Final report on pollen project

Marc'Aurelio Ranzato

26th August 2004

Database

The database used in the experiments is shown in tab.1 and 2. This database was double-checked by the expert (6 August 2004).

folder	images
*010403	195
022203	19
022303	10
022403	15
030310	22
030311	80
*032102	115
*042304_0-12	100
*042802	180
091903	30
120102	44
120202	8
*122902	158
*ChinElm	216
TrialPollen	71
TOTAL	1429

Table 1: Pollen database: list of folders and images in each folder. The * denotes the new available data in addition to the old database (March 04).

genus	number of samples
alder	**416**
ash	**329**
asterac.	6
birch	4
camplor	9
chenopod	32
chin. elm	**522**
c. myrtle	3
cypress	46
eucal.	30
ginkgo	3
jacaranda	**282**
liq. amber	44
mulberry	**111**
oak	**413**
olive	**326**
palm	39
pecan	4
pine	**705**
plane t.	10
poplar	7
sycamore	33
walnut	36
unknown	263
pistach.	0
plantain	0
elm	1
rumex	1
umbellif.	4
grass	19
TOTAL	*3686*

Table 2: List of pollen particles found in the images of tab.1. In bold characters, we highlight the species with more than 100 samples.

Detection: outline

The system is based on a filtering approach and its outline is shown in fig. 1 and 2.

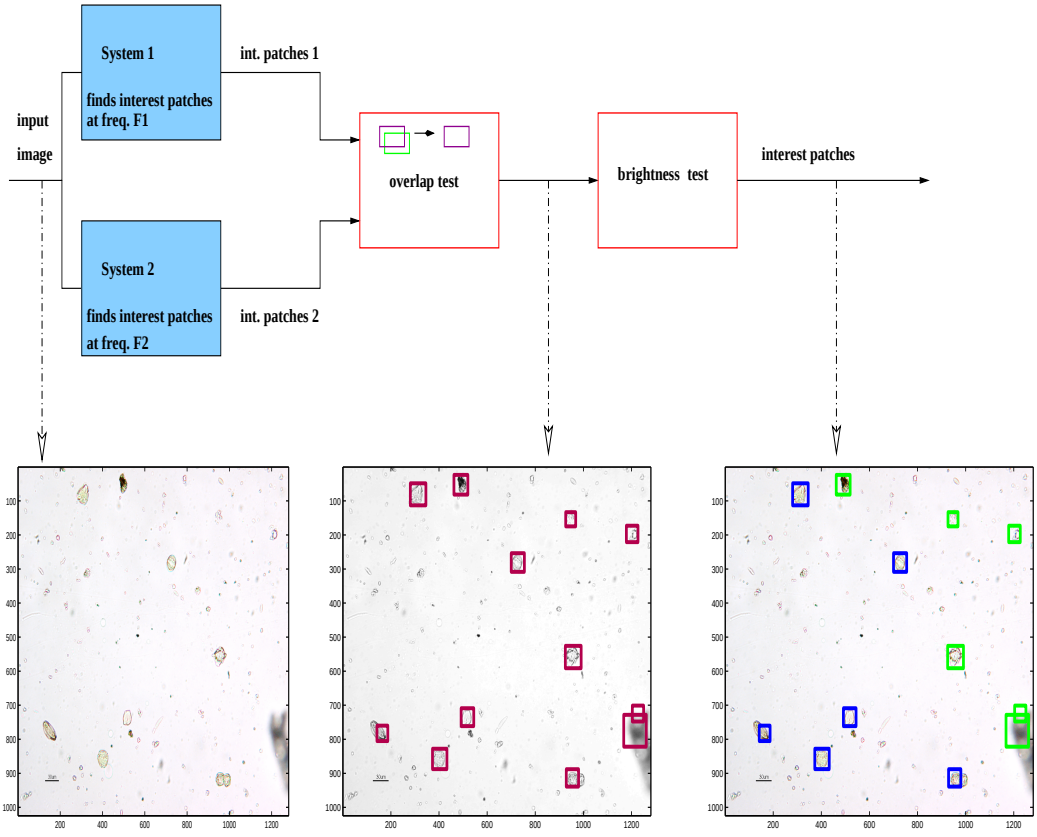

Figure 1: Outline of particle detector. Patches of interest are found by a system using two different frequencies (the light blue boxes that are described in fig. 2). Patches coming from these two systems are retained only if they pass a test on overlap and a test on average brightness (boxes with red border). For example, if two patches are too much overlapped we keep only one patch. If the average brightness of pixels falling inside the box is too high, we discard this patch because pollen particles have always some texture. The output is a set of patches that will be given to the classifier. In the example, the blue boxes are pollen grains, the green ones are false alarms.

The input image is converted in gray level values. It is blurred and sampled at two different frequencies, F_1 and F_2. In order to avoid artifacts near the image border, we normalize the background brightness to zero[1]. We apply to both

[1]The estimate of the background brightness is iterative and it is based on computation of the histogram peak of all the image pixel values.

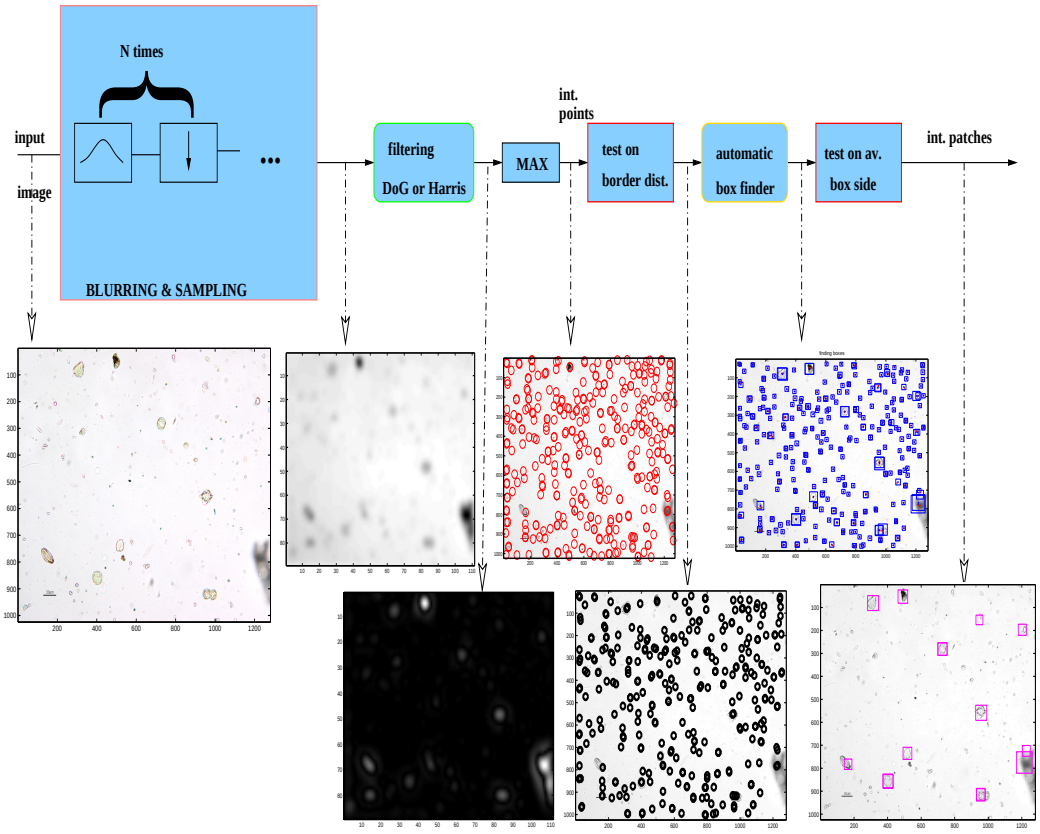

Figure 2: Zoom of the light blue boxes shown in fig. 1. The input image (after conversion to gray level values) is blurred and sampled at a certain frequency. Then, we look for the maxima of the difference of Gaussians (DoG) image. These maxima are our initial set of interest points. Then, we discard points too close to the image border. After that, we automatically compute a bounding box around each interest point. Finally, we discard the patches whose average side is too small. The output is a set of patches that might contain pollen grains.

sampled images a filtering stage: for each such image we compute a difference of Gaussians. A set of interest points is extracted looking at the extrema of the two previously found images. Then, we discard points that are too close to the image border and for the remaining ones we find a proper bounding box. After removing those patches that are too small, too much overlapped with other patches and patches whose average brightness is too high, we have a set of patches that might contain a particle of interest and then, these patches will be given to the classifier. In the following subsections, we will explain in some details the key points of the system.

The difference of Gaussians image

Given a properly blurred and sampled image, we blur the image twice with a Gaussian kernel. The difference of Gaussians (DoG) image is the difference between the outputs of the two filters.
In fig. 3, we show the plots of Gaussian kernel and difference of Gaussian (DoG) kernel and their DFT in the 1-D case.
The goal of the sampling stage is to make the smallest pollen appear a little blob of few pixels. This automatically removes dust particles smaller than pollen grains and reduces the amount of computation. With a proper choice of the variance of the Gaussian kernel σ, the output image will have a peak in correspondence of round shaped objects with a certain size. Fig. 4 shows a synthetic example in which two filters with DoG kernels having different σ are applied to an image. When σ is low, the output image has a narrow peak in correspondence of the little blob of the input image. When σ is high, the peak is in correspondence of the largest blob. The variance of the Gaussian kernel allows to select the size of the interest objects.
In order to detect how good a point of extremum is in the output image, we fit around it a paraboloid. The vertex of this paraboloid gives the exact position of the interest point and its curvature is a measure of its quality. If this value is too low then, it is likely that this extremum is generated by a strip shaped object or by an object that is smaller than the one we are looking for. If this value is high then the object in that position is round and it has a size that fits the size of our target.

