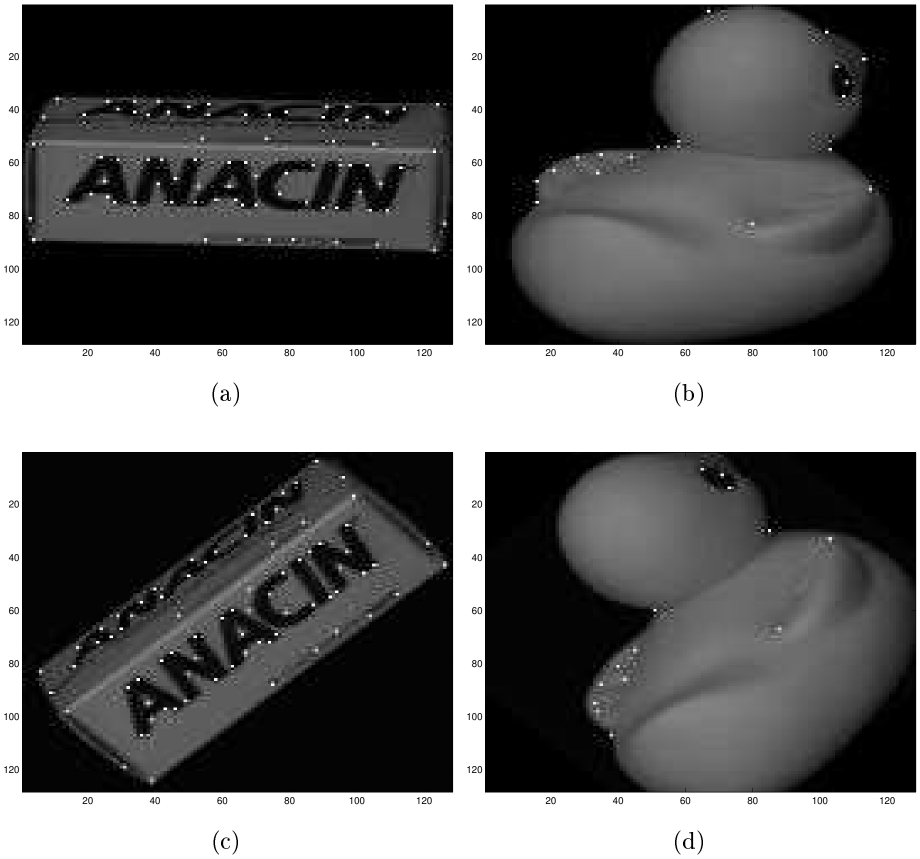


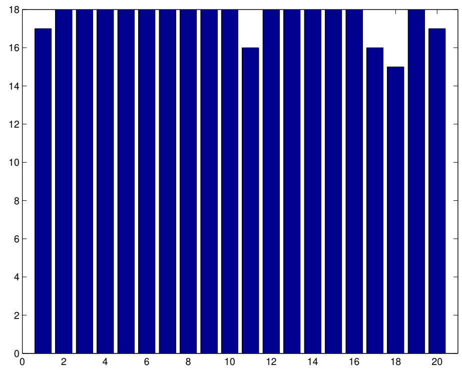
Figure 4.9: The different sensitiveness to object rotation. (a) and (b) show the original images with the detected interest points. (c) and (d) are the rotated images (bicubic interpolation). While the detections of (a) are repeated well in (c), they are displaced between (b) and (d).

For sure, a right-angled rotation of a square image does not change the intensity values of the perpendicular image structure. The pixels change their position but keep their individual values. As illustrated in the end of section 2.3 the attributes of the interest point descriptors stay in a constant manner.

An arbitrary (not perpendicular) rotation of angle α however causes a change of both position and intensity values of the pixels. At a sufficiently large scale or respectively fine resolution, these intensity changes would not affect the descriptor at all. Unfortunately the image structure of interest, namely corners are at pixel-scale. This means that they only stretch over a few pixels. Due to pixelization and quantization effects, the new intensity values might differ evidently from the former image. These differences cause localization errors and aberrant descriptor values.

Especially deviations in the corner localization might cause serious errors. Again, arbitrary rotated images of textured and cornered objects, described by a variety of salient descriptors are better recognized than the one of objects with homogeneous surfaces and a rotund shape (see figure 4.9). The intensity profiles are also affected by the rotation and deviates in their values, but the profiles on homogeneous surfaces are not affected substantially. Again they are acting as a good compensation for the errors caused by the local descriptors and spatial constraint. Because of the absence of sharp corners, the corner detection is somewhat unsteady. This error is intensified through the local descriptor and the spatial constellation.

