Institut für Informatik
Lehr- und Forschungseinheit
für Datenbanksysteme

Ludwig-Maximilians-Universität München

Diplomarbeit

Similarity search for classifying of circuit boards

Harald Kraft

Aufgabensteller:	PD Dr. Peer Kröger
Betreuer:	PD Dr. Matthias Renz
Abgabedatum:	31. Oktober 2011

Erklärung

Hiermit versichere ich, dass ich diese Diplomarbeit selbstständig verfasst und keine anderen als die angegebenen Quellen und Hilfsmittel verwendet habe.

München, den 31. Oktober 2011

....

Harald Kraft

Abstract

While storing recognized objects of a scene in a database is just a pre-step in the course of this thesis, the main task is to compare scenes and find similarities. The relevant scene has a very specific setting: it will consist of an unpopulated circuit board. Recognized objects are soldering points, which bare features like position, width and height.

Using a sweepline algorithm for two sets of database entries, i.e. soldering points of two circuit boards, it will be determined whether all soldering points exist and are correct or whether manufacturing was defect. As second result it is possible to determine the type of circuit board.

Zusammenfassung

Erkannte Objekte einer Bildszene in einer Datenbank zu speichern stellt nur einen Vorverarbeitungsschritt dieser Diplomarbeit dar, die Hauptaufgabe ist es, Szenen zu vergleichen und Ähnlichkeiten zu finden. Die relevante Szene enthält ganz spezifische Eigenschaften: sie besteht aus einer unbestückten Leiterplatte. Die erkannten Objekte dabei sind Lötpastenpads, die Features wie beispielsweise Position, Breite und Länge erzeugen.

Durch Anwendung eines Sweepline Algorithmus auf zwei Datensets von Einträgen der Datenbank, sprich von Lötpastenstellen zweier Leiterplatten, soll bestimmt werden, ob alle Lötstellen vorhanden und korrekt sind oder ob die Herstellung fehlerhaft war. Als zweites Ergebnis ist es möglich den Typ der Leiterkarte festzustellen.

Contents

Chapter 1

Introduction

When circuit boards are equipped with electronic components during manufacturing a special technique of soldering is used by the manufacturing industry. The circuit boards show a number of soldering points (depending on and defining the type of board) on which later the electronic components can be placed (and soldered).

The manufacturing industry has an interest in detecting faults during the manufacture process. It is important to have a verification whether a circuit board can be used in the step of equipment and process inspection for the soldering points. Knowing the type of board enables the companies to gather statistical information about their production. The utilization of scanners by Opdi-Tex GmbH would allow such an inspection during manufacturing but would call for additional algorithm and software approaches. Being able to classify the type of board has the advantage of the scanner being able to be used during manufacture without resetting if different types of boards are used at the same time.

In order to automatically determine the completeness and correctness of all soldering points along with the type of circuit board a comparison between a golden sample and circuit board at hand must be made. This thesis will deal with similarity features and a way of performing such a comparison. An algorithm will use a number of features that describe each single soldering joint. In order to have data about all soldering points available, a scanner will capture image data of circuit boards, perform object recognition on the points and enter their features into a database. Using the data from the database, an offline process can start to do a similarity analysis of two sets of data. The underlying algorithm for determining similarity will be based on the sweepline approach. Using that approach a sweepline will - considering one feature - progressively eat the data of each set and define a degree of similarity.

This degree of similarity will be the subject of research in this thesis. The similarity will be described by comparing features of objects and the attempt to match similar objects in pairs. A number of different matching criteria were chosen and assessed. These matching criteria will be evaluated with a number of items during an algorithm training and compared in ways of quality. The criteria with the best performance will then be verified with a set of test samples.

The result of the process can therefore identify manufacturing faults, e.g. when soldering points are missing or defaced, as well as identify the class of circuit board depending on a list of known golden samples.

Chapter 2

Preliminaries

This chapter will deal with terminology and describe the items and hardware which were of help (and were the basis) of the thesis.

2.1 Circuit boards

Circuit boards, which are also called *printed circuit boards* (abbreviated *PCB*), are one of the most fundamental parts in almost any electronic device. They consist of several layers of non-conducting material as well as paths of conducting material which connect electronic contacts that appear on the surface. Most commonly they are known as green panels with silvery pads (made out of tin) on its surface like the one visible in illustration 2.1.

However, when PCBs are used in electronic devices they will be equipped e.g. with resistances, transformer and other electronic components. There are a few techniques for assembling circuit boards where solder paste is placed onto the pads on the surface (as pictured in image 2.2[1]) to which components can then be attached.

This thesis deals with the quality of the manufacture step of applying solder paste. It will try to determine whether the application was done properly or poorly as well as classify the type of circuit board according to a number of known types.

2.2 Scanner setup

During the manufacture process, circuit boards will pass along an assembly line. When the application of the solder paste has been performed, it can pass

[1]`http://www.technolab.de/_en/solderdict/furtherexamples/faultyapplicationofsolderpaste.php`, accessed October 2011

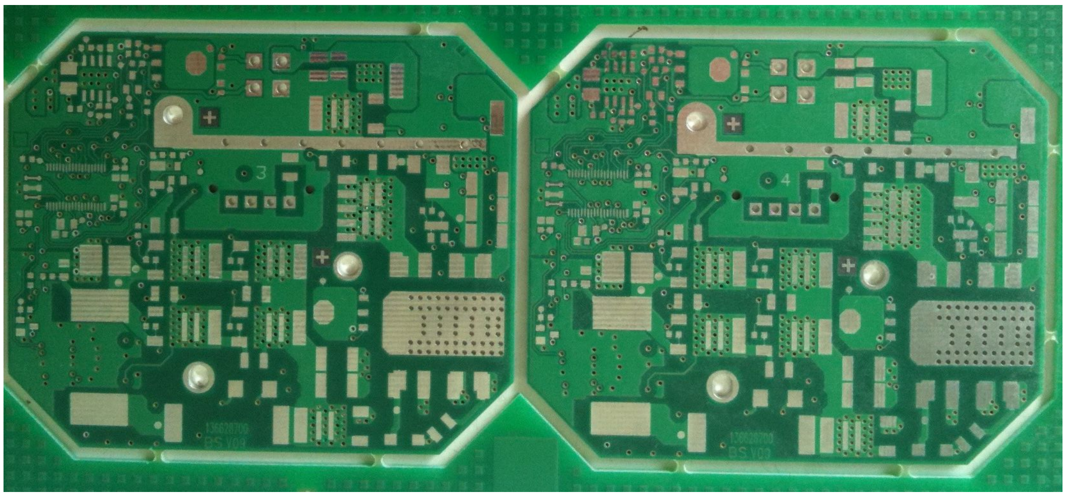


Figure 2.1: Image of a PCB with tin pads

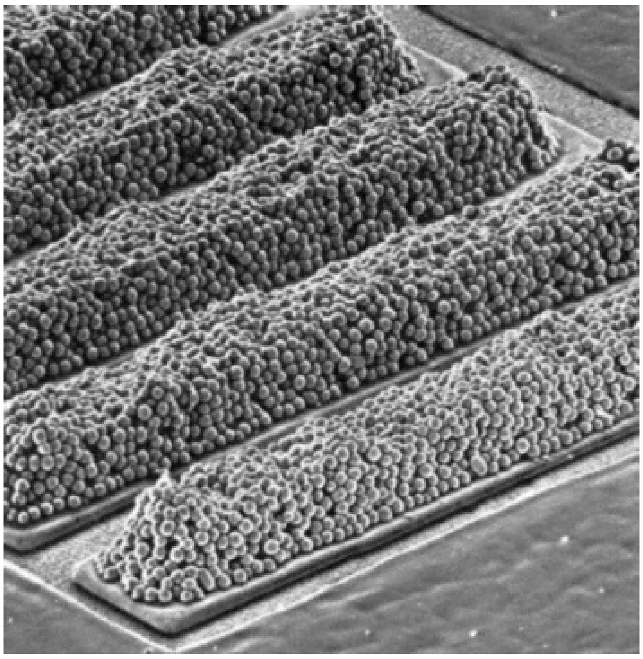


Figure 2.2: Solder paste applied to pads on the PCB surface

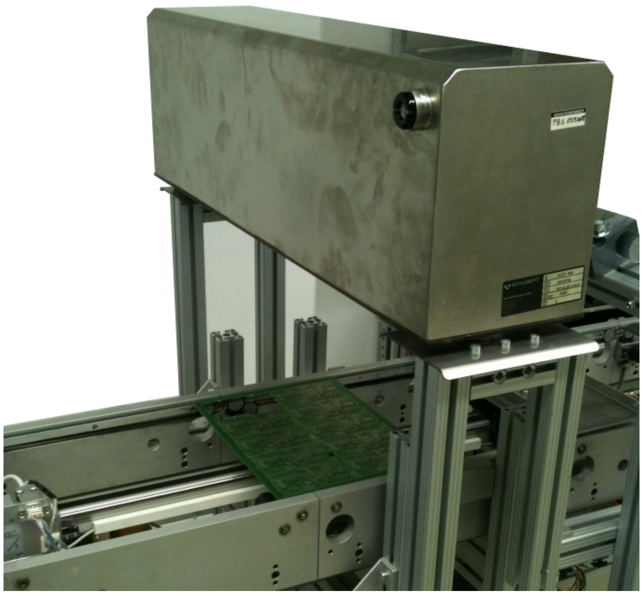


Figure 2.3: A scanner used within an assembly line

a scanner - while in movement on a conveyor belt - and can be inspected for correct application.

