Philipp PAIER

Computer-Aided Classification of Butterflies using Shape and Appearance

DIPLOMARBEIT

zur Erlangung des akademischen Grades eines Diplom-Ingenieur

Diplomstudium Technische Mathematik

Graz University of Technology
Technische Universität Graz

Betreuer:
Univ.-Prof. Dipl.-Ing. Dr.techn. Horst BISCHOF

Institut für Maschinelles Sehen und Darstellen

Graz, im März 2012

Eidesstattliche Erklärung

Ich erkläre an Eides statt, dass ich die vorliegende Arbeit selbständig verfasst, andere als die angegebenen Quellen/Hilfsmittel nicht benutzt, und die den benutzten Quellen wörtlich und inhaltlich entnommenen Stellen als solche kenntlich gemacht habe.

Graz, am ....
(Unterschrift)

Statutory Declaration

I declare that I have authored this thesis independently, that I have not used other than the declared sources/resources, and that I have explicitly marked all material which has been quoted either literally or by content from the used sources.

....
date (signature)

Abstract

The Natural Museum of History in Vienna is currently working on assembling digital image databases of their large insect collections. The purpose of this thesis is to provide and evaluate a system that makes use of such databases to ease the work of entomologists during the process of identifying new specimens. The work is divided into two smaller projects each concerning itself with the identification of butterflies. For the first project microscope scans of male genital organs are used to compare owl moths. Relational indices - measurements, that are currently determined manually by entomologists - motivate a semi-automated shape based classification approach. Structural measurement descriptors or Shape Context are used to describe certain parts of the genital organ, and a candidate list of species is calculated according to the distance between the description of a query and a groundtruth specimen. The second project concerns itself with the classification of butterflies based on the inner appearance of their wings. Color histograms and SIFT descriptors are used with a variety of region of interest detectors to extract image features. The use of spatial pyramids is proposed to incorporate spatial information and classification is done using vocabulary trees.

Evaluation for both projects is based on two different datasets respectively and recognition rates up to 90% are achieved, depending on the specific task.

Kurzfassung

Das Naturhistorische Museum in Wien arbeitet zurzeit an der Zusammenstellung von digitalen Bilddatenbanken ihrer reichhaltigen Sammlungen an Insekten. Das Ziel der vorliegenden Arbeit ist es, ein System zu entwickeln und zu evaluieren, das den Identifizierungsprozess neuer Insekten für Entomologen erleichtert. Die Arbeit ist in zwei Teilprojekte aufgeteilt, wovon sich jedes mit der Identifizierung von Schmetterlingen befasst. Für das erste Projekt werden mikroskopische Aufnahmen der Genitalorgane männlicher Eulenfalter benutzt, um Exemplare miteinander zu vergleichen bzw. voneinander zu unterscheiden. Relational Indices - Messverfahren, die von Entomologen üblicherweise zur Artunterscheidung verwendet werden - motivieren zu einem formbasierten Ansatz der Klassifizierung. Structural measurement Deskriptoren sowie Shape Context werden verwendet, um die Organe zu beschreiben und anhand der Distanz von Deskriptoren wird für ein neues Exemplar eine Liste für in Frage kommende Spezies erstellt. Das zweite Projekt befasst sich mit der Klassifizierung von Schmetterlingen anhand der Farbe und der Muster ihrer Flügel. Farbhistogramme und SIFT Deskriptoren werden zusammen mit einer Auswahl an Region of Interest Detektoren verwendet, um lokale Bildregionen zu beschreiben. Zudem werden Spatial Pyramids zur Einbindung von Ortsinformationen benutzt und Vocabulary Trees werden für die Klassifizierung verwendet.

Die Evaluierung beider Projekte wird anhand von jeweils zwei Datensätzen durchgeführt. Abhängig von der exakten Aufgabenstellung werden dabei Erkennungsraten von bis zu 90% erreicht.

Acknowledgments

First I would like to thank Univ.-Prof. Dipl.-Ing. Dr. techn. Horst Bischof, as well as Dipl.-Ing. Hayko Riemenschneider for their scientific and reliable advice during the work on my thesis.

Furthermore many thanks to Mag. Dr. Martin Lödl and his team at the Natural Museum of History in Vienna for their kind support and biological background information.

I especially like to express my gratitude to my family for their encouragement and the opportunity to pursue my studies.

Finally I would like to thank all those, who supported me in one way or the other in the last several years.

Contents

Chapter 1

Introduction

The *Museum of Natural History* in Vienna has large and constantly growing collections of all kind of insects, ranging from grasshoppers and bugs to flies. Currently the museum is working in an ongoing process on digitalizing those collections to obtain an image database, that represents the museum - the *Collection Austria*. Such a database can be used for commercial purposes, expositions and for comparisons with other collections. However, the question arises if such a database can also be used to help experts, known as entomologists, in their work. One of the typical tasks is the identification of insects new to the collection or even new to science. Given a new specimen, an entomologist tries to determine what kind of insect he is examining. For that purpose insects and organisms in general are grouped together to form so called taxa. Figure 1.1 illustrates such a taxonomical hierarchy for organisms in general, where insects would correspond to one specific class. The criteria, that are used for grouping, typically depend on the *rank* in the hierarchy and range from visual appearance to a specimens fecundity with other samples. While it is even for non-experts easy to distinguish flies from butterflies (those are in fact of different order) on first sight, it can be a very hard task, even for an entomologist, to determine genus and species of a specimen. Additionally sometimes the question arises, if a specimen is of a yet unknown species, or a group, that once formed a genus, should be partitioned further.

In a typical identification procedure, entomologists unsurprisingly make use of literature, that offers illustrations and textual descriptions to stepwise identify a specimen. This can be a very time-consuming kind of work, depending on the underlying species and on the knowledge of the expert with this particular group. With digital image databases for insects an opportunity arises to make use of computers to help entomologists in this identification procedure. Given an image of an unknown specimen, then the task is to automatically find those samples in a groundtruth database of known specimens, that are the most similar. In this thesis we propose the

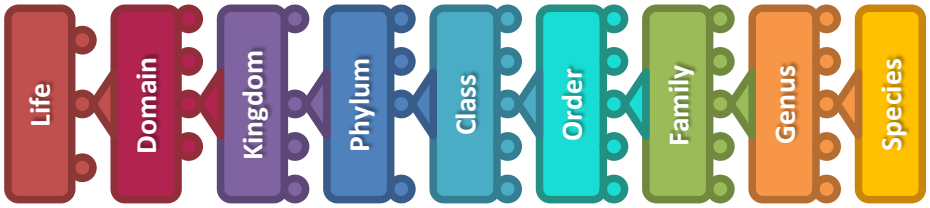


Figure 1.1: Sketch of a taxonomical hierarchy for organisms, based on 8 major ranks. From left to right, each group of one rank is divided into several groups of the next rank. E.g. the class of insects consist of approximately 30 different orders. One of those is the order of Lepidoptera (butterflies and moths), which itself is divided into several families. Note, that there might also exist ranks to further divide one major rank. E.g. a family can be partitioned into subfamilies.

use of proven methods from computer vision to implement such a system. For that reason we mainly concentrate on determining the species of butterflies and moths and ignore any other kind of insects. Two very common ways to distinguish butterflies are based on their genital organs and their wing patterns. Therefore, we divide our work into two smaller projects.

The first project concerns itself with the classification of male owl moths based on microscope scans of their genital organs: Copulatory organs in butterflies and moths turned out to be of primary systematic importance. Male and female operate their genitalia as a sort of lock-and-key system. The genital organs exhibit complex lock-and-key tools like valves, knobs and spines and therefore prove to be of special diagnostic importance on specific level. The main feature of those organs is their shape. This is why we propose the use of proven shape based methods to describe and compare certain parts of the genital organ. The second project concerns itself with the classification of butterflies based on their wing patterns. In this case texture and color are the dominant features, and we focus on typical appearance based methods to compare samples. In Chapter 2 we discuss work from computer vision that is related or similar to our work. In Chapter 3 and Chapter 4 we present our solutions for the shape and the appearance based projects respectively. Each of those two chapters also consists of individual sections of related work, experimental results and a conclusion for the respective project. Finally, in Chapter 6 we summarize the insight gained during our work and discuss possible fields for future work.

Chapter 2

Related Work

The problem of automatically recognizing one or several objects in an image, is well known in computer science. Depending on the task at hand, one might e.g. be interested if an image contains a certain object. We then speak of an object recognition problem in general. If one is also interested in the exact location of the object in the image, we typically speak of object detection. However, the problem of determining the species of a butterfly based on an image, that shows that specimen, is a typical image classification or categorization task. In image classification the goal is to assign an image to a certain group of known objects, like cars or people. Assigning a butterfly to a certain *taxon*, therefore exactly corresponds to such a problem statement. In the following we discuss the basic principles of classification as well as some typical approaches and work that is related to ours in one way or another.

2.1 Image Classification Principles

Basically we distinguish between two models of classifiers: generative and discriminative models. A generative model tries to estimate the probability $P(L|I)$ for the image I being of class L, by using training data of labelled objects to determine the joint probability $P(I, L)$ and the likelihood $P(I|L)$. $P(I, L)$ and $P(I|L)$ can then be used together with the Bayes' rule to define $P(L|I)$. Discriminative models on the other hand directly learn $P(L|I)$ or a mapping function $L = f(I)$ from the trainings data to assign a label to an image. We like to note that one very important aspect in such a scheme is how to represent the image I. A very simple possibility e.g. would be to use raw pixel values of the entire image. More subtle approaches are based on the detection of characteristic regions in the image and describing them in a suitable mathematical manner. The representation of choice is highly dependant on the task to be solved and there exists no general representation that is optimal for all kind of objects. We already mentioned that we

divided our work into two projects. This is done, because of exactly this circumstance. Genital organs are best described in terms of their shape, thus the representation of the organ part shown in an image I has to incorporate shape information, while the representation of wing pattern images should incorporate texture and color information. Therefore we discuss possible representations in the related work sections of the respective projects and concentrate in this chapter on important principles in general.

2.1.1 Nearest Neighbour and KD-Trees

The most basic discriminative classifier is called nearest neighbour classifier. A new query image I is represented by a vector v of fixed length. Distances to representations v_i of labelled images in a groundtruth database are calculated according to a suitable distance measurement d. The mapping function, that assigns a label to the query image is then given by $f(v) = l_m$, where l_m is the label of the training image with index $m := \arg\min_i d(v, v_i)$. Determining the minimum distance efficiently is therefore the main problem when dealing with large training sets. Friedman et al. [29] proposed the use of a kd-tree for efficient nearest neighbour search. A kd-tree is built by splitting the trainings data into two parts according to the component medians of the data. For each part this process is repeated until the final leaves hold the representations of the training images. Such a tree can then be used for an efficient nearest neighbour search. First the query vector *travels* along the tree, until it reaches a leaf holding an approximate nearest neighbour in the trainings data. Then, the path to the leaf is backtracked to find probably better neighbours. Because large fractions of the tree have to be considered during backtracking, when the dimensionality of the trainings data is high, kd-tree nearest neighbour search becomes rapidly time consuming for increasing dimensions. In fact, it has been shown that for more than ten dimensions an exhaustive nearest neighbour search would be as efficient.

A speedup can be achieved according to the best bin first strategy proposed by Beis and Lowe [4]. This strategy is based on kd-trees, but instead of searching for the nearest neighbour, they search for the approximate nearest neighbour to limit the search time. Basically the backtracking step of the kd-tree algorithm is done according to a priority queue based on closeness, and only a fixed number (200 in [4]) of leaves are considered as nearest neighbour candidates. As soon as that number has been reached, the algorithm is stopped, and eventually the correct nearest neighbour is not one of the considered leaves. However, in about 95% of the time this procedure finds the correct nearest neighbour, and in the other cases at least finds a good approximate to the nearest neighbour.

2.1.2 Support Vector Machines

Another frequently used discriminative classifier is the support vector machine (SVM) [96]. Given labelled trainings data, the goal is to separate two classes by the hyperplane that maximizes the margin between them. This is stated as an optimization problem an the resulting hyperplane then serves as decision boundary. Simply spoken query data is assigned a label, depending on which side of the hyperplane it belongs to. The decision function is based on inner products $\odot$ of the query vector x and the training vectors x_i. To incorporate more than two classes, an SVM is built for each class to separate it from the rest and labels are assigned according to the highest response of one SVM.

For SVMs to work out great, the data has to be linear separable. Unfortunately this assumption doesn't always hold true. However, there exists a workaround to overcome this obstacle, by mapping the data to a higher dimensional space, where linear separation can be done. Given such a mapping φ, the decision function is then based on inner products of $\varphi(x)$ and $\varphi(x_i)$. Instead of actually computing the mapping function for a query vector x, the kernel trick can be applied. Therefore a kernel K is defined such that $K(x_i, x_j) = \varphi(x_i) \odot \varphi(x_i)$ for the trainings data, and K replaces the inner product in the decision function [12].

The usage of SVMs is widespread in computer vision and pattern recognition in general. E.g. Chapelle et al. use color histograms to describe images by vectors and SVMs for classification tasks on Corel datasets of 7 and 14 categories [16]. Csurka et al. [18] use SVMs together with visual vocabularies to recognize seven classes. They showed, that SVMs outperform a Naïve Bayes approach. More recently Lin et al. [49] proposed a fast training approach for SVMs to perform large scale image classification on the *ImageNet1000* dataset of 1000 classes and achieve state of the art recognition rates.

2.1.3 Visual Vocabulary

In text retrieval, documents are represented by the frequency of words. Inspired by that principle, Sivic and Zisserman proposed [86] the usage of so called *visual words*. After image features, as we will discuss them in later chapters, are extracted from training images, kmeans clustering is used to determine a visual vocabulary. The visual words of such a vocabulary are therefore the centres of the clusters, and then an image can be represented by the frequency of occurring visual words in form of a histogram. Instead of using raw frequencies, weights are applied to each component of the histogram. In [86] this is done according to its counterpart from text retrieval, where so called *'term frequency-inverse document frequency'* (tf-idf) is used as weighting scheme. Therefore weighted frequencies are products of two

terms. The first term (term frequency) benefits words that are very commonly used in the text at hand, and are thus very representative for the text. The purpose of the second term (inverse document frequency) on the other hand is to weight words less, when they are common in general, like e.g. noun markers.

Sivic and Zisserman [86] use the principle of visual words and tf-idf to retrieve user marked objects in various frames of an entire movie. They calculate about 200.000 features from a subset of all frames and use the Mahalanobis distance for clustering to obtain the vocabulary. For object retrieval, they use stop list to suppress common visual words and additionally evaluate the spatial consistency. An inverted file structure (ifs) is used to speed up retrieval, where for each visual word the occurrences in each frame are stored. A query object is given by a subpart of one frame and compared to other frames, using the respective weighted frequency vectors and the normalized scalar product between them. The ranking of the frames is then given according to those products.

We would like to note that the principle of clustering features to obtain a vocabulary is also often termed *codebook generation* or *bag of features*, depending on the analogy that is desired. Also Leung and Malik [48] and Malik et al. [58] use that principle together with filter responses to model the atoms of human texture perception and refer to them as *Textons*.

2.1.4 Vocabulary Tree

In order to use large visual vocabularies, while still achieving fast recognition times even for large image databases, Nister and Stewenius [71] proposed a hierarchical structure together with visual words. Kmeans is used to cluster the trainings data, and this step is repeated recursively for the resulting clusters. The final result is the so called vocabulary tree with a fixed number of levels, where each leaf/path corresponds to a visual word. The benefit of such a tree is, that it serves as visual vocabulary as well as an efficient search procedure simultaneously. Thus the tree allows for faster lookup of visual words and therefore also for the use of larger and more discriminative vocabularies. Recognition is basically a hierarchical version of the original visual vocabulary. The importance of certain visual words are defined by assigning weights to leafs and nodes in the tree, according to a tf-idf scheme. For efficient scoring to classify new images, they also use inverted files for every node.

Among their experiments in [71], they achieved realtime recognition for CD covers when using an image database of 40000 CD covers. They also tested their work for databases containing up to one million training images and vocabulary trees with up to 16 million leafs and achieved sub-second retrieval times in those experiments.

2.1.5 Spatial Pyramid Matching

A drawback of the above discussed visual vocabularies is, that they don't incorporate the spatial relations between instances of visual words in an image. A histogram only gives information how often features occur and generally not where they can be found. Therefore, Lazebnik et al. [46] proposed the use of pyramid matching along with visual vocabularies. An image is partitioned into several rectangular subregions, where each can recursively be subdivided again. This results in a spatial pyramid representation of the image. Then for each subregion at each layer of the pyramid, visual word frequencies are calculated, and an entire image is represented by all those histograms, which are additionally weighted to define the importance of certain subregions.

In [46] they then use multiple SVMs for the actual classification task on three different large-scale datasets. The results showed, that the spatial pyramid representation gives an improvement over orderless image representations, when dealing with global scenes or non-cluttered and few deformable objects in canonical positioning.

2.1.6 Random Forests

Random forests are a commonly used classification scheme based on decision trees, see [2, 47, 69, 84] for some examples. The basic principle of such a tree is to stepwise reduce the number of possible candidate classes based on decisions made at every node of the tree. The term random forest is based on how such trees are built. First, the trainings data is partitioned into several subsets of randomly chosen samples, and for each subset a decision tree is built independently. For each node in the tree, the features, that are used for the splitting criteria, are pulled randomly too. The benefit of using several smaller trees (a forest) instead of one large tree, is the decrease in runtime and memory requirement. Each tree can then be seen as a classification hypothesis on its own.

The possible fields of work for random forests in computer vision are wide-ranging. Amit and Geman first used binary, randomized trees for digit recognition in [2]. Later Lepetit et al. [47] make use of randomized trees for recognizing keypoints. Therefore they state keypoint recognition as a classification task and use the support region of a keypoint to calculate simple features for splitting. In a different, context Moosmann et al. [69] proposed randomized trees instead of kmeans clustering to create visual vocabularies. Shotton et al. [84] went even further and used them for codebook creation and as classifier simultaneously. They propose similar features as Lepetit et al. [47], but use them to recognize object categories instead of keypoints and also to implement a semantic segmentation scheme.

2.2 Fine-Grained Object Categories

There exist several popular image databases, that are commonly used to evaluate object recognition and classification systems, such as Caltech 256 [32] or LabelMe [82]. However, most databases include all kind of different objects, with varying difficulty, while we have to deal with very specific object categories, namely butterfly species. Although, databases of butterflies have e.g. previously been used in [100] and [45], the species included in those sets are very different in their appearance. We on the other hand also like to be able to classify butterflies that eventually differ only a little bit from each other. When dealing with such objects we speak of fine-grained categories and classifying them is known to be a very challenging task. In the following we like to summarize some work, that dedicated themselves to such problems.

2.2.1 Identification of Insects

The eventually most similar project to ours has been worked on by O'Neill in cooperation with the Natural History Museum in London [74]. Their tool - the Digital Automated Identification SYstem (DAISY) - is very similar to what we aim for. All variants require user interaction to mark characteristic regions, which are used for classification. In its first version principal component analysis, as done for face recognition [90], was used to determine the correlation between a query scan and known classes. However, this approach has been shown to be too time consuming. To overcome this drawback, the second version used a nearest neighbour classifier and a specific pattern correlation measurement - the normalized vector difference (NVD).

Daisy has been used for various identification tasks concerning insects, like mosquitoes, wasps and also butterflies and moths. In most cases they used wings as characteristic regions to be marked by user input. They achieved very good results (>80% recognition rate according to [30]) for a dataset containing 60 species of British butterflies. While this problem setting is very similar to the one in our second project concerning wing patterns, we use a different approach. We will explain the main differences in Chapter 4.

The Automatic Bee Identification System (ABIS) has been proposed by Arbuckle et al. [3] as a suite of software tools to identify bee species. Identification of bees is done in a stepwise procedure based on high resolution images of their wings. Basically wings are seen as fingerprints. First *cells* of a wing are automatically extracted. Geometric features like lengths, angles and areas are calculated as well as appearance based features. Classification is then done either using a SVM or Kernel Discriminant Analysis (KDA). They achieve recognition rates over 95%, even when including bees, that are extremely hard to identify. There are however two drawbacks. First,

capturing bee wings has to bee done very subtle and precise in order to achieve good results. And second, during the phase of cell extraction, prior expert knowledge is incorporated into the framework. Thus, the system is very specialised on this exact task of bee identification and likely not suitable for other insect classification tasks.

Martinez-Munoz et al. [60] worked in cooperation with experts on the classification of stoneflies. Their dataset STONELFY9 consists of almost 4000 images of stonefly larvae from 9 different taxa. Their classification scheme is a two step procedure. First they use random forests together with local image features, based on keypoints and edges. But those random forests are not used to make final class decisions. Instead they are used to collect *voting evidence*. Each leaf therefore holds a histogram of the number of training samples from each category, that reached that leaf. Histograms are accumulated and in a second step forwarded to a second classifier to make the final decision. They achieved an error rate of only 5.6%, which is a very good result given the difficulty of the task. This also means a great improvement over their first work in that field [42], where they use a standard visual vocabulary approach and get an error rate of 16.1%.

2.2.2 Classification of Plants

Another fine-grained object category are plants. In [70] Nilsback and Zisserman worked on the automated classification of flowers. They use a dataset of 103 different classes of flowers with large interclass similarities and large variation within a class. Their classification procedure is based on four different aspects: local shape and appearance, shape of the boundary, overall distribution of petals and color of the flower. To combine all those features, they use a linear combination of Kernels, each corresponding to one feature type, in a SVM classifier. Nilsback and Zisserman additionally introduce weights for each Kernel and each class to define the importance of a feature for a specific flower type. In their experiments it is shown, that using multiple features give a great improvement over the single feature based classification. While the best recognition rate for only a single feature was 55.1%, the combination of all features improved the recognition rate to 72.8%.

LeafSnap is a popular iPhone and Android application to identify trees, based on just a single leaf of that tree. Therefore the leaf has to be put in a canonical position in front of white background, before being captured by the camera. The groundwork of that application has been done by Agarwal et al. [1] and Belhumeur et al. [5]. In [5] the proposal to match leafs is as follows. The shape of a leaf is represented by the Inner Distance Shape Context [51], which will be discussed in Chapter 3.2. A nearest neighbour classifier is used together with χ^2 statistic to determine the most similar leafs in a groundtruth set of known trees. They evaluated their algorithm on the Plummer Island and Baltimore Washington Woody Plants datasets,

which all contain about 30 samples per species. For evaluation Belhumeur et al. perform a leave-one-out test, where each leaf is removed from the dataset and used as query image. The correct result has been found in the top ten in about 90% of the time for the first dataset and in about 95%-97% of the time for the second dataset.

2.2.3 Visipedia

The ultimate goal of Visipedia as proposed by Perona [76] is a visual interface for Wikipedia, where queries are answered based on visual content of images. The basic functionality of such a system is desired as follows. A user captures an object of interest with a camera and sends it to Visipedia. Visipedia responds by sending information about that object back and additionally automatically segmenting it into its smaller contents. The user is then able to click on specific content displayed in the image to retrieve further information. However, such a system is still more in a theoretical stage and needs various groups of workers for realisation. Experts on specific topics are needed to provide necessary information and computer scientists are needed to provide tools that automate processes like segmentation. Additionally editors should enforce standards and annotators can help with naming and segmentation of images.