Experiments show that a perpendicular rotation does not affect the recognition at all, while arbitrary rotations decrease the recognition rate of the objects with a poor descriptor density, as figure 4.10 demonstrates. The total recognition rate is still 97.5%.

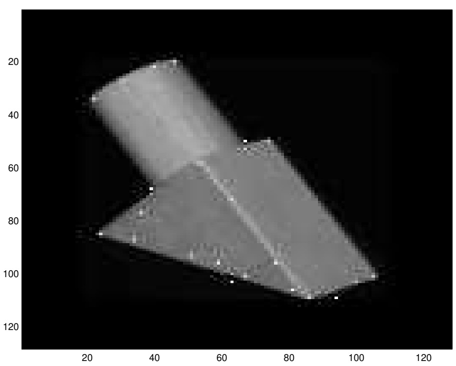


(a)

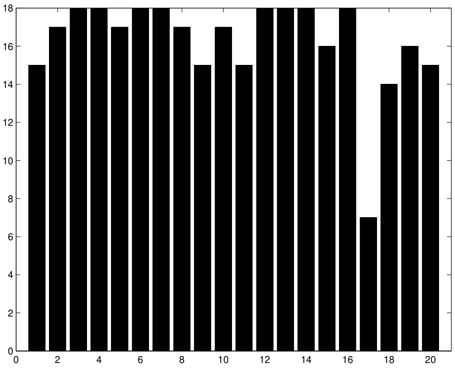
Figure 4.10: Arbitrary image rotation has only little affection on the model images with a few descriptors (e.g. 1,11,18). Recognition rate: 97.5%.

Scaling

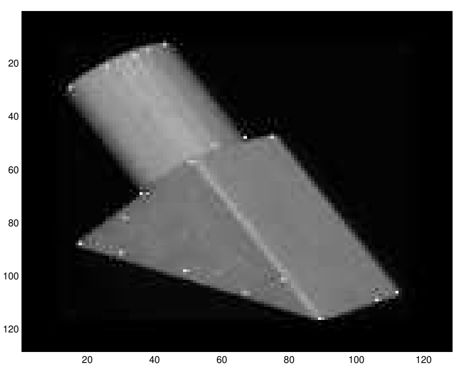
Another interesting aspect is the robustness of the recognition algorithm against variation of the image scale. For the tests, the images have been resized to scales of 0.5, 0.666 and 1.5 (see figure 4.11).



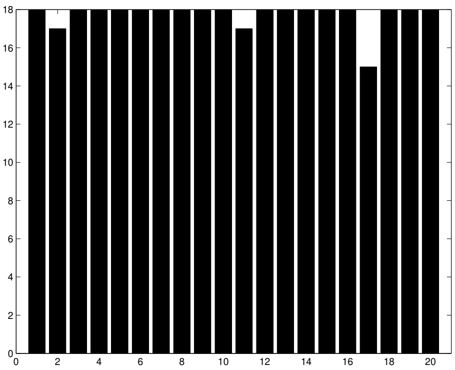
(a) $\sigma = 0.5$



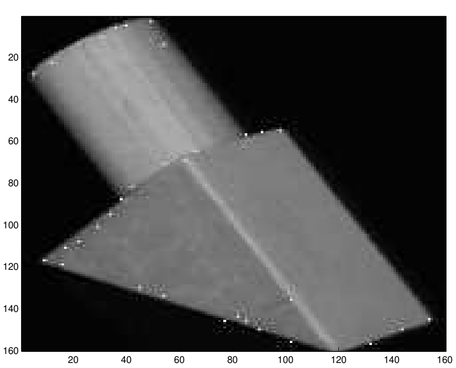
(b) $\sigma = 0.5$



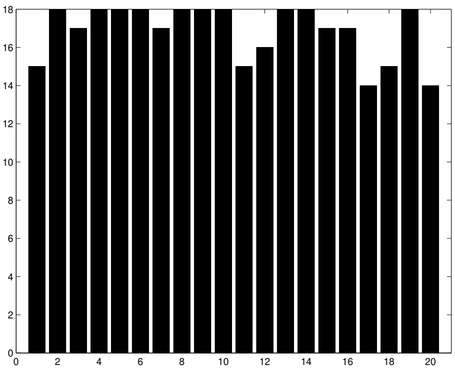
(c) $\sigma = 0.667$



(d) $\sigma = 0.667$



(e) $\sigma = 1.5$



(f) $\sigma = 1.5$

Figure 4.11: The robustness of the method against image scaling. The images have been scaled by the factors 0.5, 0.667 and 1.5 (a),(c),(e). The result distribution of the recognition is shown in (b), (d) and (f), respectively.

