

# Flower Recognition

## Semester Project Final Report

June 2010

Vincent Vuarnoz

Supervisor: Péter Vajda

Professor: Touradj Ebrahimi

# Contents

<b>1</b>	<b>Introduction</b>	<b>2</b>
<b>2</b>	<b>State of the art</b>	<b>3</b>
<b>3</b>	<b>Biology point of view</b>	<b>5</b>
3.1	Importance . . . . .	5
3.2	Books . . . . .	5
3.3	Meeting with biologist . . . . .	6
<b>4</b>	<b>System Overview</b>	<b>10</b>
<b>5</b>	<b>Segmentation</b>	<b>11</b>
5.1	Need of segmentation . . . . .	11
5.2	Watershed Algorithm . . . . .	11
5.3	Application . . . . .	12
5.4	Example . . . . .	12
<b>6</b>	<b>Features</b>	<b>14</b>
6.1	FELib . . . . .	14
6.2	Color-Based . . . . .	15
6.2.1	Color Descriptors . . . . .	15
6.2.2	Implemented features . . . . .	16
6.3	Contour-Based . . . . .	17
6.3.1	Distances vs angles . . . . .	17
6.3.2	MinMax ratio . . . . .	19
6.3.3	Projections . . . . .	20
6.3.4	AreaRatio . . . . .	21
6.3.5	Number of petals . . . . .	21
6.4	Texture-Based . . . . .	22
<b>7</b>	<b>Matching</b>	<b>23</b>
<b>8</b>	<b>Database</b>	<b>23</b>
8.1	Available databases . . . . .	23
8.2	Resulting database . . . . .	23
<b>9</b>	<b>Android</b>	<b>23</b>
<b>10</b>	<b>Results</b>	<b>25</b>
10.1	Database evaluation . . . . .	25
10.1.1	Ground Truth . . . . .	25
10.1.2	Each Feature separately . . . . .	26
10.1.3	Combining the features . . . . .	27
10.2	Final Showdown . . . . .	29

<b>11 Conclusion</b>	<b>29</b>
<b>12 Future Work</b>	<b>29</b>
12.1 Adding extra information . . . . .	30
12.2 Leaves Analysis . . . . .	30
12.3 HS Color space Features . . . . .	30
12.4 Bigger database . . . . .	31
12.5 Automatic Segmentation . . . . .	31
12.6 Distance measurement . . . . .	31
<b>13 Appendix</b>	<b>34</b>
13.1 Files organisation . . . . .	34
13.2 Software Documentation . . . . .	35
13.2.1 Tools . . . . .	35
13.2.2 Class diagram . . . . .	35
13.2.3 Paths . . . . .	36
13.2.4 Feature . . . . .	37
13.2.5 Segmentation . . . . .	38
13.2.6 DatabaseEvaluation . . . . .	38
13.2.7 Matching . . . . .	39
13.2.8 The detectExe . . . . .	39
13.3 DVD . . . . .	39

# 1 Introduction

Imagine someone hiking in the Swiss mountains who finds a beautiful flower. This person has always been bad in biology but would like to know more about that flower. What's its name? Its main features? Is it rare? Is it protected? etc. By simply taking a picture of the flower with a phone, he or she could get all these informations through an automatic flower recognition application.

As part of the semester project, the elaboration of such an application has been aimed. The recognition of flowers from photographs implies several steps, starting with the localization of the flower in the image, followed by identifying and extracting the specific characteristics of this flower, and finally finding the best match.

The solution proposed in this report includes the following elements; specific research on plants and on existing method, a segmentation algorithm based on user's inputs, implementation of several visual features suitable for flowers differentiation. It was eventually implemented in android phone.

Other applications could include educational purposes and nature preservation programs.

## 2 State of the art

Some research conducted in flower recognition and general features extraction are listed in this section.

In 1999, Das et al. [1] suggested to use domain knowledge of flowers colors to index the images. In this context they developed an iterative segmentation algorithm to isolate the flower from the background. There are indeed colors that are rarely present in flowers and the background of photographs taken in the nature could be quite similar. They used only color names and their relative proportions within the flower region as features which is a good but not sufficient mean for full recognition.

Then, Saitoh and Kaneko [2] have proposed a method that uses two input images, one of the flower and one of the leaf. In order to do so the user should place a black cloth behind the flower which is not so convenient. And even with this approach the background separation isn't straightforward, they actually used a k-means clustering method in color space (with multiple integrations). Feature-wise they considered color and shape informations for both the flower and the leaf.

Some interactive methods such as CAVIAR (Computed Assisted InterActive Recognition) [3] have been developed [4] to exploit the human perceptibility. A rose-curve is generated for the test flower and the top three candidates are proposed to the user who can then either select the right flower (if present) or modify the rose-curve to get a new set of propositions. This is done iteratively until the right flower is found, which is not suitable for our application because of these multiple user interactions.

In [5] an automatic method is developed under the assumption that the flower is focused while the background is defocused. It then uses a Normalized Cost (NC) [6] method, which needs a manual entry point on the boundary. They overcame this drawback by implementing an automatic method that minimizes the NC among a set of smartly chosen entry points. The resulting segmentation is then shown to the user who can, in case of failure (3% of the time), add a new entry point. Four shape features (number of petals, central moment, roundness and width/height ratio) and six color features taken from HS color space (x and y coordinates of the largest/second largest color cell and ratio of the largest/second largest color cell) were used yielding a recognition rate of 90%.

In general feature extractions, early methods as in [7] used HSL color coordinate system, by simply using the luminance component as a textural analysis and the hue and saturation components as chromatic analysis. A region-based color image retrieval using geometric properties is presented in [8]. Briefly, they used a region-growing technique to form color regions, then spatial relational graph and Fourier description coefficients are computed for each region. Brandt et al. [9] proposed a technique, applied on non segmented objects, using edge histograms and Fourier-transform-based features computed for an edge image in Cartesian and polar coordinate planes. Edge histograms were also used by Shim et al. [10], but where color distribution are computed for the pixels of

three types of edges and based on these distributions, three distance measurements were done. Adnan et al. [11] considered segmented objects in the image, whose geometrical shapes are estimated and compared with a predefined set of shapes of different categories.

Then some other concepts have been introduced such as the integration of a spatial distributing information of colors proposed by Li et al. [12]. They used an amelioration of HSV color model called 'color sectors', mixed with a spatial histogram generated by Hough transform applied on hue edges. Or like in [13] where two features, HSI color information (especially hue) and CSS (Curvature Scale Space) were employed. Based on dominant color in the foreground image, the method proposed in [14] used a color look-up table to divide the color space into smaller categories then Euclidean distance was used for matching. Three features, color, texture and shape information are combined together in [15] by first partitioning the image into non overlapping tiles of equal size. Then color moments and moments on gabor filter responses of these tiles locally describe the color and the texture respectively. Most similar highest priority principle is applied for matching.

Some work has been done in the domain of leaves recognition, Im et al. [16] used a hierarchical polygon approximation representation to recognize the Acer family variety. In [17] they combined different features based on centroid-contour distance curve and used fuzzy integral for the retrieval. Wang et al. [18] suggested to use a hypersphere classifier to deal with a large number of class. Prior to that they extract fifteen features (eight geometric ratios and seven Hu invariant moments). In [19] Modified Fourier Descriptor recognition program was introduced. Based on the Angle Measurement it was shown that it was efficient in shape analysis. In [20] they proposed a web-based application in which they retrieve some leaf features considering a 5-steps scheme: 1) getting leaf contour represented in Centroid-Contour Distance, 2) rotating the leaf to an horizontal position, 3) binarization, 4) analyzing leaf type, 5) detecting stem information. Zhenjiang et al. [21] used size, shape and color of petal and leaf like many others but added an object-oriented pattern recognition (OOPR) approach which mathematically deals with how to use all different features rationally in the recognition scheme.

Also, an interesting segmentation algorithm can be found in [22], it works as follows. An initial segmentation is done based on general foreground/background color distribution. After labelling some samples of pixel as foreground or background and averaging these distributions over all classes, the provisional segmentation is obtained using contrast dependant prior MRF cost function [23]. After having detected the petals (using affine invariant Hough like procedure), a new color foreground model is established and the MRF is repeated. This may be iterated until convergence.

Liu et al. [24] have explored the organization and the utilization of features in an effective way. They deduced the weights to assign to all features, texture-based or color-based, by an image feature statistic. This method appears to be superior than other methods using fixed weights.

An investigation done by Nilsback et al. [25] on the utilization of a visual

vocabulary model which explicitly represents the various aspects (color, shape, texture) has shown that the ambiguity between flower categorization can be overcome.

Finally a mobile-based flower recognition system has been implemented [26]. It uses Difference Image Entropy (DIE) and contour features of the flower. The user has to draw the exact contour of the flower himself and then the two images, the drawn and the original image, are sent to the server where the DIE is processed.

## 3 Biology point of view

### 3.1 Importance

In items recognition research a lot have been done about general features extraction or recognition between different classes of objects. In case of a specific domain recognition, taking into account the unique characteristics that belong to this category may very much improves the performance of the system.