Perona used bird identification as a possible field of work for Visipedia, such that users are able to click on certain parts of a bird in their captured image, after they send it to Visipedia and Visipedia responds to the upload. Based on that example, Branson et al. [10] proposed a multi-class recognition scheme for bird species, that combines user given information with computer vision algorithms. The basic principle is the same as for the well known 20 questions game. The user is asked several yes/no questions concerning visual attributes of the bird at hand to stepwise reduce the number of candidate species. The goal is to correctly classify the bird, while minimizing the number of questions to be asked. Therefore they also incorporate the possibility of additionally using any image classification scheme, that produces a probabilistic output over classes.

In their experiments Branson et al. [10] use a dataset of around 6000 images containing 200 different bird species, that can usually not be identified by non-experts. They use a set of 25 visual questions, such as *'HasBellyColor?'*, together with local image features, spatial pyramids and SVMs as well as other techniques to complement the questions. See [10] for implementation details. It has been shown that an average of 11.11 questions are needed to correctly identify a specimen, when no computer vision algorithms are used for support. Incorporating computer vision then reduces that number to only 6.43 questions.

The purpose of Visipedia is to gather systems, like the above mentioned. We'd like to think that incorporating different insect identification schemes,

such as one to classify butterflies, would fit very well in the overall idea of Visipedia. Experts would have easy access to such a system and could share their knowledge. This would benefit computer scientists, that work on systems for automated classification, which on the other hand even experts can use to ease their work. Additionally, users of Visipedia and annotators would help to gather large sets of trainings data.

Chapter 3

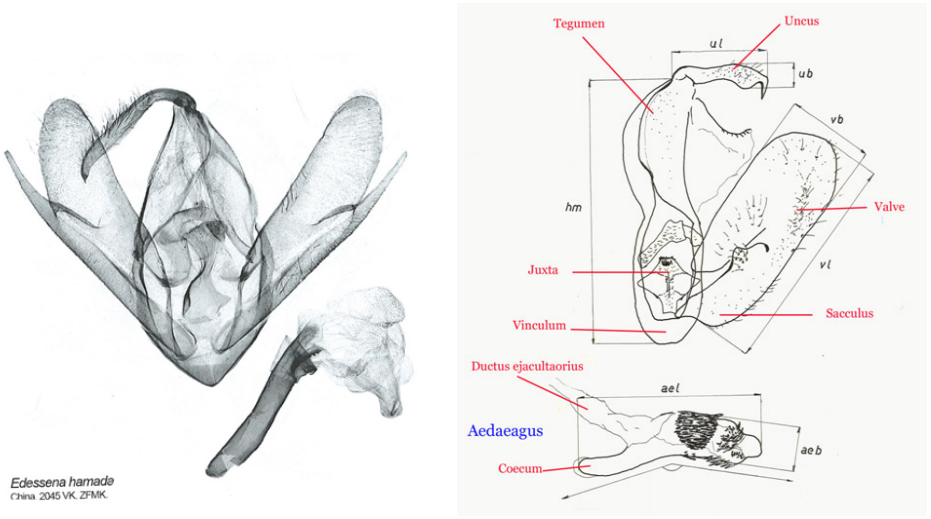
Classification based on Shape of Genital Organs

3.1 Introduction

The first project concerns itself with the classification of butterflies from the family known as *Noctuidae* (owl moths). The external appearance of most owl moths is almost identical, and for that reason entomologists need to look beneath the surface to identify a specimen. Instead of classifying by comparison of wing patterns, they dissect owl moths and have a close look at microscope scans of their genital regions. Shape and size of certain parts of the genital, as well as their relations to each other, are then compared to those of known species to determine the genus and species of a sample. Because the number of different species of the family of *Noctuidae* is estimated to go beyond tens of thousands, identifying an owl moth is a hard task even for experienced entomologists. Additionally in some cases the question may arise, if a specimen might be of a yet unknown species. Understandably a software solution that helps to narrow down the possible candidates of identity would be of great help for experts to quicken the assessment procedure. Therefore the goal of this project was to determine which means of computer vision prove to be most suitable to compare mentioned microscope scans of genital organs. In the next section we give a brief overview of what a dataset of such scans looks like and explain the biological relevant information that can be obtained from them to motivate a shape based classification approach.

3.1.1 Dataset and Biological State of the Art

Since the work on implementing their own database for the *Collection Austria* project is still ongoing, for the time being the *Museum of Natural History* provided scans of male genital organs from specialized literature [41]



(a) Typical scan of the genital organ of an *Edessena Hamada* from the subfamily of the *Hermininae*. Image taken from [41].

(b) Sketch of a genital organ explaining the most important biological terms. Image taken from [52].

Figure 3.1: The male genital organ consists of the genital corpus, with the valves to the left and to the right of the middle and the uncus on top. The tubelike aedeagus actually can be found in the middle, but is usually separated from the main corpus. To determine the species, entomologists e.g. calculate the ratio between the length of mentioned parts (*ul, ael, vl*) and their widths (*ub, aeb, vb*), compare their sizes to the size of the whole genital corpus and calculate the curvature of the aedeagus.

[35] additional to their own material. Those scans are the result of a very careful carried out procedure of preparation, where the genital organ is obtained from the gaster of the specimen, before it is stained and embedded in a microscopical mountant and then acquired by a digital microscope. Figure 3.1 shows such a genital organ of an male owl moth with an additional explanation of the most relevant regions.

Entomologists focus on certain parts of the male genital organ, especially the uncus, the valves and the aedeagus. Besides observing their shapes, they measure their length and width and calculate angles and ratios. Those so obtained attributes (called **relational indices**) are currently the main criteria for experts to determine the species of male owl moth. We therefore give an overview of just a small subset of relational indices and measurements of a much larger set, that is proposed by Lödl in [53].

Notation.

- **ub:ul** is the ratio of the width of the uncus to its length.
- **uh:ul** is the ratio of the height of the uncus to its length.
- **vb:vl** is the ratio of the width of the valve to its length.
- **aeb:ael** is the ratio of the width of the aedeagus to its length.
- **ul/vl/ael:hm** is defined as the ratio between the length of one organ part and the length of the main genital corpus.
- **tel:hm:sao** are the ratios between the length of the tegumen (upper part of the genital corpus), the length of the entire main genital corpus and the vinculum (lower part of the genital corpus).
- **oa** is the opening angle (the type of *knee*) of the aedeagus.

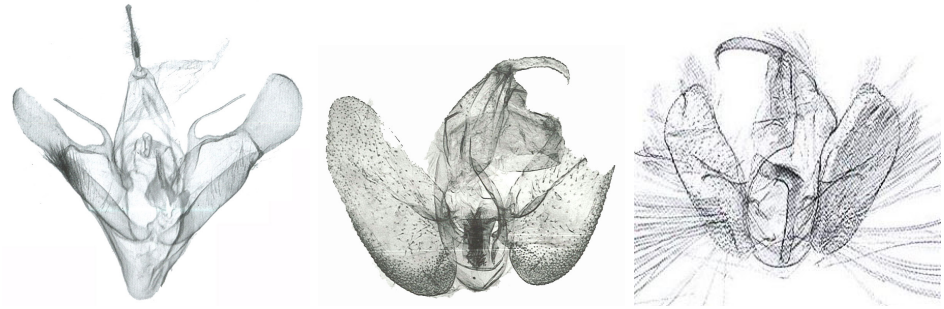
Those indices and measurements are illustrated in Figure 3.1(b) and will later serve as motivation to use specific computer vision methods for an automated classification procedure.

Although entomologists try to achieve a standardized position when dissecting a specimen, sometimes differences in position or damages to the specimen cannot be avoided. Typical positional differences in the resulting microscope scans, all of which are diagnostic irrelevant, are the orientation of the uncus, the angles between the valves and the genital corpus, or the exact position and orientation of the aedeagus. In most cases those uncertainties still allow entomologists to identify a specimen correctly, but sometimes, e.g. when the uncus is highly distorted, some parts of the genital organ become unusable for the classification process. The same can be the case due to damages like torn valves or occlusions of characteristic regions by other parts of the organ. Figure 3.2 gives examples of some obstacles of that kind.

Any software solution, with the goal to support experts during the classification procedure, needs to be robust to those factors or give the user the possibility to address them manually. In the next section we give a brief overview of our solution and explain the typical challenges that are part of its realisation.

3.1.2 Overview of Approach and Challenges

As far as we know, classifying butterflies and moths using microscope scans of their genital organs is a new territory for computer vision algorithms. Monti et al. [68] analysed Elliptic Fourier approximations of genital organ parts in microscope scans to investigate the anatomical incompatibilities of two related species, but a general approach to (semi-)automated classification based on genitals is not known. Therefore the most scientific challenges



(a) Orientation: Specimen with highly distorted uncus. As a consequence the shape of it can not be reconstructed. Image from [41].

(b) Damage: Specimen with a torn valve, which might be the result of an injury of the original butterfly or of uncarefull dissection. Image from [52].

(c) Noise: Specimen with uncleaned valves and therefore a lot of hair presence. Image from [35].

Figure 3.2: Three typical interference factors, that might occur in microscope scans of genital organs and therefore harden the classification process.

are based on the lack of experience with such datasets. The in Section 3.1.1 mentioned obstacles like occlusions, damages and noise are obvious ones. Others are a consequence of high similarities between specimens of a different genus on the one hand and few similarities of specimens of the same genus on the other hand. In such cases peculiar details are critical for entomologists to determine the species. Unfortunately the details to look for often depend on the species, and therefore identification results in an chicken-and-egg problem. While experts overcome this with lots of experience and anticipation, computers don't have that skill set yet. On the other hand, contrary to humans, computers are able to calculate and measure features very quickly and can store and access quickly a huge amount of data. Therefore the goal of this work is to provide a software solution, that helps entomologists by narrowing down the candidate-list of identity. Such a list is the result of various steps, that are all well known in computer vision.

The first task is the extraction of regions of interest in the query scan. This can be the result of an automatic segmentation or of experts manual interaction. Typical regions of interest correspond to those parts of a genital organ, which are mentioned in the previous section. In some other cases one might only be interested in certain characteristic segments to address the previous mentioned small details of high importance.

In the second stage, the task is to describe the previously gained regions of interest in a mathematical manner. This part is where the main focus of our work lies. Due to the lack of texture in microscope scans of genital organs, and because shape is the most vital feature for entomologists also, we decided to concentrate on shape based description methods from com-

puter vision research. We use different proven shape descriptors to describe enclosing contours and contour segments of regions of interest and evaluate their capabilities on our dataset. Furthermore we adapt them and combine them with each other to improve our results. We also study the qualities of some entomological relational indices and how they can be implemented as shape descriptors themselves.

In the last stage of our workflow, the calculated descriptors are compared to those of already processed specimens in one of the following ways. For contours, that enclose a complete region of interest, we compare their entire corresponding descriptors. We then speak of global retrieval, because a whole region has to be recognized. For contour segments the task is similar, but instead of retrieving a whole region, we are interested in detecting the segment of an unknown specimen *as a part* of a larger contour of a known one. Hence we speak of partial retrieval in such a case. The end result then is a list of specimens, with the highest similarities in shape to the one in the query scan.

3.1.3 Outline

In Section 3.2 we give an overview of existing proven methods from computer vision that use shape description for object detection and recognition.

Section 3.3 explains our system for Butterfly-Classification based on microscope scans in detail. We discuss several different approaches that address the given problem setting differently. Section 3.4 then provides evaluation of those systems. Therefore we arrange several experiments using datasets retrieved from various literature and compare the classification results in order to determine the most suitable solution.

Finally in Section 3.5 we summarize our procedure and discuss the experience gained from our experimental results. Furthermore we propose possible future work that addresses eventual shortcomings as well as additional features, that might be interesting for entomologists.

3.2 Related Work

Nowadays recognition systems usually consist of the three steps we described in Section 3.1.2: Region Extraction, Region Description and Matching. Shape is the main criteria for entomologists when determining the species of owl moths by their genital organs. Therefore in this section we give an overview of State of the Art techniques, that can be used for recognition systems by describing the shape of regions of interest.

3.2.1 Region Extraction

Describing the content of an image can be done in two ways. Either by describing the whole image at once, e.g. by its color values, or by determining regions of interest first and describing those in an appropriate way later. A region of interest is an area in an image that has to fullfill certain qualities to ensure the reliability of the entire recognition system, as those regions are fundamental for all following stages. They have to be distinctive for the object that is shown in the image and must be able to be retrieved in different acquisitions of the same object. Additionally detection of those regions may have to be invariant to scale, rotation, illumination and other possible factors depending on the specific task. In our case those regions have to be used to describe the shape of an organ, and therefore we are often interested in their outer contours. Those contours can be given as region boundaries from manual or automatic segmentation of the image. Another option is to use edge detection to find multiple, smaller contour fragments directly, instead of partition the entire image. Either way the results are one or several lists of image coordinates (X, Y), each corresponding to an entire region or a contour. In the following we give an overview of some methods that can be used to extract regions and detect contours for shape description either completely automatically or based on user input.

Edge Detection

One way to retrieve object contour fragments for shape description, is to detect edges in an image and interpret them as such. A very common used procedure in this field is the **Canny edge detector** [14]. The Canny edge detector is a subtle designed method, that can be seen as the result of various steps. A Gaussian kernel is used to reduce noise first. To detect intensity changes Gaussian derivative responses are then calculated for each pixel, followed by determination of orientation and magnitude of the so obtained gradients. Non-maxima suppression is then used to remove weaker responses. The final set of edges are then the result of a hysteresis thresholding, for which two thresholds are necessary. The need to manually defining those is one of the main reason for eventual shortcomings of this method, another is the fact that not all detected edges necessarily correspond to object boundaries.

The state of the art method for edge detection is the **Berkeley natural boundary detector** [59] proposed by Martin et al. In order to detect edges more likely to correspond to object boundaries, they look at local discontinuities at feature channels in regard to brightness, color and texture. Supervised learning on a groundtruth dataset of human segmentations along with several different classifiers is used to determine the optimal combination of those features. This procedure especially outperforms a simple intensity

based edge detector like Canny for natural images, where lots of texture and color information is present. The Berkeley detector is e.g. used in [22] as the base for building a contour segment network for shape matching.

Watershed

The basic principle of **Watershed**, which has been introduced to image processing by [7], is to interpret a grayscale image (usually a gradient image) as a topographical map, where small gray value regions represent basins and higher values ridges, that divide those basins. By virtually flooding the basins (by thresholding), they rise above the ridges and merge with other basins, defining a watershed. Those watersheds are then used as region boundaries. When used on gradient images the original Watershed principle tends to over-segmentation, therefore various work engages itself with overcoming this obstacle (e.g. [8, 31]).

Maximally Stable Extremal Regions

The detection of **Maximally Stable Extremal Regions (MSER)** [61] is strongly related to the Watershed principle discussed above. For a grayscale image all possible thresholds are successively used to segment the image. All pixels below the threshold are set to 0 (black) and those above the threshold are set to 1 (white), therefore defining several regions of connected pixels of value 0. Increasing the threshold leads to more black pixels and growing regions, but sometimes a region stays the same for consecutive thresholds. It is then considered to be a MSER. Given some time MSERs also merge during the thresholding procedure and might then again define a new, larger MSER, nesting the previous ones. The whole procedure can be repeated with the inverted image to retrieve a different set of MSERs. MSERs have very good repeatability properties, are invariant to photometric changes and can be calculated in linear time, as Nister and Stewenius showed in [72]. Therefore it is a very commonly used region detector for various tasks. E.g. Forssen and Lowe [26] as well as Donoser et al. [21] propose to use MSERs in combination with shape descriptors in their respective work.

Active Contours

The idea of **active contours** (also known as snakes) [39] is to align a curve C given by user interaction to the boundary of an object. Therefore the alignment problem is stated as a minimization problem of the sum of the internal energy (measuring the smoothness of C) and the external energy (representing the distance of C to the actual contour). Over various iterations a snake then approaches and eventually converges at the object contour. In [15] Caselles et al. explore a link between active contours and geodesic curves in Riemannian space - the **geodesic active contours (GAC)** - which they use

to restate the minimization problem to further improve boundary detection results. A main profit of snakes is, that they are able to deform themselves to align onto an object and are even able to detect illusory contours. Drawbacks are a strong dependence on the initial curve C and eventually slow convergence rate.

Total Variation Segmentation

In [93] Unger et al. propose to minimize the GAC energy from [15] by using the fast Total Variation minimization approach. The GAC energy is therefore stated as a weighted Total Variation problem, like in [11] and combined with an additional term to incorporate local constraints. Those local constraints represent user input, corresponding to background and foreground assignments and their reliabilities as such. The algorithm is implemented on the GPU and therefore achieves good segmentation results almost in realtime.

3.2.2 Shape Description

Once regions, contours or contour fragments have been retrieved from images, their shapes have to be described in a mathematical manner for the computer to be able to compare them. Therefore one has to specify the term *shape* and define when two shapes are similar first. Kendall [40] defined shape as the geometrical information that remains the same under certain transformations, namely translation, rotation and uniform scaling. Two shapes are then considered to be the same if one can be mapped to the other by those transformations. Based on this definition of shape, every shape descriptor should be invariant in that regard and hold characteristic geometrical information.

Basically shape descriptors can be divided into two groups. The first group uses an entire region to describe its shape, while the second group uses only its contour. In the second case we also often distinguish between global and local description of the points lying on the contour. If one point of the contour is described by its relative position to all other points, we speak of a global shape descriptor. A local shape descriptor on the other side only takes neighbouring points into account.

In the following we discuss some proven shape descriptors, that have been used for several object detection and recognition tasks, as well as in other fields of computer vision.

Moment methods

Image moments have been used to describes regions of interest in an image for a long time [25]. The basic idea is to interpret an image I as a probability density of a 2D random variable. For such a density function characteristic

statistics, called moments, can be calculated. Given a region and its discrete set of coordinates (X, Y) the raw moments $m_{(i,j)}$ of order $(i+j)$ are defined as

$$m_{(i,j)} = \sum_{(x,y)\in(X,Y)} x^i y^j I(x,y). \tag{3.1}$$

Those moments then provide shape and gray level information of the region of interest. If just the shape of the region has to be observed, instead of using I one can simply use a binary function corresponding to the region. Note that raw moments do not fullfill the necessary invariances to translation, rotation and scaling. In order to achieve those invariances Hu [36] e.g. proposed to use a set of seven adapted moments - the **Hu moments**. This is just one example from a large variety of different methods, that have been proposed to make moments invariant to certain transformations. A good summary of different moment approaches used for object recognition is given in [24].

Fourier Descriptors

A boundary C of an object can be seen as a closed curve in the complex plane. Travelling anti-clockwise along that curve can be represented by a periodic, complex function $z(t)$ with period 2π. The Fourier coefficients of the Fourier representation of $z(t)$ can then be used as shape descriptors [87]. Fourier descriptors and its derivatives to address certain invariances were introduced and especially used in the 1970s and 1980s for e.g. for digit recognition tasks [85].

Curvature Reducing Methods

According to Hoffmann and Richards [34] humans psychologically segment contours at negative local curvature minima and see the shape of the contour as the combination of the so gained segments. Therefore Mokhtarian et al. [67] make use of the curvature of a contour in the following way. They parametrize a contour C and convolve the corresponding coordinate function with Gaussian Kernels of different widths σ. Increasing σ results in smoothed contours and a decreasing number of zero crossings. The positions of those zero crossings move along the contour while σ is increased and finally meet at some point and vanish. After all zero crossings are vanished the final contour is convex. The description of the contour are then pairs of zero-crossings vanishing positions and their corresponding scale σ.

The same psychological observation is used by Latecki and Lakämper [43], but instead using the scale space, they use the tangent space to represent the curvature properties of a contour.

Shape Context and Relatives

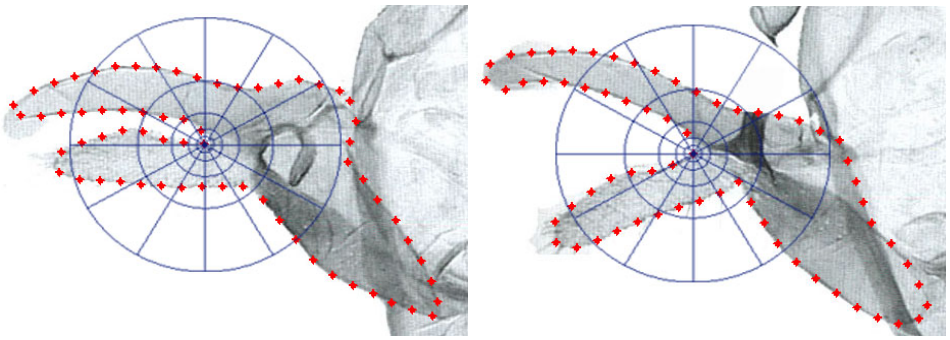
In [6] Belongie et al. propose the **Shape Context** as a shape descriptor for several recognition tasks like digit recognition and trademark retrieval. Given a contour C containing m points, they use only a smaller subset C_s of n points to describe its shape. Note that the choice of that subset does not need to correspond to points with specific characteristics, but uniformly distributed points are preferred. From one point there are $(n-1)$ vectors to all other points. Instead of using those vectors directly to describe the shape, Belongie et al. propose to calculate a histogram (the Shape Context) for every point, representing the relative distribution (distance and orientation) of the remaining points. For $p_i \in C_s$ the distribution over the relative position of the remaining $n-1$ points on C_s is then given by the histogram

$$h_i(k) = |\{p_j \in C_s : j \neq i \wedge (p_j - p_i) \in bin(k)\}|. \tag{3.2}$$

For the binning, they suggest to use 5 bins for $\log r$ and 12 bins for θ to make the descriptor more sensitive to local information and to ignore points, that are too far away from p_i. See figure 3.3 for illustration. The so obtained descriptor is by definition translation invariant. Scale invariance is achieved by normalizing all radial distances by the mean distance of all possible point pairs and rotation invariance by treating the tangent in p_i as the positive x-axis of the binning frame. The Shape Context is still commonly used for describing shape (e.g. in [56, 102]), and several other methods are influenced (e.g. [51, 57, 89]) by the work of Belongie et al.

Shape Context is sensitive to articulation, as illustrated in Figure 3.3. Therefore Ling and Jacobs [51] e.g. adapted the original variant for the recognition of objects, that can be affected by articulation, like hands or scissors. To achieve robustness to articulation they use the inner distance measure instead of the Euclidean distance when calculating the Shape Context histograms. For two points $p_i, p_j \in C_s$ the inner distance is defined as the length of the shortest path from p_i to p_j within the inner region of C. They also use the resulting **Inner Distance Shape Context** in combination with appearance based information to recognize leafs.

Toshev et al. propose another shape descriptor that is inspired by the Shape Context and use it for object detection and segmentation tasks [89]. They define a chord as a pair of boundary edges of a region of interest. For each chord, length and orientation of the vector connecting the boundary edges are calculated, as well as the orientations of the normals to those boundaries. By using the normals, that point inwards the object, they also get information about the inner structure of the object in addition to its outer shape. The distribution of those *chord features* are then again summarized in a histogram - the **Chordiogram**.



(a) Left valve of a *Eublemminae Oruza kunsuki*.

(b) Flipped right valve of the same *Eublemminae Oruza kunsuki*.

Figure 3.3: Left and right valve of the same specimen with a shape context frame with 60 bins. Note that the right valve has been flipped to especially illustrate the sensitivity to articulation. Original image from Kononenko [41].