The results show little affection (98.6% recognition) in case of a size reduction of 2/3. But a reduction to half of the original size leads to a decrease of correct recognition to 90.3%.

On the other way, increasing the scale about 50% reduces the correct recognition to 93.6%.

The results indicate, that the local descriptors are not nearly not affected by scale changes of factor 2. Beyond that, the scale of the descriptor extraction must be adapted as described in section 2.3.4, which of cause, require a priori knowledge about the object size. Scale changes are irrelevant for the intensity profiles, due to their fixed length, the estimated sub-pixel values deviate minimally.

Occlusion

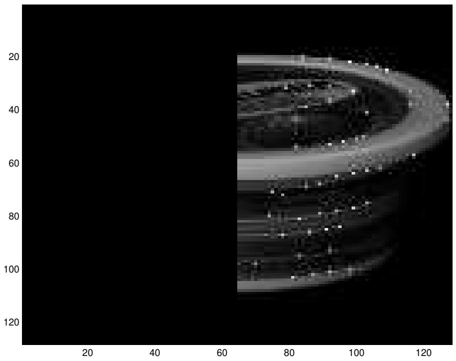
Objects might be occluded by others or their size might exceed the image borders; i.e. they might not be completely imaged. This case shows the advantage of the local characterization of the objects. As long as enough descriptors are detected and correspond to the training-data, the objects are still recognized correctly. To test the robustness against occlusion, only parts of the images are considered. The only problem are the unavoidable point detections at the new image borders. Cutting through the objects causes new corner detections near the cut, because of a dramatical change in the pixel neighborhood. Disregarding this area preserve a good object recognition, while including it produces errors (see figure 4.12).

Background

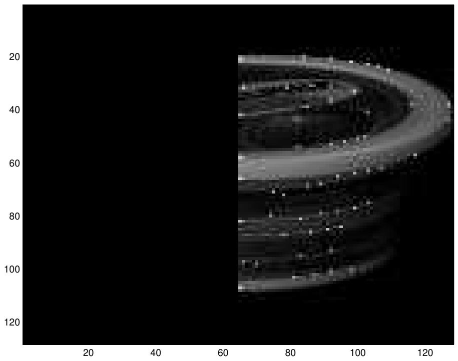
So far, the images contained only the isolated object. There was no background to consider; the remaining pixels were zero. But it would be more natural, if the objects had some kind of background, at least a table or whatever surface they lay on.

The robustness of the recognition against additive descriptors, which do not belong to the object, but are detected in the background, is tested. An example of the resulting image is shown in figure 4.13.

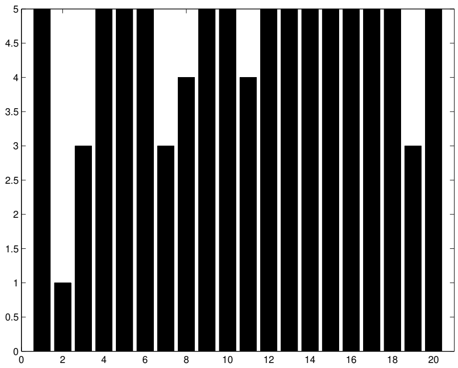
The interest point detections in the background have been increased gradually, corresponding to an intensity amplification of the background image. That means, the ratio of the intensity amplitude of the background over the intensity amplitude of the object image, denoted by I_B/I_{Obj}. The ratio in (a), for example, is 0.5/1. Thus, the intensity amplitude of the background values is 0.5, while the amplitude of the object values is 1. This leads to a few corner detections in the background, which have negligible influence