The scanner needs to be able to capture image data of moving items, which then can be used for object recognition. Such scanners are line-based, i.e. the image produced is rather a stream of image lines that can be put together to form a continuous picture. Furthermore, object recognition is necessary because the similarity search is based on comparing features of objects.

For this purpose industrial scanners produced by Opdi-Tex GmbH[2] were used, because they are able to do both capture image data of moving items as well as perform object recognition (see illustrations 2.3 and 2.4).

2.3 Gather image data of circuit boards

In order to train and test the algorithm to be developed, a series of test images are necessary. While not focusing on real world data during the beginning of the implementation, a very simple image was created manually (simulating a PCB with three solder pads, see figure 2.5) in order to verify the correct performance of all pieces of software (which are mentioned in sections 6.1.2 and 6.1.3).

[2]`http://www.opdi-tex.de/`

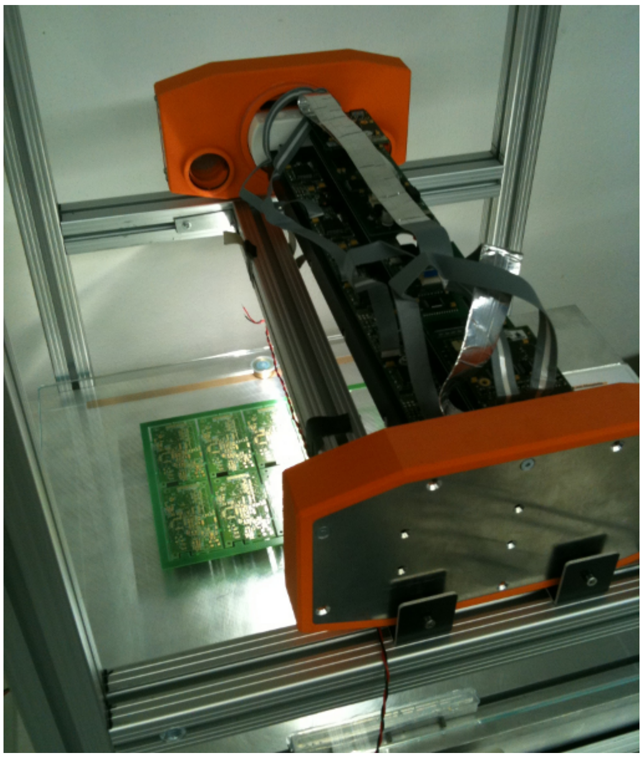


Figure 2.4: Scanner produced by Opdi-Tex GmbH for capturing image data of moving items

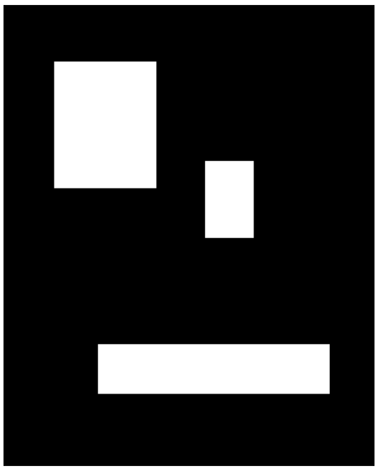


Figure 2.5: Simple image simulating a circuit board with three solder pads

To extend training and testing to real world data, two different ways of gathering image data were used:

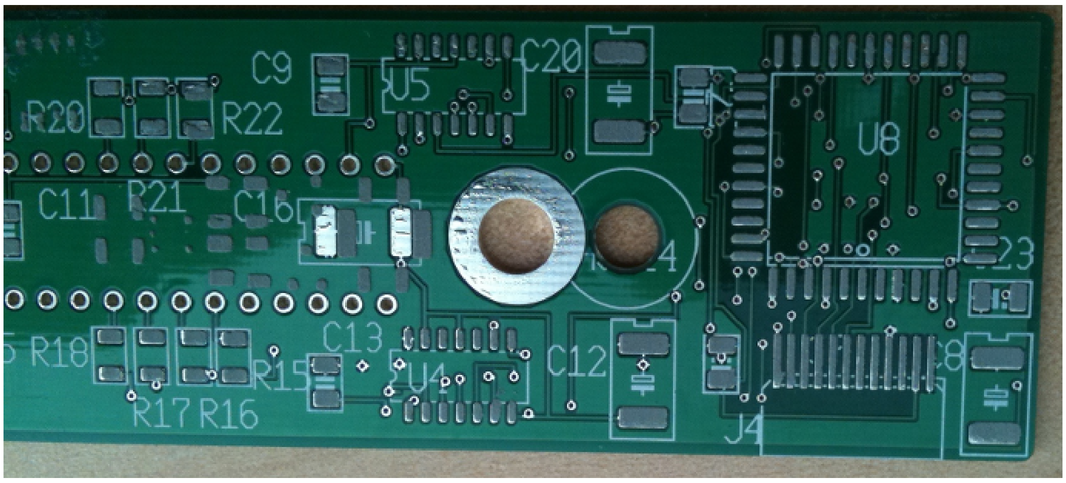
1. taking a picture of the circuit board with a common digital camera

2. capturing images with a scanner that can also be used in an assembly line

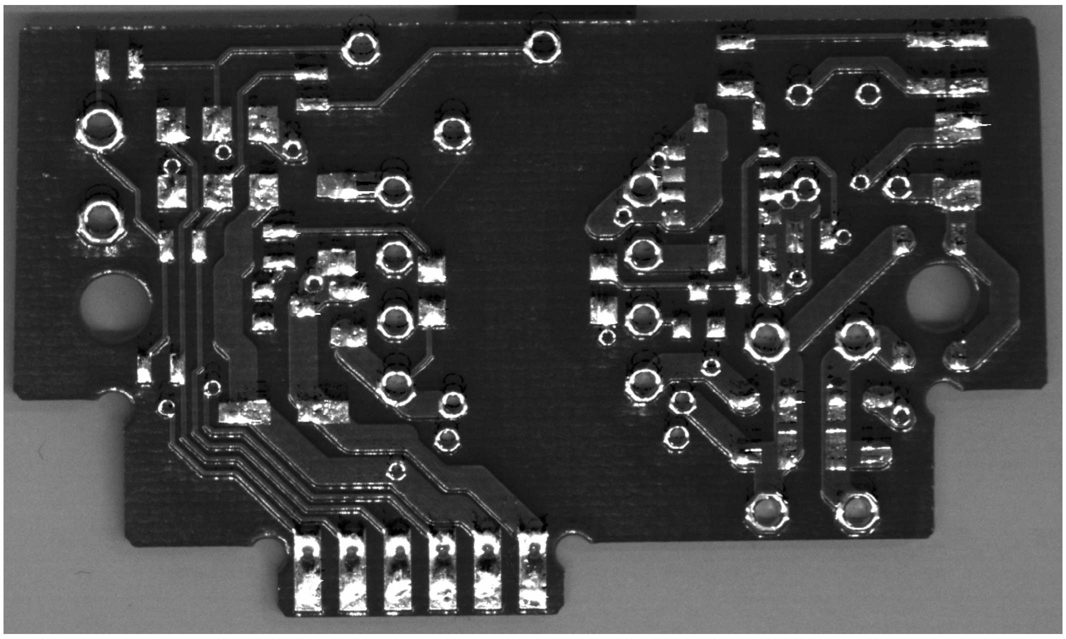
Apart from optical features, the main difference of these two ways of creating test images is, that during manufacturing a still image is not possible. Optical features include: distortions can occur if objects do not move one hundred per cent at a constant velocity as well as if circuit boards are not aligned perpendicular.