Turning Angles and Distance across the Shape

In [17] Chen et al. propose the use of **Turning Angles (TA) and Distance across the Shape (DAS)** as descriptor for partial shape matching. For a point p on the outer object contour, they use its immediate neighbour points to define the Turning Angle [88]. The bisector of that angle intersects the contour at other points and the DAS is defined as the distance of the original point p to the closest intersection point p'. A linear combination of those two features is then used as descriptor. Invariance to rotation and translation is given by the definition of the descriptor and scale invariance can easily be achieved by normalizing the DAS. Note that contrary to Belongie et al. [6], Chen et al. propose not to select the keypoints from the contour uniformly, but according to two heuristics. They sample the contour in a manner, that selects points with high curvature and simultaneously tries to maximize the distance between selected points.

kAS

Ferrari et al. [22, 23] avoid describing each keypoint on a contour and instead describe contour segments more directly. Those contour segments are usually the result of a carefully taken out preprocessing procedure - the building of a contour segment network (CSN) [23] across an image. Basically edgels are detected, linked and finally partitioned into straight contour segments. A ***k*AS** is then defined as a path of k such adjacent segments in the CSN. The final feature descriptor of k adjacent segments is a linear combination of the length and the orientation of the segments, as well as the vectors connecting the midpoints of the segments. As k grows so do the descriptors and a kAS is consequently able to represent more complex shape structures. For example,

while 2AS can define L like shapes, but not Z like shapes, a 3AS is able to do the latter. Ferrari et al. state that k should not be greater than 5, as the resulting represented structures then may not be repeatable retrieved in different acquisitions of the same objects. Instead they recommend $k = 2$ or $k = 3$.

Structural Measurement Descriptors

When a contour C is given, it is a common procedure to sample n points to obtain C_s and measure certain values between points $p_i, p_j \in C_s$. Riemenschneider propose **Structural Measurement Descriptors** [77] to store those values in a matrix $D \in \mathbb{R}^{n \times n}$ defined by

$$d_{i,j} = meas(p_i, p_j) \tag{3.3}$$

with $meas$ being a measurement of choice to describe structural information. A natural choice for example would be the length of the chords $\overline{p_i p_j}$, but basically every meaningful measurement can be used. Note that if points on C_s are ordered clockwise or counter-clockwise along the contour it is a useful quality of structural measurement descriptors, that by definition local information is stored close to and global information farther away from the main diagonal. Consequently submatrices of D represent contour fragments of the entire contour. Those properties make this kind of descriptor suitable for global as well as partial shape matching problems.

In [20] Donoser et al. propose a descriptor that is designed for shape matching of closed, outer contours of objects, by using a structural measurement matrix based on angles. Given two points p_i and p_j on the sampled contour C_s, they select a certain third reference point $p_{j-\Delta}$, $\Delta \in \mathbb{N}$ with respect to p_j and calculate the angle between the chords $\overline{p_i p_j}$ and $\overline{p_j p_{j-\Delta}}$. The respective entry in a measurement matrix Θ is then given by

$$\theta_{i,j} = \sphericalangle(\overline{p_i p_j}, \overline{p_j p_{j-\Delta}}) \tag{3.4}$$

where $\sphericalangle$ denotes the angle between two chords. A drawback is, that for $i > j - \Delta$ the reference point $p_{j-\Delta}$ is not in between p_i and p_j and therefore missing, when just a fragment of C_s is considered instead of the entire closed contour. In later work [79] Riemenschneider et al. therefore adapt the descriptor to achieve similar results for non-closed object contours. They redefine the reference point to force it to lie between p_i and p_j and adapt the measurement function in the following way

$$\theta_{i,j} = \begin{cases} \sphericalangle(\overline{p_i p_j}, \overline{p_j p_{j-\Delta}}) & \text{if } i < j \\ \sphericalangle(\overline{p_i p_j}, \overline{p_j p_{j+\Delta}}) & \text{if } i > j \\ 0 & \text{if } \mid i - j \mid \leqslant \Delta. \end{cases} \tag{3.5}$$

The resulting descriptor then has the main advantage to be self containing, making it more suitable for partial matching of contour fragments. Riemenschneider et al. use this quality for localization of objects in cluttered images. Note that both descriptors are translation and rotation invariant and for uniform sampled contours of fixed length also automatically scale invariant.

3.3 System

The main goal of our work was to investigate, which proven algorithms for Object Recognition based on shape are most suitable for datasets from Section 3.1.1 and to implement them in a software solution, which can be used by entomologists to ease the classification process. Therefore we provide a Matlab framework, where certain parts, like e.g. used descriptors, are easy to interchange and combine to be able to address diverse obstacles. The basic workflow is sketched in Figure 3.4. In a first step contours of the organ have to be provided. This are typically the outer contours of certain regions of interest delivered by manual segmentation or just a contour fragment also given by user input. From now on we will use the following notation for closed contours and contour fragments:

Notation.

- $C = (X_c, Y_c)$ set of image coordinates of a closed contour as the boundary of a region
- $F = (X_f, Y_f)$ set of image coordinates of a contour fragment

In a second step various descriptors can be calculated, based on the specific needs of the user. Those descriptors are then used to measure similarities between a query scan and scans in a groundtruth database. The final result is then a list of specimens with similar shape properties, which entomologists can use to further investigate the species of the underlying query scan. The implementation of the above mentioned steps will be discussed in detail in the following sections.

3.3.1 Manual Extraction of Regions of Interest

In most cases entomologists will be interested in the shape of the valves, the uncus and the aedeagus of an owl moth. Due to their experience, they can provide reliable segmentations of those regions, usually knowing which parts of the whole genital organ belong to which *sub*region. To give them the opportunity to mark those regions we incorporate the Total Variation Segmentation (TVSeg) tool [93] in our framework. TVSeg is a fast and comfortable way to segment images by marking *some* foreground and background pixels.

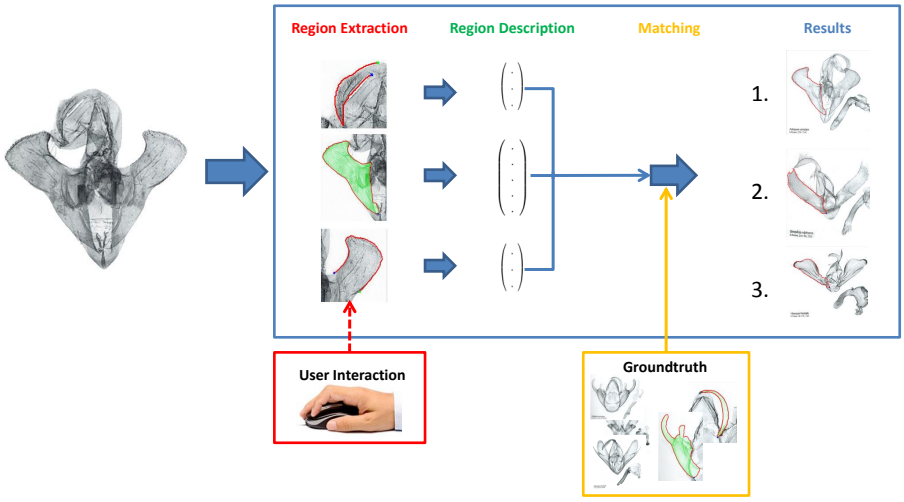


Figure 3.4: A sketch of the workflow of our system. Regions and contours are extracted, followed by the calculculation of various descriptors. Those are then used to determine scans of similar shapes in a groundtruth database.

The segmentation problem is stated as a minimization problem of an energy functional with the user input as local constraints. The result is the region of interest as a binary mask and the boundary of that region C. Note that we chose TVSeg, because it is able to provide good segmentation results in almost no time, but any other interactive extraction method, that is able to achieve similar results, can be used as well.

In some cases entomologists might be interested in just a fragment F of C, because they suspect it to be very characteristic for the species of the observed sample. Also due to a lot of noise, like e.g. hair, even experts may not be able to mark an organ part exactly. To address those aspects, we provide the possibility to define start and end points of contour fragments, that are then used for matching instead of the region boundaries containing them.

3.3.2 Describing Entire Organ Parts

In this section we focus on the description of closed contours C corresponding to entire organ parts, namely the valves and the uncus. We first give a few examples on how relational indices can be calculated automatically from C and then illustrate their connections to commonly used shape descriptors. Basically one can use the same methods for the aedeagus, but unfortunately the aedeagus is rarely given in a standardized position, often infiltrated with lots of noise or generally unusable because of preparation.

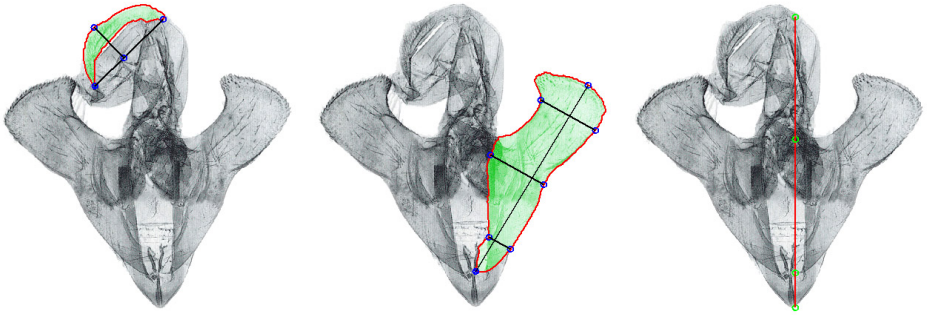
In the latter cases entomologists ignore the aedeagus, and this is recommended for our semi-automatic approach also. If the aedeagus is able to provide useful information, we recommend to use the fragment containing that information for partial matching as described in Section 3.3.3.

Implementation of Relational Indices

Given one or a few organ parts and their outer contours C, we incorporate the possibility to automatically calculate some of the relational indices proposed by Lödl [53], we mentioned in Section 3.1.1. The realisation is done in the following way:

- For **uh:ul** we first detect the peak of the uncus as the point with the highest curvature $p_1 \in C$. **ul** is then given by the Euclidean distance between p_1 and its farthermost point $p_2 \in C$. We obtain **uh** as the maximum normal distance of the vector $\overrightarrow{p_1p_2}$ to a point $p_3 \in C$. See Figure 3.5(a) for a typical result of this procedure.
- To determine **vb:vl** we calculate the maximum distance of two points $p_1, p_2 \in C$ and define it as **vl** first. **vb** is then obtained as the distance of the two intersection points where the bisector of $\overrightarrow{p_1p_2}$ meets C. Note that in some cases it is recommended to calculate not only **vb** but various widths, hence we split a valve into several sections and obtain the width for each of them accordingly. See Figure 3.5(b) for a typical result of this procedure.
- **ul/vl:hm** is calculated by determining **ul** or **vl** as described above and **hm** as the distance between the top and the bottom of the main genital corpus. Therefore two organ parts have to be extracted first.
- **tel:hm:sao** is calculated by intersecting valves with the main genital corpus and recognizing the articulation points of the valves as the topmost and lowest intersection points. See Figure 3.5(c) for a typical result of this procedure.
- Although not described in [53], we additionally calculate **us/vs/aes/hs**: **us/vs/aes/hs** as the ratio of numbers of pixels belonging to two organ parts.

The benefit of automatically retrieved indices is, that the user has to define the necessary regions of interest just once for a specimen, instead of measuring the indices separately, which is an elaborate kind of work. Nevertheless there are obvious drawbacks. The precision of the automatic calculated indices heavily relies on a robust detection of specific points. While the here implemented relational indices are in most cases robustly determined, there are a lot for which automatic retrieval would be a hard task. Unfortunately one specific index only contains a small amount of information and



(a) Ratio between uncus length and uncus height.

(b) Ratio between valve length and valve widths.

(c) Ratio between length of Tegumen, Vinculum and the entire corpus.

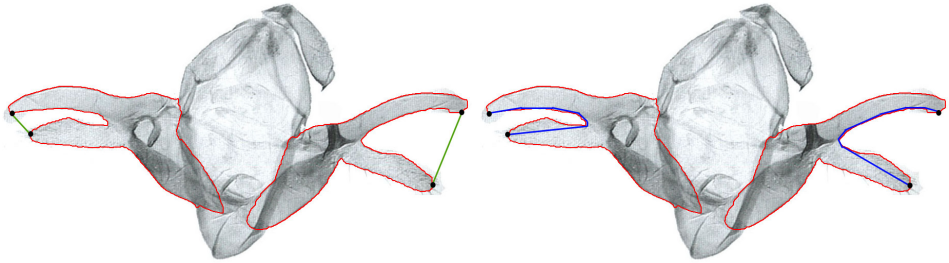
Figure 3.5: Illustration of a few relational indices for the genital organ of a male owl moth. Original image from [75].

for an accurate classification of a specimen there are more than just the five indices, mentioned above, necessary. However the above explained calculation of relational indices motivates the use of certain shape descriptors as we will explain in the following sections. We will also use them for comparison during the evaluation in Section 3.4.

Extension of Relational Indices

Instead of trying to implement the automatic calculation of all relational indices, we investigate proven shape descriptors and analyse their similarities with some indices and measurements. But first we like to record how we process a closed contour C before we describe it. The amount of points on C is usually very high, and using them all for shape description would lead to high computational effort with lots of redundant information. Therefore we sample C first to obtain C_s. This is done by choosing exactly n uniformly distributed points from C to get a fixed length feature vector for all contours. Note that those points are ordered clockwise along a contour, starting at a specific startpoint that is characteristic for the organ part and can be detected robustly. This is done to ease the later matching procedure. For the uncus we use its peak and for the valves we take the lower point of the points that define **vl** (detection of both is already described at the beginning of section 3.3.2).

Given C_s corresponding to an organ part, we propose to use a structural measurement descriptor D from Equation 3.3 instead of calculating relational indices. We use D along with the Euclidean distance of two points as measurement and normalize it by dividing it with the maximum distance between two points to achieve scale invariance. Now it can be seen that e.g.



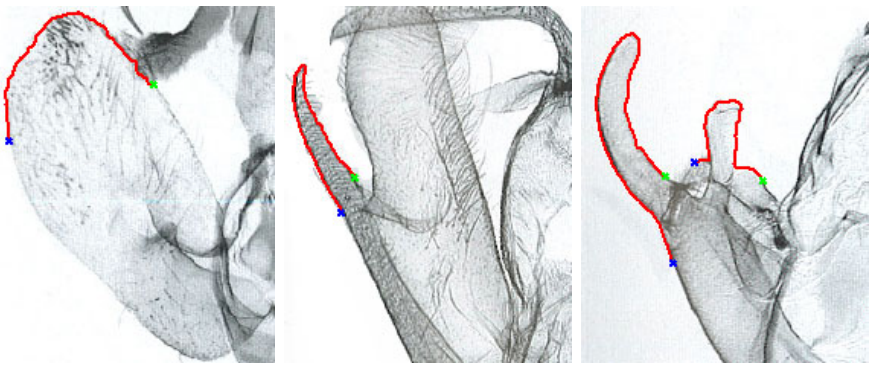
(a) Euclidean distance of two points. (b) Inner distance of two points.

Figure 3.6: Specimen, where the sensitivity of the Euclidean distance to articulation can be shown. The Euclidean distance of two peaks of the right valve is about three times as long as the distance between corresponding points on the left valve. On the other hand the inner distance is very similar. Original image from [41].

the relational index **vb:vl** is incorporated in D if C is the outer contour of a valve. As **vl** is calculated as the maximum distance of two points on C and one or various valve widths are also just distances between two points, the normalized distance matrix D holds the necessary relational index information. In fact it holds a lot more information about the shape of the contour, including other relational indices, and there is no need to detect certain points on C.

A drawback of the distance matrix as described above is, that the Euclidean distance is sensitive to articulation. A more natural way to describe objects, where articulation is irrelevant, is based on the inner distance. A distance matrix D using the inner distance instead of the Euclidean distance then is less sensitive to articulation. Figure 3.6 illustrates the sensitivity of both distances to articulation. Depending on the species, the exact articulation of certain organ parts can be a decisive feature when identifying a specimen, while for other species it is diagnostic irrelevant. In the first case the Euclidean distance is the measurement of choice, while for the second case the inner distance would be more appropriate. We will evaluate the importance of articulation sensitivity during the experiments in Section 3.4.

Although distance based relational indices similar to **vb:vl** are included, there are some indices that are not incorporated in a distance matrix, like e.g. **uh:ul**. At first this index may also seem as just another ratio between two lengths, but in fact it describes the curvature of the uncus by using the height instead of the width of the uncus. This is just one example, where an entomologist measures the curvature or a single angle to retrieve information about the specimen. Others are the curvature of the peak of the uncus or the opening angle **oa** of the aedeagus. To extend this kind of angle based information we propose to use a structural measurement matrix Θ as given



(a) Valve with no evaginations.

(b) Valve with one almost straight evagination.

(c) Valve with a curved evagination and an additional small one.

Figure 3.7: Three different kinds of valves. While the valve of the specimen on the left has no evaginations, the other two specimens do. Original images from [41].

by equation 3.4 or 3.5.

Distance based and angle based structural measurement descriptors can be used together to cover as much entomological relational indices and measurements as possible. But we also like to propose the use of descriptors that already contain both kind of information in a more summarized way to cover relational indices. Shape Context as proposed by Belongie et al. [6] fits our needs therefore, as relative positions of points are described by angles and distances. We calculate the Shape Context of every point $p_i \in C_s$ and use all the corresponding histograms as description of an organ part by concatenating them. This results in a global descriptor of size $n \cdot m$, where m is the number of histogram bins (we use $m = 60$ as originally suggested by Belongie et al. [6]). The binning frame is adapted according to the mean distance between points on C_s to achieve scale invariance. Although this is not a one-to-one realization of any relational index, it holds similar information in a more compact way. As Shape Context uses the Euclidean distance, it is also sensitive to articulation. We therefore also use the articulation insensitive variant proposed by Ling and Jacobs in [51] for comparison.

Bag of Features

While above mentioned descriptors incorporate typical relational indices in one way or another, we also propose the use of the bag of features principle [86], that has already been used together with structural measurement descriptors in [80]. This is motivated by the following example. Valves usually have varying shapes. There exist valves that have e.g. three evaginations, while others have none. Also those evaginations can be themselves shaped

differently. They can e.g. be curved or straight. See Figure 3.7 for some examples. Our idea is to use the bag of words approach to address the existence of certain fragments or points of C_s, when comparing two entire shapes.

Therefore we construct a codebook in one of the following two ways. Given various outer contours C_s from a training set of N organ parts (e.g. N left valves), we use submatrices of the corresponding structural measurement descriptor to describe all possible overlapping fragments

$$F_s = \{p_j : j = i - m, ..., i, ...i + m\} \subseteq C_s \tag{3.6}$$

of fixed length $2m + 1$ with $p_i \in C_s$ as their middle point. We then use K-means clustering on all fragments of all outer contours in the training set to obtain k cluster centres to represent our fragment codebook. The shape of an entire organ part C_s is then given by the codebook distribution of its fragments. It is important to note that the actual location of a fragment as a part of the whole contour is not relevant, as only the distribution of fragments is considered. Optimal value for k depending on N and n will be discussed in Section 3.4.

While such a codebook is based on fragments and focuses on one specific measurement, we also propose the use of Shape Context for keypoint description and construct a codebook, that incorporates angle and distance information at once, accordingly. We therefore calculate the Shape Context of every point $p_i \in C_s$ for every C_s in the training set, use again K-means for clustering and the distribution of the Shape Contexts to describe an entire organ part.

3.3.3 Describing Fragments of Organ Parts

The bag of features approach as described in the previous section already takes into account the existence of fragments or keypoints depending on the used codebook. But sometimes the question may arise, if just one *specific* fragment in a query scan, that seems to be very characteristic for the specimen, can be retrieved in known specimens. Thus resulting in a slightly different problem statement, than it would be the case when observing the distribution of *all* possible fragments. The length of fragment F of a contour, defined by a startpoint p_1 and an endpoint p_m, depends on user input. Therefore there are some small sampling adjustments necessary. When dealing with a closed contour C we can already address scale invariance by uniformly sampling exactly n points for all contours. For a contour fragment F on the other hand, the necessary number of sampled points depends on the actual length of the fragment. Therefore instead of sampling an equal number of points for every fragment, we define a sampled contour fragment $F_s \subset F$ as

$$F_s = \{p_i \in F : i = 1, 1 + d, 1 + 2d, ..., 1 + jd \leq m\} \tag{3.7}$$

where d is a parameter to define the index-distance between two points and $j = \max\{k|1 + kd \leq m\}$. We then want to match the entire user selected fragment F_s with fixed length as part of predetermined closed, outer contours of organ parts in a groundtruth database. E.g. corresponding to a problem statement like *"show me all unci with a similar peak as the one marked in the query scan"*. Therefore, additionally to the equidistant sampling method, the used descriptors for fragments and entire contours need to be comparable. This means two things. First, description of a fragment F_s must be retrievable from the description of an entire contour C_s and second, it should not depend on points on C_s, that are not part of F_s.

The angle matrix given by Equation 3.5 as proposed by Riemenschneider et al. in [79] is self containing by design and already fits our needs. For comparison we also use Equation 3.4 as proposed by Donoser et al. in [20], but ignore the first Δ columns of the description of a fragment to achieve self containment. This is possible, because the fragments are of fixed length and will not partitioned further during the matching procedure. We also evaluate the Shape Context for fragments, to see if relative point positions are better suited for such a problem than angle matrices. Therefore Shape Contexts have to be calculated with a predefined scale for the binning frame, as normalization would make fragments and entire contours otherwise incomparable. It is hereby noted, that such a description of F_s, when retrieved as a part of an entire contour, is affected by points that are not part of F_s. This can be seen as little influential noise and reduced by using a smaller scale for the binning frame, but makes Shape Context less suitable especially for description of small contour fragments.

We also like to note, that there are some additional benefits when retrieval of fragments is done, instead of a global comparison of entire closed contours. For example in regions, where valves intersect with the main corpus it can be hard to determine which pixels belong to which part. A wrongly done segmentation of the regions then might lead to a deformed outer contour, which is unsuited for a global comparison procedure. Parting the outer contour into fragments, leaving those parts out, where someone is uncertain where it belongs to, is a possible way to deal with that manner. Also e.g. for an articulation sensitive descriptor one can address this issue manually by parting the outer contour at points where articulation seems likely.

3.3.4 Matching

Once boundaries and contour fragments have been extracted from a query image and descriptors for each have been calculated, they have to be matched against a groundtruth database of labelled organs with pre-calculated descriptors. Whether we deal with entire contours or with fragments this is done differently. When given a new query scan, an entomologist is usually interested in a matching problem corresponding to a question similar to the

following:

- *"What are species with a similar valve?"* (global matching of entire, closed contours).
- *"What species have an uncus with a similar peak?"* (partial matching of selected fragments).
- *"What species have a similar valve and an uncus with a similar peak?"* (combination of the above).

We first discuss how we implement global matching for entire boundaries, followed by the partial counterpart for fragments and a proposal on how to combine more contours and descriptors for more precise results.

Global Matching

Given the segmentation of an organ part (a valve or an uncus) in an unknown query scan, we want to retrieve a list of specimens, that have an organ part of similar shape. Therefore we use its outer contour C_s of exactly n points and match it against a database of organ parts, where the same parameters have been used. As the sampled query contour and groundtruth contours then contain the same number of points n, we can simply compare their descriptors with a specific distance measurement and choose the nearest neighbours as results. Usually, to be able to do so, point correspondences have to be estimated first. As points on C are already ordered clockwise and a certain startpoint has been calculated, that is characteristic for all organ parts of the same type (uncus peak, lower maximum distance point of the valve), we are able to avoid the correspondence problem. The descriptors we use are either the ones, that are motivated by relational indices or the distribution of fragments or Shape Contexts as described in Section 3.3.2.