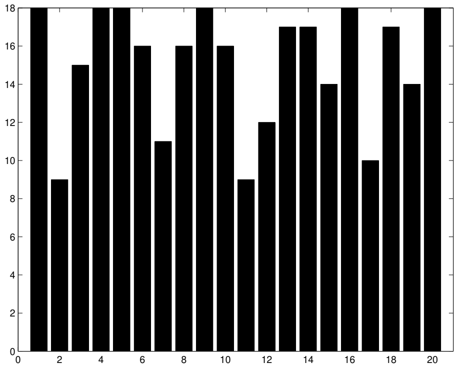
(a) 50% Occl. without boundary detections



(b) 50% Occl. with boundary detections



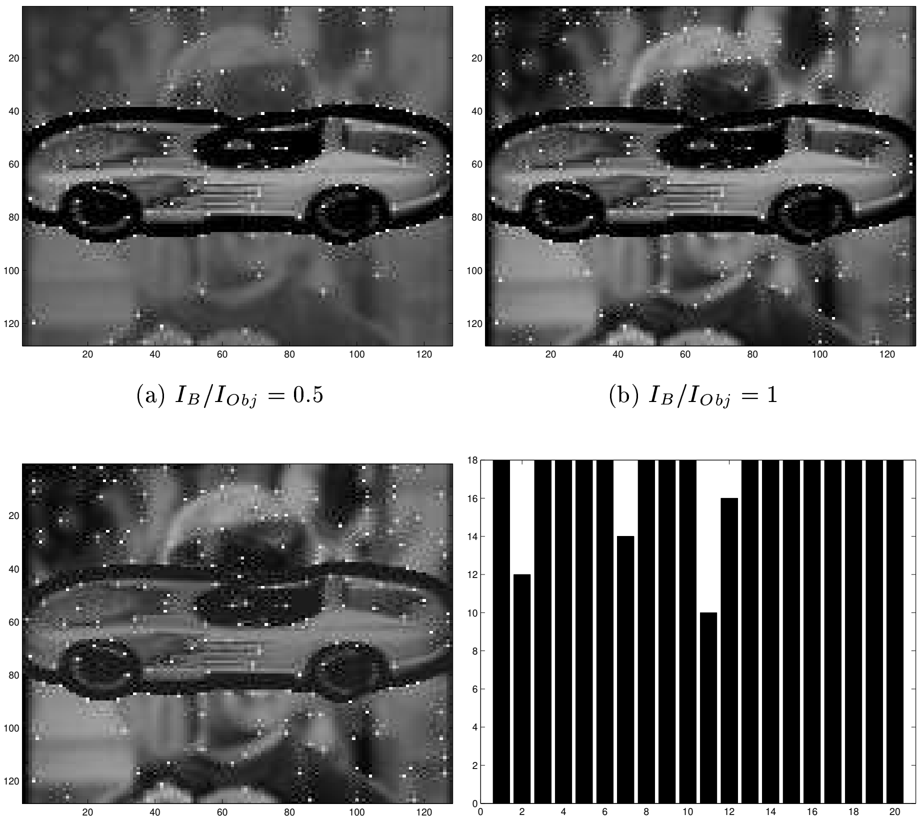
(c) 50% Occl. without boundary detections



(d) 75% Occl. without boundary detections

Figure 4.12: An example of object occlusion. (a) shows just half of an object image, the "cutting"-area is omitted in the corner detection. The recognition rate is still over 90%, which can be seen in (c). (b) illustrates the occurrence of false detections, due to boundary effects. In this case the rate is decreased to 68.9%. (d) represents the recognition rate of 75% occlusion, that means, that only 25% of the original image is used for recognition. The rate is still 83.6%

on the recognition rate. In (b) the amplitude is equal and the number of distorting descriptors, detected in the background is similar to the number of real object descriptors over the complete image. The recognition rate decreases unessential to 98.9%. Finally the ratio is 1.5/1 in (c) and the recognition is still 94.4%, even though the number of miss-detections outranges the number of real descriptors. The last result distribution is given in (d).



(a) $I_B/I_{Obj} = 0.5$

(b) $I_B/I_{Obj} = 1$

(c) $I_B/I_{Obj} = 1.5$

(d) result distribution of (c)

Figure 4.13: The disturbing effect of a background image: The objects are placed in front of a variably strong background image relative to the object intensity I_B/I_{Obj} in (a)-(c). The recognition rate is completely unaffected in the case of (a). In (b) there occur only four miss-matches (98.9%). In the case of (c) the intensity of the background is one and a half time larger than the intensity of the object. Still, the recognition is 94.4% as illustrated in (d).