Bounding box determination

The algorithm to determine a bounding box given an interest point is very straightforward. Its pseudo-code is given in fig. 5 and in fig. 6 we show an example of application of this algorithm.

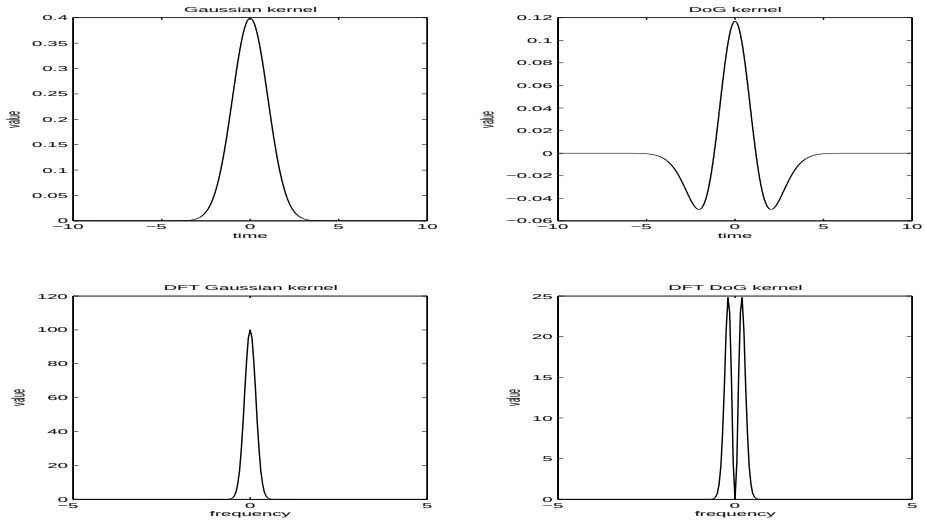

Figure 3: First row: basic kernels, namely Gaussian kernel and the equivalent kernel of the whole system. Second row: DFT of previous kernels.

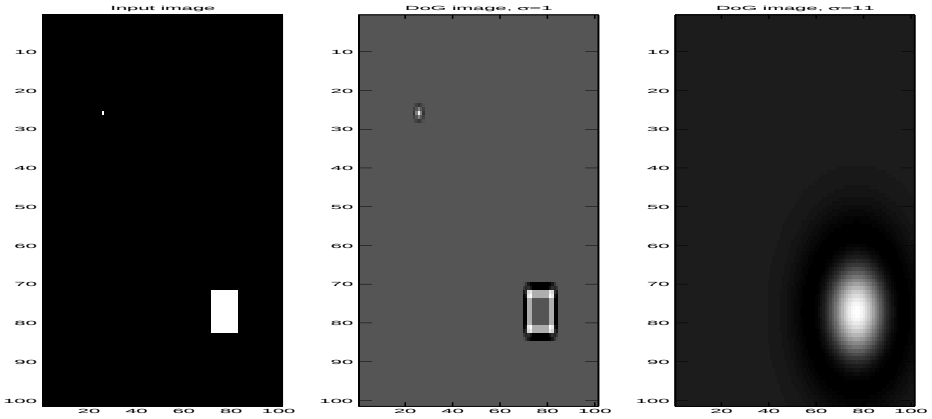

Figure 4: This example shows how the choice of scale σ is related with the dimension of objects of interest. The first image has a little blob of 1 pixel in the upper left corner and a larger one of 10 pixel diameter in the lower right corner. The second image shows a good peak in the position of the smaller blob, σ was chosen equal to 1. The third image shows a good peak in the position of the larger blob, σ was chosen equal to $8\sqrt{2}$.

ALGORITHM for bounding box determination

Input: interest point inside particle

DO:

1. consider 150 pixel values starting from your interest point in direction: N, S, E, W, NW, NE, SE, SW; you will have 8 vectors $V_1 \ldots V_8$
2. group the pixel values in each vector in set of three and compute their mean; you will have 8 vectors $v_1 \ldots v_8$, each with 50 items
3. for each vector v_i, estimate the background brightness in that direction looking at the peak of the histogram of values contained in v_i
4. the border of the object is found when the mean squared error of 4 consecutive values in v_i with respect to the background brightness is less than a certain threshold; you will get 8 points around your interest point
5. fit a circle taking three points and try many combinations (i.e. points in N-NE-E, N-SW-E, etc.)
6. the optimal bounding circle is the one with minimum error e; for each estimated circle, we compute the distances of the points not involved in the estimation (that are 5) to the circumference and we define e as the sum of the three smallest distances
7. we take the center of the circle as centroid and its diameter as side of a square bounding box.

Output: bounding box

Figure 5: Pseudo code of algorithm for determination of bounding box.

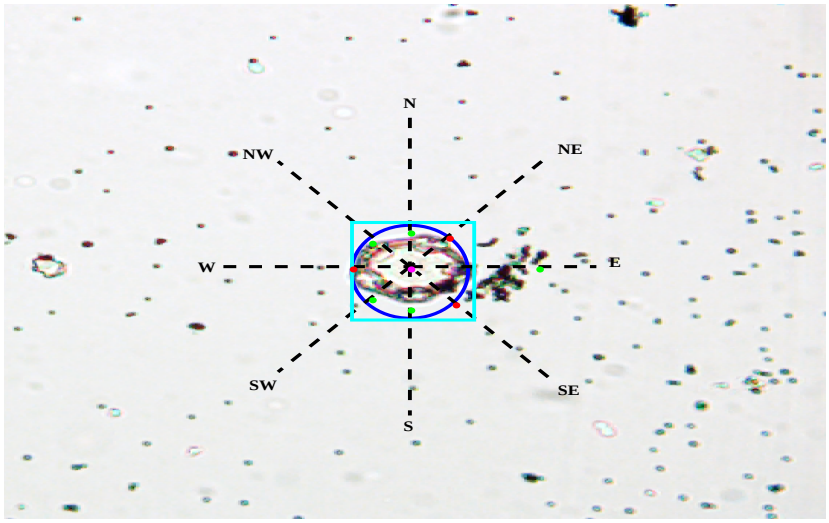

Figure 6: Example of bounding box determination. The point in magenta is the interest point. The green and red points are the estimated points around the particle border. The red points give the lowest error when fitting the circle. The resulting box is the cyan square.

We consider 150 pixel values in each direction because pine pollen (the biggest one) can be bounded by a square box of 300 pixels. Then, we compute the mean of groups of three pixel values in order to make less computations and to reduce the noise. We consider 4 values of mean for the mean squared error (point 4) because we assume that outside the pollen the signal is stable around the background value and there are at least 12 pixels of separation among particles. Error e takes into account only three directions because we allow to have two wrong points around the border of the object (think about grains that are grouped together or see the dust near the pollen of fig. 6). On the other hand, we can be confident that the circle with minimum e describes quite well 6 points out of 8 staying in the object border.

Overlap criterion

When images are very noisy in the background or pollen grains are grouped together, there can be more than one bounding box for particle or there can be a lot of bounding boxes that are very close and overlapped. Then, we have to choose which bounding box to retain and which one to remove.
Our criterion focuses on two main cases. If box A is overlapped to box B and no other box is overlapped with A and B then, we keep the box with minimum error e as defined in the previous subsection. If box A is overlapped to box B and both

boxes A and B are overlapped to box C, then, we keep the box with the smallest value of e. Finally, if box A is overlapped to box B, and only box B is overlapped to box C then, we keep box B if it has the smallest error e otherwise we remove only box B.

Examples

In order to have a better understanding of the computations performed by this system and to see its major limitations, we will take into account two images in our database. The first one is an example of very good detection (fig. 7, fig. 8, fig. 9), the latter one of poor detection (fig. 10, fig. 11, fig. 12).

Main causes of errors are:

- some pollen grains are very low contrast and they don't give any interest point
- pollen grains are sometimes grouped together and we can usually get only an interest point per group, the algorithm for bounding box determination can fail in this situation
- bounding box determination and overlap test can fail in very noisy images.

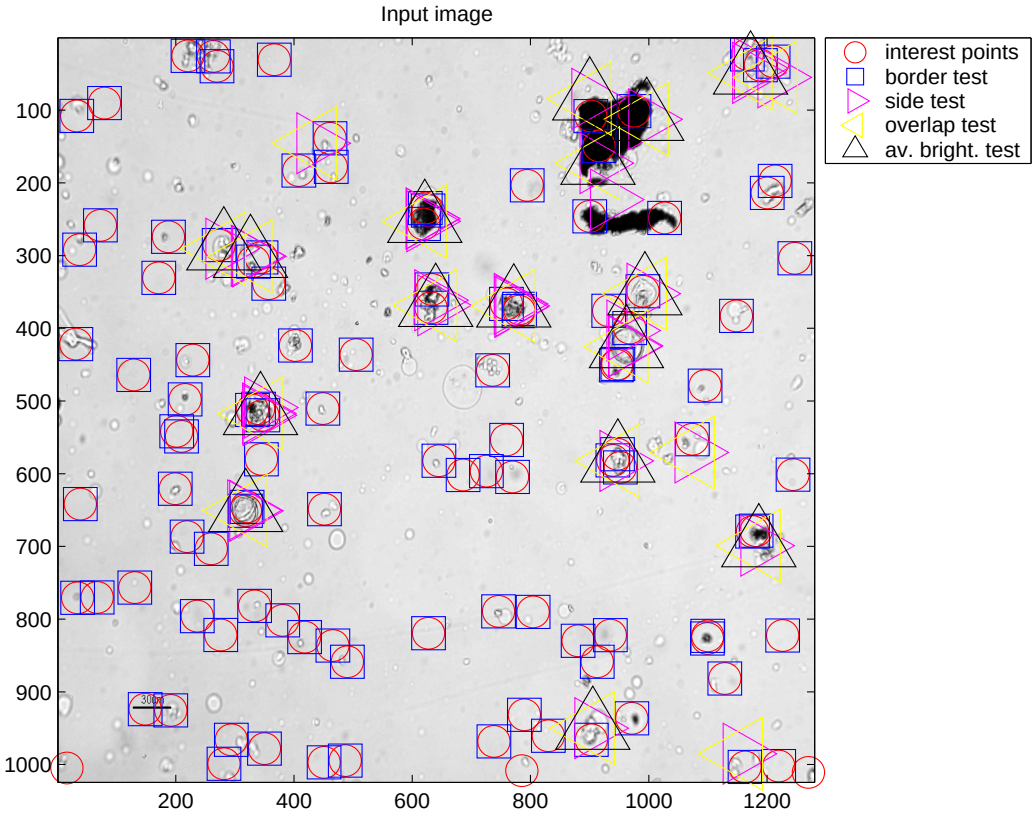

Figure 7: Example of good detection. The round red markers show the position of interest points. The square blue markers show the points that passed the test on the distance to the border. The magenta triangular markers are positioned in the centroids of those patches that have passed the test on the size length. The yellow triangular markers are located in the centroids of those patches that have passed the test on overlap. Finally, the black triangular markers show the position of the centroids of patches that passed the test on average brightness.