Note, that in the latter case the actual index of a fragment or a point holding shape information is irrelevant, as it is summarized in a histogram. Thus wrong point correspondences, as a result of insufficient segmentation, affects those description methods less than the *direct* descriptions, while on the other hand losing some spatial information. Also, when comparing two shapes, the distance measurement of choice varies, depending on the used descriptor. We use the Euclidean distance for relational indices or structural measurement descriptors and the χ^2 statistic for concatenated Shape Contexts and the Bag of Words distribution descriptions.

Fragment Matching

In case of a partial matching problem, a fragment F_s (e.g. the peak of the uncus) with fixed length of a contour is given, and one is interested in specimens that contain a similar fragment in the respective organ part. Therefore

descriptors, as proposed in Section 3.3.3 are calculated over various scales (sampling parameter d and binning frame size s in case of Shape Contexts) and matched in a nearest neighbour sense against every possible fragment of same length in a groundtruth database of closed contours. This corresponds to *placing* the query fragment along a groundtruth contour C_s and determining the best fit in terms of position and scale. Due to the properties of the discussed descriptors, such a comparison of fragments as part of closed contours is reasonable. It is redone for every contour C_s in the database, and the final result is then a list containing fragments most similar to the query fragment.

Combining Contours and Descriptors

Global and partial matching already allows to find specimens with certain shape similarities. The most interesting species however, would be those, that are similar in various aspects. E.g. for a query scan, we might be interested in all species in the groundtruth database, that have a similar left valve, a similar right valve and additionally a similar uncus peak and aedeagus curvature. Also it may be useful to use various descriptors to overcome their shortcomings. E.g. using a distance matrix and an angle matrix to compare two left valves and thus incorporating more relational indices as one alone would do. We therefore provide a way to combine multiple outer contours and fragments to measure the similarity between a query scan and the groundtruth scan and additionally give the possibility to use various descriptors for each. The overall distance between two scans is then given as a weighted linear combination of the respective global and partial results.

In the next chapter we will evaluate the capabilities of the discussed descriptors and matching procedures. We will also suggest on how to define certain parameters and what kind of organ parts or organ part fragments seem to hold the most distinctive information.

3.4 Experimental Results

In this section we present various experiments that are used to evaluate the proposed global and partial matching framework and additionally give insight on how it completes typical procedures of entomologists. Results not only show, which methods work best, but also might motivate entomologists to take things into account, that have previously been treated as less important.

The used scans of male genital organs for our experiments are taken from reference literature on that topic. In all our experiments we have looked at one to three subfamilies of *Noctuidae*, namely the *Eublemminae*, the *Herminiinae* and the *Hypeninae*. Each of those families are divided into several genera, that are themselves divided into to the actual species. The focus on

our experiments then lies on determination of the species of a query scan. Same can be done for the genus or even the subfamily, but given the same groundtruth set, those are just easier sub-problems of the former.

A main drawback for our experiments was, that most literature provides only one scan for each species. Illustrations of single specimens is often due to limited space in print matters like entomological books or journals. Additionally the illustrated specimen often represents the holotype (the so called *passport* of a species), which has been originally used to describe the species. For some species there exist several samples and different literature might also provide different scans, but those cases are unfortunately the exception. Thus creation of test and training sets based on scans from literature proved to be difficult. To increase the significance of our evaluation we therefore generated additional an set to compare left and right valves, as will be seen in Section 3.4.1.

3.4.1 Global Matching of Valves

For the first experiment, we use a set of 129 genital organs of male *Eublemminae* (40 samples), *Herminiinae* (58 samples) and *Hypeninae* (31 samples) from [41], each corresponding to a different species. We use this original set as a groundtruth and manually segment their valves. See Figure 3.8 for some exemplary scans. Note that a valve contour in this dataset usually consists of 1000 to 1500 points. We also generate a test set of the same 129 species by flipping those scans horizontally, thus making it possible to evaluate the matching of entire valves - right valves vs. left valves or vice versa. Table 3.1 shows the matching results for the global description methods discussed in section 3.3.2, when using flipped left valves as a test set and matching them against right valves as described in Section 3.3.4. If the left valve of a query scan is matched to the right one of the same specimen, the matching is interpreted to be done correctly. The percentage of correct matches are summarized in the *Best Match* columns of Table 3.1. On the other hand, the goal of our work is not necessarily to achieve perfect matching results, but to help entomologists to narrow down the list of candidate species. For that purpose, it is also sufficient if the correct specimen is among a few best matches. Therefore the *Top 3* columns indicate how often the correct specimen is one of the three best matches.

The first 3 rows in Table 3.1 correspond to the automatically calculated relational indices **vb:vl** using 1, 3 or 5 valve widths. Therefore all points are used and no sampling is done. For the other methods, we sample 30, 100 or 200 points. ED corresponds to a distance matrix using the Euclidean distance, while ID uses the inner distance. Θ 3.4 and Θ 3.5 are the angle matrices corresponding to Equation 3.4 and 3.5 respectively. For those matrices the parameter Δ has to be adapted according to the number of sampled points. We achieve the optimal results, displayed in table 3.1, when

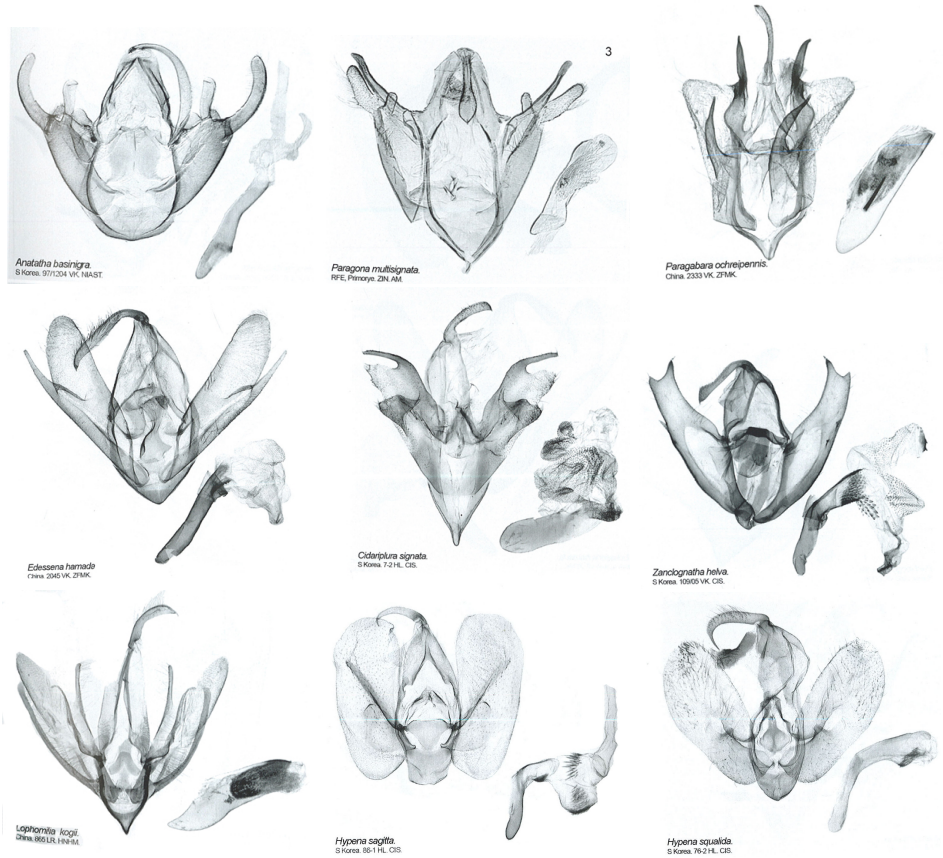


Figure 3.8: Some samples from the set, used for the experiment in Section 3.4.1. The first row corresponds to the subfamily of *Eublemminae*, second row to the subfamily of *Herminiinae* and third row to the subfamily of *Hypeninae*. Images taken from [41].

using $\Delta = 15, 30, 65$ for Equation 3.4 and $\Delta = 3, 10, 15$ for Equation 3.5. The lower values for Δ for the second matrix are due to the fact, that higher values would lead to a higher amount of zeros in Θ, thus resulting in a less discriminative description. To incorporate more relational indices into one description, we also use angle matrices along with the Euclidean distance matrix. The distance between two shapes is then defined as a linear combination of the individual distances. Shape Context and Inner Distance (ID) Shape Context are used as described in Section 3.3.2, with 12 bins for θ and 5 bins for $\log r$ as proposed by Belongie et al. in [6] and the binning frames are normalized according to the mean distance between points.

For the evaluation of the bag of features approach we use fragments as defined by Equation 3.6 and submatrices of Θ 3.4 or Θ 3.5 to describe them. Therefore in both cases, depending on the number of sampled points,

	Best Match			Top 3		
num points	30	100	200	30	100	200
vb1:vl	06.9%			17.1%		
vb3:vl	34.1%			60.5%		
vb5:vl	49.6%			62.1%		
ED	55.8%	56.6%	57.4%	71.3%	71.3%	72.1%
ID	47.3%	52.7%	52.7%	64.3%	69.8%	70.5%
Θ 3.4	65.1%	65.1%	65.1%	74.4%	72.1%	72.1%
Θ 3.5	62.1%	63.6%	61.3%	71.3%	72.1%	72.1%
Θ 3.4 + ED	65.9%	65.1%	65.1%	73.6%	72.1%	72.1%
Θ 3.5 + ED	62.8%	65.1%	63.6%	72.9%	73.6%	72.9%
SC	**73.6%**	75.2%	75.9%	**84.5%**	87.6%	86.1%
ID SC	61.2%	58.9%	62.8%	77.5%	72.1%	77.5%
CB Θ 3.4	65.1%	81.4%	85.3%	80.6%	93.1%	91.5%
CB Θ 3.5	60.5%	82.2%	82.9%	73.6%	93.1%	92.2%
CB SC	63.6%	**84.5%**	**86.1%**	82.2%	**94.6%**	**95.3%**

Table 3.1: Recognition percentages when test set of flipped left valves are matched against a groundtruth of right valves of the same specimens. Best Match columns indicate how often the correct specimen was retrieved as the one with the most similar right valve. Top 3 columns indicate how often the correct specimen was among the 3 best matches.

$\Delta = 3, 5, 5$ and the number of points defining a fragment is given by $2m+1$ with $m = 5, 15, 30$. Using those values for m correspond to fragments, that have length about $1/3$ of the entire contour, which has proven to be the most suitable choice to cover enough points to form a representative fragment. The codebook is generated using K-means clustering on all overlapping fragments of all right valves in the groundtruth. In the same manner K-means is used to generate a codebook of Shape Contexts. For both (fragment codebooks, as well as Shape Context codebooks) we use $K = 1000$, as increasing or decreasing K resulted in a drop of recognition rate. The respective results of the bag of features approaches are marked with CB in the last rows of Table 3.1.

Based on those results, one already gets some useful insight. First of all, it can be seen, that increasing the number of widths unsurprisingly drastically improves the recognition rate for **vb:vl** and as expected, using a distance matrix to complete these kind of relational indices further improves the recognition rate. However, using more than 30 points (900 point pairs) gives only slightly better results, while increasing the complexity and when using more than 200 points recognition rates will stagnate or even drop. This is the case, almost independent of the used description method and indicates that a lot of redundant information is incorporated. Also, using the inner

distance instead of the Euclidean one, leads to worse results. This strongly indicates, that articulation is in fact a distinctive feature of valves more often than not. More suprisingly is the fact that angle matrices outperform distance matrices, although there exist more distance based relational indices and measurements used by entomologists than angle based ones. However, angles seem to be more prominent features in the shape of a valve, and distance based relational indices are mainly used by entomologists, when the shape appears rather smooth, which is the case less often. Additionally, using a distance matrix along with an angle matrix also does not improve recognition results significantly if at all, indicating that an angle matrix already holds the most important information.

Shape Context outperforms structural measurement descriptors, most likely because of its ability to use angle and distance information in a compact manner, that eliminates redundant information, that can distort distance calculation. Shape Context is also less sensitive to point position uncertainties and thus less affected by inaccurate segmentations. The main benefit however results from the use of fragment or Shape Context codebooks when at least 100 points are sampled. Interpreting the valve as a combination of its *components* then proves to be the best choice. While relational indices and structural measurement descriptors describe certain aspects of the shape of a valve, the overall impression, that humans also get at first sight, is best described using the bag of words principle. In such a case there is almost no difference if an angle matrix is used to describe a fragment of C_s or Shape Context is used to describe the keypoints on C_s. We use these three description methods for codebooks, as those were the most promising, when used *directly*.

Figure 3.9 additionally illustrates the percentage of correct identification splitted into the three subfamilies to show, which subfamily profits the most from a certain method. It can be seen that the subfamily of *Hypeninae* has the lowest recognition rate in almost all cases. This is the case, because this family has the highest interspecies similarities concerning their valves. The valves of *Hypeninae* are often ellipse shaped without characteristic evaginations, which can e.g. be seen in figure 3.8 at the last two images. In such cases the exact ratio of only a few widths to the length is used by entomologists for identification. Using too many point correspondences then distorts the useful information, which is the reason for the lower recognition rate of the distance matrices compared to **vb5:vl**. Also, the reason why this family does not profit as much from Shape Context, is because Shape Context is less sensitive to point position uncertainties, but exact point positions are more important to distinguish very similar shapes. However it can also be seen, that the correct species is likely to be found, when a few best matches are taken into account, resulting in more similar results for all subfamilies, when the three best matches are observed.

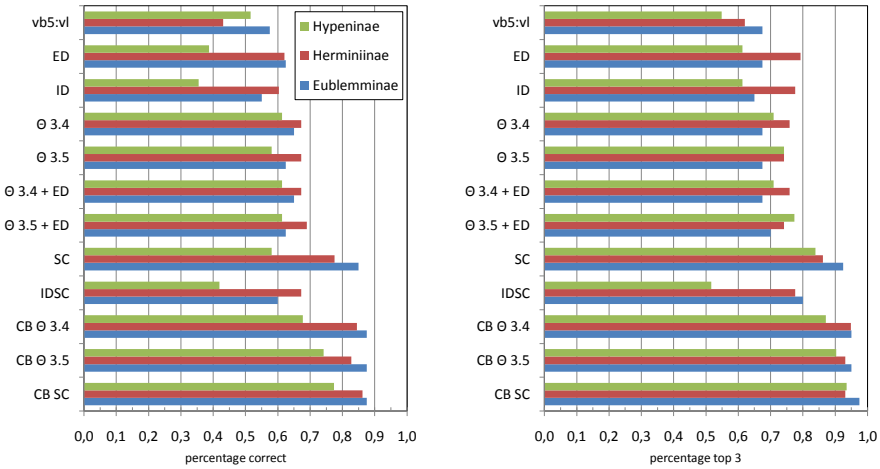
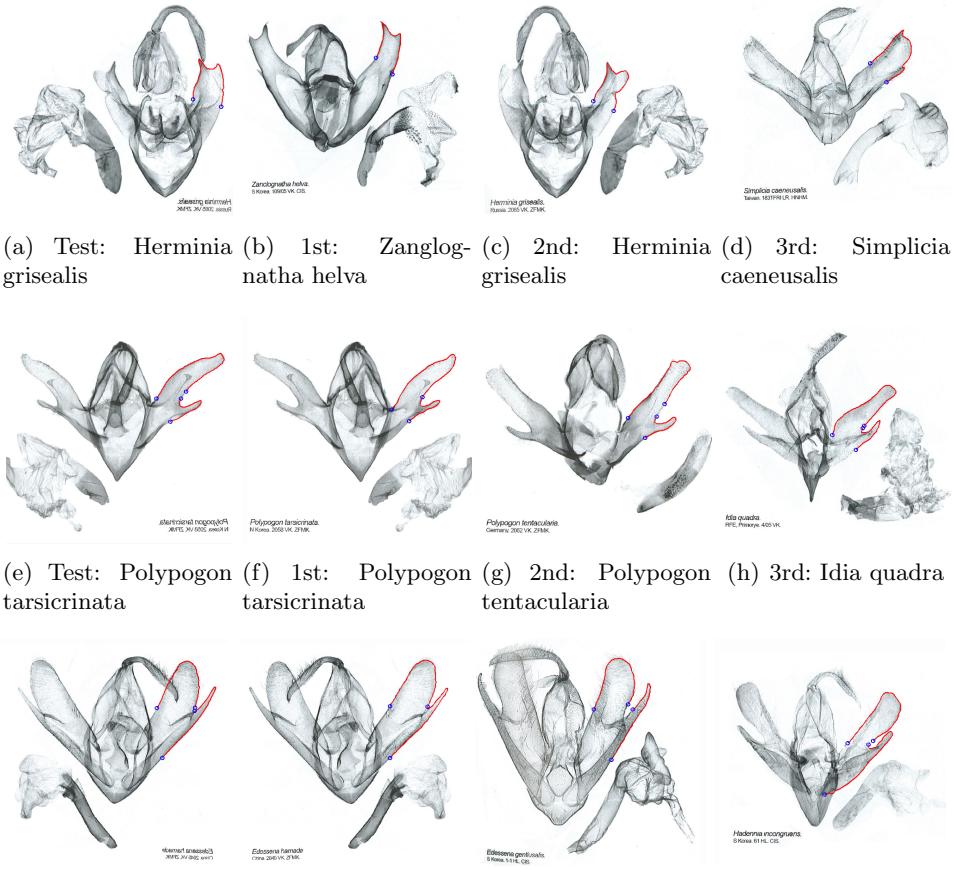


Figure 3.9: Recognition percentages splitted according to the three subfamilies *Eublemminae*, *Herminiinae* and *Hypeninae*. On the left, percentages of correct matches are shown, whereas on the right, percentages of how often the correct species was one of the three best matches are shown. Results are retrieved when 100 points are sampled.

3.4.2 Matching of Valve Fragments

For the second experiment we use the same groundtruth and test sets (129 images respectively) as for the experiment in Section 3.4.1, but this time match fragments of flipped, left valves against entire right valves. Therefore we extract between one and three segments of each left valve and use the fragment matching procedures from Section 3.3.3 and 3.3.4 to find right valves, that contain similar fragments. Samples from the groundtruth are ranked according to the sum of the distances of the query fragments and their best fits in the sample. E.g. if two fragments F_1, F_2 are specified for a query specimen, the best match in the groundtruth is the one, that contains two fragments (the best fits), whose summed distances to F_1 and F_2 are minimal. Additionally, only those groundtruth specimens are considered, where the single fragment distances are below a threshold. This is done to exclude specimens that might have a very similar fragment on the one hand, but there is no sufficient fit for another fragment on the other hand.



(a) Test: Herminia grisealis (b) 1st: Zanglognatha helva (c) 2nd: Herminia grisealis (d) 3rd: Simplicia caeneusalis

(e) Test: Polypogon tarsicrinata (f) 1st: Polypogon tarsicrinata (g) 2nd: Polypogon tentacularia (h) 3rd: Idia quadra

(i) Test: Edessena hamada (j) 1st: Edessena hamada (k) 2nd: Edessena gentiusalis (l) 3rd: Hadennia incongruens

Figure 3.10: In the left column fragments of valves in test scans are shown, as defined by user input. The other columns show the corresponding 3 best matches in the groundtruth dataset of closed valve contours. Blue circles indicate start and end points of fragments. Θ 3.5 along with $\Delta = 5$ and $d = 10$ has been used to achieve the shown results. Original images taken from [41].

The fragments for the test scans are not chosen randomly, but according to their seeming relevance. Also, we tried to avoid fragments, where one is unsure if it contains noise or segments from other organ parts or where the valve is damaged. See Figure 3.10 for illustration of chosen fragments as well as matching results. The sampling of contours and fragments is done in an equidistant manner, as given by Equation 3.7 with sampling distances $d = 5, 10, 15$. Those values would result in about 300, 150 and 100 points for an entire contour respectively. The recognition rates of the different approaches can be seen in Table 3.2. For angle matrices we use $\Delta = 10, 5, 3$.

	Best Match			Top 3		
d	15	10	5	15	10	5
Θ 3.4	70.5%	75.2%	**77.5%**	83.7%	86.1%	**88.4%**
Θ 3.5	**72.1%**	**76.7%**	**77.5%**	**84.5%**	**87.6%**	**88.4%**
SC	64.3%	63.6%	63.6%	74.4%	72.9%	70.5%

Table 3.2: Recognition percentages when test set of flipped left valve fragments are matched against a groundtruth of entire right valves of the same specimens. Best Match columns indicate how often the correct specimen was retrieved as the one with the right valve containing the most similar fragments. Top 3 columns indicate how often the correct specimen was among the 3 best matches.

To take multiple possible scales into account, one can do the sampling of query fragments with various d_i. E.g., if the groundtruth contours have been sampled with $d = 5$, we can sample a query fragment with $d = 4, 5, 6$ and choose the best fit in a groundtruth valve not only according to position but also according to the *best* sampling distance. Note, that this is not necessary for this experiment, as left and right valves of the same specimen have usually the same size. However, we also tested matching over multiple scales $d_i \in \{d-2, d-1, d, d+1, d+2\}$, which lead to similar results.

It can be seen, that the angle matrices are better suited for fragment matching, which is the reason because of their self containment ability. The Shape Context representation of a fragment in an entire closed contour on the other hand is affected by points that are not part of the fragment, thus leading to different descriptions, than for a similar query fragment. Also matching of just a few segments outperforms matching of entire valves in case of angle matrix description. This is explained by two circumstances. First, global valve matching depend on start point calculation, while fragments are *positioned* at every possible location on a groundtruth valve. In case of global matching of very similar shapes (like simple ellipsoids) slightly different start points can then lead to different results. Second, and more important, test fragments are defined, so they do not contain parts of a valve that are affected by noise, damages or distortions due to preparation, while those parts are contained in a global representation of the entire valve. However, using one to three explicitly defined fragments are themselves still outperformed by all overlapping fragments in combination with codebook representation.

3.4.3 Classification using multiple Organ Parts

The goal of the last experiment was to evaluate the importance of the individual organ parts to correctly identify a new specimem rather than comparing different methods. For that purpose we use the same groundtruth

	Best Match			Top 3		
organ part	U	L	R	U	L	R
uh:ul	02.6%	-	-	05.2%	-	-
vb5:vl	-	18.4%	23.7%	-	39.5%	39.5%
CB Θ 3.4	**28.9%**	**50.0%**	**63.2%**	**60.5%**	**68.4%**	**76.3%**
CB Θ 3.5	26.3%	44.7%	60.5%	47.4%	65.8%	**76.3%**
CB SC	23.7%	39.5%	50.0%	47.4%	68.4%	68.4%

Table 3.3: Recognition percentages for matching based on one of three specific organ parts - the uncus (U), the left valve (L) and the right valve (R). Best Match columns indicate how often the correct specimen was retrieved as the one with the most similar organ part. Top 3 columns indicate how often the correct specimen was among the 3 best matches. We use 100 points for sampling and the same parameter values as for the experiment in Section 3.4.1.

dataset as in the first two experiments (129 specimens of different species), but also manually segment the unci and aedeagi in the contained scans. The test set however differs this time. We searched a set of scans, that are taken from [75], and gathered those samples, for which also one sample exists in the groundtruth dataset, that is of the same of the same species but not the exactly same specimen. The resulting test set unfortunately consists of only 38 scans, all of which are from the subfamily of *Herminiinae*. However, the following experiment illustrates, how multiple organ parts can be used to determine the species of a new sample, instead of using only the valves. Also, for the first experiments intraclass variability was less an issue, as right and left valves of one and the same specimen usually won't differ much. This time, by using specimens in the test set, that are not exactly the same as those in the groundtruth dataset, we are also able to see how intraclass variability might affect recognition results. Table 3.3 shows those recognition results for individual organ parts. For that purpose we use the global matching procedures based on codebooks, that worked best for the experiment in Section 3.4.1 and compare them to relational indices that can be retrieved from those organ parts.