4.2 Test Results

The various result distribution demonstrates at once the dependence of the recognition success from the number of salient descriptors. Since the complete method is based on the detection of corners, it fails easily at uniform objects without a diversified shape and surface. In the case of large homogeneous object surfaces, these *are* salient image structures considering the intensity profiles, but the corner detection is locally unstable and this error is reproduced through the entire recognition process. Furthermore, the number of detections is too poor to compensate the fuzziness of the descriptors.

On the other hand, the method works very well with complex object shapes and textures, which provide a lot of salient interest points. These objects are still recognized correctly, even though their appearance is strongly distorted (see figure 4.14).

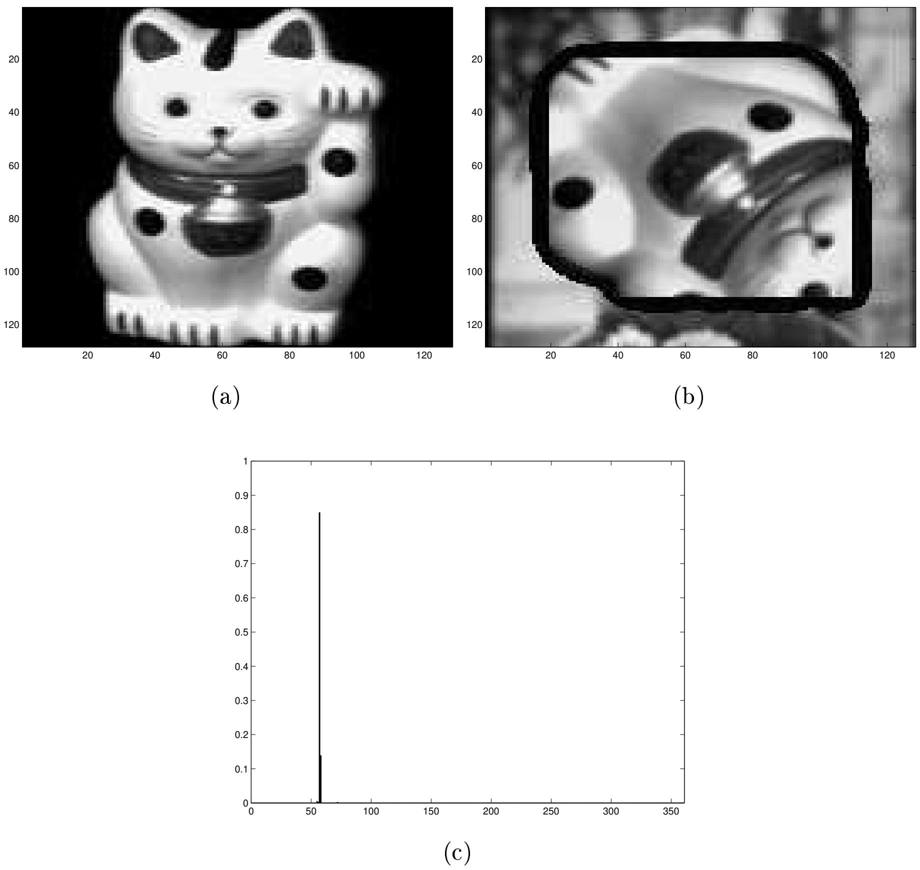


Figure 4.14: A combination of several appearance disturbations: The image in (a) is only partial visible, scaled, rotated and imaged in front of a background (b). Still, it is recognized correctly (c). Though, the recognition rate for the entire dataset is below 40% in this case.

Chapter 5

Conclusion

Due to the lack of perfection of current camera technology and the limits of computational power, computer vision abilities are very restricted. So, most important for any recognition method are the robustness and tolerance against errors and deviations. One of the most basic countermeasures to oppose the natural variety, is the individual adaptation of the method to the requested task of recognition.

Recognition derived from the maximum of a probabilistic model is the underlying principle for a good error tolerance. Local restriction of the object description and the independence among the descriptors enable also the assignment of object parts to the complete model.

This approach is based on several recognition techniques and combines their ideas to an efficient and error tolerant recognition method. The different model descriptors and their imposed constraints have all their strengths and weaknesses, but they reciprocally support each other.

The modular build-up of the algorithm enables the user to adapt the system to a specific recognition task and required error tolerance. The threshold values and the weight of each module for the entire probability estimation can be optimized for every task.

The first-level recognition is based on the very local interest point descriptors. Differential invariants are extracted from image structures at pixel-scale and stored as model features.