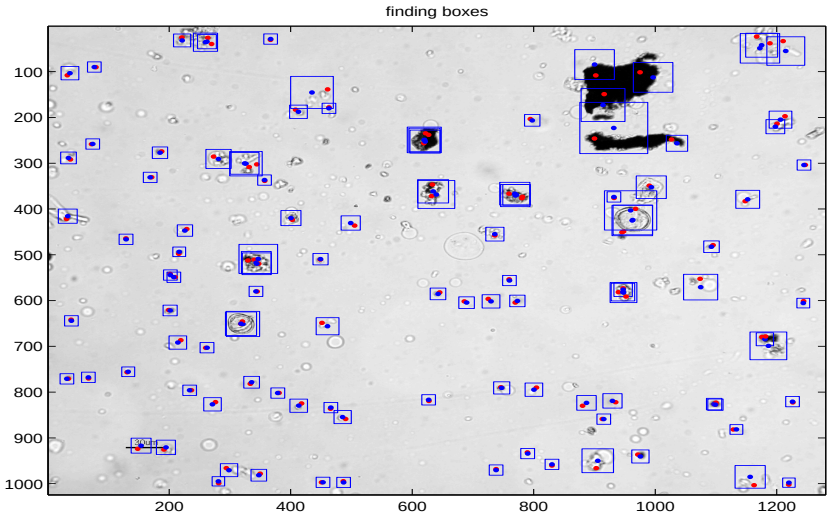

Figure 8: Example of good detection. The red points are the interest points that passed the test on border distance. The blue square boxes are the automatically computed bounding boxes. The blue point is the centroid of these bounding boxes.

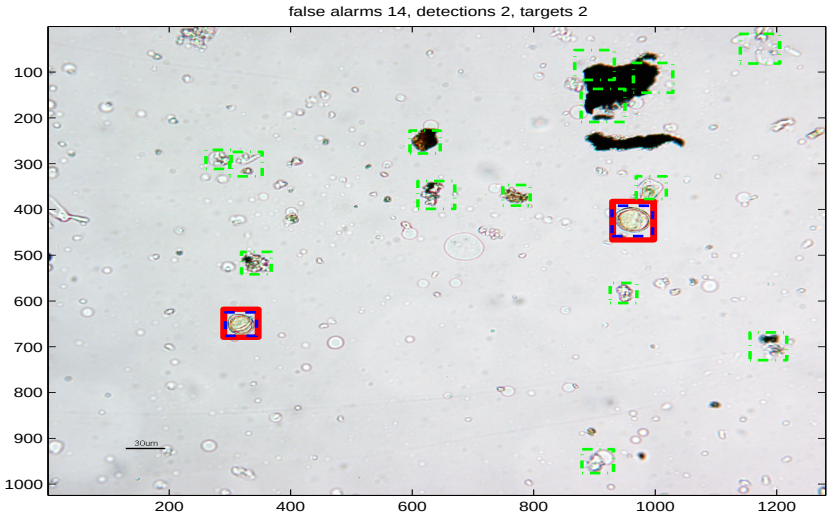

Figure 9: Example of good detection. The red boxes are dragged by the expert. The blue boxes show the patches computed by the system that match the expert's ones. The green boxes are patches computed by the system that are false positives. In this image, all pollen grains are detected and other 14 particles that are not pollen grains are detected.

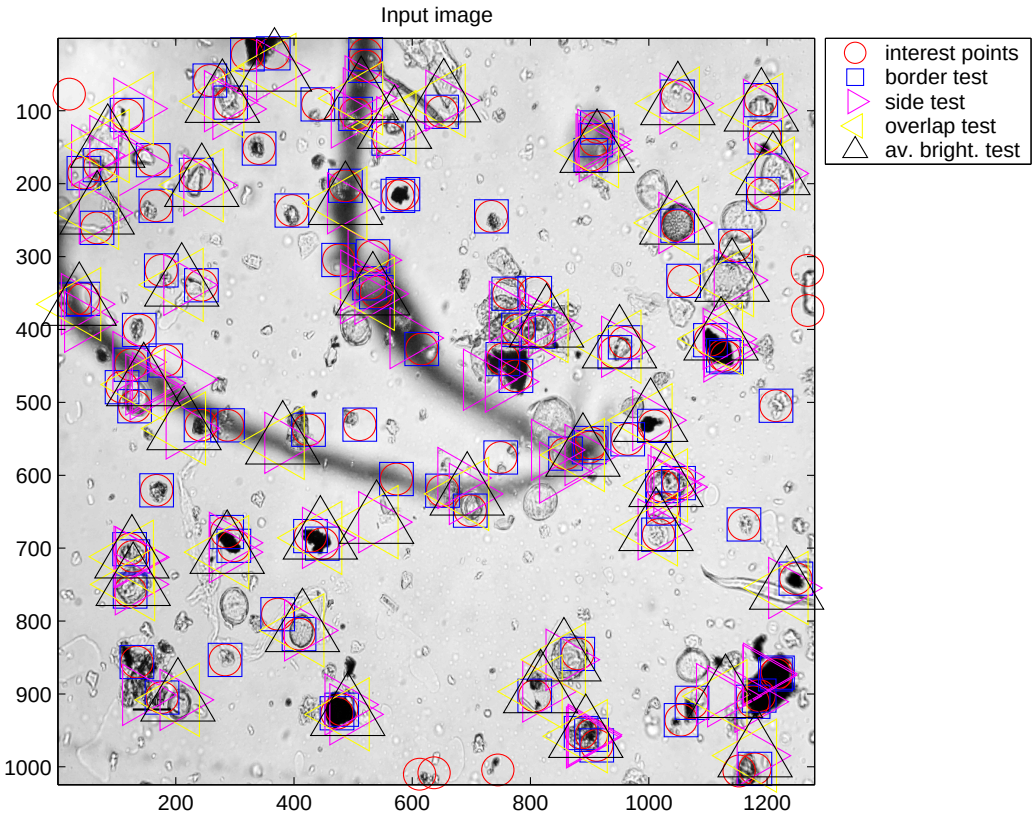

Figure 10: Example of bad detection. The round red markers show the position of interest points. The square blue markers show the points that passed the test on the distance to the border. The magenta triangular markers are positioned in the centroids of those patches that have passed the test on the size length. The yellow triangular markers are located in the centroids of those patches that have passed the test on overlap. Finally, the black triangular markers show the position of the centroids of patches that passed the test on average brightness.

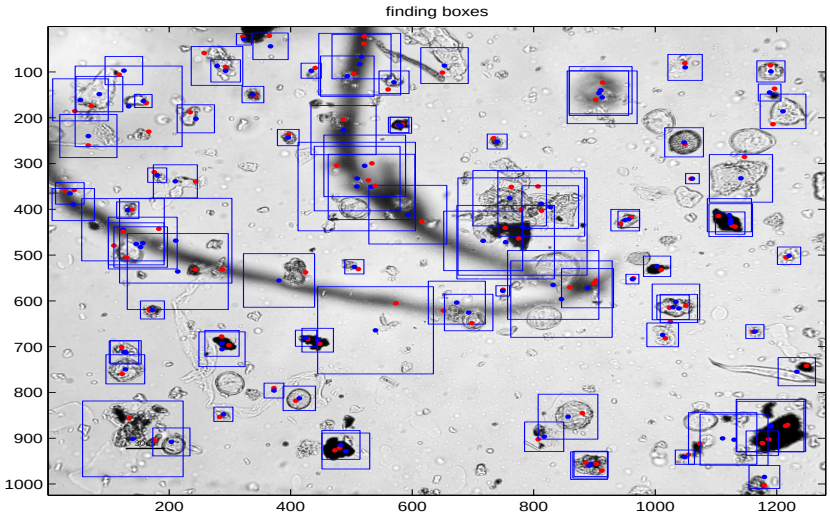

Figure 11: Example of bad detection. The red points are the interest points that passed the test on border distance. The blue square boxes are the automatically computed bounding boxes. The blue point is the centroid of these bounding boxes.