We have three observations. First the recognition rates drop significantly for matching based on valves, compared to the experiment in Section 3.4.1. This is most likely because of intraclass variability and the fact that only one specimen represents an entire class in the groundtruth. Another reason is, that for this test set it turned out to be harder to segment the boundaries of organ parts correctly, as there was more noise and damages noticeable in the scans. The second thing we notice is, that right valves lead to better recognition rates than left valves. This is pure coincidence and in general left and right valves hold the same kind of information. Again, this seems to be

	Best Match			Top 3		
organ part	LR	LRU	LRUA_f	LR	LRU	LRUA_f
rel. ind.	28.9%	36.8%	-	50.0%	60.5%	-
CB Θ 3.4	**65.8%**	**76.3%**	**76.3%**	78.9%	84.2%	84.2%
CB Θ 3.5	57.9%	68.4%	71.1%	**81.6%**	**92.1%**	**92.1%**
CB SC	57.9%	68.4%	68.4%	73.7%	86.8%	84.2%

Table 3.4: Recognition percentages for matching based on organ parts used together. Best Match columns indicate how often the correct specimen was retrieved as the one with the most similar organ part. Top 3 columns indicate how often the correct specimen was among the 3 best matches. L and R stand for left and right valve respectively and U and A_f stand for uncus and an optional aedeagus fragment. We use 100 points for sampling entire organ parts and the same parameter values as for the experiment in Section 3.4.1. For the aedeagus fragment we use the fragment matching procedure with $d = 10$.

more of a test set issue and also shows, that the results for such a small test set has to be interpreted cautiously. The third thing, that can be observed, is that recognition rates based on the uncus are smaller than those based on valves. This is rather unsurprising, as valves have a way more characteristic shape than unci. In fact, species of the same genus will often have almost identical unci. Also, the relational index based on the ratio of uncus height and uncus length (**uh:ul**) is by far too few information to correctly identify a specimen on its own.

A natural way to improve recognition rates, would be to add more scans for each species in the groundtruth dataset to cover a larger variety of each species. However, due to the fact that most of the time only one sample is available at the moment, this is part of future work on dataset creation. Another natural way to overcome possible variability within a class, is to use more organ parts together to overcome the shortcomings of each individual one. To do so, we simply sum the individual distances of each query organ part to their counterparts in groundtruth scans, and rank the candidate species according to that summed distances. One can also weight each organ part differently, but our experiments have shown that we achieve best results when uncus and valves are weighted the same or at least very similar. The recognition rates, when both valves are used together and the uncus is also added, are shown in Table 3.4. The used relational indices for comparison purposes are **vb5:vl**, **vl:hm**, **tel:hm:sao** and **vs:hs**, when both valves are used and we add **uh:ul**, **ul:hm** and **us:hs** to incorporate the uncus. Matching is the done by summing up the individual Euclidean distancen of relational indices. Note that for some of those indices the main genital corpus has to be extracted too, and those indices are therefore themselves not

based on just one organ part. The results confirm, what we have expected. The more organ parts can be used for classification, the better. In almost all cases we see an increase after adding another organ part.

In a last step to further improve the recognition rate, we wanted to incorporate the aedeagus. We already mentioned, that the entire aedeagus is not well suited for matching, because it is harder to determine its boundary. However it is sometimes possible to specify fragments that are part of the aedeagus for sure, if the aedeagus is prepared proper. Therefore we extracted such aedeagus fragments if possible, and considered it additionally to the uncus and the valves for classification. To match the aedeagus fragments we use various sampling scales $d_i \in \{5, 6, ..., 10\}$ and the fragment matching procedure, that corresponds to the descriptor that is used for the uncus and the valves in their codebook representation. However, a simple sum of the individual distances to take all organ parts into account is not reasonable, as different measurements are used for fragment matching and global matching. Therefore we divide the individual distances by a constant factor, so that each is in the range $[0, 1]$. After that, it is again possible to weight the individual distances and to add them up. For the methods involving angle matrices we weight uncus, valves and the optional aedeagus fragment the same, and for the Shape Context approaches it has proved to weigh the influence of the aedeagus slightly less.

Unfortunately, as Table 3.4 shows, the additional aedeagus fragment doesn't really improve recognition results. This is not the reason, because the aedeagus is unimportant generally. More likely the reason is, that the shape of the aedeagi are often very similar and there basically are just a few different types of aedeagi. Those types are further partitioned by entomologists observing small details, that seem not to be captured precise enough by the used descriptors. However, those are just assumptions, and exact reasons can only be given, if one can work on larger databases of sufficiently segmented aedeagi.

3.5 Conclusion and Future Work

During our work on the presented project, we motivated the use of proven shape concepts from computer vision by establishing a connection to measurements, that are typically used by entomologists to identify a butterfly by its genital organ. The whole project should be seen as a *proof of concept* and illustrate how a software solution could aid entomologists at their work. We already got some useful insights during our experiments, which we would like to summarize now:

- Although most relational indices are based on distances, we have seen that angle matrices seem to be more reliable than distance matrices.

This might motivate entomologists to explore some new relational indices based on angles.

- The Euclidean distance has proven to be a better choice than the inner distance during our experiments, when distances between points are measured. This strongly indicates, that articulation is an important feature and should not be considered impertinent when valves with evaginations are compared. But we also like to note, that this might vary from subfamily to subfamily and that future experiments on other datasets might therefore lead to the opposite conclusion.

- Entire organ parts should only be used, if one is able to sufficiently segment their boundaries. Otherwise it is safer to use just fragments of the boundary.

- Methods that are based on the existence of specific segments in an entire contour (bag of feature approaches) seem to outperform methods that are motivated by simple relational indices.

- Generally, the more organ parts or fragments can be used, the better. Although the valves seem to be the most distinctive organ part, adding the uncus further improves recognition rates. On the other hand, to incorporate the aedeagus one has to be more carefully and specific research seems necessary.

However, there still exists a large variety of possible fields of work for the future. For one, automatic segmentation of certain organ parts would make user input unnecessary and therefore would make the creation of groundtruth databases a lot easier. Also, though this concerns another field of work, the construction of a large collection containing scans of various specimens of the same species, would make it possible to perform more meaningful experiments. Such a collection could then also be used with learning procedures, to determine which shape features are especially characteristic for what kind of species. Future work might not necessarily have to consider itself with the identification of the species of a sample. Reliable automatic determination of the genus could already be of great help for entomologists.

Chapter 4

Classification based on Appearance of Wings

4.1 Introduction

The second project concerns itself with the classification of butterflies based on the color and the patterns of their wings, which is probably more easily comprehensible for non-experts. But to be able to do so, the wings of the observed species need to provide distinctive features, which is not the case for the previous discussed family of *Noctuidae*. The species treated in the present chapter are therefore rich of wing-texture. The difficulty in identifying those species lies in the fact, that wings of some different species look similar for the human eye at first sight, because of their repetitive patterns. It is a hard task to keep track of how often a certain pattern appears, but unfortunately the exact number of occurrences is the vital feature, that distinct one species from the other. And again the high number of different species makes it hard for experts to remember the exact pattern of a species. Thus identifying a specimen results in a high amount of research time, comparing it to various samples from literature. Therefore our goal was again to help experts by narrowing down the list of candidate species as good as possible using proven algorithms from computer vision, but this time the focus lies on appearance based description methods. In the next section we give some examples of butterfly wings from preserved specimens and use them to illustrate the complexity of the problem and how our work differs from others.

4.1.1 Dataset and State of the Art

When identifying a specimen based on their appearance, entomologists can use frontwings and hindwings as well as upper and underside of the wing. See Figure 4.1 for illustration of the terminology. Some samples e.g. are almost identical on the upperside of the wing, while the underside is *just* similar. An

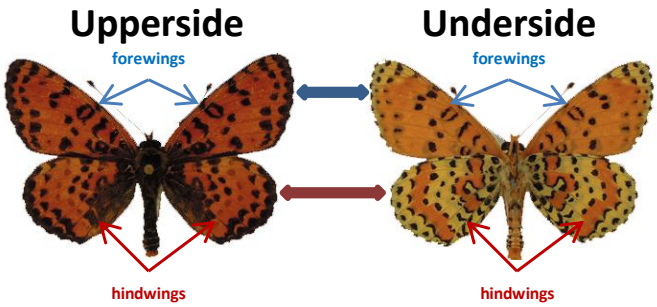


Figure 4.1: Upper and underside of a Melitaea didyma. Original images of the specimen provided by the Museum of Natural History Vienna.

entomologist may be able to determine the genus of a specimen by a simple look at the upperside, but to determine the correct species he has to analyse the underside very carefully. Thus, depending on the given specimen, the task itself and the knowledge of the expert, classification should be done using one specific or both sides of the wing. Regardless which part of the wing is observed, the main problems are interclass similarities and intraclass variability again.

Figure 4.2 shows some samples of preserved specimens in a the standardized view as used for our experiments and illustrates the challenge of such a classification task. Patterns seem to be very similar overall and entomologists then look for certain details, like the existence of a specific circle at a certain location of the entire pattern. If there exist various species with the same circle, then entomologists e.g. look for another pattern at a specific location and do so until they are certain of the underlying species. This stepwise procedure is done, according to a so called *key*. An entomological key is an instruction to stepwise reduce the number of candidate species. For each step, a feature is described textually, and depending if such a feature exist one or several times or even doesn't exist at all in a specimen, certain species are excluded and the following step may differ. A good key therefore, is a key that leads to correct classification way more often than not. Sometimes such keys are not described textually, but by images, exemplary showing a feature, instead of describing it. Such keys are called *pictorial keys*.

This motivates the use of an automatic classification system based on images of butterfly wings for various reasons. First, a pictorial key usually is more describing than textual keys, thus it seems natural to automatically compare images or certain parts of images with each other. Second, counting features can be done very efficient by a computer. And third, the idea behind keys and pictorials is very similar to hierarchical structures like vocabulary and decision trees, that are widely used in computer vision, and training algorithms could be used to determine optimal keys based on image

(a) Melitaea didyma (b) Melitaea didyma (c) Melitaea phoebe (d) Mellicta athalia

(e) Melitaea didyma (f) Melitaea didyma (g) Melitaea phoebe (h) Mellicta athalia

Figure 4.2: Three different species of brush-footed butterflies (family of Nymphalidae, subfamily *Melitaeinae*), where two specimens are shown of the same class to demonstrate intraclass similarity compared to interclass similarity. Images (a)-(d) show the respective upper sides, while images (e)-(h) show their bottom sides. Different species seem to be very similar, due to the repetitive occurrence of certain patterns, making it hard for the human eye to focus on the decisive features. Images are provided by the Museum of Natural History Vienna.

features. In our work we mostly concentrate on the evaluation of typical image features in terms of their suitability to describe wing patterns. We also make a proposal on how to use those features for classification, motivated by the principle of entomological keys. In the next section we give a brief overview of our solution. The basic principle is similar to the one for microscope scans of genital organs, but this time a specimen is not described by its shape, but by its appearance. Therefore the related work Section 4.2 will give an overview of methods to detect and describe regions of interest, that contain patterns.

As wings of butterflies can be very rich of texture, it is without surprise, that such datasets already have been used to evaluate several recognition systems. Wang et al. e.g. use a dataset of ten different butterfly species in [100], where each image shows a butterfly in its natural environment with no standardized view. The same is the case for the dataset of 7 different species used by Lazebnik et al. in [45]. While the challenges for those datasets occur as a result of the natural environment and the limited and changing view, the contained species have very different wing patterns. We, on the other hand, also want to distinguish species that have similar wing patterns, but therefore make use of the fact that an entomologist can record images of preserved specimens in a standardized view.

We also mentioned the insect identification system DAISY [74] in Section

2.2.1 and noted, that this system has already been used to identify British butterflies with success. However, their system needs the user to mark the boundary of a butterfly wing to work properly. We on the other hand want to give the user the opportunity to decide how much effort he wants to put in and aim for a system, that is able to do classification based on none to very small interaction. Also, while DAISY is based on measuring correlation between a query wing and a groundtruth wing, entomologists concentrate on the occurrence of certain local features. Given the success of local image features in various classification tasks, this inspires us to base our approach mostly also on the occurrences of features.

4.1.2 Outline

In Section 4.2 we give a brief overview of different techniques commonly used for recognition tasks, based on region of interest detection and appearance description. Then, in Section 4.3 we explain our system for butterfly identification in more detail. Therefore, we propose the use spatial pyramid representations and various local image features together with visual vocabularies, as well as a simple global approach based on global color histograms.

Evaluation of the proposed system is done in Section 4.4 and based on two different datasets of butterfly wings. We evaluate color histograms and a variety of SIFT variants as well as different region of interest detectors by comparing the resulting recognition rates.

Finally in Section 4.5 we summarize our procedure and discuss the experience gained from our experimental results. Furthermore we propose possible future work that addresses eventual shortcomings as well as additional features, that might be interesting for entomologists.

4.2 Related Work

The principle of a classification system based on local appearances, is the same than its shape counterpart. Regions of interest have to be extracted and described and are then used to classify a query image. However, this time we are not interested in outer contours of objects, but in patterns in the inner regions of a butterfly wing. For that purpose a large variety of proven methods exist to detect and describe such regions. A good discussion on this topic is given in [81]. In the following sections we give an overview of a variety of methods, some of which we will use as features to describe butterfly wings.

4.2.1 Region of Interest Detectors

The fundament of most appearance based recognition systems is the automatic detection of local regions of interest in an image, that hold appearance

information. Detection of such regions has been proven to be more reliable detected automatically, than contours, which often need some user guidance to give similar results. This is probably the main reason, why appearance based methods in general are more popular than shape based ones, although this is of course task dependant. The reliability of a detector is given by a hand full of properties that are desired [28]:

- **Repeatability**: Regions of interest should be able to be detected in different acquisitions of the same scene. Therefore, the detector has to be invariant to geometric transformations and to illumination changes.
- **Accuracy**: Location and shape of regions of interest should be estimated accurate.
- **Completeness**: Detected regions should cover as much structural information as possible to represent the object or scene.
- **Interpretability**: More of an optional property of a region of interest is its interpretability. Meaning, that regions, that represent well known structures (e.g. regions around corners and circles) should be favoured.

A very simple method to extract regions of interest, is to simply use *every other* point in the image and a fixed scale to define squared or circular image patches. This method is called **dense sampling** and in its basic implementation has no invariance properties. We already discussed the MSER detector in Section 3.2. This detection method is in fact frequently used for appearance based recognition also (e.g. in [78]). Therefore the center of a MSER represents its location and an ellipse is fitted into it to approximate its shape.

Because of the importance of the reliability of detectors, there exist several papers, that evaluate the strengths and shortcomings of different detection methods. Among those, the work of Mikolajczyk and Schmid [63] and Mikolajczyk et al. [65] is highly recommended. In the following we discuss some detectors in little more detail.

Harris Corner based Detectors

The probably most popular point of interest detector is based on the *cornereness* of points in an image - the **Harris Corner Detector** proposed by Harris and Stephens in [33]. Points are considered to be corners in the following way. The second moment matrix (structure tensor) T is calculated for every point p in the image I:

$$T(p) = \begin{pmatrix} I_x^2(p) & I_x I_y(p) \\ I_x I_y(p) & I_y^2(p) \end{pmatrix} \tag{4.1}$$

where $I_x(p)$ and $I_y(p)$ are first intensity derivatives of I at the point p. Analysing the eigenvalues of the structure tensor gives then insight on the cornerness of p. If both eigenvalues are high an the ratio between them is low, p is likely a corner. However, eigenvalue calculation can be circumvented. Harris and Stephens therefore propose to use the function $c(p) = det(T(p)) - k(trace(T(p)))^2$, with k usually being around 0.05. If that value is a local maximum and greater than a certain threshold, p is considered a corner. While corners can be detected very fast in that way, are rotationally invariant and have good repeatability properties, they don't give insight on scale and shape of the structure around them.

To provide a *characteristic scale* of a structure, one can combine the Harris detector with automatic scale selection methods, which are mostly inspired by the work of Lindeberg on scale-space representation [50]. Mikolajczyk and Schmid [63] e.g. use the Laplacian of Gaussian (LoG) as a scale selection criterion along with a scale adapted version of the Harris cornerness measurement $c(p)$ to detect points of interest and automatically determine their characteristic scale. It is given by

$$|LoG(p, \sigma_n)| = \sigma_n^2 |I_{xx}(p, \sigma_n) + I_{yy}(p, \sigma_n)| \tag{4.2}$$

where σ_n denotes the standard deviation of the Gaussian kernel used for scale space representation. Extrema over scale then correspond to blog like structures of the same scale. The overall procedure of detection and scale estimation is then basically divided into two steps. Keypoint detection using the scale adapted Harris measurement is done first, followed by characteristic scale estimation and location refinement through additional consideration of equation 4.2. The resulting detector (**Harris-Laplace Detector**) is then scale and rotation invariant.

However, the estimated scale has uniform size in all directions. To further improve the capabilities of a Harris based detection method, Mikolajczyk and Schmid [63] propose another extension to achieve invariance to affine transformations, that are commonly caused by viewpoint distortions. Consequently, in addition to keypoint location and scale, the affine shape of the surrounding region has to be estimated. This is done by iteratively approximating the transformation matrix U, that maps the image patch around a point p to an isotropic patch around the normalized point p^* and is usually represented by an ellipse. The overall procedure of the entire detection can then be seen as the result of various steps [65]:

- Scale adapted Harris detection delivers initial regions.
- Shape of a region is estimated by using the corresponding second moment matrix.
- The affine region is normalized to a circular region, for which location and scale are updated.

- Step 2 and 3 are redone until eigenvalues of the current second moment matrix are almost equal.

The resulting **Harris-Affine Detector** gives excellent results [65], but has the disadvantage of increased runtime in comparison to its scale invariant counterpart.

Hessian based Detectors

While first derivatives are used to detect corners, second derivatives can be used in a similar manner to detect blob an ridge like structures. For this purpose, the Hessian Matrix H is given by

$$H(p) = \begin{pmatrix} I_{xx}(p) & I_{xy}(p) \\ I_{xy}(p) & I_{yy}(p) \end{pmatrix} \tag{4.3}$$

where I_{xx}, I_{yy} and I_{xy} denote second, partial derivatives of the image intensity I. Local maxima of the determinant of $H(p)$ indicate blob like structures while additionally penalizing longer structures with small second derivatives in one direction. Like the basic Harris corner detector, a Hessian matrix based detector is also rotationally invariant, with the same drawbacks, but can be made scale and affine invariant in the same manner as its Harris counterpart [65].

Difference of Gaussian Detector (DoG)

Very similar to the Harris and Hessian based scale invariant approaches is the **Difference of Gaussian (DoG) Detector** proposed by Lowe [54, 55]. It is part of a well and subtle designed approach for feature detection and description - the **Scale Invariant Feature Transform (SIFT)**. Therefore Lowe uses the difference of Gaussian filtered images to approximate the Laplacian of Gaussian

$$D(p, \sigma_n) = (G(p, \sigma_{n+1}) - G(p, \sigma_n)) * I(p) \tag{4.4}$$

where $G(p, \sigma_n)$ $G(p, \sigma_{n+1})$ are Gaussian kernels, corresponding to two adjacent scales σ_n and σ_{n+1}. Initial points of interest are then retrieved as local extrema over location and scale of $D(p, \sigma_n)$. To ensure stability and repeatability, points with low contrast are discarded and edge responses are detected with the help of the Hessian matrix of D. Additionally orientations for each keypoint are assigned, based on the local image gradients. Thus, the detector is invariant to scale and rotation and gives similar results to the Hessian-Laplace detector, but the Difference of Gaussian can be computed faster than the Laplacian of Gaussian.

Edge and Intensity based Regions

While most detection methods try to estimate the scale or affine shape around an interest point and use that estimation to define a frame for further calculations, Tuytelaars et al. [91, 92] proposed to retrieve the frame more directly. In [91] they first detect corners and edges. Then, for every corner p they select the two nearest edges and walk along those edges and stop at points p_1 and p_2 respectively, according to a stopping criterion based on photometric quantities. A parallelogram is defined by $\overline{pp_1}$ and $\overline{pp_2}$ and serves as an affine frame of the region of interest.

The basic idea of the approach in [92] is similar, though not based on corners and edges. Instead, an initial point p is retrieved as a local intensity extrema and the algorithm studies the intensities along radially symmetric rays starting at p. Based on a specific intensity function, a stopping criterion is defined, leading to end points p_i for each ray. Finally an ellipse is fitted around those points to approximate the region of interest. While this method is faster than the edge based variant, both have rather long runtime.

Scale Invariant Feature OPerator (SFOP)

Two main problems may arise when dealing with low textured images. Either just a few keypoints can be detected or/and the detected regions of interest are hard to interpret. Therefore, unsatisfied with the results of the above discussed state-of-the-art region detectors in that regard, Förstner et al. [28] designed a method that is able to detect keypoints of different types and simultaneously classifying them. They extended a scale-space adapted version of Förstners previous work [27] on junction like keypoints to a variant that is able to detect additional types of spiral features, especially stars and circular structures. The main idea is to find points, where the consistency of an image region is locally optimal with respect to a spiral model. Thus, they are able to not only determine location and scale, but also the type of detected regions. The repeatability of the so called **Scale Invariant Feature OPerator (SFOP)** is similar to Lowe's detector [55], but it is able to retrieve more stable keypoints in images with low texture content. Runtime on the other hand is noticeable longer than in case of Lowes detector.

4.2.2 Description of Regions of Interest

While we already discussed how to proper describe the shape of object contours in Section 3.2, in this section we give an overview of state-of-the-art methods that describe the visual appearance of *inner* object regions. Detection and description are strongly related, as not every description method is suited to describe all kinds of regions of interest. Therefore work in that field often doesn't concern itself with just one of those two tasks, but with the combination of both. The overall process of detection and description is then

referred to feature extraction. However detection and description methods can be interchangeable, in which cases the descriptor often inherits properties of the used detector. In the following we explain some appearance based descriptors, independent of the used detector, in a little more detail.

Grayvalue and Color Statistics

A simple way to describe the image region around an interest point, is to use the intensity or color values of each pixel in the region and to concatenate them into one vector. The statistical measurement cross-correlation can then be used to determine the similarity between two descriptors. However, such a simple representation is not invariant to any image transformations. For that purpose the used detector has to estimate the transformation and the corresponding region must be transformed into a normalized *shape*.

Another way to describe an image region, is to use normalized, intensity or color histograms instead of raw values. To achieve additional invariance to specific illumination changes, different color models (e.g. RGB, Opponent, HSV) have to be considered. Burghouts and Geusebroek [13] as well as van de Sande et al. [94] discuss a variety of different color models and their properties in their respective work. Van de Sande et al. [94] also evaluate the corresponding color histograms in terms of their quality as descriptors.

We already mentioned image moments in Section 3.2 for shape description when used with a binary mask, corresponding to a region of interest. However, they also incorporate gray level information, when used with a gray level image. Also, so called color moments and color moment invariants can be used to extend the image moment theory for color models [66].