Despite the high technical aspect of this project, dealing with flowers gives it a biological connotation. Some basic knowledges about flowers have to be learned and concepts about how the biologists themselves recognize flower to be studied. To this purpose some books have been borrowed and a meeting with a real biologist was booked. The two next paragraphs are devoted to these experiences.

### 3.2 Books

Different books have been loaned from the EPFL library in Lausanne. “Aspects of morphogenesis of leaves, flowers and somatic embryos” is a documentation edited by the National Botanic Garden of Belgium and treats advanced themes as genetic aspects of flower development or molecular aspects of somatic embryogenesis in conifers. “Organization des plantes à fleurs” also appears to be quite a bit too much micro-biology oriented, a good find to note anyway: the plants with flowers are part of the class named the *Angiosperms* which hold the majority of the plants on earth with more than 200'000 species! The last book entitled “Botanique Systématique des plantes à fleurs” relates the history of the botanic classification, the evolutions of different parts of the flowers through time before providing descriptive tables of families. A family, as the one described in Figure 1 holds more than 400 genus and about 12'000 species. Looking at the definition of the family gives a good idea about the complexity of defining a plant, e.g. under the section flower more than 20 qualificatifs are used (such as cyclic, polystemon, shortened sepals, staminodes present, etc.) and by looking at the other page it seems the plants are looking completely different despite being in the same family.



Figure 1: Cover of “Botanique Systématique des plantes à fleurs”, and extracted pages from the book showing the enormous number of genus and species contained in one family.

### 3.3 Meeting with biologist

A meeting with Pascal Vittoz, biologist and Chief of Research Project in UNIL, has been arranged on the 23rd of March. He first of all introduced the categorization of flowers, organized in a hierarchy : Embranchement, Class, Family, Genius and finally Species. He then explained a few methods used on the field to recognize flowers, one of them is the use of a key i.e. a guide with a set of iterative questions with multiple answers that leads to the wanted flower. This Swiss book is called “Flora Helvetica” and a sample is exhibited in Figure 2.

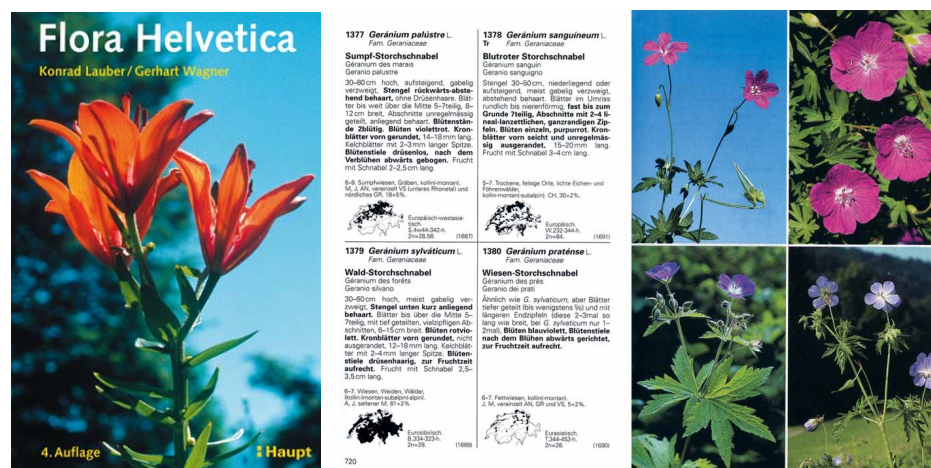


Figure 2: Book “Flora Helvetica” considered as the bible of the botanist

Note that about 150 questions are needed just to find the correct family! It

shows once again how complex the differentiation of the species is and gives an idea of how similar the flowers of a same genus might be. There are about 3000 species in Switzerland but 1000 of them represent 90% he explained, and then searched his computer to provide a complete list of all the names of plants in this country. He also removed all plants without a blooming flower that yielded a number of candidates of about 1300. They're all listed in an excel table with many other informations such as the family, the level of protection and even the endangered rate by regions! These informations could be of great help in a future development of the application and this document will be present in the attached DVD.

Another interesting thing he shared was which features does he use based on his everyday life; color and shape of the flowers were part of the most obvious, but the concept of symmetry in the flower was a bit less straightforward and said to have non-negligible influence. In Figure 3 the first flower has multiple axis of symmetry while the one on the right has apparently only one.



Figure 3: The symmetry in the flower has a great impact in the recognition

Also the disposition of the ovary or the number of stigmates play an important role. And even if it would be a little bit harder to apply them in a computer vision application it's always good to have possible solutions to explore. Here in Figure 4 the left flower is said to have a superior ovary while on the right it is an inferior ovary, pay attention of the sepals which are below the ovary on the left and above it on the right.



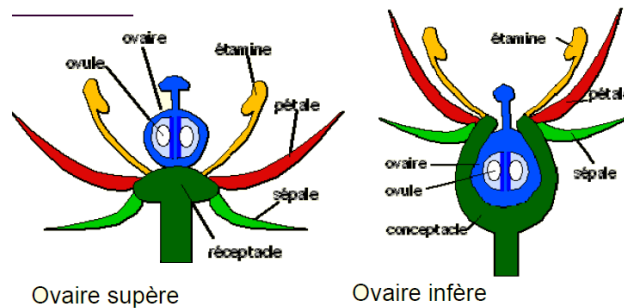


Figure 4: The left flower holds a superior ovary which lays on the receptacle and the right flower holds an inferior ovary which is surrounded by the conceptacle.

While showing his slides about Floristic<sup>1</sup>, course he teaches at the uni, he also emphasises the importance of leaves; indeed their size, their shape, their disposition can vary very much and be a good mean for differentiating similar blooms. The disposition of the leaves on the stem can be alternate, opposed or whorled as illustrated in Figure 5a).

The nervation of the leaf can be of different types, there are leaves with dichotomic, parallel, palmate, pinnate nerves. These four features can be found respectively at the top-left, the top-right, the bottom-left and the bottom-right of the Figure 5 b).

Finally being in presence of an integer limb, a toothed limb, a crenate limb or sinuate limb will certainly not conclude to the same flower and this characteristic is also a critical factor in a recognition process (see Figure 5 in order).

<sup>1</sup>The slides will also be added on the DVD, Floristique\_09.pdf

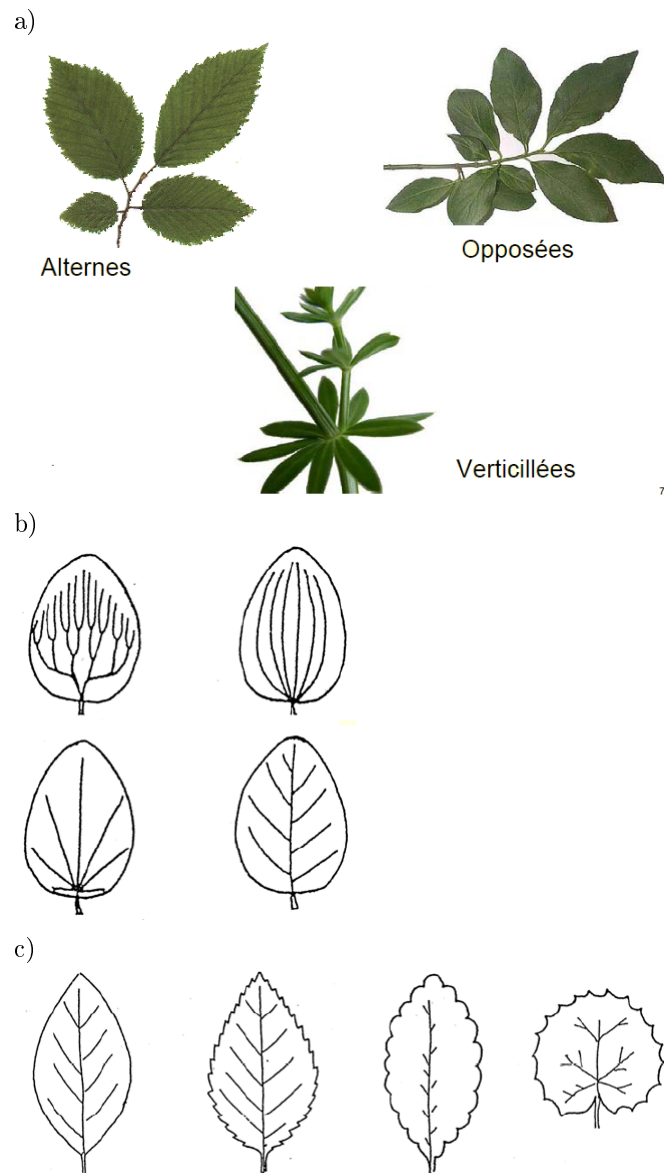


Figure 5: Importance of leaves: in a) the different dispositions, in b) the nervatures, in c) the shape of the limb

In conclusion this meeting has been of great use as it helped understand how professionals approach the sensitive task of recognizing flowers. Many factors come into play, whether the color, the shape, the symmetry of the flower or some more subtle as the number of stigmates or the position of the ovaries. The

accent has been put on the important role of the leaves; with their alternate disposition or their palmate nerves they offer many clues for recognition. After all, every concepts introduced were as many ideas of features for the project! The main thing to remember is that most of the time it is a complex combination of many characteristics that leads to a correct decision.