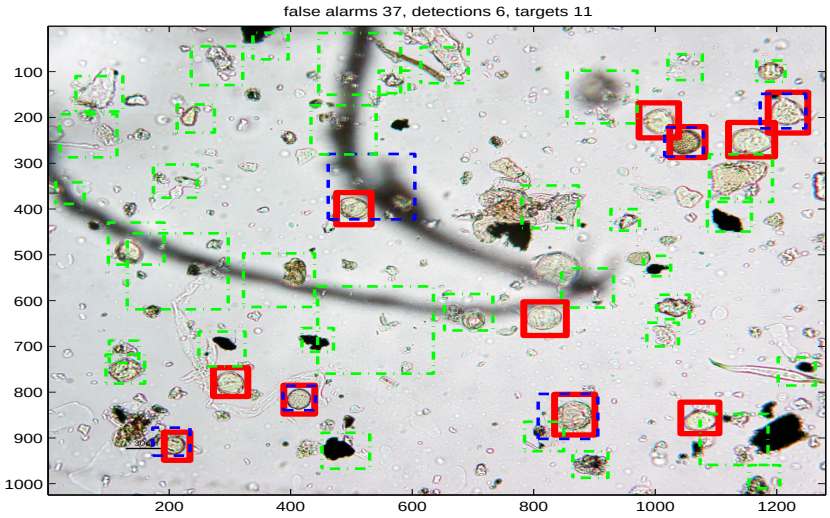

Figure 12: Example of bad detection. The red boxes are dragged by the expert. The blue boxes show the patches computed by the system that match the expert's ones. The green boxes are patches computed by the system that are false positives. In this image, there are 6 detections, 5 missed detections and 37 false alarms.

Detection: experiments

We studied the performance of this detector changing the values of the sample frequency and the variance of the Gaussian kernel. With the best achieved performance, we run many experiments varying the parameters MIN_K and MAX_ERR in the function that computes the interest points. Finally, we tried many values for the threshold used in the brightness test. With the best parameters that have been found (see tab. 3), we run a final experiment on all the images of pollen database. The result is shown in tab.4. We are able to detect more than 90% of pollen particles with a percentage of false alarms around 1000% using the DoG image.
Note that the system is quite fast because it takes about 5 sec. to evaluate an image 1024x1280 pixels on machine working with a 2GHz processor.

Parameter	**DoG**
F_1	8
F_2	$8\sqrt{2}$
σ	$\sqrt{2}$
MAX_ERR	0.2
MIN_K	0.1
$Bright.\ thres.$	0.97

Table 3: Parameters of detector that have been found during the tuning stage.

folder	**images**	**nr. grains**	**DoG**	
			%det	%false al.
*010403	195	959	98.0	515
022203	19	35	88.6	3030
022303	10	11	100	3140
022403	15	28	96.4	1610
030310	22	56	92.9	1560
030311	80	102	98.0	1820
*032102	115	426	89.9	742
*042304_0-12	100	500	90.0	519
*042802	180	520	86.3	1236
091903	30	66	93.9	1090
120102	44	56	98.2	1560
120202	8	12	91.7	2390
*122902	158	273	98.2	2470
*ChinElm	216	424	97.2	1000
TrialPollen	71	235	96.6	950
TOTAL	*1263*	*3703*	***93.9***	***995***

Table 4: Detection performance on each folder of the current database. In the last row, the final result is given.

Analysis of results

We have already observed that the main causes of errors are due to dust particles that are attached or very close to pollen grains or to pollen particles that are grouped in a single blob. In these situations, we are not always able to have an interest point for each pollen grain and the algorithm to find a proper bounding box can fail.
Since our final goal is to classify the detected patches, we have built a database with the detected patches (using DoG images) and we have sorted them according to the expert's reference list. Then, we have checked the patches belonging to the most numerous classes in order to discard the *bad* samples. Bad samples are produced by a mistake of our bounding box algorithm in the situations above mentioned. This procedure will give us a more reliable percentage of detection and it will allow us to run experiments of classification. In this way, we do not risk to consider as pollen particles some false alarm samples both in training and test stage.

According to this analysis, the new final results are shown in tab.5.
Examples of good and bad patches for some of the most numerous classes are shown in fig. 13 and following.

genus	**expert's**	**detected**	**% det.**	**detected good**	**% det.**
alder	416	402	96.6	355	85.3
ash	329	323	98.8	303	92.1
chin. elm	522	506	96.9	466	89.3
jacaranda	282	259	91.8	223	79.1
mulberry	111	88	79.3	74	66.7
oak	413	384	93.0	329	79.7
olive	326	288	88.3	220	67.5
pine	705	694	98.4	606	86.0
TOTAL	*3104*	*2944*	*94.9*	*2576*	*83.0*

Table 5: Performance of detector using DoG images when considering the most numerous classes. The number of detected patches is given and also the percentage of detection with reference to the number of expert's patches for each class. The last two columns refer to the patches that can be reasonably classified.

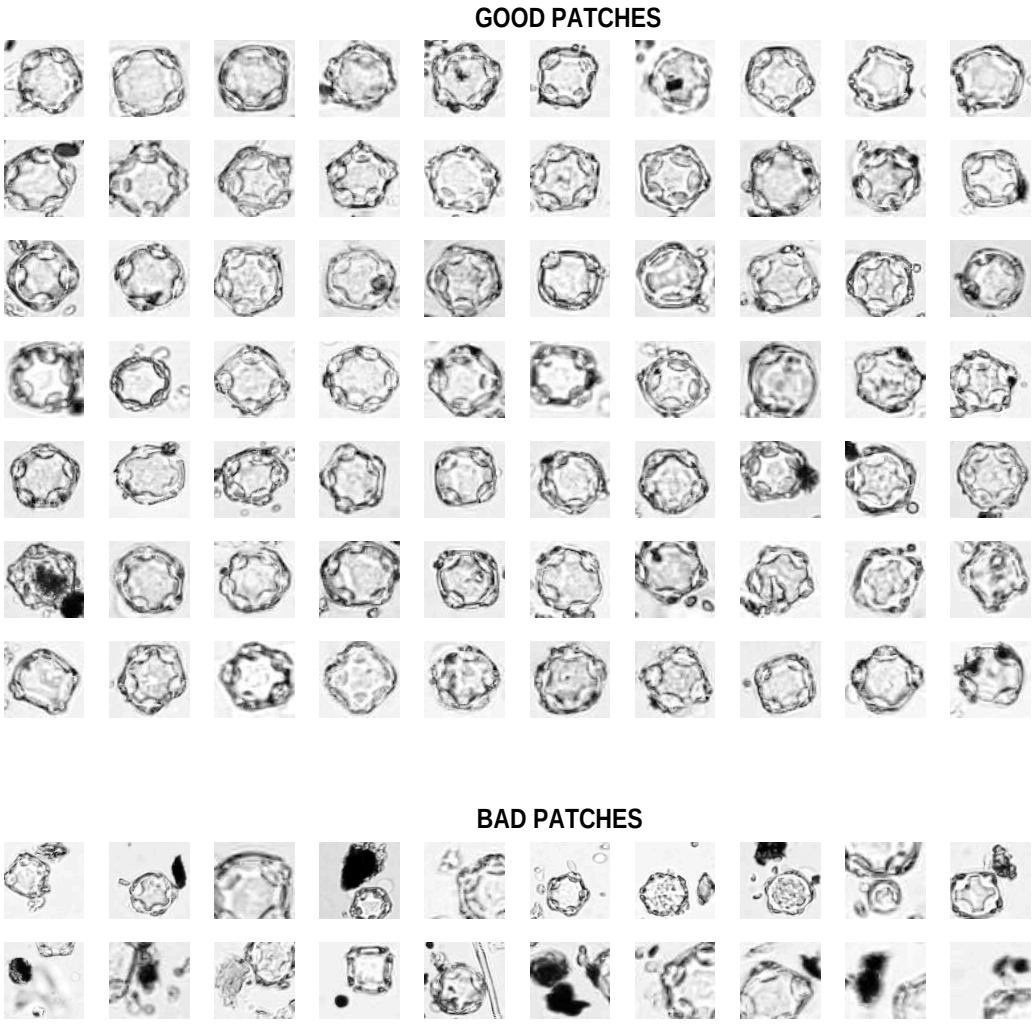

Figure 13: Example of 90 patches of alder pollen extracted by detector. The first 7 rows show examples of good segmentation. The last 2 rows show samples that have been removed because of a bad segmentation. Problems usually arise when there is junk attached to the pollen or when pollen grains are grouped in a single blob. Note that images have been rescaled to a common resolution in order to fit the grid.