For this thesis no tests were performed during a live manufacturing process, therefore image data of circuit boards needed to be captured offline so it could be replayed at a later point in time. Opdi-Tex GmbH uses a simulation software component, which is able to use image files as source data and can simulate a real scanner.

A few different kind of circuit boards were available and could be used. Two examples can be seen in the illustration 2.6.



(a)



(b)

Figure 2.6: PCBs used in training and testing

Chapter 3

Relevant features

In order to be able to compare objects, a number of features need to be defined which will describe an object.

This thesis uses the term "object" to describe a smaller or larger part of the image data separating itself from the surrounding image data and its set of features that an object recognition software creates about it.

While performing object recognition, Opdi-Tex's software defines a whole list of features for each object. In a visualizing software component, recognized objects are marked (as seen in image 3.1) and the most significant features can be viewed (screenshot 3.2). For this thesis a selection of features from that list was made, which would be suited in the process of similarity search.

The scanner software determines all features that involve a measurement of some sort of length in number of pixels. To go in compliance with the software's axis orientation: the X axis ranges in the image from the outermost left pixel to the outermost right pixel, whereas the Y axis progresses with time and increments with each new image line that is captured (*image line* refers to the concept described in section 2.2). The specific hardware setup of the scanner defines both the amount of pixels of an image on the X axis as well as on the Y axis: the full range of the X axis depends on the width of the scanner hardware, the Y axis depends on the frequency with which new image lines are captured. A sample rate of 1kHz means that a picture stream with the height of 1000 pixels is captured each second. As an additional note, the Y coordinate contains an absolute value originating at the start of execution of the scanner software and will increment infinitely until either the scanner is switched off, the software crashes or the datatype runneth over.

The following features are the most important and prominent ones describing the uniqueness of an object.

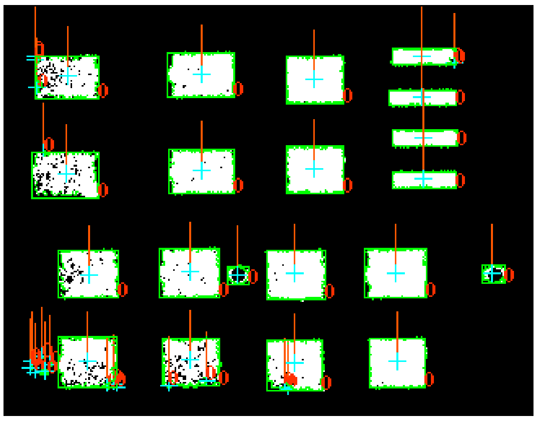
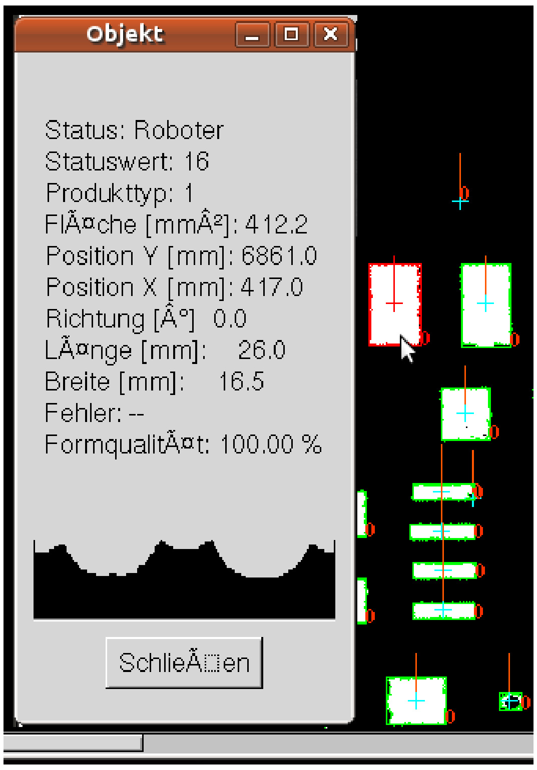


Figure 3.1: Defined objects



Status: Roboter
Statuswert: 16
Produkttyp: 1
FlÃ¤che [mmÂ²]: 412.2
Position Y [mm]: 6861.0
Position X [mm]: 417.0
Richtung [Â°] 0.0
LÃ¤nge [mm]: 26.0
Breite [mm]: 16.5
Fehler: --
FormqualitÃ¤t: 100.00 %

Figure 3.2: A list of features

Center of gravity The most obvious feature of an object as far as location is concerned is the *center of gravity*. It describes an object's location (within an image stream) in matters of X and Y coordinates - as mentioned measured in pixels. For example, an object's center of gravity (x=500, y=667788) appears 500 pixels from the left border of the scanner in the 667788th image line the scanner software captured.

Dimensions Next to the location of the object, its dimensions are prominent features. Both *length* and *width* are determined by the scanner software and are independent from the object's orientation. If an object is not a perfect rectangle, length and width are those of the minimal bounding rectangle.
An object's *area* is the actual number of pixels it comprises of (and can therefore be a smaller number than multiplying length and width if the object is not a rectangle).

Bounding box Because objects can occur with a rotation, i.e. the minimal bounding rectangle not having edges parallel to the axes, the *bounding box* describes the maximal expansion of an object in all directions.

Rotation If an object's minimal bounding rectangle is not parallel to the axes, a rotation occurs and is described as an angle. However, this feature was not used in the course of this thesis.

Chapter 4

Defacing origins

In the course of checking solder pads for similarity, one has to be aware of the different kinds of deformation which can occur and cause an object to fail in similarity. A difference in one or several features of solder pads, which were mentioned in chapter 3, can occur for various reasons. The different reasons can origin from a fault during the application process of the solder paste or they can origin from scanner issues and its way of capturing images.

4.1 Manufacturing faults

The following faults have their origins in incorrect application of the solder paste.

4.1.1 Solder paste on pad is missing

The most obvious reason for a non-matching pad is if no solder paste was applied and therefore the object is actually missing. This can happen for example if the mask - with which the solder paste is applied - is clogged at some holes.

4.1.2 Two soldering points merged together or touching

Especially when solder pads are located in close proximity, it happens that solder paste of two solder pads merges into the other and/or touches at certain places. For example if too much solder paste was used or if the mask is dirty at the appropriate places, such a fault can appear.

4.1.3 Smeared soldering joints

Solder paste can be smeared at its actual location and exceed the borders of the solder pad resulting in a differently shaped object. When the mask is not totally clean or is not removed carefully enough, faults like these can occur.

4.1.4 Aligned at slightly different coordinates

An offset for a number of solder pads can occur when the master plate (i.e. the mask) is not placed at the exact position or is moved during application or when being removed.

4.2 Errors originating from scanner or conveyor belt

When the PCB moves on a conveyor belt, various incidents can lead to differences in relevant features.

4.2.1 Circuit board is rotated by a certain angle

Even though a very good alignment of the PCB can be taken for granted, a perfect alignment is usually not possible. This is caused by the fact to allow the PCB to actually move on the belt without the risk of it being stuck. However, the difference in rotated alignment will occur only to a very small extent.

4.2.2 Uneven/different belt velocity during PCB capture

While a circuit board passes the scanner (and image data is captured), variations in speed of the belt can occur. This might happen if the conveyor belt is stopped (for example for safety reasons) or has to be sped up.

This will result in objects appearing smaller (faster belt) or larger (slower belt) in the Y dimension. Furthermore, if the PCB is not in perfect alignment with the axes, a distortion of the object in X dimension will appear.

Chapter 5

Objects considered for classification

As described in section 2.1, during manufacture tin pads on the surface of a PCB are covered with solder paste (so that electronic components can be soldered onto the PCB). The basic procedure of classifying a circuit board is to compare it to a golden sample, which is a circuit board where solder paste was perfectly applied on all solder pads. The idea then is to perform similarity checks on such pads (those of the golden sample to those of the board to be inspected), while the pads will appear as objects (recognized by Opdi-Tex's scanner software). When a PCB passes along the scanner, the software is able to detect different kinds of result pads (see image 5.1 for a visual example):

- A solder pad covered with gray solder paste (correctly **A** or incorrectly **B**)
- A tin pad, where solder paste is missing **C**
- A tin pad, which has intentionally not been covered with solder paste **D**

There are several possible approaches concerning inspection of objects, which are suitable to be used for similarity matching.