## 4 System Overview

The whole process starts with the user taking a picture of a flower, after having indicated the flower region (see Section 5 ), these informations are sent to the server. Then the segmentation is done, it outputs the mask of the flower which is a binary image with ones on the flower region and zeros in the background. The original image and the mask are needed for the features extraction which stores the values for each feature separately. Then the data are compared to the database's for the matching. Finally the result is sent back to the user. This scheme is illustrated in Figure 6.

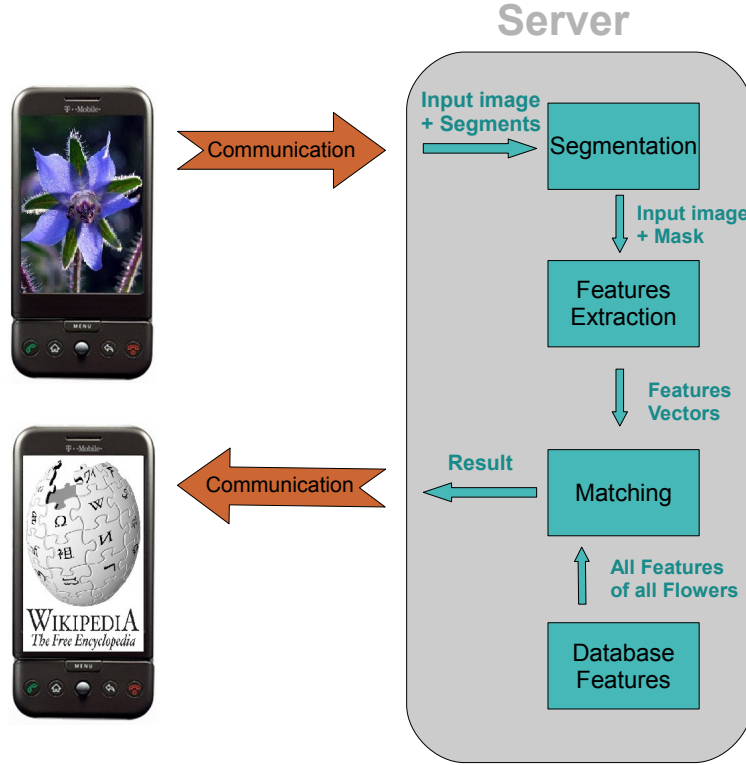


Figure 6: System Overview. The image of the flower is sent to the server where segmentation, feature extraction and matching are processed.

## 5 Segmentation

### 5.1 Need of segmentation

The main purpose of segmentation is to determine regions in the image that belong together, in this application only two regions are needed; one defining the flower and another corresponding to the background. This is a very important step since all the features should be extracted targeting the flower area only.

### 5.2 Watershed Algorithm

This algorithm is based on the gradient of the image which is viewed as a topographic relief where high gradient values correspond to mountains while low values correspond to valleys as shown in Figure 7.

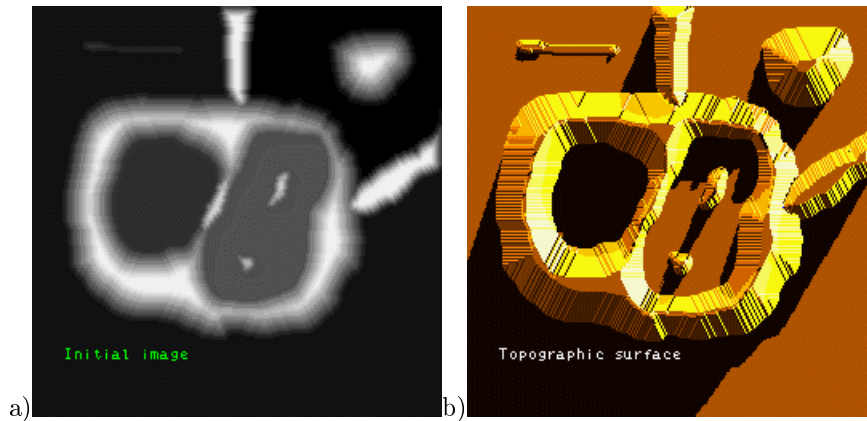


Figure 7: a) Gradient of an image, b) Topographic interpretation

Now let's imagine the valleys are filled with water from their local minima (see Figure 8 a)). The merging of waters coming from different sources is prevented by building a watershed, which in English means “a ridge that divides two different river systems”. These lines of contacts when the water of one source meets another's are known as watershed lines and the regions they create are called catchment basins. The Figure 8 b) shows the resulting separations.

In the example above the algorithm was performing well but sometimes due to the presence of too many local minima, it could yield in an over-segmentation. An amelioration of the algorithm called marker-controlled watershed counters this little drawback by allowing the user to choose the locations of the water sources. These locations are called markers and they basically say which regions have to belong together in the final result. This functionality is very useful as it can be specified which part of the image is of interest and which is not.

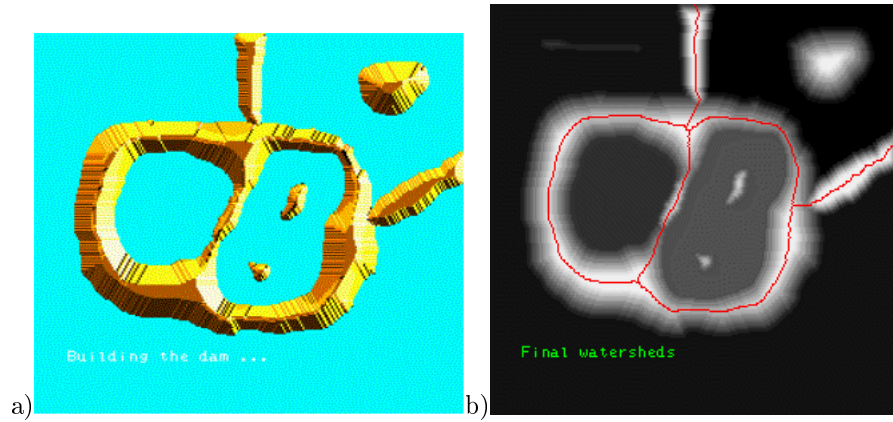


Figure 8: a) Water being flood, b) Resulting watershed lines (in red)

### 5.3 Application

One can think of using this markers facility in an iterative way i.e. the result of segmentation can be shown to the user and if the result does not satisfy him, he could add new markers, then the result is shown, and so on until he's happy.

The main drawback of the watershed, and not a negligible one, is that the whole database has to be treated manually. Indeed to extract the features the mask output by the segmentation is needed and thus has to be done a priori. This could represent a very tiring job for a large database !

Note that this algorithm is already implemented in OpenCV which is this time a significant advantage.

### 5.4 Example

An example of the operations is presented here. First of all the original image is displayed to the user. Then he may draw on the image to indicate the different regions, flower region or background. Finally the watershed algorithm is performed. Figure 9 goes through these steps visually for a better understanding.

a)



b)



c)

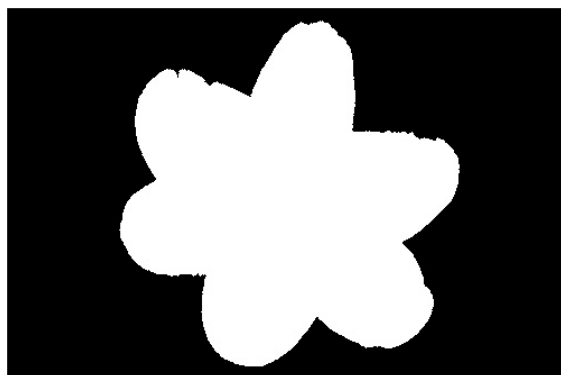


Figure 9: a) input flower, b) markers drawn by the user, c) resulting mask

## 6 Features

This section lists the all 18 features extracted for the recognition. First a quick explanation about the FELib is presented, then the features have been grouped by different categories: the features based on colors (which may also come from the Felib), the features based on the contour of the flower, the features based on the texture and finally two others. This section is structured according to this organization which is illustrated in Figure 10.