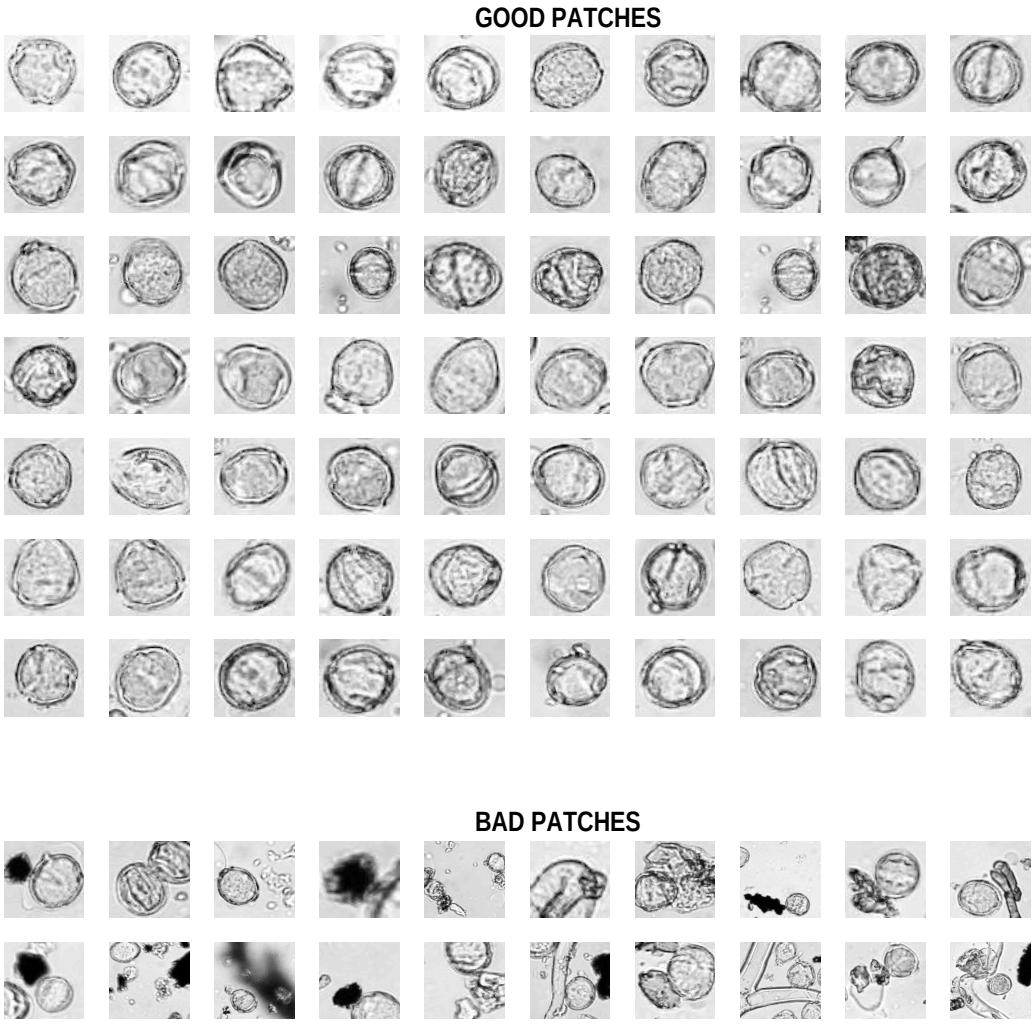

Figure 14: Example of 90 patches of jacaranda pollen extracted by detector. The first 7 rows show examples of good segmentation. The last 2 rows show samples that have been removed because of a bad segmentation. Problems usually arise when there is junk attached to the pollen or when pollen grains are grouped in a single blob. Note that images have been rescaled to a common resolution in order to fit the grid.

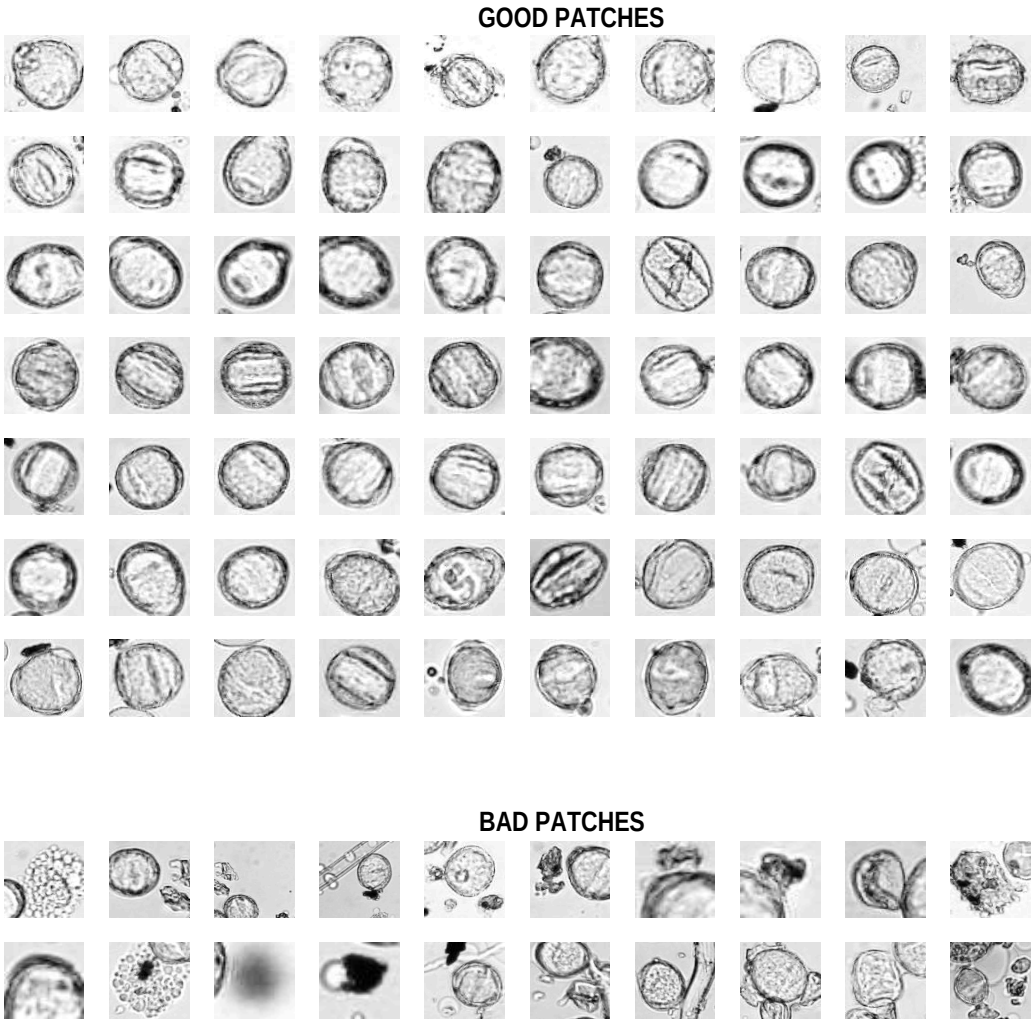

Figure 15: Example of 90 patches of oak pollen extracted by detector. The first 7 rows show examples of good segmentation. The last 2 rows show samples that have been removed because of a bad segmentation. Problems usually arise when there is junk attached to the pollen or when pollen grains are grouped in a single blob. Note that images have been rescaled to a common resolution in order to fit the grid.

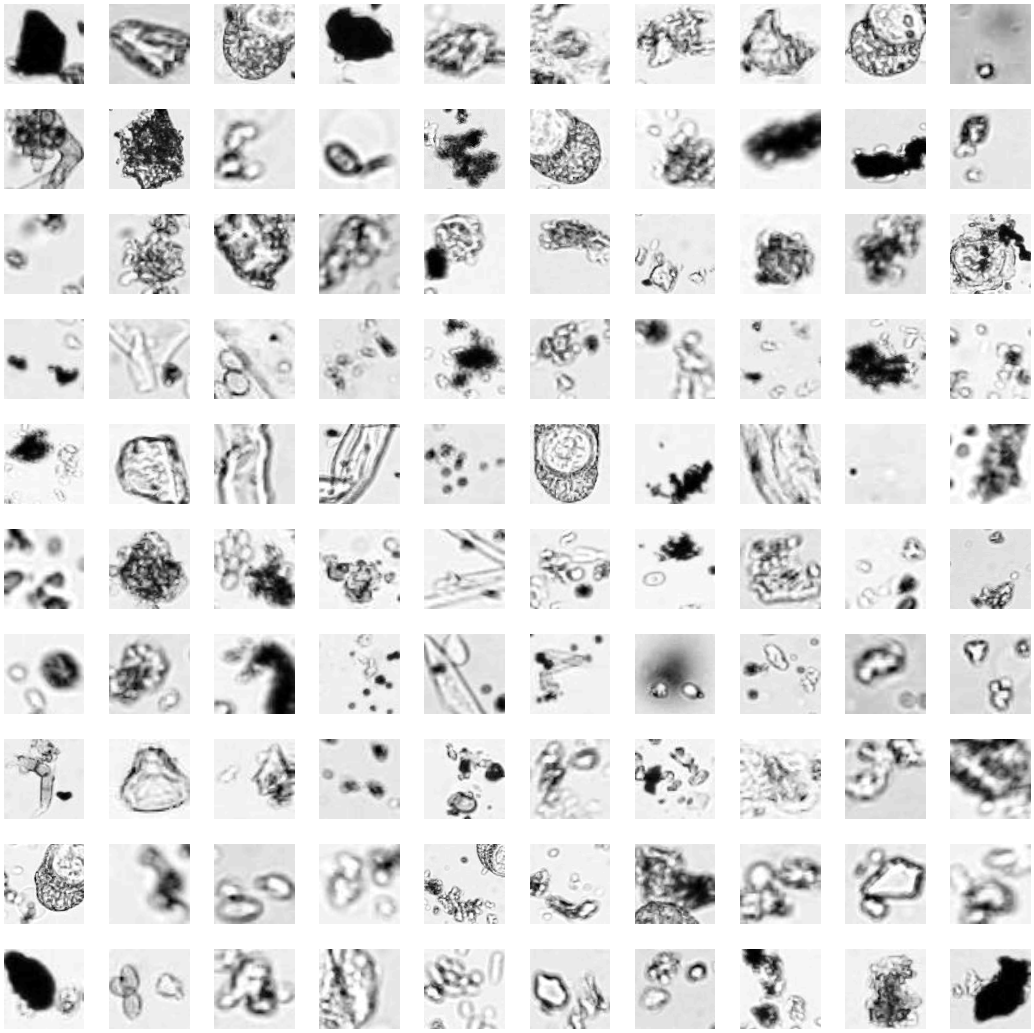

Figure 16: Example of 100 patches extracted by detector and belonging to the false alarm class.

Classification

We show the performance of classifier using MoG with feature based on local jets. The databases we are considering is the database of extracted patches derived from the expert's reference list and the database built while running the detector using DoG images. This last set of images has been checked in order to remove samples with bad segmentation.
The pollen particles we have used for classification are listed in tab.5.
We randomly extracted from each class the 10% of images to perform the tuning. Then, we run 100 experiments to assess the performance of the system. At each iteration, the 10% of images was randomly selected for the test stage.