Spin Images

Johnson and Hebert originally proposed the use of so called **Spin Images** in a 3D shape based framework for multiple object recognition [37]. This idea has been adapted for 2d images by Lazebnik et al. for texture matching in [44]. They create a 2D histogram for a region of interest according to the intensities of pixels and their distance from the region center. They use 10 bins for the gray value binning and five bins for the radial distances. The resulting descriptor has size $5 \cdot 10 = 50$. Spin Images are by design invariant to rotation, and affine illumination changes can be addressed by histogram smoothing and normalization.

SIFT and Color-SIFT Descriptors

Along with the DoG detector, discussed in Section 4.2.1, Lowe also proposes [54, 55] the usage of a **SIFT** descriptor to represent the detected regions of interest. For a keypoint, its scale is used to determine a window around the point, which is then used for description. Image gradients and orientations

are calculated inside that window and weighted according to a Gaussian weighting function. Those values are then accumulated in 16 orientation histograms, each corresponding to one of 4×4 subregions of the window. Therefore the coordinates of the points inside the window and their gradient orientations are rotated relative to the previously estimated keypoint orientation to achieve rotation invariance. For the orientation histograms 8 bins are used, thus the final descriptor has a size of $4 \times 4 \times 8 = 128$. To reduce the effect of linear illumination changes, the vector is finally normalized to unit length and thresholding is done to handle non-linear illumination changes. Its descriptive power makes SIFT one of the most popular descriptors and has been used for a large variety of tasks (e.g. in [10, 60, 70]). It is scale and, if wanted, rotation invariant, but not invariant to affine transformations. This obstacle can be overcome when using the SIFT descriptor together with affine invariant detectors like e.g. Harris-Affine.

Strongly related to SIFT is the **Gradient Location-Orientation Histogram (GLOH)**, which is therefore also called **Extended SIFT**. The overall principle is the same, but Mikolajczyk and Schmid [64] use a radial and angular grid to divide an image region into 17 subregions instead of the original proposed 4×4 grid. For binning local gradients, they use 16 orientations, which results in a $17 \times 16 = 272$ descriptor, that is reduced in dimension to 128 by using Principal Component Analysis (PCA). It is shown in [64] that GLOH slightly outperforms SIFT.

However, those methods use only the intensity in a region, although there is no doubt, that color has high discriminative power. The challenge is to incorporate color, while remaining invariant to illumination changes. For that purpose different color models have been investigated by Burghouts and Geusebroek [13] as well as van de Sande et al. [94] and combined with SIFT descriptors. Basically they define various channels for a color model and calculate SIFT descriptors on each channel. E.g. the **RGB-SIFT** detects keypoints in the intensity image and then calculates the SIFT descriptor on each RGB channel around those keypoint positions. Consequently, the final descriptor for one keypoint has size $128 \times 3 = 384$. It is shown in [94], that RGB-SIFT is invariant to the most important illumination changes. However, depending on the desired invariance properties, different color models can be used.

Textons

Inspired by the work of Julesz [38] on human texture perception, Leung and Malik [48, 58] propose to use a filterbank to model the receptive fields of simple cells in the visual cortex. The filterbank consists of 48 [48] or 40 [58] respectively, and is able to detect edges, ridges and blobs at different scales and orientations. **Textons** are then defined in the following way. Filteresponses are usually calculated for every pixel in all intensity images of a

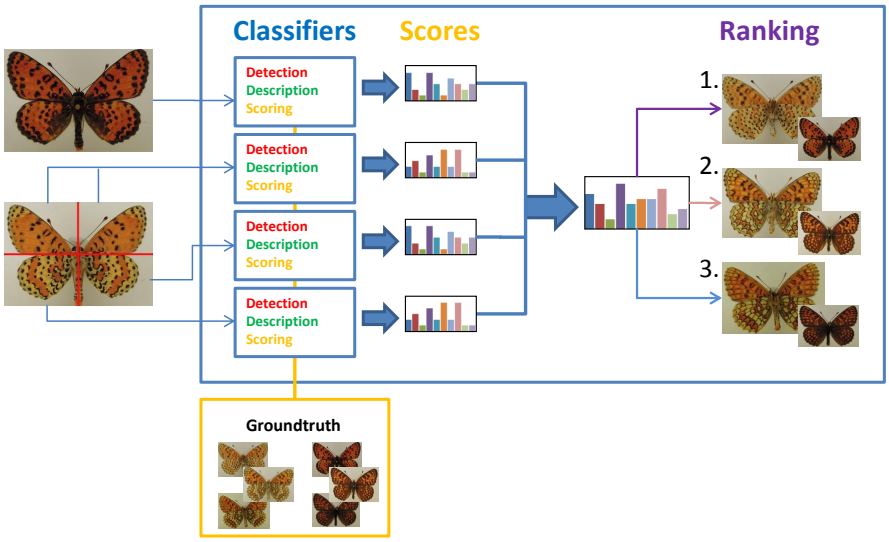


Figure 4.3: A sketch of the workflow of our system. Given images of the upper- and/or underside of the query specimen, subsystems based on different image regions and computer vision methods are used to collect voting evidence in form of scores, represented by bar diagrams. Those are then summed up to get an overall hypothesis. Original images of the butterflies provided by the Museum of Natural History Vienna.

training set. Kmeans is used to cluster the 48/40-dimensional filter response vectors and the resulting cluster centres are termed Textons, as they represent the *atoms* of texture perception. Such a codebook of Textons can then be used for recognition and detection purposes.

In other work, different filterbanks have been proposed to model receptive fields and to reduce dimensionality (e.g. [19, 83, 98]). However, in [97] Varma and Zisserman discuss if filterbanks are necessary for texture representation and come to the conclusion that in some cases intensities in the neighbourhood of a pixel are sufficient.

4.3 System

A drawback of a typical entomological key is the fact, that once an incorrect decision has been made, it will lead to the wrong species name. In most cases an entomologist realizes sooner or later that he probably made an incorrect decision at one point and backtracks the (pictorial) key to find its cause. Depending on the experts knowledge, processing a key already can be a work of hours and additional backtracking would even add to that. We therefore propose to avoid early decisions, that completely exclude

some species. In our framework we want to collect *voting evidence* instead of making *voting decisions* and use that evidence to create a list of candidate species, similar to the idea behind the work on stonefly identification by Martinez-Munoz [60].

Given a number of certain features x_i in a query image and a particular classifier, that models the probability $P(y_j|x_i)$, that the query image belongs to class y_j, can be seen as voting evidence. Therefore classes with a higher probability get a higher score s_j, than those with low. Using different features and different classifiers lead to various voting evidence, which can be summed up to give one overall hypothesis. Also evidence can be collected according to different parts of the wings, like forewing and hindwing and if images of both, the upperside and the underside, are available, they can both be used too. This basic principle is illustrated in Figure 4.3. It differs from the work of Martinez-Munoz [60], as they use a second stage classifier, while we simply sum up the individual results, mostly because of the current lack of data to train such a classifier. In the following we discuss the different techniques we incorporate in our framework to collect evidence.

4.3.1 Preprocessing

For all of the following procedures, the butterfly has to lie in a canonical pose and in front of a homogeneous background. Most of them also need the bounding box, containing the specimen as a reference frame to incorporate spatial information. In some cases, we even like the exact binary mask representing the butterfly regions. Therefore the background has to be removed. As we assume, that the background is always a homogeneous region, we use region growing to extract it. From the binary mask, corresponding to the butterfly, we are then also able to retrieve its bounding box. Alternatively we also incorporate the TV Segmentation tool [93] in case a homogeneous background, for whatever reason, can not be guaranteed.

4.3.2 Scoring based on Global Color Histograms

Looking at images of butterfly wings, probably the first thing that is recognized by humans, are their colors. We would then say e.g. the butterfly is orange although there might be parts that are not. The overall impression is, what leads to such absolute statements. To model such an impression we propose the use of color histograms. Loosely spoken, such histograms count how often a color occurs. However, interpretation of such a histogram depends on the color space of the image. We use some of the histograms discussed in [94] for comparison. The RGB histogram e.g. combines the 1D histograms of each RGB channel, but the RGB space is not similar to our sensation of colors. Closer to the human way of seeing is the HSV space, for which histogram binning is done according to the hue channel and weighted

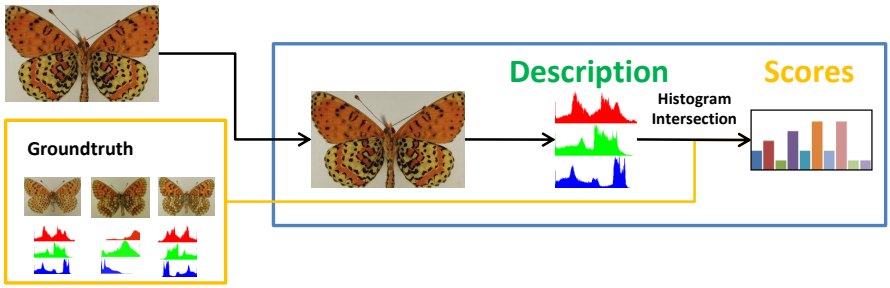


Figure 4.4: Scoring block of an overall classification scheme. Scoring is based on simple comparison of RGB color histograms. Original images of the butterflies provided by the Museum of Natural History Vienna.

by the corresponding saturation.

Other models we investigate are the opponent and transformed color histograms, each providing different invariances to illumination changes. See [94] for details. Observing the influence of illumination invariances is not only interesting, because images might not be acquired under the same light conditions all the time, but also because the wings of a specimen usually lightens in color after the specimens death. Also there exist species with a certain grade of intraclass variability in that regard, often depending on the gender of a specimen (e.g. males of a certain species might always be darker than their female counterpart). After a color histogram is calculated and normalized to uniform length, we use histogram intersection *hint* to compare the color histogram to the ones of the specimens in a groundtruth database. The actual scores are then given by

$$s_j = \frac{hint(h, h_j)}{\max_j(hint(h, h_j)} \tag{4.5}$$

where h is the color histogram of the query image and h_j the histograms of the groundtruth images. In that way, the groundtruth image with the most similar color histogram gets a score of one and all the other images get a score according to their ratio to the best match. See Figure 4.4 for illustration. However, looking at the initial examples for a wing dataset in Figure 4.2, even color as understood by humans is generally not sufficient to determine the species of a specimen and can at best give a rough indication. Consequently the results given by global color histograms are unsurprisingly very poor, as can be seen in Section 4.4. This motivates to a more subtle approach, which will be discussed in the next section.

4.3.3 Scoring based on Visual Vocabularies

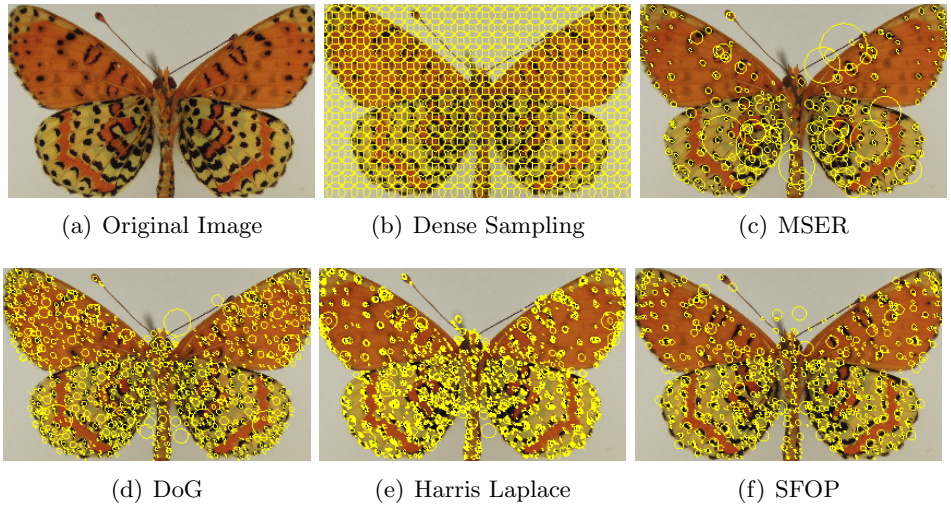
The subtle entomological procedure of identifying a specimens species is rather based on the occurrence of certain patterns on the wings, than on color alone. Therefore we propose to use region of interest detectors and descriptors to extract and describe those patterns together with visual vocabularies as an alternative to the simple color histogram approach. Therefore image features are detected and described first, spatial pyramids are used to incorporate spatial information and vocabulary trees are used to define visual words, which are then used in two different manners for scoring.

Local Image Features

We use a variety of different detectors and descriptors, for one to evaluate them in terms of their suitability to capture the essential parts in a butterfly wing, and for second to combine their corresponding voting evidence to get an overall result. As images are acquired in an standard canonical view, we do not need affine or even rotational invariance. In fact, exact orientation is a discriminative property of the patterns, that should not be ignored. However, scale invariance is necessary.

The first method we use to extract local image regions, is simple dense sampling. In order to select the sampling distance and the support regions around the resulting points, the scale of the butterfly has to be estimated. For that purpose we use the wingspan of the butterfly and define sampling distance and support region scale dependent on that value. Another detector we incorporate is the MSER detector[61], but instead of the ellipse, that is usually used to define the shape of the support region, we use the enclosing circle of a MSER. This is sufficient as we do not need any additional information about the region besides scale. Also we apply Harris Laplace [63], DoG [55] and SFOP [28] detectors without estimating orientation. Figure 4.5 shows the results of the various detectors applied to the underside of a butterfly.

Once regions of interest are detected, they have to be described in a suitable manner. We basically use two different kinds of features, that are both motivated by how humans distinguish butterflies: colors and patterns. For the first, we use local color histograms, which are similar to their global counterpart, but are only calculated for the support region of a keypoint. To represent patterns, we compare a variety of the gradient based SIFT descriptors. The first one is the original one, based on pixel intensities, as proposed by Lowe [54, 55]. However, the original SIFT descriptor makes only use of image intensities. Therefore, we also use different color SIFT variants, that already have been discussed in various work, to eventually achieve higher discriminative descriptors, by calculating gradients on each color channel independently. In case of the RGB-SIFT [94], the SIFT descriptors are cal-



(a) Original Image (b) Dense Sampling (c) MSER

(d) DoG (e) Harris Laplace (f) SFOP

Figure 4.5: An overview of the used detectors applied to the underside of a Melitaea didyma. Original image provided by the Museum of Natural History Vienna.

culated for each RGB channel, and concatenating them results in a 384 dimensional vector. The same is done for the HSV-SIFT [9], the Opponent-SIFT [94] and C-SIFT [13] according to their corresponding channels.

It is hereby noted, that though all those variants make use of color channels in one way or another, they are mainly designed to achieve higher discriminability, while remaining invariant to most important illumination changes, rather than incorporating color as humans perceive color. A good comparison of their invariances is given in [94]. However, the need for invariances to intensity changes is of course task dependant, and we will see in Section 4.4, which will work best for our problem setting.

A simple way to design a descriptor, that contains gradient information as well as color information, is by concatenating a SIFT descriptor with a local color histogram. The variant we use in that regard is the Hue-SIFT [95], that concatenates the original SIFT with the local Hue histogram as described above.

Spatial Pyramid Refinement

Usually, when local image features are detected, they are used together with the bag of features principle to build codebooks, that can then be used for classification. However, the basic bag of features approach does not incorporate any kind of spatial information, as only the distribution of features are used. While the occurrence of certain patterns in general is a very important indication, that a specimen might belong to a certain species, it is also very

(a) Original image, corresponding to spatial pyramid level 0.

(b) Spatial pyramid level 1, dividing the image into 2×2 subregions.

(c) Spatial pyramid level 2, dividing the image into 4×4 subregions.

Figure 4.6: Spatial pyramid partition of the underside of a Melitaea didyma. Especially level 1 is interesting, as it roughly divides the butterfly into left and right fore- and hindwing. Original image provided by the Museum of Natural History Vienna.

important, where the patterns occur. A butterfly with two points on the left forewing e.g. might not be of the same species as a specimen that has similar points on the right hindwing.

To incorporate spatial information, while not being dependant on precise user input, we make use of spatial pyramid representations [46]. Therefore the bounding box, containing the specimen is successive divided uniformly into subframes as illustrated in Figure 4.6. At each level of the pyramid, the features are then assigned to the subframe, containing them. Thus we obtain representations of parts of a butterfly wing independently. At level 1 of the spatial pyramid, where the frame is subdivided into 2×2 regions, this corresponds to a rough partition into left and right fore- and hindwings. The Spatial pyramid also gives the user the possibility to interact by defining which frames he wants to base his identification on, without subtle and precise outlining of wing parts. The following steps of codebook creation and classification are then based on the subregions individually, thus leading to different voting evidence.

Visual Vocabulary

Based on trainings data, we build our visual vocabularies independently for each level and subframe of a spatial pyramid. Therefore we make use of the hierarchical kmeans as proposed by Nister and Stewenius in [71]. A vocabulary tree is built in the following way. Features derived from the trainings data are clustered using kmeans, and the resulting clusters correspond to a node in a vocabulary tree and are then further partitioned until a certain level in the hierarchy is reached. At each node the cluster center is stored and the leaves hold the final visual words. See Figure 4.7 for illustration.

For one, the vocabulary tree serves as an efficient manner to search for approximate nearest neighbour visual words. Therefore a feature *travels* along

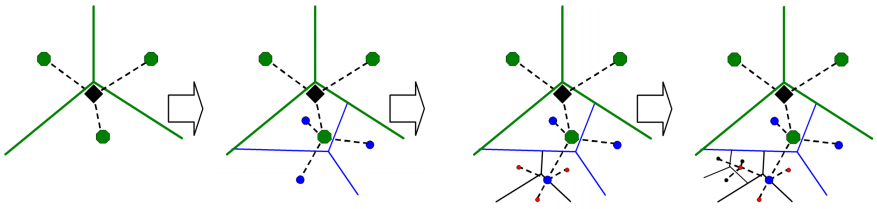


Figure 4.7: Illustration of hierarchical kmeans as used for vocabulary trees. At each step the data is partitioned into three clusters, represented by its center, and the procedure is repeated until the final tree has depth of four. Image taken from [71].

the tree according to its nearest center at each level of the tree, until it stops at a final leaf, respectively visual word. In that manner we can create visual word histograms for trainings specimens as well as query ones and can then compare those histograms using the χ^2 statistic. We then define the score as:

$$s_j = \frac{\tilde{s}_j}{\max_j(\tilde{s}_j)} \tag{4.6}$$

with

$$\tilde{s}_j = 1 - \frac{\chi^2(h, h_j)}{\max_j(\chi^2(h, h_j))} \tag{4.7}$$

where h is the visual word histogram of the query image and h_j are the histograms of the groundtruth images. In that way, again the groundtruth image with the most similar histogram gets a score of one and all the other images get a score according to their ratio to the best match. Due to the fact, that not always the same amount of features are extracted, the histograms are normalized to uniform length to make comparison reasonable.

However, using the tree in that manner, it *only* serves as an approximate nearest neighbour search scheme for a visual vocabulary. In the original work of Nister and Stewenius [71], they simultaneously use the vocabulary tree for scoring by using inverted file lists at each node and leaf. Such a list holds the indices of those training samples that have descriptors that belong to the corresponding node or leaf. Basically, for each feature in a query image, that passes certain nodes and reaches a specific leaf, *points* are distributed to the training images, depending on how often their features pass the same nodes and leaves. Using inverted file lists implements that kind of scoring very efficient and fast.

For our purposes, we only distribute scores according to reached leaves in the vocabulary tree. This is called flat scoring, contrary to scoring that incorporates all nodes, which is called hierarchical scoring. Therefore, during training phase, we assign each pair (i, j) of leaf index i and training image

label j weights in the following manner. For each descriptor, that belongs to j and arrives at leaf i we adapt the weight matrix W, which initially is a null matrix, according to

$$W(i,j) = W(i,j) + \frac{1}{w_j} \tag{4.8}$$

where w_j is the overall number of descriptors, that belong to j. This normalization is important, because training images can hold different numbers of features. On a side note, one can also see, that column j then represents the visual word distribution of training image j, normalized according to the L_1 norm. In a second step W is entropy weighted and the final scoring matrix S is then given by

$$S(i,j) = W(i,j) \log\left(\frac{N}{n_i}\right) \tag{4.9}$$

where n_i is the total number of labels that reach leaf i at least once, and N is the overall number of labels in the training dataset. In the same way the scoring can be expanded to be based on all nodes and not just the leaves, by doing the same procedure for each level of the vocabulary tree. However, to benefit from hierarchical scoring, that incorporates the nodes, a more subtle approach is needed and nodes have to be weighted differently than leaves. Nister and Stewenius discuss and evaluate several voting strategies based on leaves and nodes in [71]. They concluded, that weighing the nodes becomes more important, when the needed visual vocabulary would grow too large, in order to give the necessary distinctiveness. We tried using scoring nodes too, in the same manner as leaves, but it did not have a recognizable impact on our experiments. Most likely because of the rather small dataset we use, where we have only one training sample per class and thus also a different scheme for evaluation. Therefore we leave that kind of hierarchical scoring open for future work, when datasets become large and bigger visual vocabularies become necessary.

For a query image I the scoring is then done according to the matrix S. Let L be the number of visual words and D_i be the set of descriptors retrieved from I that reach leaf i. The score of label j is then given by

$$s_j = \frac{\tilde{s}_j}{\max_j(\tilde{s}_j)} \tag{4.10}$$

with

$$\tilde{s}_j = \sum_{i=1}^{L} |D_i| S(i,j). \tag{4.11}$$

We again use the maximum norm to force the label with the highest score to get one point and all the other labels a value according to their ratio

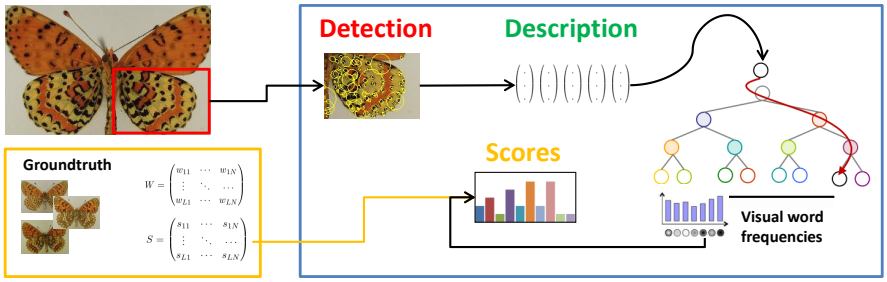


Figure 4.8: Scoring block of an overall classification scheme. First, MSERs are detected on the underside of the right hindwing and described (e.g. using the SIFT descriptor). Then a vocabulary tree is used to provide scoring evidence. Original images of the butterflies provided by the Museum of Natural History Vienna.

to the best score. The resulting scores serve as voting evidence. Figure 4.8 illustrates the here described workflow, which serves as one *block* of the overall framework as described in Figure 4.3. Note, that in the illustration only the underside of the right hindwing has been used together with MSER detection and the scoring matrix S. Based on other features and e.g. both forewings we can also use another block, that uses the χ^2 statistic along with visual word histograms, to get different voting evidence.

We also like to point out the main differences between using the vocabulary tree for scoring and the χ^2 statistic on visual words histograms. Though both are based on the occurrences of features, the former only rewards labels, that have similar features as the query image. However it does not explicitly *punish* those, that have features, that don't occur in the query image, though this circumstance is attenuated by the weighting procedure. On the other hand, the χ^2 statistic of two histograms is directly influenced by discrepancy between occurrences, but the main drawback is that distances to all the training histograms have to be calculated, which leads to higher computational effort during runtime when the training dataset is very large.