- The local descriptors are invariant against Cartesian coordinate transformation, that means translation and rotation. They can be adapted to scale changes with a multi-scale representation or an individual assimilation of σ respectively to the scale.
- They are conditionally invariant against Gaussian noise, since the extracted invariants are full contractions of Gaussian derivatives of local

image structures. The automatic blurring effect of the Gaussian kernel counteracts the noise.

- The descriptors can handle partial occlusion, as long as enough interest points are detected on the visual part of the objects. And in a similar way, the recognition is not affected by an obtrusive background. In the former case, some interest points are missing and in the latter, there are too many. In both cases the recognition is still working correctly, as long as the number of correct detections dominates over the arbitrary miss-matches.

The individuality and independence of each descriptor are the strength of this model. Considering the statistical distribution of correspondences gives a good error tolerance. Descriptors might deviate in both position and attributes, which causes wrong correspondences or the skip of practically correct correspondences. But it is not necessary and rather impossible to match every descriptor. The entire conformity of the object with the model is given, as long as most correspondences vote for the correct model.

The problem of the local description is the variety of object appearance. A single model cannot be derived from each individual object to recognize it in every possible appearance. It is impossible to derive descriptors from 2D-images, which are invariant against 3D-transformations of spatial objects in the real world. During the projection of the 3D-object on a 2D-image information is lost, which cannot be recovered. The more complex the shape and surface of the object, the more manifold is its way of appearance from different points of view. The only option is a manifold of model images, to be able to appeal every possible appearance. The descriptors are extracted from every model image and stored in a database as object model.

The beneficial side effect of this technique is the contemporary estimation of the object pose as required, if e.g. the object should be grab by a robot arm.

Doubtlessly every object recognition task could be solved, if the model database stored enough descriptors to reproduce absolute every possible object appearance. But due to the infinity of appearance variations, this is impossible. Therefor and to save computational time the amount of stored data should be quite sufficiently to accomplish the special task.

The intuitive formula of recognition rate and calculation costs has to be optimized. That means, reducing the redundancy of stored information until the limit of error tolerance is reached.

This is done during the training by varying the amount of images, the number of interest points per image and finally the number of differential invariants per descriptor. This information-"thinning" require a greater tolerance regarding the necessary similarity of a correct object-model correspondence. As long as these tolerance fields around the descriptors not intersect each other a correct recognition is guaranteed.

Since the conformity of the query object with the model is based on the statistical distribution of every individual descriptor correspondence, the deviation limit can be further extended. The occurrence of single miss-matches does not lead to a false result, as long as enough correspondences votes for the correct model.

On top of that considering the more global, spatial constellation of the interest points of an object ensures the result. Different constraints toward this constellation can be posted. The test results showed, that imposing simple angle and distance thresholds between interest point correspondences decrease evidently the number of miss-matches. The approach of Matas ensure the matching in a comparable manner. The combination of both constraints provides the best results.

Possible Extensions

In the beginning of this thesis recognition techniques were divided into two groups. One using geometrical features of the object and the other using only image structures without any constructional knowledge. But the two different approaches could also be combined.

The recognition approach, presented here, is based on the extraction and comparison of interest point descriptors. The interest points are object corners. Thus, they are also geometrical features of the objects and their spatial constellation could e.g. be connected to a simple graph structure. This gross model would describe the geometrical shape of the object.

In this way the geometrical features could extend the overall object description. Combined with a stereo camera the system could derive the 3D-world coordinates of the corners and calculate a simplified structure model with exact measures. The corner descriptors could be used as a preselection of possible models. The descriptor correspondences and the spatial constellation of the matched corners would reduce efficiently the manifold of possible object poses. In combination with a graph algorithm the system could quickly estimate, whether or not the object would fit the model shape or vice versa, if it were possible to project the model shape in a way, that would fit the object image.

The variety of recognition tasks and approaches to solve them let conclude, that there is no “best” method. The different techniques could be implemented into modules, as already done in this approach. A modular structure of a probabilistic model containing a weighted coalition of every possible image feature description is an well adaptive and robust tool.

Appendix A

Program Structure

The program structure is given in the diagrams of Fig. A.1 and A.2, respectively for the training and testing process.

Training

The training program to calculate the model descriptors and to build the database is split up into several program files.