5.1 Consider only pads covered with solder paste

Only pads covered with solder paste (i.e. types A and B) are taken into account.

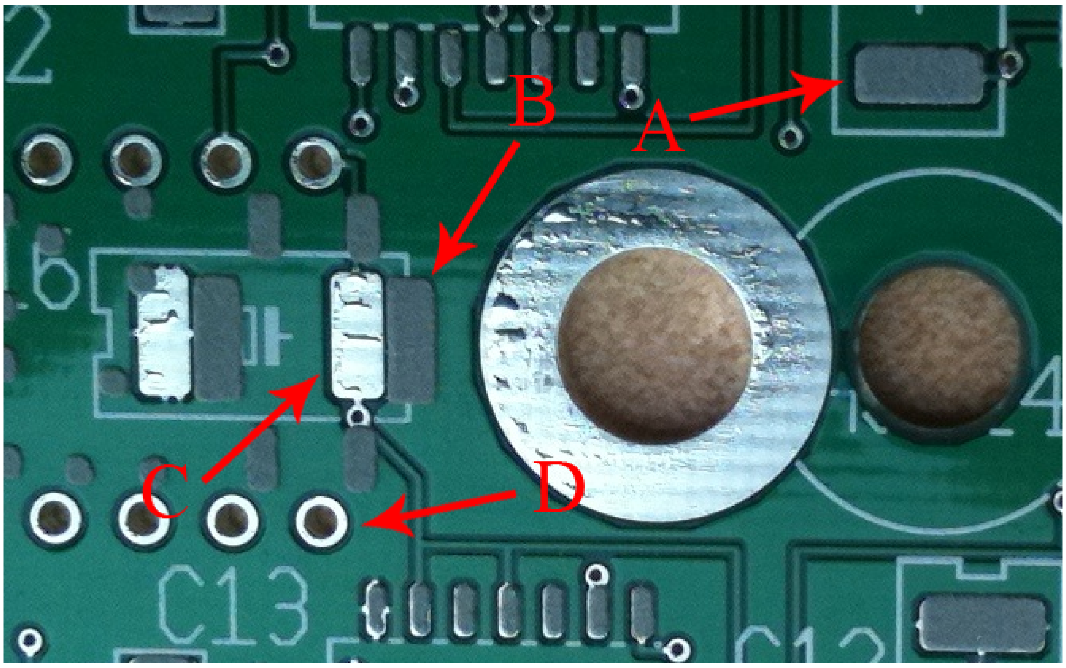


Figure 5.1: Different sorts of objects that can be detected by the scanner software

5.1.1 Extract only objects resulting from gray solder paste

With this approach the scanner software will detect objects (and describe the features) only resulting from points on the PCB that have been covered with solder paste. The software will detect all such points regardless of whether the solder paste was applied at the right location or dislocated. Pads that remain silver from tin (intentionally or by faulty application of solder paste) are disregarded. To make it clear: If solder paste has been dislocated and left a tin pad uncovered, the gray solder paste is recognized as an object whereas the tin pad is not. This means, that from a golden sample the number of extracted objects matches the number of pads that need to be covered with solder paste. Considering the faults that can occur during application of solder paste (as discussed in section 4.1), a circuit board that has not been manufactured correctly will have an equal number or less objects. Speaking in terms of similarity match: a perfectly produced PCB will match all objects from the golden sample.

5.1.2 Extract only objects from tin pads

This approach will consider all mentioned types of resulting pads as objects and store their features. In order to then retrieve all objects that had been detected from gray covered pads, an additional pre-step and dataset is necessary: Objects from all tin pads of a PCB need to be detected *before* solder paste is applied. By then subtracting objects from just tin pads in a PCB (that has passed the step of applying solder paste) leaves the dataset of those objects (from the pre-step PCB) which will later be covered with solder paste and can be compared to the same result of the golden sample.

However, the method of subtracting a dataset from another adds a certain degree of complexity to the issue at hand.

5.2 Consider only tin pads

As opposed to considering only gray pads in order to perform similarity matching of a golden sample to a circuit board, one could also just consider tin pads.

If only tin pads are detected as objects (and gray covered pads disregarded), the golden sample will produce only such objects resulting from tin pads which are not covered with solder paste intentionally. In other words, if there are no pads for a certain type of PCB that remain uncovered on purpose, the golden sample would produce zero objects - and in return any inspected circuit board that produces one or more objects (from uncovered or partially covered pads) would render faulty. Another effect - for obvious reasons - is that it would be quite hard or even impossible to determine the class of such a PCB if no objects are produced.

5.3 Chosen approach

The approach chosen to perform similarity checks is the one which extracts objects only from pads covered with gray solder paste (described in section 5.1.1). The choice was met for a number of reasons. The most important one was that without objects, a type classification cannot be made. Another reason was not to introduce unnecessary complexity to the problem.

This means for one, that all the following chapters will deal with objects that describe a point in the PCB covered with solder paste, and for another that in order to be a perfectly manufactured PCB its objects need to match all objects from the golden sample.

Chapter 6

A Framework for object matching

This chapter deals with the actual realization, implementation, training and testing of similarity search.

After explaining which steps had to be done before using the matching software, the performance of the sweepline algorithm will be explained. Based on the sweepline approach, training with a number of different matching criteria could be done to evaluate the quality of the criteria. When the choice of criteria had been completed, tests with new data (different from training items) were run.

6.1 Preliminary steps

Before the actual process of similarity matching could begin, a few one-time steps had to be undertaken.

6.1.1 Database

Using a database for storing and exchanging object data (i.e. relevant features) was chosen for two reasons: On the one hand a database is a very efficient and simple way to exchange large amounts of data between processes. On the other hand by using a database it was possible to run the different steps time-independent from one another: First, run the scanner software which will gather object data and store it in a database, second, run the similarity checks using offline data retrieved from the database.
The database schema was developed to contain all relevant features of the objects with suitable data types. The layout is displayed in appendix A.

6.1.2 Database filling software

Opdi-Tex's scanner software will create and use object features for object recognition, but it will discard the values after a certain amount of time. A short piece of software was written, which would use all object features as input and store the relevant features in the database.
An important goal of this small program was to be kept simple and reliable, so that the similarity check algorithm could rely on the data it was fed. Because of the nature of the already existing scanner software, this piece of software was written in C and named *minidaemon*.
Before classification (and matching) can begin for a specified circuit board, the database needs to be filled with objects retrieved from golden samples of all types of PCBs that need to be classified.

6.1.3 Visualization

In order to be able to visualize the results which the algorithm would produce, a small web interface was written in PHP. The interface could display the input data used by the scanner (i.e. show an image) and the output produced by the algorithm (i.e. colored markings of matching and not matching objects).
The interface offers a way to select the PCB to be analyzed. In this thesis, the PCBs that can be selected are basically the training items categorized in PCB class (i.e. which type of PCB) and fault class (see chapter 4). Section 6.3 will deal with more information on the different training items.
When the user selects a certain PCB he wishes to classify and hits *submit*, the sweepline algorithm (described in the following, section 6.2) will perform the matching and classification steps. The web interface will show the user the name of the golden sample that matched best along with its image as well as mark all objects in the picture that did not have an appropriate match. Objects from the golden sample are marked in a different color than objects from the circuit board which is analyzed. A screenshot of the interface can be seen in image 6.1

6.2 Sweepline algorithm

The algorithm for classifying of circuit boards and for determining which soldering pads have been incorrectly manufactured uses a sweepline approach (just like described in [SX08] and [DBCvK08]). The sweepline will move over a set of objects retrieved from a PCB's golden sample and try to match it to a corresponding set of objects retrieved from the PCB to be analyzed.