4.3.4 Final Score

Once voting evidence has been collected by using different combinations of color histograms, detectors, descriptors and/or by using different regions of the spatial pyramid of upper- and underside of a specimen, an overall score can be established. Let $s_{j,k}$ be the score of label j according to a scoring scheme k, the final score is simply a weighted sum of the individual ones:

$$s_j = \sum_k w_k s_{j,k}. \tag{4.12}$$

Weights w_k are introduced to control the importance of a specific scheme, e.g. to weigh results based on the hindwings more than those of forewings. In the next section we will evaluate different score schematics on two dataset of butterflies individually as well as in combination with each other.

4.4 Experimental Results

We tested our framework on two different datasets. The first dataset consists of various species of Austrian butterflies, and during implementation our framework was mainly tested on this dataset, as we believe that a framework that correctly identifies a specimen of the Austrian fauna would be a good illustration to show the benefits of an automated classification system for butterflies. However, we also evaluated our framework on a dataset of American butterflies of the family of *Hesperiidae* to see if it can be used on a completely different set of butterflies also.

4.4.1 Austrian Butterflies

For the first experiments we used a training set of 134 species provided from the Natural Museum of History in Vienna, originally produced for the Austria Forum [62], where each species is represented by one image of the upper and one image of the underside of a specimen in canonical pose. Thus the training set consists of only one sample for each species. The Natural Museum of History Vienna additionally provided us images of specimens, for which a sample of the same species is given in the training set. This set consists of 113 specimens of 17 different species from 3 (sub-)families (*Heliconiinae, Melitaeinae, Satyrinae*). We already gave some examples in Section 4.1.1 in Figure 4.2. We use that set as the test set for our experiments. We like to note, that only one sample per species in the training set is obviously not optimal to capture the variety of a species. Due to the current lack of data, we could have only expanded the training set by moving test samples to the training set. However, we choose not to, because we prefer a larger test set, and we could have expanded the training set only for a few classes, while others would still have been limited to only one sample. We evaluate the discussed techniques by comparing recognition rates, representing how often the correct species is the one with the highest overall score, and rates of how often the corrects species was among the Top X.

As the underside of the wing is more characteristic for most species, the first experiments, that compare classifiers based on different detectors and

descriptors, are based on images of the underside only. However, in the last experiment we use the most promising techniques on both sides of the wings.

Comparing Image Representations

In the first experiment we compare classifiers that are based on visual vocabularies using sift descriptors or local color histograms and classification based on global color histogram intersection. Therefore we use only images of the underside of the wings. We use the Harris Laplace detector for keypoint detection and spatial pyramids to incorporate spatial information. We also compare scoring using the χ^2 statistic of two visual word histograms and flat scoring based on inverted file lists as given by Equation 4.10. For the global and local RGB, Opponent (OPNT) and transformed color (TC) histograms, we use 15 bins per channel and for the Hue histogram we use 36 bins for the Hue channel, with each pixel belonging to a certain bin is additionally weighted by its saturation.

For building the vocabularies we use a tree depth of 4 levels, with 10 cluster centres at each level, which results in 10000 leaves/visual words. We also tested a tree depth of 5, which resulted in better recognition rates, when no spatial pyramid is used. However, using pyramids with a larger vocabulary then did not improve recognition rates, which is the reason why we recommend the smaller trees together with spatial pyramids.

We tested spatial pyramids at 3 levels. Level 0 corresponds to classification based on the entire image only. Level 1 is based on the entire image and on its subdivision into 2×2 rectangles. Level 2 further divides those rectangles into 4×4 image regions. The classification for level 2 is then e.g. given by the individual scores based on the entire image, the 2×2 and the 4×4 sub-rectangles. Those scores are simply summed up, without additional weighting or normalisation. We also tried out to weigh each level of the pyramid differently (e.g. $1/4, 1/4, 1/2$ as in [46]), but generally best results in automated tests have been achieved with no additional weighting. However, when an entomologist has a *feeling*, that e.g. the forewings of a specimen might not be very characteristic compared to the hindwings, weighing the corresponding regions in the spatial pyramid representation differently might further increase the chance of identifying the specimen correctly. We also made use of spatial pyramids together with global color histograms. Therefore a global color histogram is simply calculated for every region in the spatial pyramid representation and histogram intersection is done for each region independently, and individual scores are later summed up to give the overall score.

The results of our experiments are shown in Table 4.1 and 4.2. We recognize that color histograms, global as well as local ones, perform very poorly. While it is understandable, that they do not achieve the high discriminative power of SIFT descriptors, it is surprising, that the results are not

Global	Histogram Intersection		
pyramid levels	0	1	2
RGB Hist	02.65%	05.31%	05.31%
HUE Hist	03.54%	03.54%	07.08%
OPNT Hist	05.31%	06.19%	08.85%
TC Hist	**09.73%**	**08.85%**	**10.62%**

Table 4.1: Recognition rates according to how often a query image was assigned to the specimen of the same species in the training set using global color histogram intersection and spatial pyramids up to level 2.

Figure 4.9: The undersides of two different specimens of the species *Eurebia ligea*. The images illustrate the variability in terms of brightness and color that can occur within a species. Images provided by the Museum of Natural History Vienna.

better. There are various reasons, that could explain those results. First of all, though the images in the training set have originally also been made by the Natural Museum of History in Vienna, they could not guarantee the same acquisition conditions for the later made images in the test set. Also, there exists intraclass variability concerning colors (e.g. a female specimen might be lighter than a male one) and the color of a specimen might additionally fade over time. See Figure 4.9 for illustration. As HUE histograms and transformed color histograms have the most invariances to intensity and color changes [94], this would also explain, why they perform slightly better. The fact that only one training sample is available per species further affects this circumstance.

We recognize a similar effect for SIFT descriptor variants. The variants, that achieve best results (original SIFT, Opponent-SIFT, RGB-SIFT), are those with most invariances to light intensity and/or color changes. The greatest benefit in recognition rates either way results from incorporating more levels of the spatial pyramid. We also see, that the χ^2 statistic outperforms the flat scoring given by the vocabulary tree. This can be explained

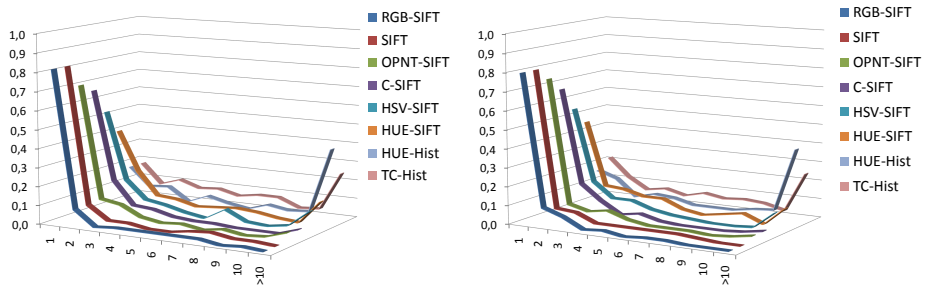
Local	TF-IDF Flat Scoring			χ^2-Statistic		
pyramid levels	0	1	2	0	1	2
RGB Hist	05.3%	07.1%	09.7%	04.4%	05.3%	07.1%
HUE Hist	**10.6%**	12.3%	**17.7%**	07.1%	10.6%	15.9%
OPNT Hist	05.3%	09.7%	10.6%	05.3%	07.1%	09.7%
TC Hist	**10.6%**	**14.2%**	**17.7%**	**14.2%**	**18.6%**	**21.2%**
SIFT	44.3%	**69.1%**	**81.4%**	58.4%	74.3%	**79.7%**
HUE-SIFT	16.8%	31.9%	39.8%	30.9%	35.4%	45.1%
HSV-SIFT	21.2%	41.6%	52.2%	20.4%	44.3%	53.9%
OPNT-SIFT	49.6%	58.4%	69.9%	53.9%	65.5%	73.5%
C-SIFT	36.3%	47.8%	65.5%	31.9%	45.1%	66.4%
RGB-SIFT	49.6%	**69.1%**	**81.4%**	61.9%	**77.9%**	**79.7%**
All SIFT	**53.9%**	**69.1%**	76.9%	**62.8%**	76.1%	**79.7%**

Table 4.2: Recognition rates, according to how often a query image was assigned to the specimen of the same species in the training set based on local features. In all cases the Harris Laplace detector was used for keypoint extraction, as well as spatial pyramids up to level 2. Additionally TF-IDF scoring and χ^2 statistic based classification was compared. *'All SIFT'* corresponds to the sum of the individual scores given by each SIFT descriptor.

by the fact, that missing features are not explicitly punished by the second. However, this advantage of the χ^2 statistic vanishes, when more levels of the spatial pyramid are used. Therefore we recommend TF-IDF scoring, which is faster (though for the current small datasets time benefit was not a concern) and gives similar results when used with spatial pyramid representation of level 2.

In order to combine all SIFT variants we simply summed up the normalized individual scores given by each descriptor. Again, this slightly improves recognition rates, when no subdivision of the image is done and the benefit vanishes in most cases when using spatial pyramids. Therefore SIFT or RGB-SIFT alone already achieve best performances and other variants do not give additional insight. We noticed that in case of no spatial pyramids being used, every single SIFT descriptor was necessary to achieve an improvement. Even by ignoring the poorly performing HUE-SIFT, the combination of all the others would lead to a small drop from 53.9% to 51.5%. However, the overall conclusion is, that using the *right* descriptor with spatial information improves recognition rates more than using different descriptors together.

As it would already help an entomologist, if the list of candidate species of a specimen is narrowed down to a reasonable size, we also tested how often the correct species of a query sample is among the Top 10 in terms of their score. Figure 4.10 gives an illustration in that regard. Therefore the x-axis corresponds to the position in the candidate list, ranging from place



(a) Results based on TF-IDF flat scoring. (b) Results based on the χ^2 statistic.

Figure 4.10: Illustration of how often the correct training image was at a certain rank or outside the Top 10 according to their score. Scoring was done using spatial pyramid level 2.

1 to 10 and > 10 if the species was outside of the Top 10. The y-axis then represents how often the correct species occupied a certain position in the candidate list. Consequently good curves have high values at position 1 and flatten as they approach > 10. Ideally the value at > 10 is 0%, in which case the correct species was among the Top 10 for all test images. For the illustration in Figure 4.10 we used spatial pyramid levels up to 2 and ignored the lesser performing color histograms for clarity reasons. The results show, that SIFT and RGB-SIFT were able to place the correct species in the Top 10 for all test images.

In the original work of Nister and Stewenius [71], they had four images per class in the training set and evaluated how many of the correct class were among the Top 4. We would have liked to rebuild that setup, but as only one specimen per class is given in the training set, we choose the above described additional evaluation variant.

Comparing Detectors

An essential part for describing a butterfly is to *capture* those patterns on the wings, that are characteristic. Different keypoint detectors deliver different regions of interest, and the purpose of the following experiment was to determine which detectors lead to the best recognition rates. Again this is done on just the underside of the butterfly wing, as they deliver the most characteristic features for our dataset, according to entomologists. The detectors we use are dense sampling, MSER, Harris-Laplace, DoG and SFOP. For description we use the RGB-SIFT method and otherwise the same setup for pyramid representations and vocabulary building as in the experiment concerning descriptors.

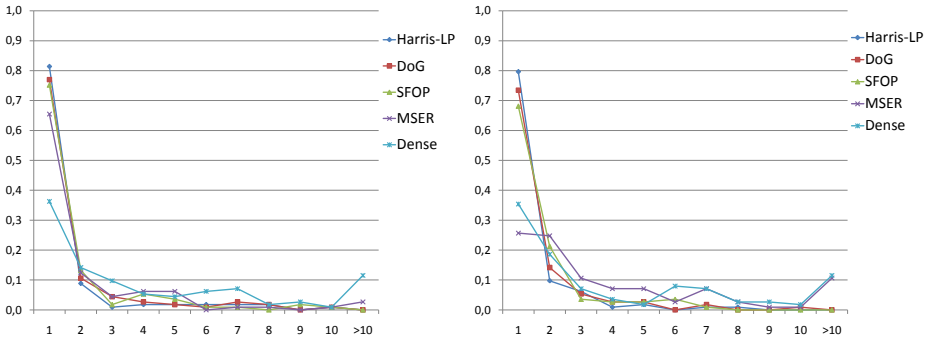
The results can be seen in Table 4.3. We make similar observations as for

the first experiment. The usage of spatial pyramids is important to achieve good results and combining the results given by each detector separately improves recognition rate, when no spatial refinement is done. The benefit however is recognisable greater than in case of combining different SIFT descriptors. Different detectors deliver different regions of interest and therefore are more likely to complement each other. However this benefit again vanishes, when using more levels of the spatial pyramid. We therefore believe that recognition rate of 81.4% given by the Harris-Laplace detector together with the RGB-SIFT description, can only be improved by adding more training images per class.

We make two additional observations concerning dense sampling and MSERs. First, dense sampling doesn't benefit from spatial pyramid representations and is generally less qualified for our task. This illustrates the importance of detecting characteristic regions on a butterfly wing, instead of more or less *randomly* sampling them. Second, we notice a significant drop of recognition rate, when MSERs are used together with spatial pyramid level 2 and the χ^2 statistic for classification. We believe this is related to the fact, that MSERs detectors deliver a rather small amount of regions of interest compared to most other detectors. Therefore it is more important, that those regions lie in the correct subrectangle and it seems that a subdivision into 4×4 regions is too specific in this case and the χ^2 statistic additionally punishes then missing features in a rectangle. Figure 4.11 again additionally illustrates how often the correct species was among the Top 10 matches.

RGB-SIFT	TF-IDF Flat Scoring			χ^2-Statistic		
pyramid levels	0	1	2	0	1	2
Dense	37.2%	36.3%	37.2%	29.2%	34.5%	35.4%
MSER	46.1%	56.6%	65.5%	55.7%	67.3%	25.7%
Harris-LP	49.6%	**69.1%**	**81.4%**	61.9%	**75.2%**	**79.7%**
DoG	52.2%	63.7%	76.9%	60.2%	68.1%	73.5%
SFOP	41.6%	68.1%	75.2%	50.4%	68.1%	68.1%
All	**65.5%**	**69.1%**	76.9%	**69.1%**	71.7%	73.5%

Table 4.3: Recognition rates according to how often a query image was assigned to the specimen of the same species in the training set based on local features. In all cases the RGB-SIFT descriptor was used for descrription, , as well as spatial pyramids up to level 2. Additionally TF-IDF scoring and χ^2 statistic based classification was compared. *'All'* corresponds to the sum of the individual scores given by each detector individually.



(a) Results based on TF-IDF flat scoring. (b) Results based on the χ^2 statistic.

Figure 4.11: Illustration of how often the correct training image was at a certain rank or outside the Top 10 according to their score. Scoring was done using spatial pyramid level 2 and RGB-SIFT was used for description.

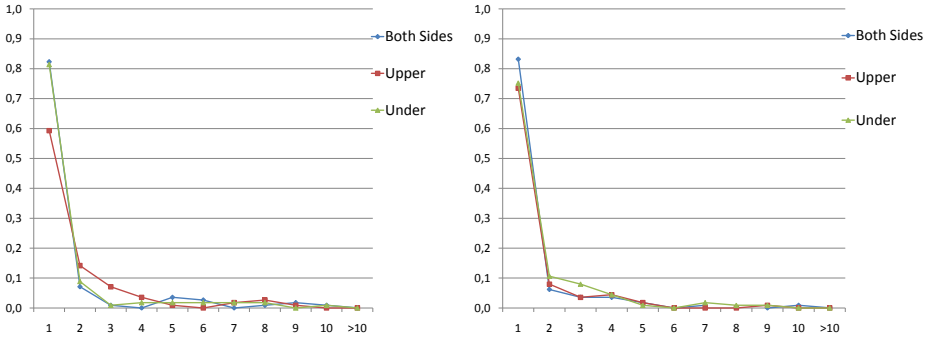
Using Upper- and Underside

In most cases butterflies that have similar uppersides, differ on their underside, if they are not of the same species. On the other hand, if they are very similar on the underside, they will also be very similar on the upperside. However, as this observation is based on the human perception of butterfly wings, we investigated, if this also holds true for our framework or if the uppersides can complement the undersides.

We use TF-IDF scoring, spatial pyramids of level 2 and the RGB-SIFT descriptor as this setup gave best results for undersides of butterfly wings. However we tested three different detectors (Harris-Laplace, SFOP, MSER) to see if uppersides of wings need different detectors to extract relevant regions of interest. Additionally we summed up the scores of results based on upper- and underside in such a setup to see if they complement each other. The results can be seen in Table 4.4 and Figure 4.12. We notice a smaller discrepancy regarding the results based on different detectors when used on the uppersides and also a benefit when using all three detectors together. We finally also get a small improvement by using upper- and underside, instead of only one side, but as expected the underside turns out to *hold* more characteristics and the gain from incorporating the upperside is rather small. We also like to note a general observation, which holds true independently of the used method or if classification is based on upper- or underside of the wings: recognition rate for specimens of the subfamily of *Satyrinae* (like the specimen in Figure 4.9) was smaller than recognition rates for the other two families. This is due to the fact, that those butterflies have little amount of pattern compared to other families. The experiments on the second dataset therefore concerned themselves with more species with small amount of wing-patterns.

RGB-SIFT	Under	Upper	Upper + Under
MSER	65.5%	59.3%	79.7%
Harris-LP	**81.4%**	59.3%	82.3%
SFOP	75.2%	60.2%	77.9%
All	75.2%	**73.5%**	**83.2%**

Table 4.4: Recognition rates according to how often a query image was assigned to the specimen of the same species in the training set based on local features. Spatial pyramid representation of level 2 was used together with RGB-SIFT descriptors and TF-IDF flat scoring. Classification based on all three detectors combined (*'All'*) and classification based on both wing sides were achieved by summing up the individual scores.



(a) Results based on Harris Laplace key-points only

(b) Results based on Harris Laplace, DoG, MSER and SFOP regions of interest

Figure 4.12: Illustration of how often the correct training image was at a certain rank or outside the Top 10 according to their score. Scoring was done using spatial pyramid level 2 and TF-IDF flat scoring. RGB-SIFT was used for description.

4.4.2 Hesperiidae of America

Though during our work we mainly experimented on Austrian butterflies, we made an additional experiment with a complete different set of butterflies. This should be seen as a side *project*, as we did not specifically tune our framework for this dataset, but were interested, if our framework might give good results anyway. According to experts specimens of the family of *Hesperiidae* are considered to be hard to identify their species. A lot of species e.g. are simply brown or dark orange and hold just a few, small geometric forms. Those species are then distinguished by the exact amount, position, shape or orientation of those forms. See Figure 4.13 for some examples. The Natural Museum of History in Vienna selected a variety of different species from that family from the large collection of *Butterflies of America* [101].

(a) Quas. fieldi (b) Quas. nayana (c) Thesp. dalman (d) Nyct. nyctelius

(e) Quas. fieldi (f) Quas. nayana (g) Thesp. dalman (h) Nyct. nyctelius

Figure 4.13: Four different specimens of American Hesperiidae. Top row corresponds to their uppersides, while bottom row corresponds to their undersides. Images taken from *Butterflies of America* [101].

Out of those, we then used those species for which more than three images in canonical pose were available, to build a training and a test set. The resulting training set consists of three images respectively for upper- and underside per species and 48 different classes. We therefore chose the six training images, such that the intra class variability is captured as good as possible. The test set then contained the remaining images, resulting in one to four test specimens per class and 103 specimens overall.

As we were able to obtain 3 training images this time, we again investigated some *color* descriptors that didn't perform as good as RGB-SIFT or SIFT for the Austrian butterflies. Therefore we chose the Opponent-SIFT variant as an additional color SIFT variant to RGB-SIFT and local transformed color histograms for comparison. All our results were achieved by using MSER, Harris-Laplace, DoG and SFOP detectors. We noticed that using all four detectors together outperformed any variant using just a subset. Again different detectors are combined by summing up their resulting individual scores. For the scoring we use only flat TF-IDF scoring and spatial pyramids of level 2 ($1 \times 1 - 2 \times 2 - 4 \times 4$ rectangles). The recognition rates can be seen in Table 4.5. Also Figure 4.14 shows how many training images of the correct species were among the Top 3 matches on average.

We notice, that local transformed color histograms have significant better recognition rates as for the first experiments and seem to benefit from the fact, that more training images are available. This confirms our assumption, that to make use of color, the variability of a species in that regard has to be captured by various training images. We also see that Opponent-SIFT outperforms RGB-SIFT for this dataset, which also indicates that we are able to achieve higher discriminability when we use a descriptor with fewer

	Under	Upper	Upper + Under
TC Hist	66.9%	63.1%	79.6%
OPNT-SIFT	87.4%	77.7%	84.5%
RGB-SIFT	86.4%	73.8%	83.5%
All	**91.3%**	**78.6%**	**89.3%**

Table 4.5: Recognition rates according to how often a query image was assigned to the specimen of the same species in the training set based on local features. Classification is done by TF-IDF flat scoring and by using spatial pyramid representation of level 2. In all cases we use MSER, Harris Laplace, DoG and SFOP detectors combined. We compare three different descriptors and also combine their results by summing up the individual scores, indicated by *'All'*.

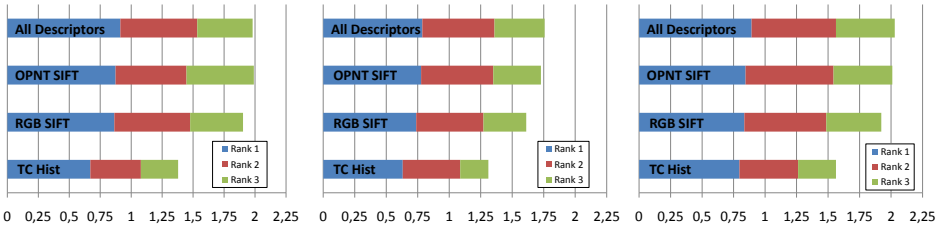
color change invariances, as long as the intra-class variability of color is captured by the training images. We also get an improvement when using the three mentioned descriptors together The final observation is, that results based on the upper sides doesn't necessarily improve the results that are given by the underside alone. Again it seems, that using both sides only improves recognition rates if there is *room* for improvement. Loosely spoken if the underside already gives *enough information*, as it is the case for the SIFT variants, using the upperside additionally likely results in a small drop in recognition rate.

Considering the fact, that this group of butterflies is considered to be hard to classify, the results look very promising. However, the dataset is still very small and the overall number of different species of the family *Hesperiidae* goes beyond 4000. Thus it will be interesting to see, if recognition rates can be remained, while the number of species in the dataset increases.

4.5 Conclusion

We presented a framework for butterfly identification based on the appearance of their wings. Therefore we proposed to collect voting evidence in terms of scores and to rank training specimens according to one or several scores combined. We compared individual results based on a simple global approach, as well as different visual vocabularies [86] based on various detectors and descriptors. An essential part of our system is the use of spatial pyramids [46] to incorporate spatial information, that is necessary to improve recognition rates. We made several observations, that are summarized in the following.

Unsurprisingly, any classification scheme based on local image features outperformed classification based on global color histograms. Additionally, even local color histograms performed poorly, especially compared to the in-



(a) Results based on the wing undersides only.

(b) Results based on the wing uppersides only.

(c) Results based on the wing upper- and undersides.