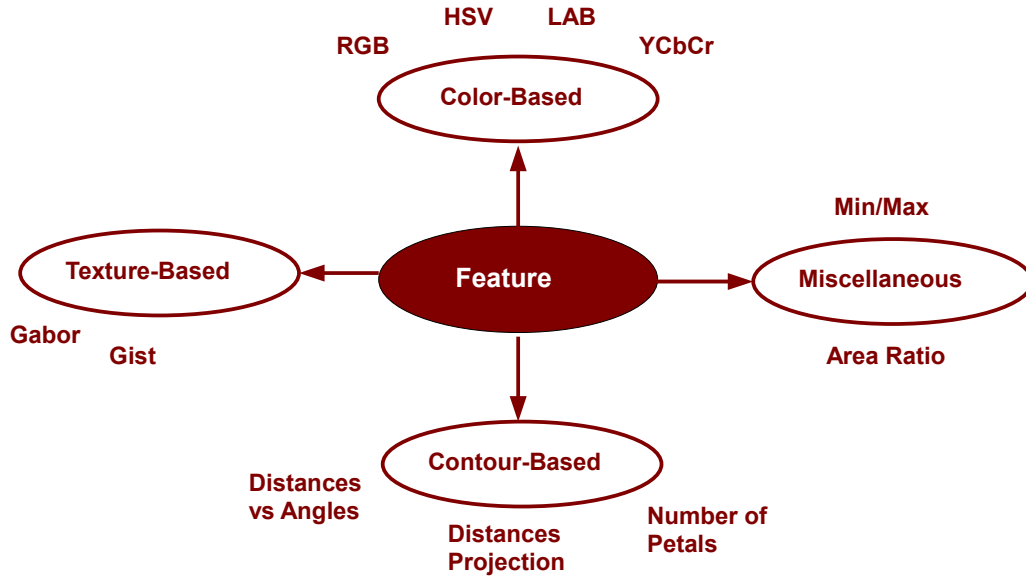


Figure 10: The Feature class and its children organized in different categories

### 6.1 FELib

FELib is a Feature Extraction Library created by Jianke Zhu<sup>2</sup>. This library, which can be downloaded at [32], provides tools to extract the following features:

- Color Histogram, Color Moment
- Edge Histogram

---

<sup>2</sup><http://www.vision.ee.ethz.ch/~zhuji/>

- Gabor wavelets transform
- Local Binary Pattern
- GIST

## 6.2 Color-Based

By following the intuition, and this was confirmed by the biologist, it can be said that color is one of the most important aspect of a flower. This is indeed the first element that anyone would say to characterize a bloom. So in this paragraph, first a bit of theory about general color features extraction can be found, followed by an enumeration of the implemented features.

### 6.2.1 Color Descriptors

There are many different ways of extracting the color informations from an image, some of the basic ones are color histograms and color moments. They both can be evaluated from different color spaces, so the main ones are listed below.

#### Color Spaces

- RGB : this color space comes from an additive model in which the three primitive colors red/green/blue are added together to reproduce the all range of colors. Similar in a way to the HVS it is widely used and present in all CRT monitors.
- HSV : stands for Hue, Saturation, Value, it is a cylindrical-coordinate representation and is known for its intuitivity i.e. close to how a person would describe a color.
- LAB : Lab is a color-opponent space, L is for lightness and a and b for the color-opponent dimensions, based on non linearly compressed CIE XYZ color space coordinates. Because of its opponent process that better match how the informations from the cones are processed by the HVS, this color-space is known to be perceptually uniform.
- YCrCb : Luma, blue difference, red difference. YCrCb is not an absolute color space, it is more a way of encoding RGB informations. More efficient than RGB for storage and communication.

**Color Histograms** A color histogram is a representation of the distribution of colors in an image. With digital images, the most common method to create a histogram is to equally subdivide the color space into a number of small bins, and then count the number of pixels falling into each of these bins. The advantages of color histograms are its robustness to rotation, scaling or partial occlusion, and its low computational cost. However, color histograms are liable



to lose the spatial information. Moreover they have a high sensitivity to noisy interference such as lighting intensity changes and quantization errors which is not a wanted characteristic in image retrieval.

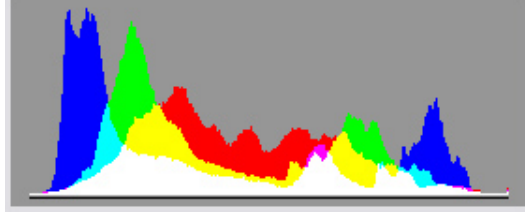


Figure 11: Color Histogram showing the three channels (red, green, blue)

**Color Moments** Considering RGB, an image can be seen as a combination of three functions:  $I(x, y) = (R(x, y), G(x, y), B(x, y))$ . The generalized color moments of order  $p + q$  and degree  $a + b + c$  can be defined as follows:

$$M_{pq}^{abc} = \int \int x^p y^q R(x, y)^a G(x, y)^b B(x, y)^c dx dy$$

Note that moments of order 0 don't contain any spatial information being thus rotationally invariant and moments of degree 0 don't contain any photometric information. By using high values for orders or degrees leads to instability, typically order 0 and 1 combined with first and second degrees are used. This would represent 27 different moments (after having removed the useless degree 0). Moreover by subtracting the average in all input channels before the computation, the shift-invariance is achieved. [30]

### 6.2.2 Implemented features

By taking advantages of the nice tools provided in OpenCV, color histograms can be quite easily computed. They consist of three 1-D histograms (one for each color dimension) that are finally put following each other in the feature vector, whose length equals 768 ( $3 \cdot 256$ ). The option of inputting a mask for the histogram calculation is offered and has plainly been used in order to discard the background. Different color spaces were tested namely; RGB, HSV, LAB and YCrCb.

The FELib also provides some color-features :

- HSV Color Histogram
- HSV3D Color Histogram
- RGB Color Moment
- LAB Color Moment
- RGB Pyramid Color Moment

Note that the source code isn't available with this library so it's hard to say how the extraction works, especially for color histogram whose vector has a length of 9. Some documentation has been found for the color moment though. Here is how they work: the image is partitioned into 3x3 grids and for each grid three values are extracted : the color mean, the color variance and the color skewness.

The color mean is defined by  $\mu_i = \frac{1}{N} \sum_{j=1}^N p_{i,j}$  and the color variance and skewness by respectively the second and third color moment when defined as follows:

$$M_i^h = \left( \frac{1}{N} \sum_{j=1}^N (p_{i,j} - \mu_i)^h \right)^{\frac{1}{h}}$$

where  $i = 1, 2, 3$  (for each color dimension). This finally yields in a 81-dimensional grid color moment vector. [27][31]

Also note that no mask facility is proposed with these features, however for the sake of good order, all non-flower pixels have been set to zero prior to extraction.

### 6.3 Contour-Based

The main feature after the color is the shape of the flower. In this section a few self-implemented features to get some informations about the form of the flower are presented. It is basically based on the mask image whose contour is detected. Then several features are deduced from that.

#### 6.3.1 Distances vs angles

The most straightforward is to go through the contour for all angles between 0 and 2PI for a given step size. The distances from the contour point and the center of gravity are then computed. The figure 12 gives a good representation of this process.

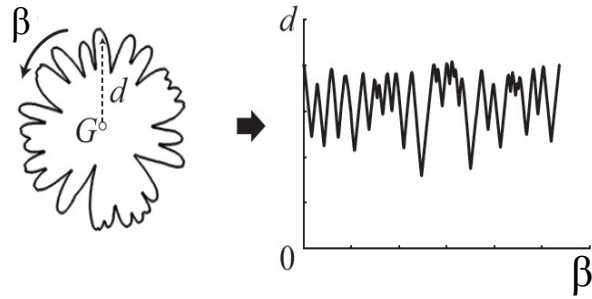
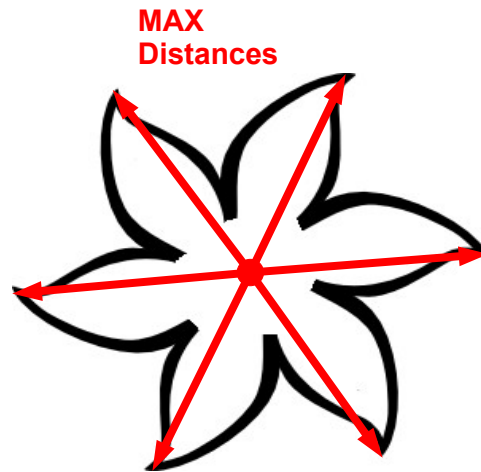


Figure 12: Getting a relation between the angles and the distances

Now an interesting issue has to be taken care of, it is the determination of the starting point which is of huge influence on the results. One can suggest to always start at the horizontal line from the center of gravity or any other fixed point. This would unfortunately yield in a system sensible to rotation which is not likely to be effective. The solution proposed here is to find the maximum distance and to attach the origin of angles to it. Thus the rotation independancy is obtained in most of the cases going from all symmetric flowers (as in Figure 13 a)) to even more specific ones like in Figure 13 b). These two are suitable since the origin will be set on the maximum distance yielding to the same feature vector.

a)



b)

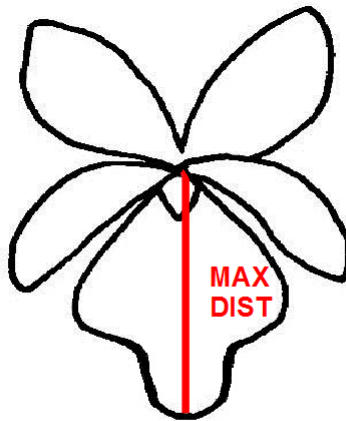


Figure 13: The feature Interpolation by setting the origin to the max distance becomes rotation invariant. Indeed both in with a symmetric flower and in b) with an asymmetric the starting point would be on a peak.