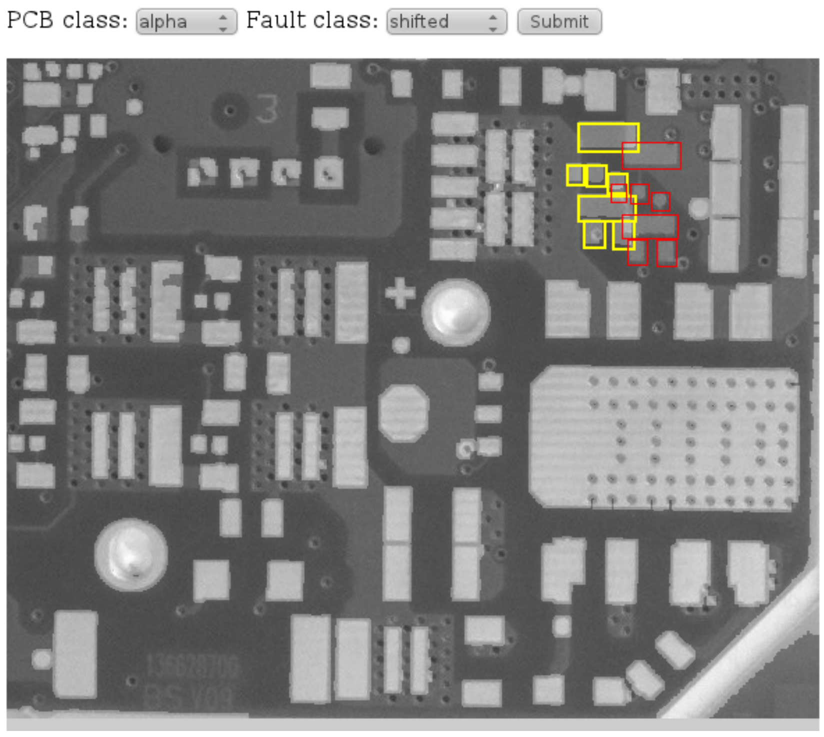


Figure 6.1: Marking of unmatched objects in simple web interface

6.2.1 Procedure

While the software is at work, the following steps will be run:

1. Retrieve object set of golden samples and object set of current PCB
2. Run sweepline over object sets and apply a matching criteria for each object
3. Evaluate overall similarity for each golden sample
4. Run sweepline again with best matching golden sample

The sweepline algorithm as well as the visualization interface were written in PHP, because PHP offers a number of functions for efficient database access and easy usage of result arrays. The software queries the database for all available golden samples and performs the matching steps for each of them. All objects of the current golden sample are retrieved from the database as well as all objects from the PCB to be classified. The two result sets are combined and sorted by the starting Y coordinate of the bounding box, so the sweepline can progressively handle objects by their bounding box order.

As the algorithm advances the sweepline with ascending Y coordinate it collects objects from the combined result set and tries to match golden sample objects with PCB objects. One of the matching criteria which were used in this step was if the center of gravity of both objects did not differ by more than 5 pixels in each axis (see section 6.3 for the different criteria). If two objects matched they were removed from the sweep line status set. an object was also removed if the sweepline advanced past its bounding box (and the object remained unmatched).

After the sweepline reached the end of all objects, the remaining (unmatched) objects were combined with those that had not had any match during the run. A score for classification is determined by dividing the number of unmatched objects by the total number of objects (both numbers refer only to objects from both golden sample). The smaller the score the more similar the object sets were (according to a certain matching criteria). Details and evaluation of this score will be discussed in sections 6.3 and 6.4.

For visualization purposes, after the sweepline matching was completed for all available golden samples, the algorithm is run again for the best matching golden sample in order to view all unmatched objects in the interface.

6.3 Training

In order to train the algorithm and evaluate the results of the training, a number of different circuit boards where used as training items. These items therefore describe the different classes of PCB that are to be classified. Furthermore, each PCB class consisted of different versions of a circuit board, each of which showing one of the discussed faults. As described in section 6.2.1, a score is used for classification of boards: dividing the number of unmatched items by the total number of objects to be matched. It meant a circuit board performed better in classification, if its golden sample could match more objects than any other. However, as soon as two objects matched they were removed from the sweepline set and therefore made the matching absolute. If the matching criteria was loosened (less strict), logic dictates that more matches will occur - no matter whether they are correct. There is no method during the actual run in production (after training and testing) to determine with unknown data to, whether a matching was correct or not. So in order for the score to be a good classificator, during training it was important to maximize the quality of the matching by applying different matching criteria and compare the results.

6.3.1 One-time pre-steps

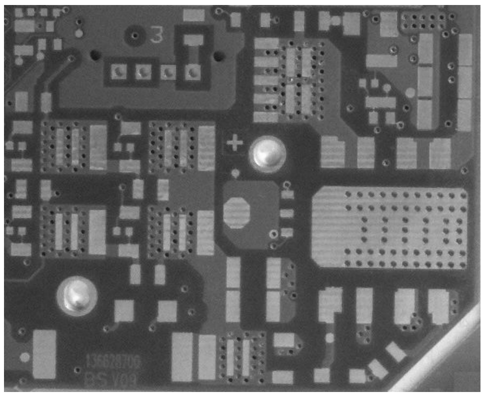
In the beginning of the tests, object data from all training circuit boards (i.e. all golden samples and all faulty boards) had to be collected an stored in the database. Therefore the Opdi-Tex software (including the mentioned add-on *minidaemon*) was run with each of the circuit boards and data was gathered. Even though the software produces continuously object data, only one dataset was kept in the database for each PCB (because of the nature of the software, the object sets would be exactly 100% the same).

6.3.2 Training itemset

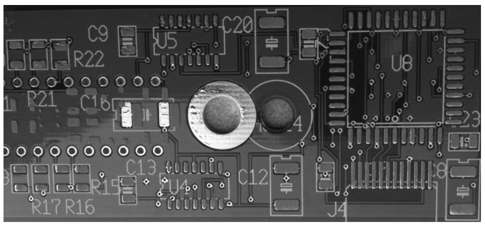
PCB classes The training itemset consisted of a number of circuit boards (including the artificial one mentioned in section 2.3) depicted in the images 6.2 and 6.3.

Fault classes All circuit boards of the training set were used in its original form and versions that resembled one of the faults described in chapter 4. These faults were categorized under the following six labels:

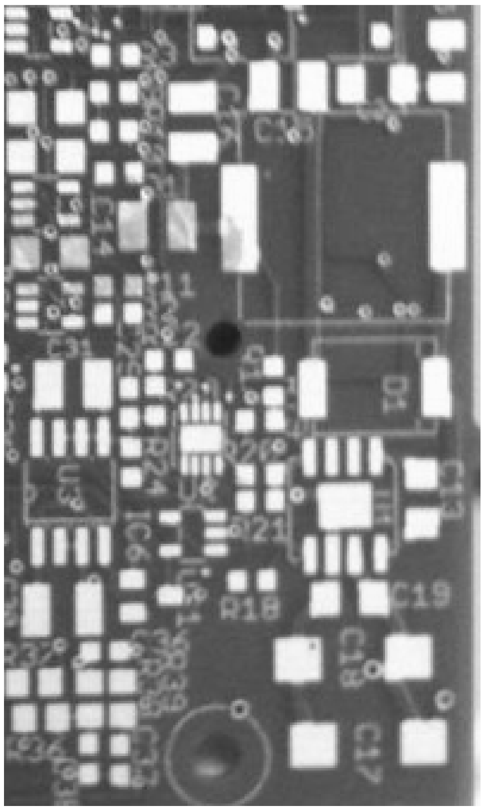
- Soldering points on PCB are missing (henceforce called *missing*)
- Solder paste of points is merged together (henceforth called *merged*)
- Solder paste is smeared outside of solder point (henceforth called *smeared*)



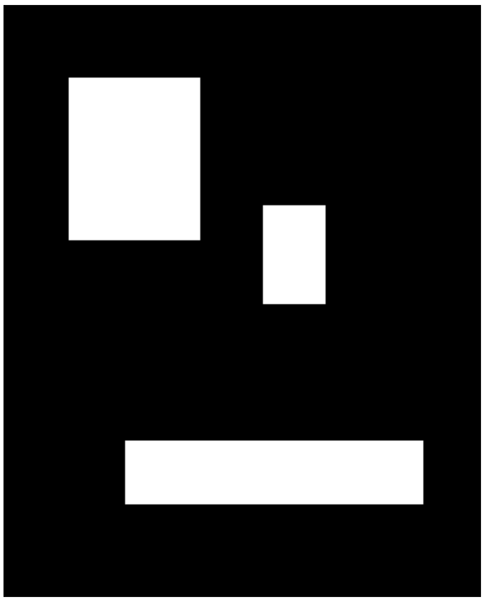
(a)