Classification using expert's patches

We have computed the performance using the expert's patches in order to find a lower bound for the error rate of the classifier as a part of the system for real-time pollen count. Why are we getting a lower bound? First, experiments with detector patches use less samples to train the system. Secondly, the automatic segmentation of grains is not always perfect. Last but not least, we have to deal with a lot of false alarms in any real situation.

In order to tune the system, we have run many experiments varying the number of parameters: dimensions of feature space (form 6 to 18), number of components in each mixture of Gaussians (from 1 to 10 when appropriate[2]), structure of covariance matrix of each component, initial conditions for EM algorithm. The optimal parameters can be found in tab.6. Finally, we have run 100 experiments. Each time, we have randomly extracted from each class the 10% of samples for test, the rest was used to perform the training. The average result on 100 experiments is shown in fig. 17.

parameters	**value**
dimensions	14
nr. components	2
covariance	diag.

Table 6: Parameters found thanks to tuning stage which also provided us a good set of initial conditions for the random number generator used to perform EM algorithm.

[2]This means that we tried to keep the number of parameters to less than 50 because *mulberry* class has about 100 samples.

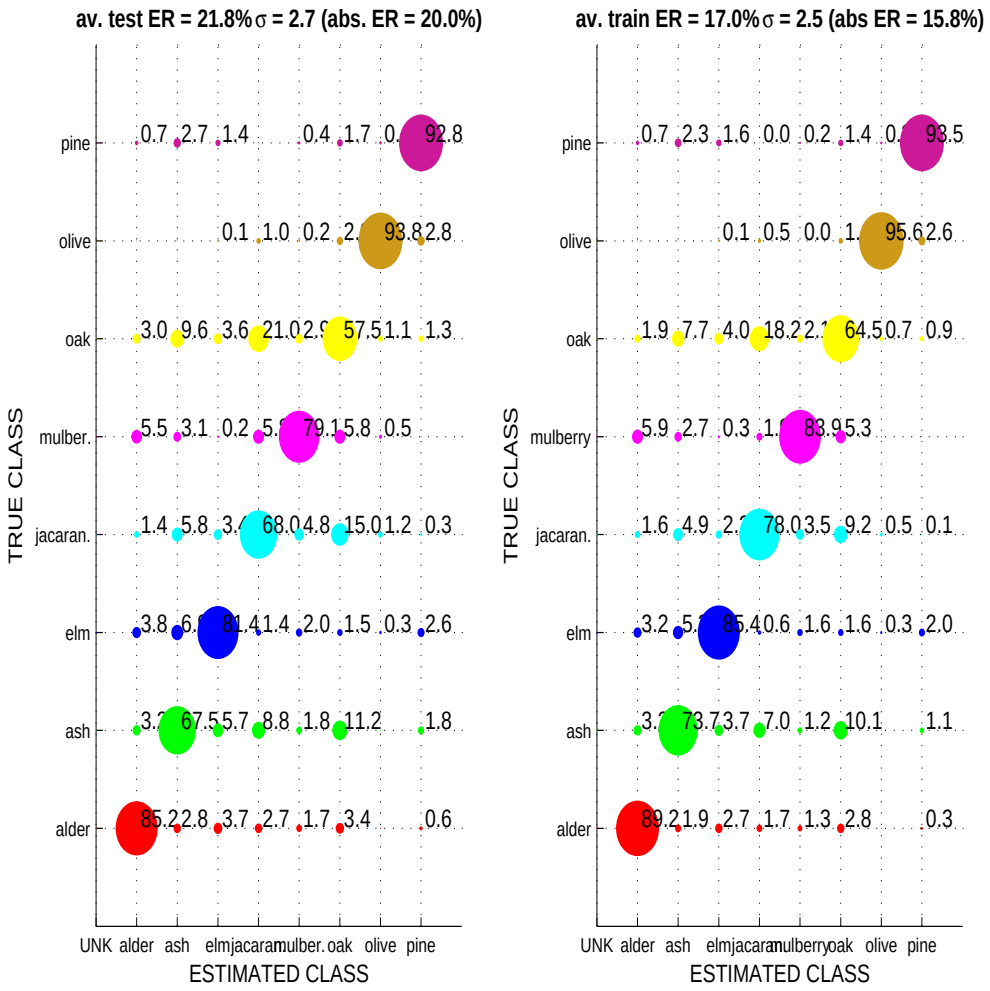

Figure 17: Average test (left side) and training (right side) confusion matrices of 100 experiments in which expert's patches were used (see second column of tab. 5). The proportional error rate is the sum of the error rates of each single class; these error rates are the ratio between the number of misclassified patches belonging to that class and the product of the number of tested classes by the total number of samples belonging to that class. The error rate shown in parenthesis is the ratio between the total number of misclassified images and the total number of tested images. The proportional error rate is useful when we run experiments using a different number of test samples for each class. In these experiments, we have randomly extracted the 10% of images for test at each iteration.

Classification using detector patches without false alarms

The tuning stage gave us the parameters that are shown in tab.7. We present the result of classification using detector patches without the false alarm class in fig. 18. This is useful to assess the quality of our automatic segmentation, even though the performance is a little worse due to the minor number of images used in training. We observe that the tuning stage provided us the same parameters that were found using the expert's patches, and that also the performance is very close to that one (see fig. 17). Thus, we can conclude that the automatic segmentation is good. Finally, given the strong similarity of pollen grains belonging to certain classes (i.e. ash, jacaranda, oak) we can be satisfied about an error rate of nearly 20% because this classification has been performed using a classifier developed for the cells in urine.

parameters	**value**
dimensions	14
nr. components	2
covariance	diag.

Table 7: Parameters found thanks to tuning stage which also provided us a good set of initial conditions for the random number generator used to perform EM algorithm.

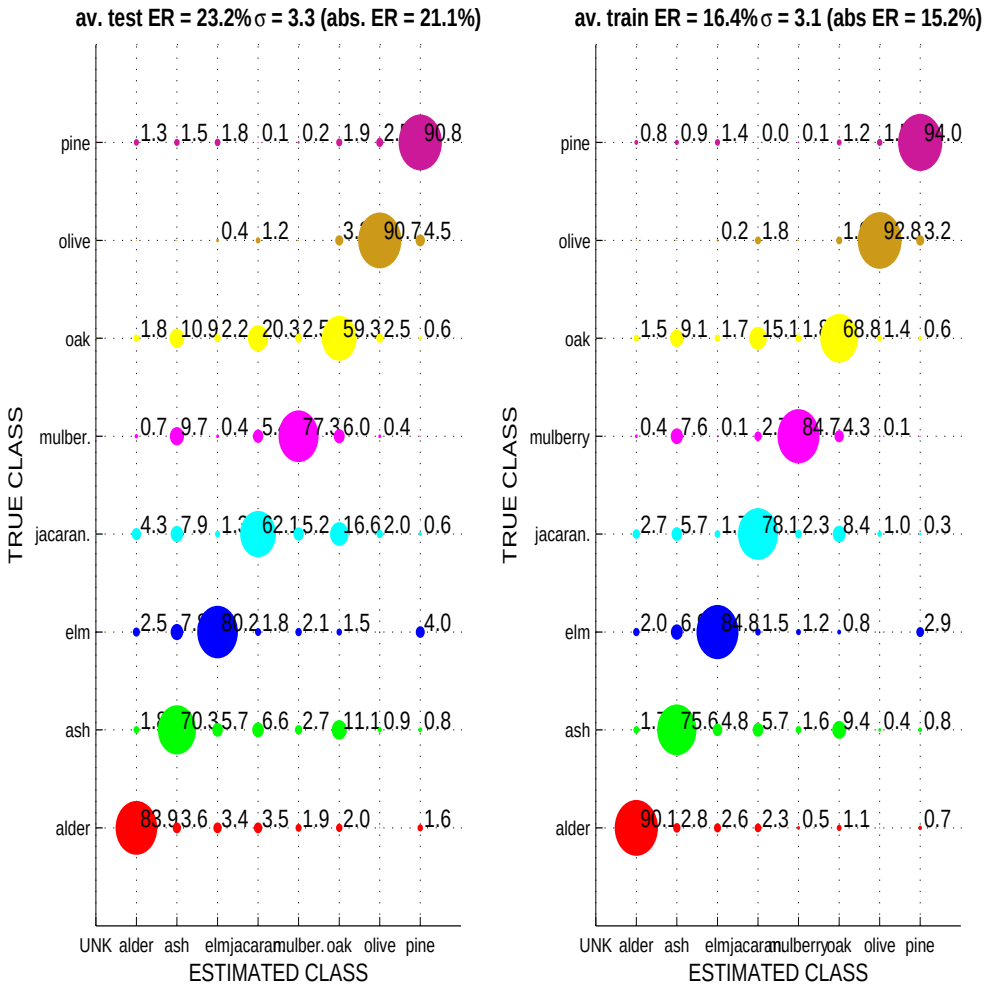

Figure 18: Average test (left side) and training (right side) confusion matrices of 100 experiments in which detector patches were used (see third column of tab.5). In these experiments, we have randomly extracted the 10% of images for test at each iteration.

Classification using detector patches and considering the false alarm class

Given the detector performance, we assume that for each found pollen there are 10 false alarms. Thus, we are using in our experiments 2576 pollen samples and 26000 samples of false alarms. Beside the pollen categories and the false alarm class, we refer also to another class: "*unknown*" category. This is the class where we put the images whose maximum posterior probability is less than a certain threshold. Of course, we assume that the system is right when he assigns a false alarm sample to either the false alarm class or the unknown one. We apply the Bayesian classifier using features based on local jets. This classifier is one of the best we have found but it needs a lot of images to estimate parameters in the MoG. For this reason, we consider only categories with more than 70 images, namely: alder, ash, elm, jacaranda, mulberry, oak, olive, pine and obviously the false alarm class.