The main procedure *training.m* loads the training images one by one and uses the sub function *calc_desc.m* to estimate the descriptors of each image. *calc_desc.m* calls the function *corner_detection.m* to get the coordinates of the interest points in each image.

With a complete set of the descriptors of three images in a row as input, the sub function *matching.m* reduces the number of descriptors. Only the strong ones that are repeated in all 3 images are returned to the main program.

Finally, as a look-up table for the semi-local constraints and intensity profiles *find_desc_neighbors.m* estimates the n nearest neighbors of these remaining descriptors. The descriptors are then inserted into the database by *insert_desc.m*.

Testing

The test program is also divided into several modules.

testing.m loads a query image from the training set and calls the same sub functions (*calc_desc.m* and *corner_detection.m*) to calculate the image descriptors as the training program.

These descriptors are compared to the model database. *range_query.m* returns every database entry similar to one of the descriptors. The spatial

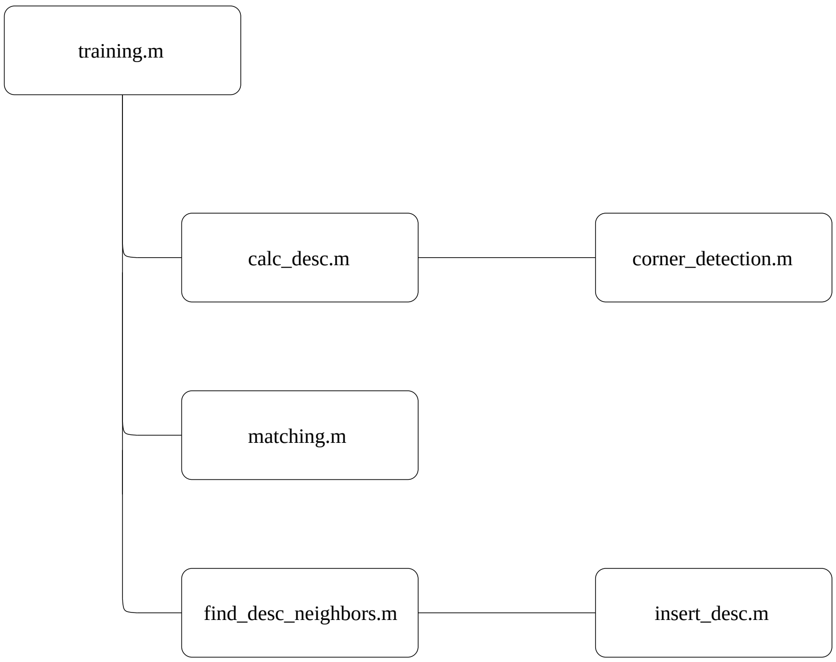


Figure A.1: The structure of the training program

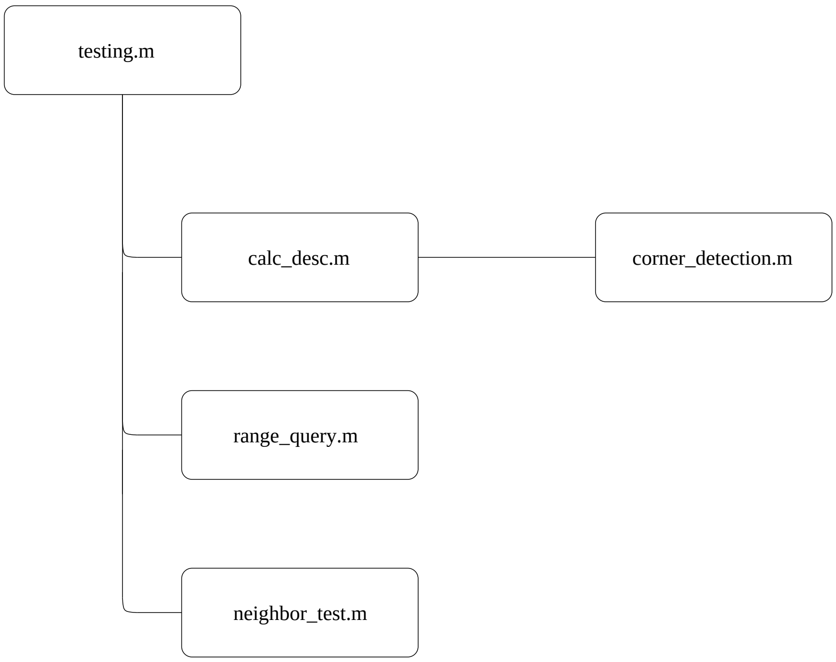


Figure A.2: The structure of the test program constraints and intensity profiles for every correspondence is then checked by *neighbor_test.m* and the remaining number of correspondences, the final votes for every model, is returned.