### 6.3.2 MinMax ratio

From the distances vector it is easy to find the maximum and the minimum distances, then by taking their ratio a new feature is obtained. It gives a good

information about the spacing between two petals and would be very good at differentiating the two flowers shape shown in Figure 14.

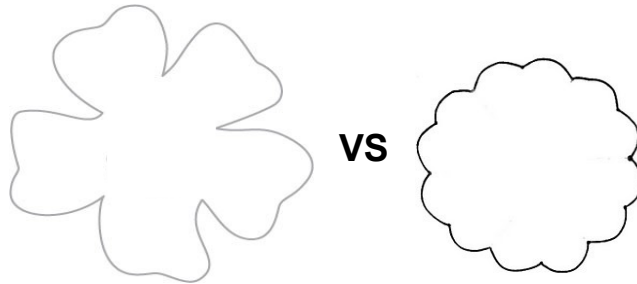


Figure 14: The feature MinMax would differentiate these two images

### 6.3.3 Projections

Another feature deduced from these distances is this time a projection of the distances. It can be considered as an histogram of distances and produces the precious information of how the distances are distributed around the flower. If it's not clear see Figure 15 which very simply illustrates this concept.

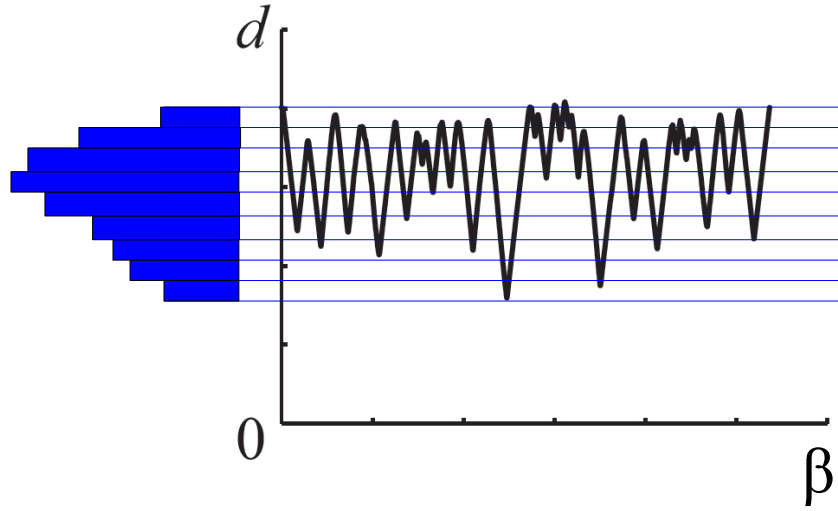


Figure 15: Projections of the distances

#### 6.3.4 AreaRatio

This feature just needs the maximum distance and then computes the area of the flower region over the area of the circumscribed circle. This feature furnishes a mean to get an idea of the repartition of the petals. It is again a very nice information to have.

#### 6.3.5 Number of petals

The last but not least feature derived from these distances, is a feature that counts the number of petals! It works as follows: the mean of the distances is calculated and then going around the flower, the number of petals is based on the number of times this mean distance is crossed. This feature works very well as long as the roundness of the flower is high (i.e. the mean value is consistently between the max and the min value). Examples of one suitable and one unsuitable flower are shown in Figure 16.

a)



b)



Figure 16: Good elements a) and bad element b) for the Number of petals feature

#### 6.4 Texture-Based

In order to take into account the texture of the flower two features taken from the FELib, Gabor wavelets and Gist, were inserted in the framework. A gabor wavelet transform is applied to a rescaled version of the image down to 64x64. Different levels and orientations are used, respectively 5 and 8, yielding in 40 sub

images. Then 3 moments are computed: mean, variance and skewness for each sub image, finally producing a 120-dimensional vector. [27] The idea behind Gist descriptors is to develop a low dimensional representation of the scene, which does not require any form of segmentation. A set of perceptual dimensions (naturalness, openness, roughness, expansion, ruggedness) is proposed. They should represent the dominant spatial structure of a scene and it has been shown that they are reliably estimated using spectral and coarsely localized information. The image is divided into a 4-by-4 grid for which orientation histograms are extracted. [28]

## 7 Matching

## 8 Database

### 8.1 Available databases

The Visual Geometry Group of Oxford University has been working on flower recognition and has therefor created a dataset consisting of 102 flower categories and more than 8000 images! The flower mainly come from United Kingdom. Downloads or more infos can be found at [33]. This database is very attractive since at least 40 images of the same category are present, which is essential for a good recognition at a large scale.

The second database has been suggested by Pascal Vittoz, biologist, and can be found on the website of the “Centre du Réseau Suisse de Floristique” (CRSF) [34]. It gathers only flowers growing within switzerland which is great, but the extraction had to be done manually and could once again become tiring for a larger application.

Last source of images was to go around the EPFL and take several pictures of some flowers with a digital camera. This demarche really brings an asset to the database as the testings with the phone will employ such self-taken pictures.

### 8.2 Resulting database

Finally by mixing a bit the three sources exposed in the previous paragraph, the final database contains 110 images. The repartition is roughly the following 60% from the Oxford dataset, 30% from the CRSF website and 10% of manual shots. Within each of the 29 categories there are between 2 and 6 flowers. A good variety of colors and shapes was obtained as it can be observed in Figure 17 at the end of the report.

## 9 Android

The application has been written in JAVA using Eclipse and it uses a Socket Server to run the different functions written in C++ and also holds the database features. First of all, a big thanks to Péter Vajda who created a nice interface to





Figure 17: Samples of images present in the database

conveniently run the application. When it is launched the camera is displayed and a photo may be taken and is then showed on the screen. Step 2, the user draws the markers on the image roughly delimiting the flower region from the periphery. When it's done, these two images (the original and the markers) are sent to the server where an executable is run for the segmentation (see Section 13 Appendix 13.2.5 for more info). The resulting segmented image is sent back and showed to the user who can either accept it or decides to draw new markers. In the latter case it goes back to step 2, otherwise the matching is run on the server (again from an executable) and the result for each feature is shown.



Figure 18: The android phone used for the implementation, a Samsung-i7500

## 10 Results

### 10.1 Database evaluation

In order to test the efficiency one can collect additional pictures of flowers present in the database and see if the system recognize them. But to have significant results another set of suitable test images would have to be found. So a ground truth evaluation of the database has been conducted.

#### 10.1.1 Ground Truth

It consists of going through all the images in the database and search the second best match (the first one obviously being the same image). If the flower indicated is part of the same category as the flower under test then it's a successful recognition. By doing this for the whole database, the performance of the system can be evaluated by establishing the recognition rate. The figure 19 illustrates this procedure.

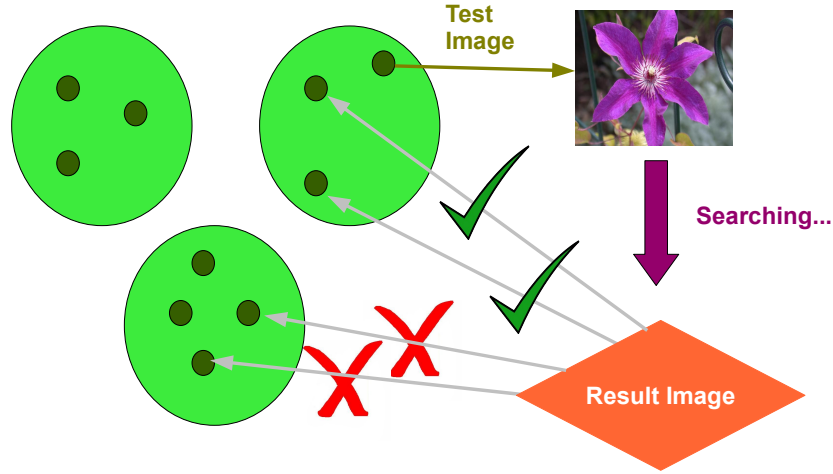


Figure 19: Ground Truth Schema, if the resulting image belongs to the same category as the test image it's a match

#### 10.1.2 Each Feature separately

This part has been done in Matlab, M-file `eval_feature_sep.m`. The code works as follows, a for loop that goes through all flower names runs the executable `groundTruth_euD.exe` (euD for euclidean dist). This executable returns the second best match for each feature and write it to a file which is read directly in the next statement by matlab and stored in a big cell called 'ResDist'. it is a 2x1 cell containing two 1x10 cell with each element being a 18x1 cell containing the result for the different features.