Figure 4.14: Illustration of how many out of three correct training samples were among the Top 3 on average. Additionally the colored parts indicate the relative value of how often a correct training sample was at a certain rank (blue = 1, red = 2, green = 3). TF-IDF was used together with spatial pyramids of level 2. All results are based on four detectors (MSER, Harris Laplace, DoG, SFOP) and separated by the used descriptor. *'All Descriptors'* represents the results based on the sum of the scores of the three individual ones.

vestigated SIFT variants. The main reason therefore is the variability within a class in terms of color compared with the current lack of trainings data. This is confirmed by the tests on American *Hesperiidae*, where three training images per species were available, which resulted in higher recognition rates for local transformed color histograms. Using the χ^2 statistic to compare visual word frequencies gives better recognition rates than flat TF-IDF scoring, if few spatial pyramid levels are used. This is the case, because the χ^2 statistic punishes missing features.

However, this benefit vanishes, when the image is parted into 4×4 rectangles, as local uncertainties then affect the χ^2 statistic more because of the same reason. Therefore, we propose flat TF-IDF scoring together with spatial pyramid representations of level 2. When only one training image was available SIFT and RGB-SIFT gave the best recognition rates, due to their invariance to intensity changes. However, the second experiment indicated, that when the training set becomes larger, other color SIFT variants or even local color histograms might become interesting again. Harris Laplace, DoG and SFOP detectors seem to be a better choice to extract characteristic wing patterns than MSER detectors. Most likely this is the case, because those detectors generally deliver more keypoints and loosely spoken, the more features, the better the wing pattern can be described. However, results based on uppersides showed smaller discrepancy in recognition rates and improvement when detectors are combined with each other. This indicates that different kind of wings respond differently to certain detectors and they can complement each other. We therefore believe, that any identification system based on butterfly wings should incorporate a variety of detectors.

Collecting scoring evidence by using various detectors, descriptors and images of both sides of the wings often improves recognitions rates, that are achieved by only one individual classifier. However, if one method already gives very good results, it becomes more likely that it is *dragged down* by other results. Therefore, when in usage, we propose to use our framework in the following way. First get results based on the best method (e.g. for Austrian butterflies this would be Harris Laplace detector and RGB SIFT description). If those results already make it possible for the user to identify the specimen, the goal has been achieved. If not, he can collect scoring evidence based on the next best method, and the scores are added to the previous ones, and so on.

The above proposal is suitable for the current framework, where it is easy to incorporate various classifiers and combine their results by a late fusion process. Due to lack of trainings data, we were not able to make use of well known learning algorithms. Those can be designed to make the above mentioned stepwise procedure obsolete by learning which features are characteristic for which species. We think, that once training databases grow to a suitable size, this would greatly benefit our system. Especially classification based on random forests [2] seem to be a natural choice, as textual and pictorial keys, that are currently used by entomologists, are themselves decision trees.

Another possible field of work, which we believe could be interesting for the future, is the implementation of regions of interest detectors, that are especially designed to extract typical wing patterns. In this thesis we used well known detectors, but it would also be interesting to see, if more complex wing structures could be captured by specialized detectors. We also like to note how the current visual vocabulary trees could be optimized. As our trainings data was rather small, we were able to design large enough visual vocabularies to describe butterfly wings. However, the larger the training database would become, the more it would benefit from larger vocabularies, resulting in increasing runtime. The alternative is to incorporate the nodes for scoring, as suggested in [71], instead of using simple flat scoring. Also it would be interesting to incorporate a punishing term into scoring for missing features, such that the score for training species, that have certain features, that the query specimen has not, is additionally decreased. In any case, the most important field of work for the future is to build a large training database, that would allow us to become more reliable experimental results as well as would make us able to make our system more robust.

Chapter 5

Remarks on Implementation

Implementation of both projects was done using Matlab. Because of the explorative nature of this thesis Matlab seemed to be a good choice, as lots of code already exists for Matlab, evaluation can be done very easily, and it allowed us to simultaneously build a small GUI, for which evaluated methods were incorporated with small effort necessary. As our datasets were rather small, runtime was not an issue, and we evaluated our systems only in terms of their recognition rates. We hereby like to note, that depending on the project and the used algorithms, matching/classification needed between five and 20 seconds on a Quad Core @2.87GHz with 8GB RAM using Windows Vista. However, as soon as datasets will become larger, the framework should be adapted to be stand alone.

We made use of publicly available code and binaries in addition to our own code. For both projects the highly recommended VLFeat tools [99] of Andrea Vedaldi were used. For the first project we additionally used implementations of TVSeg [93], Pacem [79], Shape Context [6] and Inner Distance Shape Context [51] and eventually adapted them to our needs. For the second project we additionally made use of the binaries of the color descriptor tool [94] of Koen van de Sande and the SFOP Matlab implementation [28] by Wolfgang Förstner.

Chapter 6

Summary

The goal of the thesis at hand, was to investigate the possibility of implementing a butterfly classification system, as well as providing a first version of a framework, that does so. The framework was designed to show entomologists the benefit of an (semi-)automated classification system, and how it can be used to ease their work. Therefore the thesis was parted into two smaller projects. The first one concerned itself with matching user marked contours of butterfly genital organs by their shape, and the second project concerned itself with classifying butterflies based on the appearance of their wings. The main part of the work was explorative, while simultaneously building a system, that can already be used by entomologists. We therefore compared a variety of computer vision techniques and evaluated them on butterfly datasets given by literature and the Natural Museum of History Vienna. Results of both projects look promising and showed, that the task at hand certainly seems to be manageable. However, there is a need for more training and test data for both projects to continue work on that system.

We already proposed possible fields of future work for each project in the respective chapter, but we would also like to note, that there is a variety of other tasks that can be investigated together with the Natural Museum of History in Vienna, as they own large collections of all kind of insects. Of special interest would be e.g. the identification of flies based on head structures or the identification of beetles based on their surface. Once research is done for a variety of different tasks, it will also become interesting if the respective work can be assembled to implement one universal tool for insect identification, similar to DAISY [73].

Bibliography

[1] Agarwal, G., Belhumeur, P., Feiner, S., Jacobs, D., Kress, W., Ramamoorthi, R., Bourg, N., Dixit, N., Ling, H., Mahajan, D., Russell, R., Shirdhonkar, S., Sunkavalli, K., and White, S. (2006). First steps toward an electronic field guide for plants. *Taxon*, 55(8):597–610.

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

[3] Arbuckle, T., Schroder, S., Steinhage, V., and Wittmann, D. (2001). Biodiversity informatics in action: identification and monitoring of bee species using ABIS. In *Proceedings of the 15th Int. Symp. Informatics for Environmental Protection*, volume 1, pages 425–430.

[4] Beis, J. and Lowe, D. (1997). Shape indexing using approximate nearest-neighbour search in high-dimensional spaces. In *Proceedings of the Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1000–1006.

[5] Belhumeur, P., Chen, D., Feiner, S., Jacobs, D., Kress, W., Ling, H., Lopez, I., Ramamoorthi, R., Sheorey, S., White, S., and Zhang, L. (2008). Searching the World's Herbaria : A System for Visual Identification of Plant Species. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 116–129.

[6] Belongie, S., Malik, J., and Puzicha, J. (2002). Shape matching and object recognition using shape contexts. *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, 24(4):509–522.

[7] Beucher, S. and Lantuejoul, C. (1979). Use of Watersheds in Contour Detection. In *International workshop on image processing, real-time edge and motion detection (1979)*, pages 2.1–2.12.

[8] Bieniecki, W. (2004). Oversegmentation avoidance in watershed-based algorithms for color images. In *Proceedings of the International Conference on Modern Problems of Radio Engineering, Telecommunications and Computer Science*, pages 169–172.

[9] Bosch, A., Zisserman, A., and Muñoz, X. (2008). Scene classification using a hybrid generative/discriminative approach. *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, 30(4):712–727.

[10] Branson, S., Wah, C., Schroff, F., Babenko, B., Welinder, P., Perona, P., and Belongie, S. (2010). Visual recognition with humans in the loop. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 438–451.

[11] Bresson, X., Esedoglu, S., Vandergheynst, P., Thiran, J., and Osher, S. (2005). Global Minimizers of The Active Contour / Snake Model. *Free Boundary Problems (FBP): Theory and Applications*.

[12] Burges, C. J. (1998). A tutorial on support vector machines for pattern recognition. *Data Mining and Knowledge Discovery*, 2(2):121–167.

[13] Burghouts, G. and Geusebroek, J. (2009). Performance evaluation of local colour invariants. *Computer Vision and Image Understanding (CVIU)*, 113(1):48–62.

[14] Canny, J. (1986). A computational approach to edge detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, 8(6):679–98.

[15] Caselles, V., Kimmel, R., and Sapiro, G. (1997). Geodesic active contours. *International Journal of Computer Vision (IJCV)*, 22(1):61–79.

[16] Chapelle, O., Haffner, P., and Vapnik, V. (1999). Support vector machines for histogram-based image classification. *IEEE Transactions on Neural Networks (NN)*, 10(5):1055–64.

[17] Chen, L., Feris, R., and Turk, M. (2008). Efficient partial shape matching using Smith-Waterman algorithm. In *CVPR Workshop on Non-Rigid Shape Analysis and Deformable Image Alignment (NORDIA)*, pages 1–6.

[18] Csurka, G., Dance, C., Fan, L., Willamowski, J., and Bray, C. (2004). Visual categorization with bags of keypoints. In *ECCV Workshop on Statistical Learning in Computer Vision (SLCV)*, pages 1–22.

[19] Cula, O. and Dana, K. (2004). 3D Texture Recognition Using Bidirectional Feature Histograms. *International Journal of Computer Vision (IJCV)*, 59(1):33–60.

[20] Donoser, M., Riemenschneider, H., and Bischof, H. (2009). Efficient partial shape matching of outer contours. In *Proceedings of the Asian Conference on Computer Vision (ACCV)*, pages 281–292.

[21] Donoser, M., Riemenschneider, H., and Bischof, H. (2010). Shape Guided Maximally Stable Extremal Region (MSER) Tracking. In *Proceedings of the International Conference on Pattern Recognition (ICPR)*, pages 1800–1803.

[22] Ferrari, V., Fevrier, L., Jurie, F., and Schmid, C. (2008). Groups of adjacent contour segments for object detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, 30(1):36–51.

[23] Ferrari, V., Tuytelaars, T., and Van Gool, L. (2006). Object detection by contour segment networks. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 14–28.

[24] Flusser, J. (2006). Moment invariants in image analysis. In *Proceedings of World Academy of Science, Engineering and Technology*, volume 11, pages 196–201.

[25] Flusser, J., Zitova, B., and Suk, T. (2010). *Moments and Moment Invariants in Pattern Recognition*. John Wiley Sons.

[26] Forssén, P.-E. and Lowe, D. (2007). Shape descriptors for maximally stable extremal regions. In *Proceedings of the International Conference on Computer Vision (ICCV)*, pages 1–8.

[27] Förstner, W. (1994). A framework for low level feature extraction. In *Proceedings of the European Conference on Computer Vision (ECCV)*, volume 2, pages 383–394.

[28] Förstner, W., Dickscheid, T., and Schindler, F. (2009). Detecting interpretable and accurate scale-invariant keypoints. In *Proceedings of the International Conference on Computer Vision (ICCV)*, pages 2256–2263.

[29] Friedman, J., Bentley, J., and Finkel, R. (1977). An algorithm for finding best matches in logarithmic expected time. *ACM Transactions on Mathematical Software (TOMS)*, 3(3):209–226.

[30] Gaston, K. and O'Neill, M. (2004). Automated species identification: why not? *Philosophical Transactions of the Royal Society of London. Series B, Biological Sciences*, 359(1444):655–667.

[31] Gies, V. and Bernard, T. (2004). Statistical solution to watershed oversegmentation. In *Proceedings of the International Conference on Image Processing (ICIP)*, volume 3, pages 1863–1866.

[32] Griffin, G., Holub, A., and Perona, P. (2007). Caltech-256 Object Category Dataset. Technical Report 7694, California Institute of Technology.

[33] Harris, C. and Stephens, M. (1988). A combined corner and edge detector. In *Proceedings of the Alvey Vision Conference (AVC)*, pages 147–151.

[34] Hoffman, D. and Richards, W. (1984). Parts of recognition. *Cognition*, 18(1-3):65–96.

[35] Holloway, J. (2008). The Moths of Borneo 17: Noctuidae: Rivulinae, Phytometrinae, Herminiinae, Hypeninae, Hypenodinae. *Malayan Nature Journal*, 60:1–268.

[36] Hu, M. (1962). Visual pattern recognition by moment invariants. *IRE Transactions on Information Theory*, 8(2):179–187.

[37] Johnson, A. and Hebert, M. (1999). Using spin images for efficient object recognition in cluttered 3D scenes. *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, 21(5):433–449.

[38] Julesz, B. (1981). Textons, the elements of texture perception, and their interactions. *Nature*, 290(5802):91–97.

[39] Kass, M., Witkin, A., and Terzopoulos, D. (1988). Snakes: Active contour models. *International Journal of Computer Vision (IJCV)*, 1(4):321–331.

[40] Kendall, D. (1984). Shape manifolds, procrustean metrics, and complex projective spaces. *Bulletin of the London Mathematical Society*, 16(2):81–121.

[41] Kononenko, V. and Han, H. (2007). Atlas genitalia of the Noctuidae in Korea (Lepidoptera). *Insects of Korea*, 11:462 pp.

[42] Larios, N., Deng, H., Zhang, W., Sarpola, M., Yuen, J., Paasch, R., Moldenke, A., Lytle, D., Ruiz Correa, S., Mortensen, E., Shapiro, L., and Dietterich, T. (2008). Automated insect identification through concatenated histograms of local appearance features. *Machine Vision and Applications (MVA)*, 19(2):105–123.

[43] Latecki, L. and Lakämper, R. (1999). Convexity Rule for Shape Decomposition based on discrete Contour Evolution. *Computer Vision and Image Understanding (CVIU)*, 73(3):441–454.

[44] Lazebnik, S., Schmid, C., and Ponce, J. (2003). A sparse texture representation using affine-invariant regions. In *Proceedings of the Conference on Computer Vision and Pattern Recognition (CVPR)*, volume 2, pages 319–324.

[45] Lazebnik, S., Schmid, C., and Ponce, J. (2004). Semi-local affine parts for object recognition. In *Proceedings of British Machine Vision Conference (BMVC)*, pages 959–968.

[46] Lazebnik, S., Schmid, C., and Ponce, J. (2006). Beyond bags of features: Spatial pyramid matching for recognizing natural scene categories. In *Proceedings of the Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2169–2178.

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

[48] Leung, T. and Malik, J. (1999). Recognizing surfaces using three-dimensional textons. In *Proceedings of the International Conference on Computer Vision (ICCV)*, volume 2, pages 1010–1017.

[49] Lin, Y., Lv, F., Zhu, S., Yang, M., Cour, T., and Yu, K. (2011). Large-scale image classification: fast feature extraction and SVM training. In *Proceedings of the Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1689–1696.

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

[51] Ling, H. and Jacobs, D. W. (2007). Shape classification using the inner-distance. *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, 29(2):286–299.

[52] Lödl, M. (1994). Revision der Gattung Hypena Schrank, 1802 s.l., der äthiopischen und madagassischen Region, Teil 1 (Insecta: Lepidoptera: Noctuidae: Hypeninae). *Annalen des Naturhistorischen Museums in Wien*, 96B:373–590.

[53] Lödl, M. (2001). Morphometry and relation patterns in male genitalia of noctuids (Lepidoptera: Noctuidae). *Quadrifina*, 4:5–33.

[54] Lowe, D. (1999). Object recognition from local scale-invariant features. In *Proceedings of the International Conference on Computer Vision (ICCV)*, volume 2, pages 1150–1157.

[55] Lowe, D. G. (2004). Distinctive Image Features from Scale-Invariant Keypoints. *International Journal of Computer Vision (IJCV)*, 60(2):91–110.

[56] Lu, C., Adluru, N., Ling, H., Zhu, G., and Latecki, L. (2010). Contour based object detection using part bundles. *Computer Vision and Image Understanding (CVIU)*, 114(7):827–834.

[57] Lu, C., Latecki, L., Adluru, N., Yang, X., and Ling, H. (2009). Shape guided contour grouping with particle filters. In *Proceedings of the International Conference on Computer Vision (ICCV)*, pages 2288–2295.

[58] Malik, J., Belongie, S., Leung, T., and Shi, J. (2001). Contour and Texture Analysis for Image Segmentation. *International Journal of Computer Vision (IJCV)*, 43(1):7–27.

[59] Martin, D., Fowlkes, C., and Malik, J. (2004). Learning to detect natural image boundaries using local brightness, color, and texture cues. *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, 26(5):530–549.

[60] Martínez-Muñoz, G., Larios, N., Mortensen, E., Zhang, W., Yamamuro, A., Paasch, R., Payet, N., Lytle, D., Shapiro, L., Todorovic, S., Moldenke, A., and Dietterich, T. (2009). Dictionary-free categorization of very similar objects via stacked evidence trees. In *Proceedings of the Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 549–556.

[61] Matas, J., Chum, O., Urban, M., and Pajdla, T. (2002). Robust wide-baseline stereo from maximally stable extremal regions. In *Proceedings of British Machine Vision Conference (BMVC)*, pages 384–393.

[62] Maurer, H., Brandstaller, T., Diem, P., and Wolf, H. (2009). Austria-Forum. http://www.austria-lexikon.at/.

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

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

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

[66] Mindru, F., Tuytelaars, T., Gool, L., and Moons, T. (2004). Moment invariants for recognition under changing viewpoint and illumination. *Computer Vision and Image Understanding (CVIU)*, 94(1-3):3–27.

[67] Mokhtarian, F., Abbasi, S., and Kittler, J. (1996). Efficient and robust retrieval by shape content through curvature scale space. In *Proceedings on the International Workshop on Image Databases and Multi-Media Search*, pages 51–58.

[68] Monti, L., Baylac, M., and Lalanne-Cassou, B. (2001). Elliptic Fourier analysis of the form of genitalia in two Spodoptera species and their hybrids (Lepidoptera: Noctuidae). *Biological Journal of the Linnean Society*, 72(3):391–400.

[69] Moosmann, F., Triggs, B., and Jurie, F. (2007). Fast discriminative visual codebooks using randomized clustering forests. *Neural Information Processing Systems (NIPS)*, 19:985–992.

[70] Nilsback, M. and Zisserman, A. (2008). Automated flower classification over a large number of classes. In *Proceedings of the Indian Conference on Computer Vision, Graphics and Image Processing (ICVGIP)*, pages 722–729.

[71] Nister, D. and Stewenius, H. (2006). Scalable recognition with a vocabulary tree. In *Proceedings of the Conference on Computer Vision and Pattern Recognition (CVPR)*, volume 2, pages 2161–2168.

[72] Nister, D. and Stewenius, H. (2008). Linear Time Maximally Stable Extremal Regions. In *Proceedings of the European Conference on Computer Vision (ECCV)*, Electric Power Engineering Series, pages 183–196.

[73] O'Neill, M. (2007a). Daisy. In *Automated Taxon Identification in Systematics*, Systematics Association Special Volumes, pages 101–114. CRC Press.

[74] O'Neill, M. (2007b). DAISY: A Practical Computer-Based Tool for Semi-Automated Species Identification. In *Automated Taxon Identification in Systematics*, chapter 7, pages 101–114.

[75] Owada, M. (1987). *A taxonomic study on the subfamily Herminiinae of Japan : Lepidoptera, Noctuidae*. National Science Museum, Tokyo.

[76] Perona, P. (2010). Vision of a Visipedia. *Proceedings of the IEEE*, 98(8):1526–1534.

[77] Riemenschneider, H. (2012). *Object detection by partial shape matching, category models and joint segmentation*. Phd thesis, submitted, Graz University of Technology.

[78] Riemenschneider, H., Donoser, M., and Bischof, H. (2008). Online object recognition by MSER trajectories. In *Proceedings of the International Conference on Pattern Recognition (ICPR)*, pages 1–4.

[79] Riemenschneider, H., Donoser, M., and Bischof, H. (2010). Using partial edge contour matches for efficient object category localization. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 29–42.

[80] Riemenschneider, H., Donoser, M., and Bischof, H. (2011). Image retrieval by shape-focused sketching of objects. In *Proceedings of the Computer Vision Winter Workshop (CVWW)*.

[81] Roth, P. and Winter, M. (2008). Survey of appearance-based methods for object recognition. Technical Report ICG-TR-01/08, Graz University of Technology.

[82] Russell, B., Torralba, A., Murphy, K., and Freeman, W. (2008). LabelMe: a database and web-based tool for image annotation. *International Journal of Computer Vision (IJCV)*, 77(1):157–173.

[83] Schmid, C., Mohr, R., and Bauckhage, C. (2000). Evaluation of interest point detectors. *International Journal of Computer Vision (IJCV)*, 37(2):151–172.

[84] Shotton, J. and Johnson, M. (2008). Semantic texton forests for image categorization and segmentation. In *Proceedings of the Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1–8.

[85] Shridhar, M. and Badreldin, A. (1984). High accuracy character recognition algorithm using fourier and topological descriptors. *Pattern Recognition*, 17(5):515–524.

[86] Sivic, J. and Zisserman, A. (2003). Video Google: a text retrieval approach to object matching in videos. In *Proceedings of the International Conference on Computer Vision (ICCV)*, volume 2, pages 1470–1477.

[87] Sonka, M., Hlavac, V., and Boyle, R. (2008). *Image processing, analysis, and machine vision*. Thompson Learning, 3rd edition.

[88] Tanase, M. and Veltkamp, R. (2005). Part-based Shape Retrieval. In *Proceedings of the ACM International Conference on Multimedia*, pages 543–546.

[89] Toshev, A., Taskar, B., and Daniilidis, K. (2010). Object detection via boundary structure segmentation. In *Proceedings of the Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 950–957.

[90] Turk, M. and Pentland, A. (1991). Face recognition using Eigenfaces. In *Proceedings of the Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 586–591.

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

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

[93] Unger, M., Pock, T., Trobin, W., Cremers, D., and Bischof, H. (2008). TVSeg-Unteractive Total Variation based Image Segmentation. In *Proceedings of British Machine Vision Conference (BMVC)*, pages 40.1–40.10.

[94] van de Sande, K., Gevers, T., and Snoek, C. (2010). Evaluating Color Descriptors for Object and Scene Recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, 32(9):1582–1596.

[95] van de Weijer, J., Gevers, T., and Bagdanov, A. (2006). Boosting color saliency in image feature detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, 28(1):150–156.

[96] Vapnik, V. (1999). *The Nature of Statistical Learning Theory (Information Science and Statistics)*. Springer.

[97] Varma, M. and Zisserman, A. (2003). Texture classification: Are filter banks necessary? *Proceedings of the Conference on Computer Vision and Pattern Recognition (CVPR)*, 2:691–698.

[98] Varma, M. and Zisserman, A. (2005). A statistical approach to texture classification from single images. *International Journal of Computer Vision (IJCV)*, 62(1):61–81.

[99] Vedaldi, A. and Fulkerson, B. (2010). VLFeat: An open and portable library of computer vision algorithms. In *Proceedings of the International Conference on Multimedia*, pages 1469–1472.

[100] Wang, J., Markert, K., and Everingham, M. (2009). Learning Models for Object Recognition from Natural Language Descriptions. In *Proceedings of British Machine Vision Conference (BMVC)*, pages 2.1–2.11.

[101] Warren, A., Grishin, N., Pelham, J., Davis, K., and Stangeland, M. (2012). Butterflies of America. http://butterfliesofamerica.com/.

[102] Zhu, Q., Wang, L., and Wu, Y. (2008). Contour context selection for object detection: A set-to-set contour matching approach. *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 774–787.