(b)

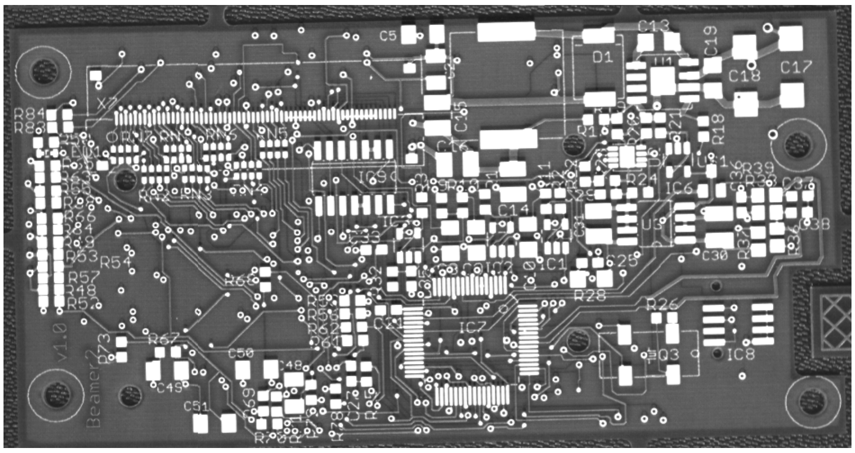


(c)

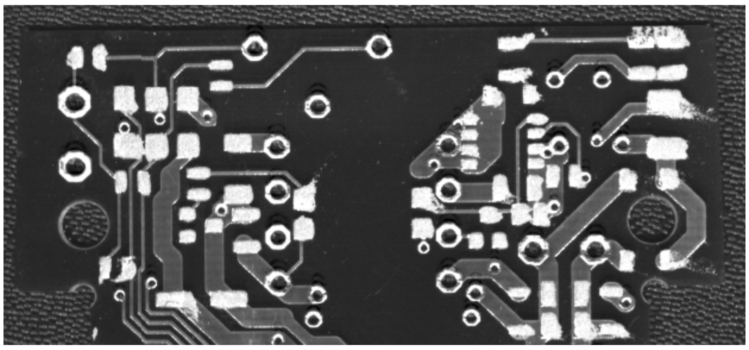


(d)

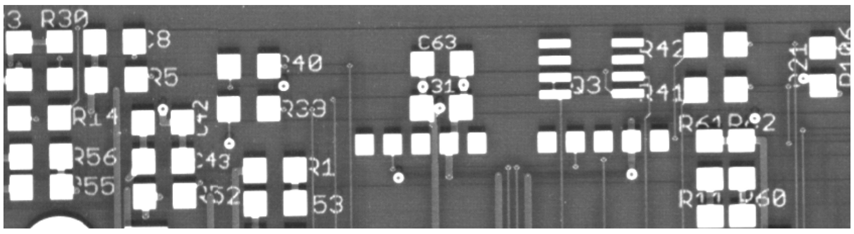
Figure 6.2: Images of circuit boards used during the training and evaluation process (part1)



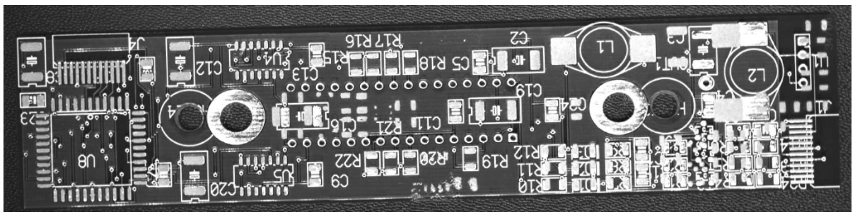
(a)



(b)



(c)



(d)

Figure 6.3: Images of circuit boards used during the training and evaluation process (part2)

- Soldering points are shifted from their original position (henceforth called *shifted*)
- PCB is rotated (along with all solder points) by a certain angle (henceforth called *rotated*)
- Solder points are stretched caused by a difference in belt velocity (henceforth called *streched*)

6.3.3 Matching criteria

In the course of this thesis, a number of criteria - which stated whether two objects being compared by the sweepline algorithm were actually called a *match* - were developed and evaluated.

Tolerance in center of gravity The criteria was that while two objects were compared, the X coordinate and Y coordinate must not differ by more than a certain amount of pixels in both dimensions in order to be a match. Following amounts were used:

- $\Delta x < 5$ && $\Delta y < 5$
- $\Delta x < 20$ && $\Delta y < 20$

Tolerance in area size For two objects to be a match, their area size must not differ by more than a certain amount of pixels, i.e.:

- $\Delta area < 50$

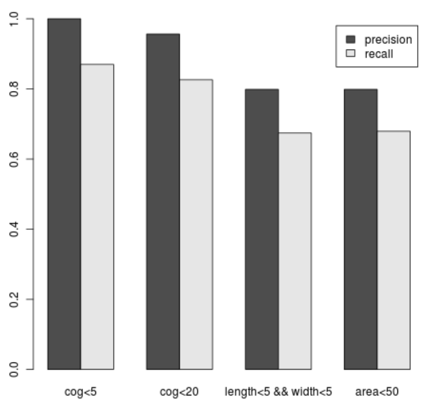
Tolerance in length and width Due to the fact that the value of *area* is different from multiplying length and width (in the way that the scanner software deals with those values), another matching criteria can be formed:

- $\Delta length < 5$ && $\Delta width < 5$

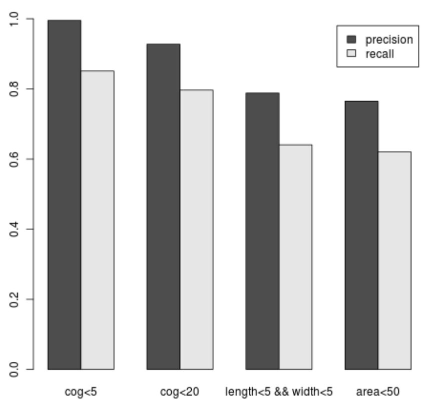
6.3.4 Training evaluation

In order to evaluate how a certain matching criteria performed, a known perfect matching was necessary. The reason for this is that only by knowing which two objects are corresponding (in the golden sample and in the PCB to be analyzed) one is able to determine whether a match (made by the algorithm with the current matching criteria) was truly correct. For that matter preknowledge of the used data was taken into account - however, it was not

Figure 6.4: Plot of precision and recall for fault classes *missing* and *merged*



(a)



(b)

	$cog < 5$	$cog < 20$	$length < 5$ && $width < 5$	$area < 50$
precision	0.960	0.924	0.760	0.739
recall	0.8193	0.812	0.648	0.633

Table 6.1: Average precision and recall for different matching criteria

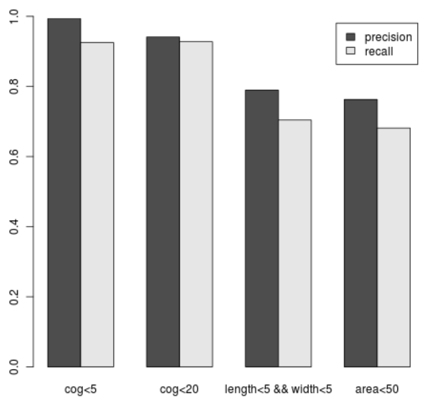
possible to create and/or use a 100% perfect matching (see problem chapter 7 for more details) and therefore an "almost perfect matching" was being used.

The target was to find a matching criteria which would achieve the best precision and recall (as defined in [HK06]) values when using the training items. A high precision in this context meant that the algorithm was matching the objects it could match correctly. A high recall meant that the items which needed to be matched were matched correctly.

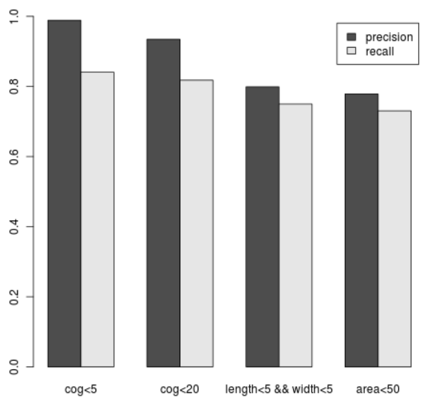
During the evaluation process, all circuit boards from the training set (with all appearing fault classes) were run with the different matching criteria. By using the "almost perfect matching", it was possible to record both precision and recall for all training items. There was no way, however, to determine these values for the fault class *rotated* - because not even an almost perfect matching could be used -, therefore no values were recorded for this fault class.