Then by going through these results and checking if the category of the flower under test is the same the percentage of recognition for the invidual feature is obtained and showed in the Figure

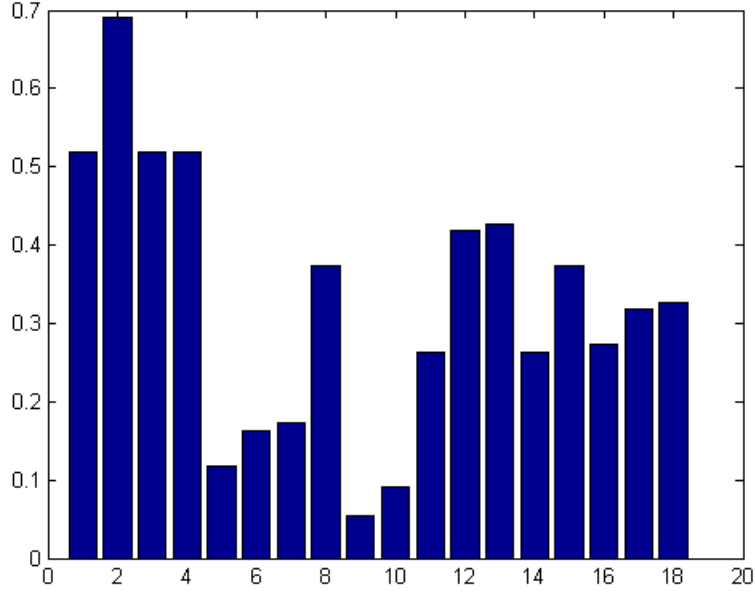


Figure 20: Results for each features separately, respectively 'ColorHistoRGB' 'ColorHistoLAB' 'ColorHistoHSV' 'ColorHistoYCrCb' 'Moments' 'Contour.Interpolation' 'Contour.Interpolation01' 'Contour.Projection' 'Contour.MinMax' 'Contour.AreaRatio' 'Contour.NumberPetal' 'Felib.ColorFeaturesHSV' 'Felib.ColorFeaturesHSV3D' 'Felib.ColorFeaturesRGB' 'Felib.ColorFeaturesLAB' 'Felib.ColorFeaturesRGBpyr' 'Felib.Gabor' 'Felib.Gist'

We quickly notice that the Color Histograms (the first four) are above all the others with the best result at 69% with HSV color histogram. We also note that the Moments, the Interpolations have a rate below 20% and even worst the two ratios, areas and min/max, have less than 10% of correct decisions. Being situated around 30-40% the FELib features are in the average.

Now the question is, would it be possible to improve these results by combining some features together?

### 10.1.3 Combining the features

This part has been done in matlab in `eval_all_features_distances.m`. This time a big 3D matrix is created holding all the euclidean distances between all the images and for all features! That means the matrix has a size of 110x18x110 and all the distances are normalized. Note that no C++ code is called here, the features are directly read from the files.

Then weights are assigned for each features, let's give them all a weight of 1

to begin with. All the distances are sum up along the features dimension yielding in a 2D 110x110 matrix with the score obtained from the distance calculated for all flowers compared to all flowers. To clarify, all the diagonal elements will be null as they represent distances between the same flower.

Now by taking the second smallest score for each flower and again checking if their category are identical will allow to deduce a recognition rate based minimal weighted distances.

The result by taking all features weighted equally is: 73.6 %. We see that there is already an improvement compared to the best feature alone (which was 69%). But now by playing around with these weights a better combination can be found. For instance, by taking only four features (HSV color histo, moments, projection and hsv-based Felib) a recognition rate of 79% is reached!

In the previous test the four features had the same weight, now by varying the weights through 3 loops (one weight was set to one as only their relative values are important). The obtained best weights were finally 1, 1.1, 1.5, 1.2 for respectively HSV Color Histo, Moments, Projections and HSV-based Felib.

A final recognition rate of 80% has then been achieved increasing of 10% the best feature taken alone as illustrated in Figure .

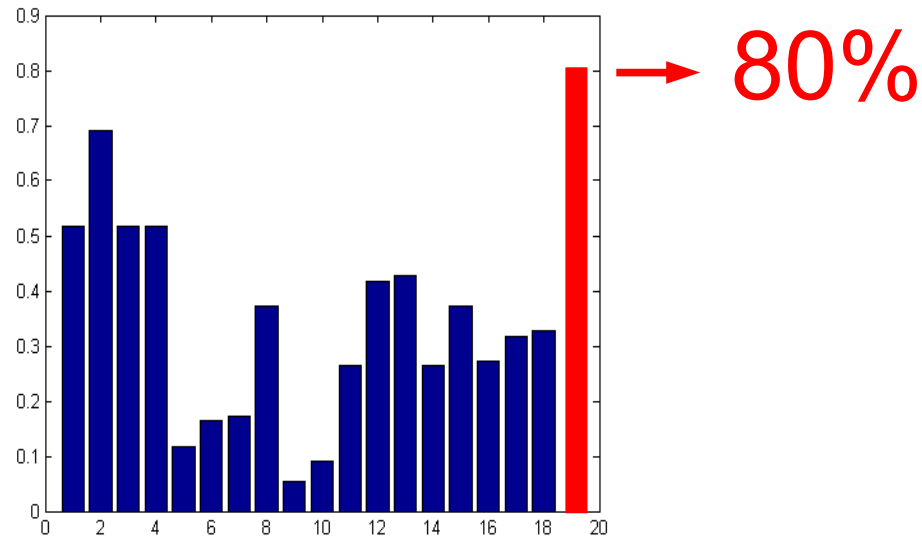


Figure 21: Final recognition rate of 80% reached with a combination of features

Note that surprisingly the feature Moment is part of the best solution even if its rate alone is very low ( $\sim 10\%$ ). This can be explained by the fact that this feature is not good at identifying similarities but is useful to exclude candidates.

## 10.2 Final Showdown

A few samples of flowers present in the database were gathered in order to test the framework on a real plant. The experiment was conducted indoors although the original pictures had been taken outdoors. And yet, after a good segmentation, often about 4 or 5 features named the right flower. Also during the oral presentation the experience has been offered to the jury to take a photo of a photo and, despite some color modifications through the printing, almost all features have resulted the correct flower!

## 11 Conclusion

This very ambitious project has been a source of constant learning for the author. The early stages consisted of investigating the biology dimension of the problem. With some documentation and the advices of a professional biologist the concepts were quickly clarified, even some ideas of features were already thought up. Then the review of the state of the art has helped a lot to understand the difficulties and the solutions proposed up to date. The elaboration of a nice framework has taken quite a bit of time but was worthwhile as when it was in place the addition of a new element (feature, distance, etc..) was very convenient. The segmentation step has been proven to be crucial and a marker-controlled watershed algorithm has been used. After having imported some features and implemented some others for a total of 18 different features, the testing phase could begin with the use of ground truth evaluation. Most of the features gave quite disappointed results by rarely being above 40%. But then by combining all the features together a promising rate of 73% was obtained. And further playing with the different ways of using these features a final recognition rate of 80% has been reached.

It would be very interesting to continue working on this project as this framework could serve as a kernel while exploring new solutions. Many new issues would arise if considering a finite product application which would make it even more motivating.

## 12 Future Work

In this section some ideas are thrown that could improve or complete the current system.

## 12.1 Adding extra information

The implementation on android has until now only brought additional difficulties or constraints. What if for once one can actually benefit from additional information available in a phone like when or where the picture was taken. Indeed flowers usually have a specific period of the year when they bloom or some can only grow on mountains or close to a lake, etc. Anyway having GPS coordinates or time of the year could extremely reduce the number of possible candidates.

## 12.2 Leaves Analysis

As explained by Pascal Vittoz (see Section 3.3) the leaves play an important role in the process of flower recognition. So the possibility should be given to the user to take one picture of the bloom and one picture of the leaf (if available).

Different shapes of leaves mainly in the teething exist as it has been shown in Figure 5. Also combined with an analysis of the nervature could significantly increase the recognition rate. In the figure 22 a photo of a leaf taken around EPFL, the nerves are clearly visible and the teeth well marked.



Figure 22: Photo of a very characteristic leaf

## 12.3 HS Color space Features

In [5] their color features are only based on the observation of some values of the HS plane which they separate in 72 regions as shown in the Figure 23 below.

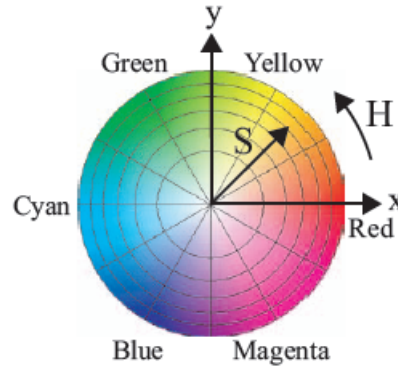


Figure 23: HS space divided in 72 cells

Then they identify the segment which has the largest entry and deduce many features, their best one (they say) being 'the x coordinate of the largest color cell'. This approach is worth exploring as it seems to give extremely good results.

## 12.4 Bigger database

Obviously if this application pretends to enter the market a much bigger database would have to be collected (maybe organized by continents or countries).