We aim to find the performance of the system considering all the available pollen categories. It has been experimentally proved that a better performance is achieved modeling the false alarm class and not just only the pollen categories. Since we have available a lot of samples belonging to this class and since these samples show an extreme variability in appearance, we can build for this class a more complicate model. The tuning stage provided us the parameters listed in tab. 8.

In order to find an average error rate we run 100 experiments and at each iteration we randomly select the 10% of images for test. The promising performance is shown in fig. 19.

parameters	**value**
dimensions	14
components poll. classes	2
covariance poll. classes	spher
components f.a. class	4
covariance f.a. class	full
prob. thresh.	0.5

Table 8: Classification of 8 pollen species when modeling the false alarm class: parameters found thanks to tuning stage which also provided us a good set of initial conditions for the random number generator used to perform EM algorithm.

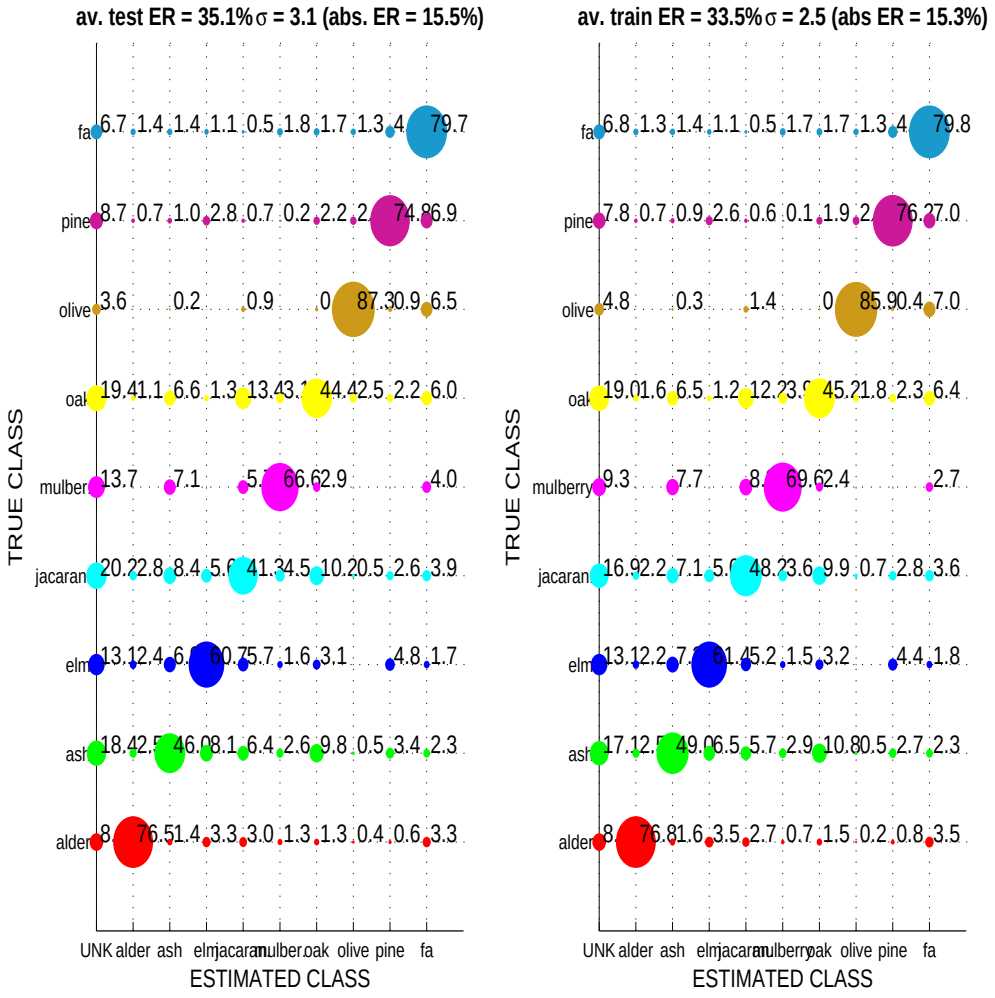

Figure 19: Classification of 8 pollen species when modeling the false alarm class: average test (left side) and training (right side) confusion matrices of 100 experiments in which good detector patches were used. In these experiments, we have randomly extracted the 10% of images for test at each iteration.

Classification during specific seasons

In any practical application of airborne pollen recognition, it is useful to know what types occur together at various times of the year in the area of interest. In Pasadena area, in September there is chinese elm and almost nothing else; in January, ash, alder, pine pollen grains are commonly found. We will never encounter a period of the year with all the 8 kinds of pollen considered in the previous section.
Then, it is interesting to see what the performance would be in a particular season.

In September, we have to distinguish between chinese elm and false alarms. Since we have 466 samples of this pollen, we take 4660 samples belonging to the false alarm class. We proceed like in the previous section modeling the false alarm class. The tuning stage provided us the parameters listed in tab. 9. Choosing a threshold value of 0.65, we achieved an error rate of about 6% as shown in fig. 20.

parameters	**value**
dimensions	8
components poll. classes	1
covariance poll. classes	diag
components f.a. class	5
covariance f.a. class	full
prob. thresh.	0.65

Table 9: September month: parameters found thanks to tuning stage which also provided us a good set of initial conditions for the random number generator used to perform EM algorithm.

In January, we have to distinguish among alder, ash, pine and obviously false alarms. Since we have 1264 samples of these pollen grains, we take 12600 samples belonging to the false alarm class. The tuning stage provided us the parameters listed in tab. 10. Choosing a threshold value of 0.7, we achieved an error rate of about 17% as shown in fig. 21.

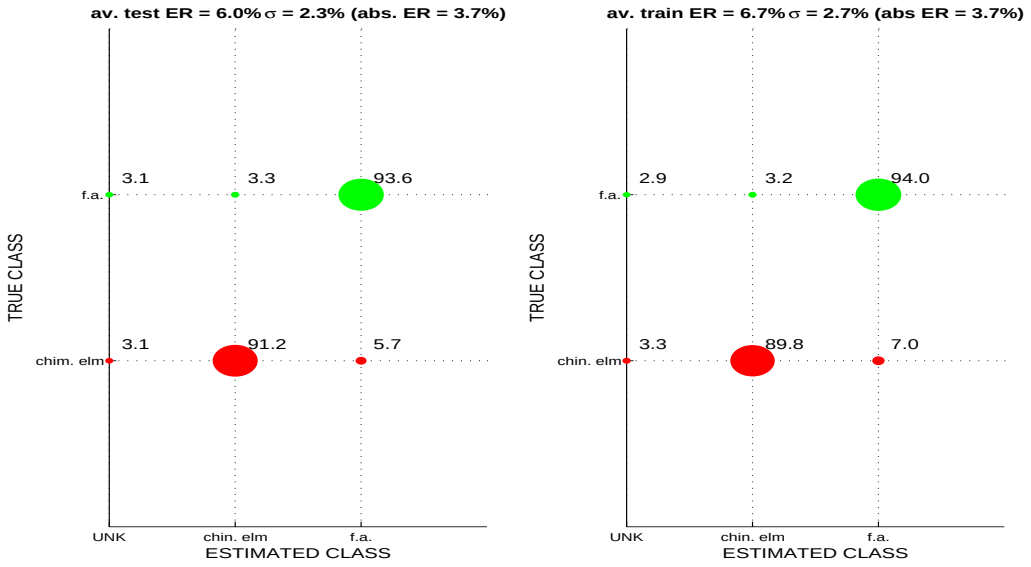

Figure 20: September month: average test (left side) and training (right side) confusion matrices of 100 experiments in which good detector patches were used. In these experiments, we have randomly extracted the 10% of images for test at each iteration.

parameters	**value**
dimensions	8
components poll. classes	1
covariance poll. classes	full
components f.a. class	10
covariance f.a. class	full
prob. thresh.	0.7

Table 10: January month: parameters found thanks to tuning stage which also provided us a good set of initial conditions for the random number generator used to perform EM algorithm.