A mean of precision and recall (over all types of PCBs) was calculated and plotted for the different matching criteria. The charts in illustrations 6.4, 6.5 and 6.6 show those values for each of the fault classes.

Figure 6.5: Plot of precision and recall for fault classes *smeared* and *shifted*



(a)



(b)

Figure 6.6: Plot of precision and recall for fault class *stretched*

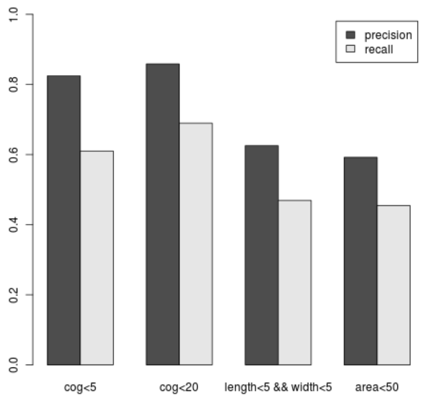
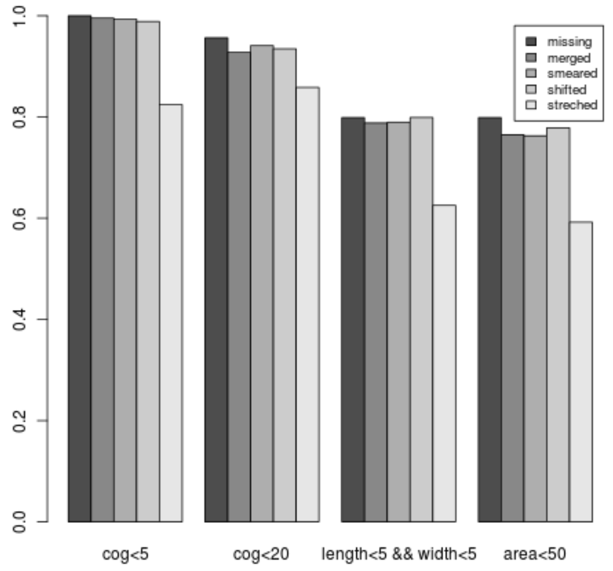
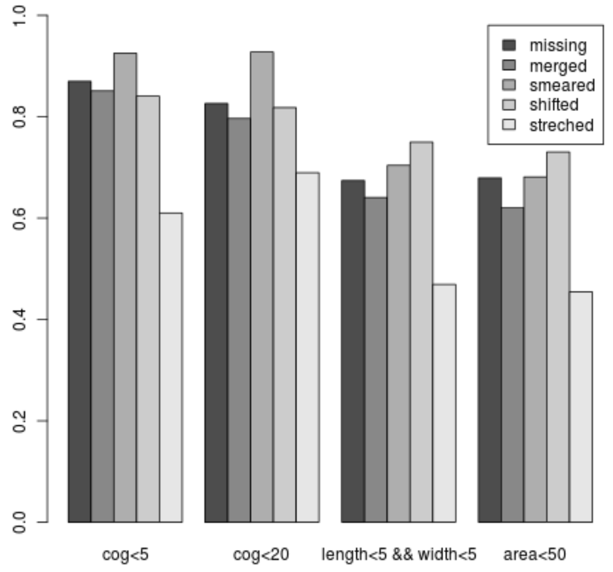


Figure 6.7: Precision and recall for all classes and matching criteria



(a)



(b)

6.3.5 Evaluation result

As mentioned in the introduction of this section (see beginning at 6.3), by using less strict matching criteria, the number of unmatched items would decrease but the quality of the matching (whether two objects were matched correctly) could also decrease. However, a matching criteria still needs to be able to classify the circuit board at hand, therefore the matching criteria will need to match a certain amount of objects. By comparing precision and recall of all classes (see illustration 6.7) and their mean values (in table 6.1), it is obvious that the matching criteria which tolerated a difference of the center of gravity of less than 5 pixels performed the best.

6.4 Testing

The results which were retrieved from the training step (as described in section 6.3) needed to be tested for effectiveness on data which was not used during training.

In order to do this datasets from another number of PCBs were used as test items. Namely these were items from PCB classes that were used during the training, but the image data was retrieved in an additional run of the scanner (and therefore similar but new object data was created). This means that these items resemble good cases of test items because they provide object differences as they appear in reality.

Chapter 7

Encountered problems

During the work on this thesis a few obstacles had to be conquered. The most prominent ones are being discussed in this chapter

7.1 Problems

Number of training and test items It was rather difficult to gather a large number of training and test items (and therefore objects for the matching cause). This was due to the fact that at the current state in the industr no automated process of checking for properly applied solder paste exists, yet. Also it is not easy to collect circuit boards that have purposely been produced faulty.

Perfect matching In order to calculate precision and recall, a perfect matching (see section 6.3.4) is necessary to determine which matches actually are true positives. Because the items used in the thesis created several thousands of objects, it was not possible to create a perfect matching.

7.2 Solutions

In order to simulate the errors resulting from the conveyor belt (as discussed in section 4.2), training object images have been manually defaced. With the help of image manipulation software, the binarized training image (7.1) has been altered in various ways (images of the alterations can be seen in 7.2 and 7.3):

- Simulate missing gray pads, image 7.2(a)
- Simulate merged gray pads, image 7.2(b)

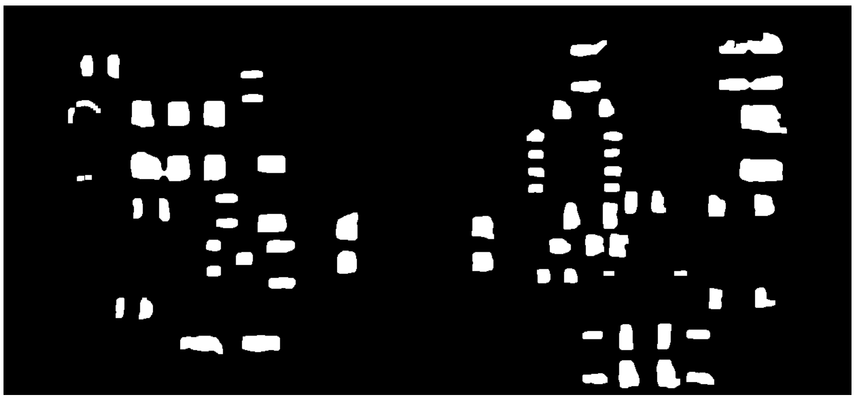
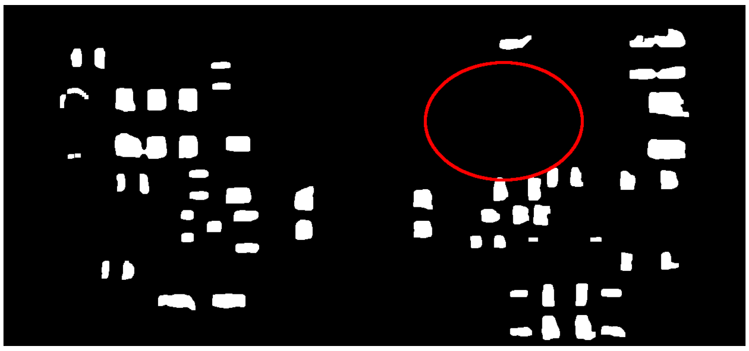


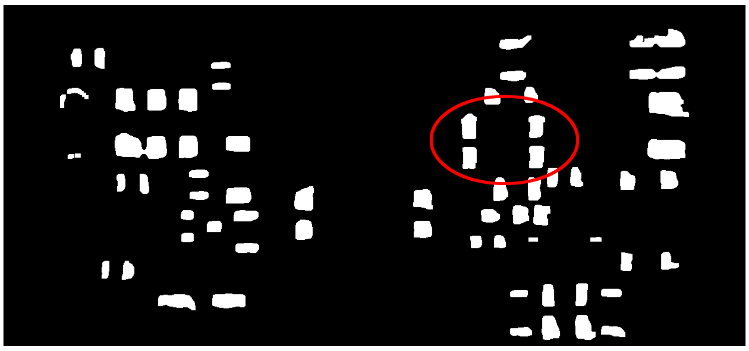
Figure 7.1: Image of a binarized circuit board in its original form without manual defacing

- Simulate smeared solder paste on pads, image 7.3(a)
- Simulate a shift of solder pads along the X and Y axis, image 7.3(b)
- Simulate a rotation of a PCB, image 7.3(c)
- Simulate a deviation in conveyor belt velocity, image 7.3(d)

As far as the perfect matching problem is concerned, based on some knowledge actually available during the training period it was possible to create an almost perfect matching for most fault classes. As mentioned in this section, fault classes of the different circuit boards were simulated by manually defacing the original images. One effect of this approach was that when manually creating faults (e.g. removing objects in order to simulate fault class *missing*) objects which were left untouched kept their exact feature data as in the golden sample: Take for example the fault class *merged*, where pairs of objects were manually merged together. This meant that all other objects would - among other features - appear actually at the same center of gravity compared to the golden sample. Because of this, it was possible - after the sweepline algorithm had finished the matching - to check matched pairs of items if their center of gravity was equal and so determine whether the match was a true positive or false positive. Having this information available enabled the calculation of precision and recall.