## 12.5 Automatic Segmentation

To continue with the previous point, a bigger database will introduce another problem which is the segmentation. Until now it was done manually but would be quite unfeasible for a larger database. Some methods could be explored for automatic segmentation such as improving the method employed in [5] which makes the assumption that the flower is focused and the background defocused, then uses a normalized cost method. To date no perfect method has been found but to have a self-sufficient segmentation process is something worthwhile.

## 12.6 Distance measurement

In this project mainly Euclidean distance has been used. Some tests with scalar product distance has been done but no relevant results were obtained. However the use of a new distance measurement such as the Mahalanobis distance could be of great use. Indeed this kind of distance measurement takes into account the neighboring values. While Euclidean distance would be rapidly big for slightly different histograms, Mahalanobis distance would be especially efficient as the data are highly correlated.



## References

- [1] Das M.; Manmatha R.; Riseman E.M.; “Indexing flower patent images using domain knowledge,” IEEE Intell Syst 14:24–33, 1999
- [2] Saitoh, T.; Kaneko, T.; “Automatic recognition of wild flowers,” Pattern Recognition, 2000. Proceedings. 15th International Conference on , vol.2, no., pp.507-510 vol.2, 2000
- [3] Nagy, G.; Jie Zou; , “Interactive visual pattern recognition,” Pattern Recognition, 2002. Proceedings. 16th International Conference on , vol.2, no., pp. 478- 481 vol.2, 2002
- [4] Jie Zou; Nagy, G.; , “Evaluation of model-based interactive flower recognition,” Pattern Recognition, 2004. ICPR 2004. Proceedings of the 17th International Conference on , vol.2, no., pp. 311- 314 Vol.2, 23-26 Aug. 2004
- [5] Saitoh, T.; Aoki, K.; Kaneko, T.; , “Automatic recognition of blooming flowers,” Pattern Recognition, 2004. ICPR 2004. Proceedings of the 17th International Conference on , vol.1, no., pp. 27- 30 Vol.1, 23-26 Aug. 2004
- [6] Saitoh T.; Aoki K.; Kaneko T.; , “Automatic extraction of object region from photographs,” In Proc. of 13th Scandinavian Conference on Image Analysis, volume LNCS 2749, pages 1130–1137, Jul. 2003.
- [7] Yang, C.C.; Chan, M.C.; , “Color image retrieval based on textural and chromatic features,” Systems, Man, and Cybernetics, 1999. IEEE SMC '99 Conference Proceedings. 1999 IEEE International Conference on , vol.4, no., pp.922-927 vol.4, 1999
- [8] Ing-Sheen Hsieh; Kuo-Chin Fan; , “Color image retrieval using shape and spatial properties,” Pattern Recognition, 2000. Proceedings. 15th International Conference on , vol.1, no., pp.1023-1026 vol.1, 2000
- [9] Brandt, S.; Laaksonen, J.; Oja, E.; , “Statistical shape features in content-based image retrieval,” Pattern Recognition, 2000. Proceedings. 15th International Conference on , vol.2, no., pp.1062-1065 vol.2, 2000
- [10] Seong-O Shim; Tae-Sun Choi; , “Edge color histogram for image retrieval,” Image Processing. 2002. Proceedings. 2002 International Conference on , vol.3, no., pp. 957- 960 vol.3, 24-28, June 2002
- [11] Adnan, A.; Gul, S.; Ali, M.; Dar, A.H.; , “Content Based Image Retrieval Using Geometrical-Shape of Objects in Image,” Emerging Technologies, 2007. ICET 2007. International Conference on , vol., no., pp.222-225, 12-13 Nov. 2007

- [12] Taijun Li; Qiuli Wu; Jiafu Yi; Cheng Chang; , “Color Sectors and Edge Features for Content-Based Image Retrieval,” *Fuzzy Systems and Knowledge Discovery*, 2007. FSKD 2007. Fourth International Conference on , vol.3, no., pp.234-238, 24-27, Aug. 2007
- [13] Jeong-Yo Ha; Gye-Young Kim; Hyung-Il Choi; , “The Content-Based Image Retrieval Method Using Multiple Features,” *Networked Computing and Advanced Information Management*, 2008. NCM '08. Fourth International Conference on , vol.1, no., pp.652-657, 2-4, Sept. 2008
- [14] Krishnan, N.; Banu, M.S.; Christiyana, C.C.; , “Content Based Image Retrieval Using Dominant Color Identification Based on Foreground Objects,” *Conference on Computational Intelligence and Multimedia Applications*, 2007. International Conference on , vol.3, no., pp.190-194, 13-15, Dec. 2007
- [15] Hiremath, P.S.; Pujari, J.; , “Content Based Image Retrieval Using Color, Texture and Shape Features,” *Advanced Computing and Communications*, 2007. ADCOM 2007. International Conference on , vol., no., pp.780-784, 18-21, Dec. 2007
- [16] Im C.; Nishida H.; Kunii T.L.; , “Recognizing Plant Species by Leaf Shapes-a Case Study of the Acer Family,” *Proc. Pattern Recognition*. 2 1171–1173, 1998
- [17] Wang, Z., Chi, Z., Feng, D.: Fuzzy Integral for Leaf Image Retrieval. *Proc. Fuzzy Systems*. 1 (2002) 372–377
- [18] Wang X.F.; Du J.X.; Zhang G.J.; , “Recognition of Leaf Images Based on Shape Features Using a Hypersphere Classifier,” In *Advances in Intelligent Computing*, Vol. 3644, pp. 87-96, 2005
- [19] Zhenjiang Miao, M.-H. Gandelin, and Baozong Yuan, “A new image shape analysis approach and its application to flower shape analysis”, *Image and Vision Computing*, vol. 24, (10), pp. 1115-1122, October 2006
- [20] Yanhua Ye; Chun Chen; Chun-Tak Li; Hong Fu; Zheru Chi; , “A computerized plant species recognition system," *Intelligent Multimedia, Video and Speech Processing*,” *Proceedings of 2004 International Symposium on* , vol., no., pp. 723- 726, 20-22, Oct. 2004
- [21] Zhenjiang M.; Gandelin M.H.; Baozong Y.; , “An OOPR-based rose variety recognition system,” *Engineering Applications of Artificial Intelligence archive*, Vol. 19 , pp. 79-101, 2006
- [22] Nilsback M.E. and Zisserman A., “Delving into the whorl of flower segmentation,” *Proceedings of the British Machine Vision Conference*. 2007
- [23] Boykov Y.Y. and Jolly M.P., “Interactive graph cuts for optimal boundary and region segmentation of objects in N-D images,” In *Proc. ICCV*, volume 2, pages 105–112, 2001.

- [24] Liu, Pengyu; Jia, Kebin; Wang, Zhuozheng; , "A New Image Retrieval Method Based on Combined Features and Feature Statistic," Image and Signal Processing, 2008. CISP '08. Congress on , vol.2, no., pp.635-639, 27-30 May 2008
- [25] Nilsback, M.-E.; Zisserman, A.; , "A Visual Vocabulary for Flower Classification," Computer Vision and Pattern Recognition, 2006 IEEE Computer Society Conference on , vol.2, no., pp. 1447- 1454, 2006
- [26] Jung-Hyun Kim; Rong-Guo Huang; Sang-Hyeon Jin; Kwang-Seok Hong; , "Mobile-Based Flower Recognition System," Intelligent Information Technology Application, 2009. IITA 2009. Third International Symposium on , vol.3, no., pp.580-583, 21-22 Nov. 2009
- [27] Zhu J.; Hoi S.C.H.; Lyu M.R.; Yan S.; , "Near-Duplicate Keyframe Retrieval by Nonrigid Image Matching," ACM Multimedia 2008, pp. 41-50, 2008
- [28] Douze M.; Jégou H.; Sandhawalia H.; Amsaleg L.; Schmid C.; , "Evaluation of GIST descriptors for web-scale image search," CIVR 09 Santorini, 2009
- [29] D. Lowe, "Distinctive image features from scale-invariant keypoints," 2003
- [30] K.E.A. van de Sande, T. Gevers, and C. G. M. Snoek, "Evaluation of color descriptors for object and scene recognition," in IEEE Conference on Computer Vision and Pattern Recognition, Anchorage, Alaska, USA, 2008
- [31] Maheshwary P. and Srivastav N.; "Retrieving Similar Image Using Color Moment Feature Detector and K-Means Clustering of Remote Sensing Images," In Proceedings of the 2008 international Conference on Computer and Electrical Engineering, pp. 821-824, 2008
- [32] <http://www.vision.ee.ethz.ch/~zhuji/felib.html>
- [33] <http://www.robots.ox.ac.uk/~vgg/data/flowers/102/index.html>
- [34] <http://www.crsf.ch/?page=fotos>
- [35] Gary Bradski and Adrian Kaehler, Learning OpenCV, O'Reilly Media, September 2008

## 13 Appendix

### 13.1 Files organisation

The working directory should contain the following elements

- a text file with all the names of the flowers (listImage.txt)

- a folder 'images' with all the flowers with their names starting with 'flower.' (exemple: flower.sunflower.jpg)
- a folder 'segments' with all the segments with the same names as the flowers but this time starting with 'segments.' (exemple: segments.sunflower.PNG)
- a folder 'masks' with all the masks also with the same name but starting with 'mask.' (exemple: mask.sunflower.jpg)
- a folder 'features' with all the features for each flower, exemple: feature.sunflower.Gabor.txt

## 13.2 Software Documentation

In this section you will first find a sketch of the tools and the general class diagram. Then all classes present in the system architecture are explained one by one, presenting their basic utilisation and their fonctionnalités are roughly described.