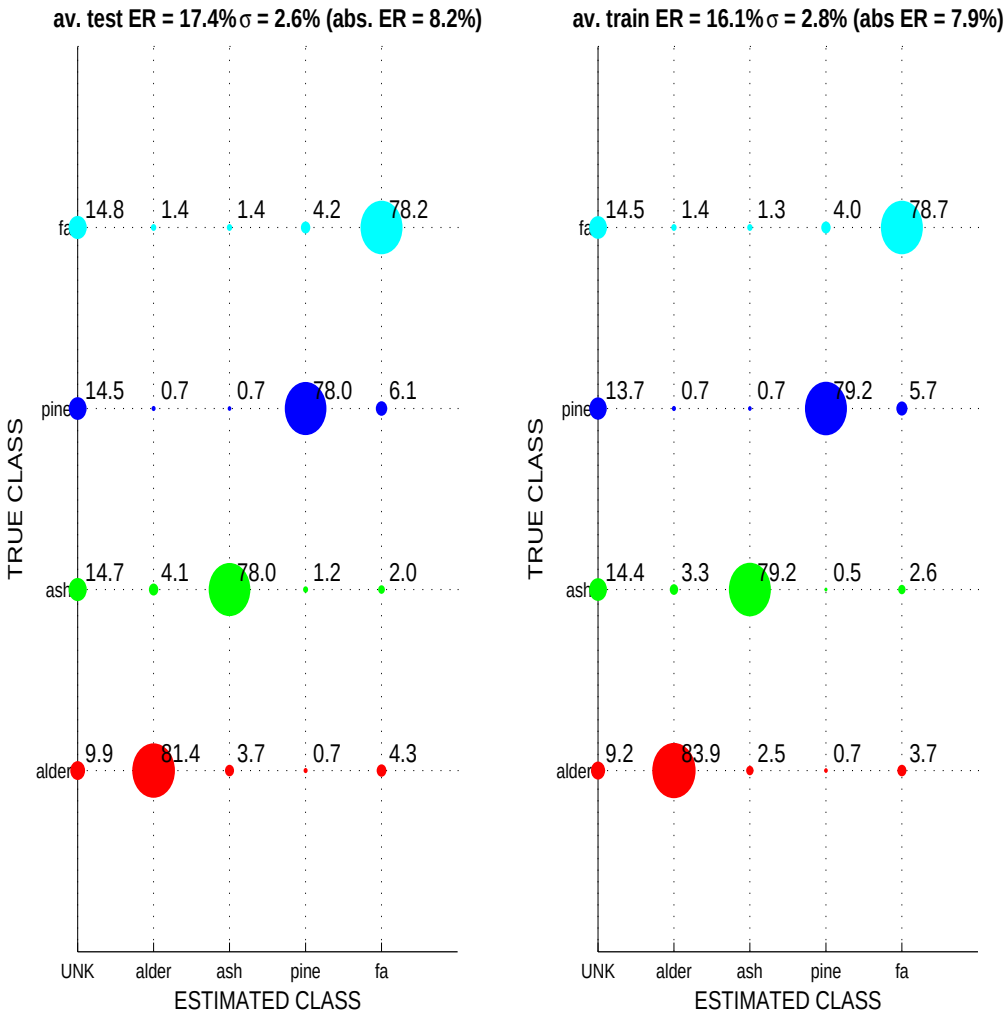

Figure 21: September month: average test (left side) and training (right side) confusion matrices of 100 experiments in which good detector patches were used. In these experiments, we have randomly extracted the 10% of images for test at each iteration.

Conclusion

The detector is able to detect with probability of 0.94 all pollen grains. It is able to determine a proper bounding box for the 83% of the pollen grains that were labeled by experts. It detects also for each pollen ten other particles that are not pollen. With the spatial resolution of our images this results are quite good.

The classifier has an error rate of about 21% on 8 pollen categories. A worse performance is achieved when false alarms are introduced; in this situation the error rate increase up to 35%. The false alarm class is described by a more complex model and we achieve on this category the best correct classification rate: nearly 90%.

The experiments are done in conditions which are very close to a real situation. Particularly, we keep the ratio between pollen and junk particles at one out of ten and the classifier receives objects in the right proportion. In this way, we overcome the difficulty to select images for training and to classify pollen particles that are not modeled because their number is too low.

This system has a better performance when considering few pollen categories. Thanks to the counts of this last years in Pasadena area, we know what kind of pollen particles can be released in each season and we can use this information to improve the performance of the system. For example, in September there is mainly a release of chinese elm and we achieve an error rate of about 6%; in January there are only three kinds of pollen and the error rate is around 17%. These promising results make the system already attractive for a practical application in airborne pollen recognition.

Future work

We should try to improve the system in the way in which we acquire the data. A proper standardized procedure to set up the spore trap machine will allow to get a database with images having a less variable background. A higher spatial resolution will give a better performance in detection, a better segmentation of the pollen grain and also a better performance in classification (no junk around the particle). In Appendix, there is a list of “good” images that are already in our database and can be used to set up the spore trap for this purpose.

Detection can be improved trying to decrease the number of false alarms with other tests. For example, we could insert a threshold on the minimum value of the average brightness inside the patch; this should remove black objects. Color information seems to carry a lot of information and it has still to be used. Pollen

grains are brown, cyan, green and red while other particles are typically black or pink (i.e. bubble of glue). The segmentation has to be improved in order to deal with elongated particles. For example, pine pollen has two blob on the sides. One of these blobs usually falls out the bounding box because the interest point is on the other blob and we are trying to fit a circle looking at 5 points around the interest point.

Classification could be improved looking for other kinds of feature that might take into account information about color and shape of the particle. A very strong effort should be done to improve the performance on the false alarm class. When the error rate on this category will be less than 5% then, the system will be ready for a practical application given the current performance of the detection system.

APPENDIX

Images that could constitute the kernel for the future database having a better spatial resolution are listed in tab. 11. The patches that were extracted by the detector using these images, are shown in fig. 22. Each image in this set gives no more than 15 false alarms with the current system and has no more than three pollen grains. These images are the 5% of the current database and they give the performance of detection shown in tab. 12.

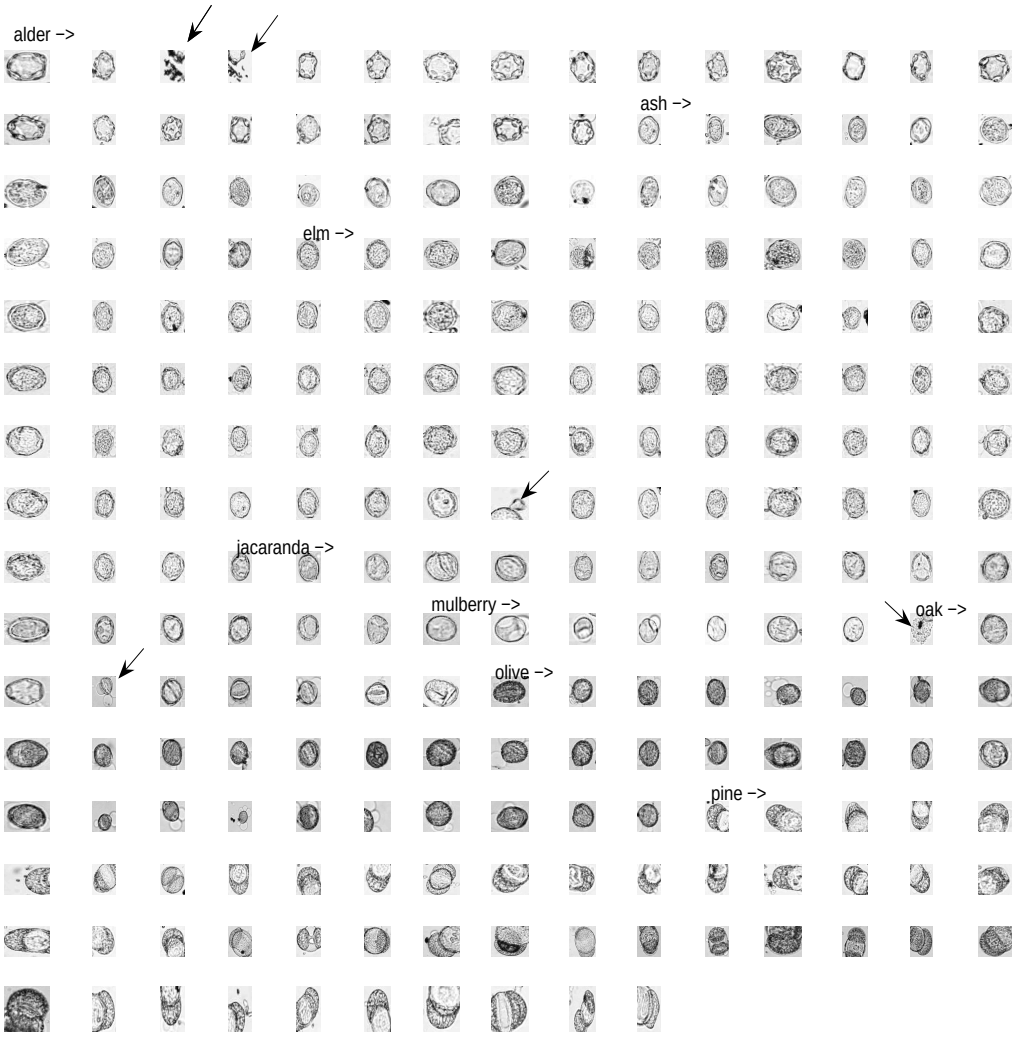

Figure 22: Patches extracted automatically by the detection system from a set of “good” images (i.e. images with a good spatial resolution). The arrows show the patches that do not have a good segmentation, they are 6.

folder	indexes of good images
*010403	(4,9,10,13,20,21,23,24,25,26, 28,29,37,38,39,54,66,67,98,106, 109,115,123,124,127,135,168,195)
022203	-
022303	(2)
022403	(3,13)
030310	-
030311	(12,30,31,32,34,67,68)
*032102	(69,89,90,91,101, 104,114)
*042304_0-12	(5,18,20,21,22, 24,33,46,49,55, 66,70)
*042802	(12,22,26,27,48,57,58,59,60,61, 63.64.65.66.80,81,86,87,89,90, 92,93,95,96,97,98,114,115,117,118, 119,121,122,123,124)
091903	(13,18,22,23)
120102	(1,7,20,24,28,30,35,38,40,41, 42,43)
120202	-
*122902	(1,6,10,13,17,18,19,20,21,22, 151,152,157)
*ChinElm	(17,18,19,30,32,33,34,35,43,48, 59,62,63,66,71,76,80,84,86,87, 91,94,98,99,100,102,103,104,106,114, 115,126,139,156,160,161,162,169,178,184, 185,186,199,202,210)
TrialPollen	(19,22)
TOTAL	168

Table 11: Indexes of “good” images in each folder of the current database.

genus	detected	detected good
alder	24	22
ash	25	25
chin. elm	74	73
jacaranda	18	18
mulberry	7	7
oak	9	7
olive	33	32
pine	45	45
TOTAL	*236*	*230*

Table 12: Detection on "good images": the last columns refers to images that have been well segmented. The percentage of detection is 100%. If we remove the 6 patches that have a bad segmentation, the percentage of detection is 98%. The percentage of false alarms is 565%.