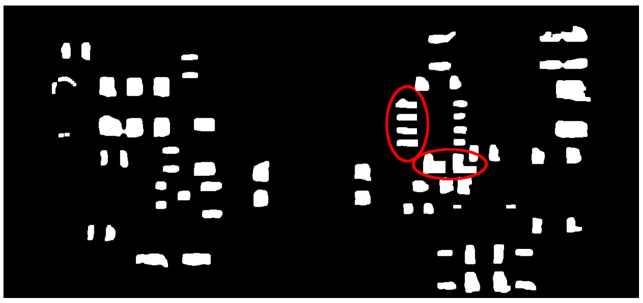


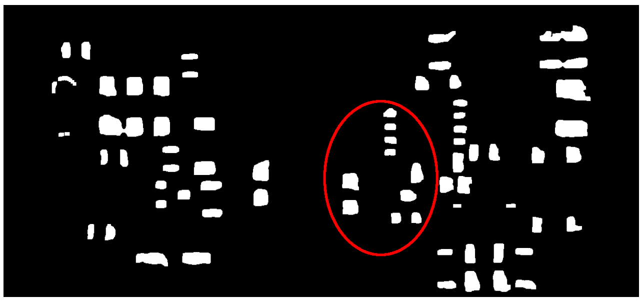
(a)

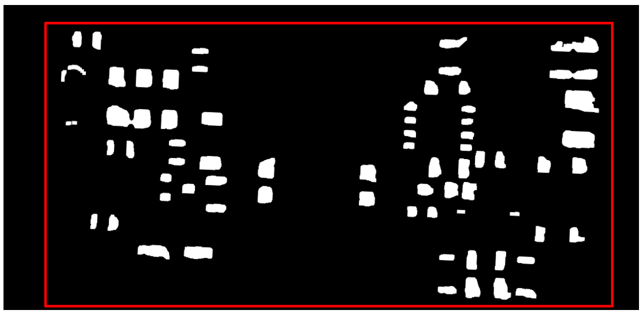


(b)

Figure 7.2: Images of binarized PCBs manually defaced (part 1)


(a)


(b)


(c)

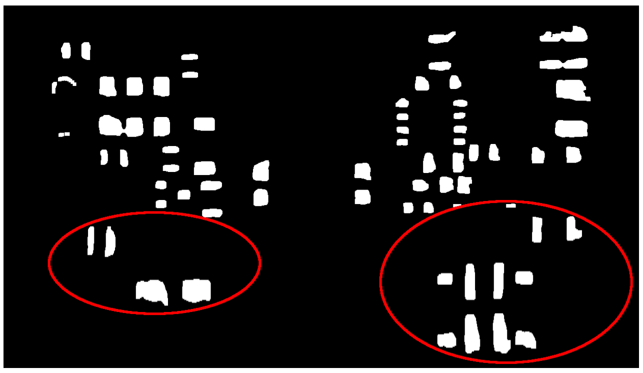

(d)

Figure 7.3: Images of binarized PCBs manually defaced (part 2)

7.3 Still existing problems

While it was possible to find an almost perfect matching for most fault classes, a perfect matching for the fault class *rotated* did not exist. Because during a rotation practically all features are modified, there is no way (except an almost impossible manual matching) to create such a perfect matching. For this reason that fault class was not taken into account when determining precision and recall.

Chapter 8

Discussion

The aim of this thesis was to develop an algorithm which would be capable to classify circuit boards during their manufacturing, as well as determine the correctness of the manufacturing process concerning the soldering points.

After extracting object information about the soldering joints, a sweepline approach compared all objects to corresponding objects of a golden sample and checked object pairs for similarity. Depending on a certain matching criteria, two objects were matched if a certain degree of similarity occurred.
The thesis evaluated a number of such matching criteria, all of which used different features about the objects, and determined which one was the best to use.

With the criteria of comparing the center of gravity of two objects and calling it a match if it did not differ by more than 5 pixels, the best result could be achieved during training and verified with a test set.

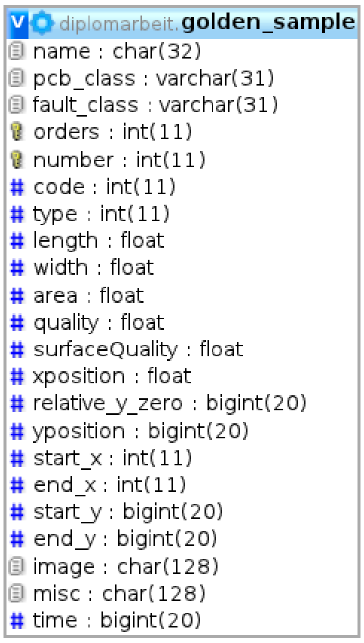
The way the sweepline algorithm determined that a match was absolute, it used the first appearing match of the items currently being compared in the sweepline group. Even if there was a closer match, e.g. according to the matching criteria the object would have a smaller distance, this fact was disregarded. One possible research extension to this thesis would be to consider all elements in the current sweepline group and use the best match instead of the first.

The topic of matching criteria can be discussed even further. The matching decision could be performed in a number of steps and weigh the results. In such a way a more tolerant matching might be achieved.

Appendix A

Database schema

The database which was used to store object information consisted of three tables *golden_sample*, *training_item* and *test_item*. All of the tables had the same schema, which is illustrated in A.1 and described in table A.



```
diplomarbeit.golden_sample
name : char(32)
pcb_class : varchar(31)
fault_class : varchar(31)
orders : int(11)
number : int(11)
code : int(11)
type : int(11)
length : float
width : float
area : float
quality : float
surfaceQuality : float
xposition : float
relative_y_zero : bigint(20)
yposition : bigint(20)
start_x : int(11)
end_x : int(11)
start_y : bigint(20)
end_y : bigint(20)
image : char(128)
misc : char(128)
time : bigint(20)
```

Figure A.1: Schema of a table how it was used in the database for storing objects

column	description
pcb_class	Type of circuit board (e.g. *alpha*, *bravo*, ...)
fault_class	Class of fault (e.g. *missing* or *golden_sample*)
number	Counter for objects within the same PCB
length	Length of the object in pixels
width	Width of the object in pixels
area	Number of pixels of the object's surface
xposition	X coordinate of the object's center of gravity
relative_y_zero	absolute Y coordinate of the PCB's top edge (in which an object occurred)
y_position	absolute Y coordinate of the object's center of gravity
start_x	X coordinate of an object's bounding box beginning
end_x	X coordinate of an object's bounding box end
start_y	absolute Y coordinate of an object's bounding box beginning
end_y	absolute Y coordinate of an object's bounding box end

Table A.1: Important and used attributes of the database schema

List of Figures

List of Tables

Bibliography

[DBCvK08] M. De Berg, O. Cheong, and M. van Kreveld. *Computational geometry: algorithms and applications.* Springer, 2008.

[HK06] J. Han and M. Kamber. *Data mining: concepts and techniques.* The Morgan Kaufmann series in data management systems. Elsevier, 2006.

[Inc11] Google Incorporated. *Google Books, http://books.google.com/*, accessed October 2011.

[SX08] Shashi Shekhar and Hui Xiong, editors. *Encyclopedia of GIS.* Springer, 2008.