Note that the Doxygen documentation will be attached on the DVD.

### 13.2.1 Tools

The computer language used for this project is C++, compiled with Microsoft Visual C++ 2008 Express Edition. The computer vision library OpenCV has also been employed.

In a first stage the idea was to code everything in JAVA so as to implement the application on android. The importation of OpenCV library into JAVA was not straightforward and even if a way has finally been found, its utilisation then was inconvenient as a lots of functions did not work the same way and the documentation was very limited. To save a lot of time the decision was taken to code everything in C++ and the application to be run on the server.

### 13.2.2 Class diagram

As shown in Figure 6 there are three main steps :

- Segmentation
- Features Extraction
- Matching

So it seems like a rational decision to create a class for each of these tasks, they will be the core of the class diagram which is shown in Figure 24.

The class Feature, which is the base class for all other features, and the class Segmentation are both necessary for the database evaluation as well as for the matching. An additional class called Paths deals with all the paths in the code to have a nice environment-independant architecture for all the other classes.

Usually the code is well commented and should be self explanatory.

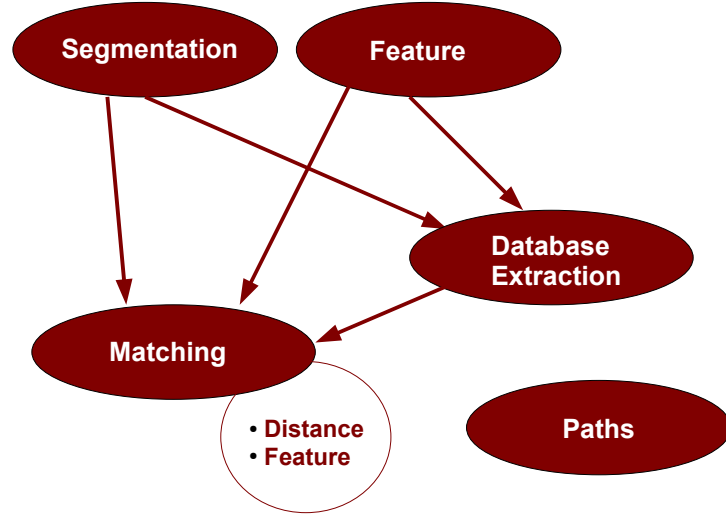


Figure 24: Class diagram

### 13.2.3 Paths

The class Paths impose the organisation of the folder and files and can be modified to the wish of the user. At the moment a folder 'images', 'segments', 'masks' and 'features' have to be present in the working directory which is the main member. The second member is 'listFile' and correspond to the name of the file containing all the flowers names. It should be within the working directory. Then all the other members are paths for respectively the flowers, the segments, the masks and the features; always from the working directory. The constructor takes 6 string arguments to assign these values. While the constructor is called the values for each of the arguments is stored in another member in order to remember its initial value while calling the method 'generatePaths'. This is very usefull when treating the all database not to have filename like flower1.jpg.flower2.jpg etc.

The method 'generatePaths' is used to initialise all the path when the name of the flower to be treated is known. The last method 'setFeatureType' is used after the declaration of a new Feature in order to get suitable features file names.

### 13.2.4 Feature

This class is the based class for all features. Its main member is `myFeatVec` which is a vector of float. It contains the values extracted by a given feature. Other members are `'img'`, the flower image, `'mask'`, the mask image, essential for feature extraction. The constructor takes the paths of this two guys as arguments and load them within the class. Every derived class has to call the constructor of the base class! A getter has been created for `myFeatVec`, called `'getFeatVec()'`.

`'readNames'` is used to read the flowers names from the `txtFile` and put them in a vector of string. Two other methods `'findMax'` and `'findMin'` return the index of respectively the maximum and the minimum element in a vector of float.

All classes derived from `Feature` should have a `'extractFeature()'` method which assign the member `'myFeatVec'` to the current features values it also write the features to a file with the suitable name. The first line is a header which indicates how many elements follow.

**ColorHisto** The constructor of this class has an additional argument to specify in which color space the histogram has to be computed, “RGB”, “LAB”, “HSV”, “YCrCb”. Its `extractFeatures` uses the histograms facility of OpenCV and produces a vector of length  $3 \cdot 256 = 768$ .

**Contour** This class has a few children and incorporates new members that are `'distancesVec'`, `'anglesVec'`, `'maxDist'` and `'minDist'`, which are respectively a vector of float representing the distances for each point on the contour, a vector of float representing the angles corresponding to the distances, the maximum distance and the minimum distance. The method `'assign_DisAngVecs()'` will find the contour and assign the two vectors members.

Note that the Interpolation in order to get a regular angles axis is made in the derived class `Interpolation` which will inheritate of these `distancesVec` and `anglesVec`. All other children can use the already computed values.

**Felib** The incorporation of this library in the architecture was not straightforward since it has been created for the language C and some linker error recursively appear unless the keyword

```
extern "C" { }
```

embraced every declarations of the functions implemented in the library! Moreover with the source code not being available it is often hard to check what such-and-such function does and how to correctly use it.

Without the source code it is also impossible to explain how the extractions are done, but one thing for sure is that it doesn't give the possibility to input a mask which is quite an important drawback because it will take the background into account for each image !

Note that for the sake of good order all non-flower pixels have been set to zero.

**ColorFeatures** The constructor has an additional argument to specify which features to use (default is HSV), the codewords are : “HSV”, “HSV3D”, “RGB”, “LAB”, “RGBpyr”.

### 13.2.5 Segmentation

**The class** The class Segmentation has 3 members: img, segments and mask. img is the original flower, segments are the user input markers, mask is the resulting mask.

The constructor takes the full path of the flower as argument and load it into the member 'img'.

The method genSegments is meant to be the result of the user's drawing. In this case the segments have already been established (yielding in a good mask) and the function just loads the image into the member 'seg'.

The method getMask is a getter for the mask.

The method SegmentMe actually does the segmentation. Its input is the path of the mask where the output should be written to. It mainly uses the OpenCV function 'cvWatershed' (see [35] for more information) with a little trick. The cited function takes a pointer to a IplImage as argument but the depth should be a signed 32-bits integer but all images have unsigned 8-bits integer depth. This transformation is far from being straightforward and should use the function cvFindContour to find the contour in the first image (the U. 8-bits) and draw these contours into the other image (S. 32-bits) using cvDrawContour (more infos about these functions again in [35]). Finally the output mask is painted after the watershed has been called.

**The executable** In the final application implemented on the android phone, the segmentation will be called from the server in JAVA. An executable had to be created to do so. Its name is segmentationExe and takes 3 arguments that are : path of the flower image, path of the segments image (just drawn by the user), path where the output mask should be saved.

### 13.2.6 DatabaseEvaluation

The DatabaseEvaluation class acts as a main but for the full set of flowers. It surely need a Paths to be able to write the features in the right place with the right names. The txtFile is also a member and will be used to go through all the flowers in the database. listNames has already been met and is also assigned with the method 'setListNames' called within the constructor. The constructor takes a Paths as argument to assign the member myPathsDB.

The main function is 'doIt()' which will go through all the image extract all the features and write them all in multiple files with the correct names. After

that the Matching class will go read the values written directly in these files without having to recalculate everything.

### **13.2.7 Matching**

This class is probably the most important, it has many members one of them being a typedef 'FeatureVector' which is a vector of Feature. It was created to be able to add as many Feature as wanted for the matching. This is done by using the method 'addFeature()'. It also contains a distance as different measurement can be used, default is Euclidean. The method 'getResult()' will go through the featureVector and for each feature separately compare the test flower with all the others in the database. Another method 'compare()' care about this task and will read the features in the file find the closest one (method present in Distance) and returns the resulting flower.

### **13.2.8 The detectExe**

To run the whole framework in JAVA an executable has also been done for the detection. Its name is detectExe and it was created in another project to deal with some issues. It should take 3 arguments the path of the image, the path of the mask, and the path of the file where the results have to be written (for each feature). It also e.g. doesnt write to files the features of the flower under test.

## **13.3 DVD**

In the DVD the following items can be found:

- The database (images, segments, masks, features)
- EVALUATION folder
- Source Code: segmentExe, detectExe, FlowerRec (original source code)
- Doxygen documentation
- Final Presentation
- Papers
- Meeting with biologist material : Floristique\_09.pdf, legend.xls, Plants with Flower in Switzerland by Pascal Vittoz.xls
- FELib library original "source" code
- Final Report.pdf
- README.TXT (for evaluation)