Bibliography

[1] N. Duffy. Overview of apperance based methods in computer vision. Technical report, INRIA Rhone-Alpes, 1999.

[2] L.M.J. Florack. *The syntactical Structure of Scalar Images.* PhD thesis, Universiteit Utrecht, 1993.

[3] L.M.J. Florack, B.M. ter Haar Romeny, J.J. Koenderink, and M.A. Viergever. General intensity transformations and differential invariants. *JMIV*, 4(2):171–187, 1994.

[4] C. Harris and M. Stephens. A combined corner and edge detector. *Alvey Vision Conference*, pages 147–151, 1988.

[5] D. Hilbert. Über die vollen Invariantensystemen. *Math. Annalen*, 42:313–373, 1893.

[6] B. Jähne. *Digital image processing.* Springer-Verlag, Berlin, third edition, 1995.

[7] J.J. Koenderink and A.J. van Doorn. Representation of local geometry in the visual system. *Biological Cybernetics*, 55(6):367–375, 1987.

[8] J. Matas, J. Burianek, and J. Kittler. Object recognition using the invariant pixel-set signature. In *Proc BMVC*, volume 2, pages 606–615, London, UK, September 2000.

[9] H. Murase and S.K. Nayar. Visual learning and recognition of 3-d objects from appearance. *IJCV*, 14:5–24, 1995.

[10] J. Pauli. *Verarbeitung und Verstehen von Bildern*, chapter 8, pages 219–259. B. Radig, München, 1993.

[11] S. K. Nayar S. A. Nene and H. Murase. Columbia object image library (COIL-20). Technical report, Columbia university, February 1996.

[12] H. Samet. The quadtree and related hierarchical data structures. *Computing Surveys*, 16:187–260, 1984.

[13] R. Schalkoff. *Pattern Recognition - Statistical, Structural and Neuronal Appraoches.* John Wiley & Sons, New York, 1991.

[14] C. Schmid. A structured probabilistic model for recognition. *Computer Vision and Pattern Recognition*, 2:485–490, june 1999.

[15] C. Schmid and R. Mohr. Local grayvalue invariants for image retrieval. *T-PAMI*, 19:530–535, 1997.

[16] C. Schmid, R. Mohr, and C. Bauckhage. Comparing and evaluating interest points. In *ICCV 98*, pages 230–235, 1998.

[17] M.J. Swain and D.H. Ballard. Color indexing. *IJCV*, 7:11–32, 1991.

[18] Suk T. and Flusser J. Convex layers: A new tool for recognition of projectively deformed point sets. In Franc Solina and Ales Leonardis, editors, *Computer Analysis of Images and Patterns: 8th International Conference CAIP'99*, number 1689 in Lecture notes in Computer Science, pages 454–461, Berlin, Germany, September 1999. Springer.

[19] M. Tarr and S. Pinker. Mental rotation and orientation-dependence in shape recognition. *Cognitive Psychology*, 21:233–282, 1989.

[20] B.M. ter Haar Romeny, L.M.J. Florack, A.H. Salden, and M.A. Viergever. Higher order differential structure of images. *IVC*, 12:317–325, 1994.

[21] M.A. Turk and A.P. Pentland. Face recognition using eigenfaces. *CVPR*, pages 586–591, 91.

[22] O. Vornberger and O. Müller. Datenbanksysteme, Vorlesung SS 2001. http://www-lehre.informatik.uni-osnabrueck.de/∼dbs/2001/skript/node34.html.

Acknowledgements

I am grateful to all those who contributed to this diploma thesis, especially to my supervisor Josef Pauli here in Kiel and my second reviser Jens Arnspang, who invited me to write this thesis mainly in Copenhagen. I appreciate the assistance of other professionals of the computer vision groups of Kiel and Copenhagen.

Thanks to the language skills of Nadine Hansen and the final reading of Carsten Frandsen and after all the encouragement of my friends and especially the patient support of my parents.

Erklärung:

Hiermit versichere ich, daß ich die vorliegende Arbeit selbständig verfaßt und ausschließlich die angegebenen Hilfsmittel und Quellen verwendet habe.

Kiel, den