3-28

MVA2007 IAPR Conference on Machine Vision Applications, May 16-18, 2007, Tokyo, JAPAN

A SVM based Method to Detect Color Shift Defects in IC Packages

R.M.C.B. Ratnayake, Craig Hicks, and M.A. Akbari
Kokusai Gijutsu Kaihatsu Co. Ltd.
2-3-9 Amanuma Suginami-ku Tokyo 167-0032 Japan
{chanaka, hicks2, akbari}@kgk.co.jp

Abstract

Automated Visual Inspection (AVI) is an essential part in the manufacturing process of Integrated Circuit (IC) packages. Contamination a common defect type found in IC packages appears as a shift in color. One of the main difficulties of this kind of inspection is manual parameter tuning, considering the fact that metallic areas change their colors from product to product, and depending on the IC package material and AVI System lighting.

The main target of this paper is to overcome this limitation by automating setting the decision rule, which is very difficult due the large number of parameters and their multidimensional behavior. For this purpose a novel parameter learning system based on Support Vector Machines (SVM) is proposed here to solve this problem.

1. Introduction

The performance of electronic equipment is improving rapidly. Portable electronic equipment requires smaller and thinner packaging systems for saving space and miniaturization. In addition, they need to be defect free to protect the integrity and performance of electronic equipment. As a result of these considerations, Automated visual inspection of IC packages is an essential part during its manufacturing process, and often is the preferred or the only choice available.

There are many algorithms and optimized implementations of those algorithms used for Automated visual inspection. Moganti [1] categorizes these algorithms in three types: reference-based, non-referential, and hybrid. In practice, two common considerations for choice of algorithm are speed, and a result equivalent or better than a human expert visual inspector can make given the same visual data and sufficient time. It is notable that human experts will inspect mostly without a reference pattern, inferring defects from a priori knowledge and the regularity of the pattern. A defect is usually an "obvious" irregularity, obvious being defined by the a priori knowledge. Algorithms, which capture this paradigm, have an advantage in approximating a human expert. Such algorithms belong to the class of non-referential or hybrid algorithms.

Contamination is a common defect found in metallic pads (wire bonding areas and solder land areas) of IC packages. It could be stain, discoloration or chemical residue on the surface added in the manufacturing process of IC packages, and the presence of contamination could affect the electrical conductivity.

Contamination appears as a shift in color in images taken from color CCD cameras. Appearances of pads are different from product to product depending on the IC package material characteristics. They also change with the inspection machine lighting. The presence of colors makes the task of parameterizing a defect quite difficult, because some combinations of colors may be ok while others are not, and the necessary number of parameters becomes quite large.

In conventional methods for detecting contamination defects, setting the decision rule is a difficult task: Whether to use a weighted sum of differences, or a logical formula, etc., the choice seems arbitrary. We have been using a relatively simple method to detect contamination defects in IC packages. We measured 6 features to inspect each pad: the medians of R, G, and B for each pad, and the medians of median R, G, and B of all the pads in the region of interest. In this method choosing the decision rule is very difficult due to multidimensional nature of these features.

To overcome this problem we introduce a machine learning based solution. Our method uses Support Vector Machine (SVM) [2] classifier, which is well known among the researchers for its accuracy as well as the ease of use compared to neural networks. We used n-fold cross validation evaluation method over the training data sets. Considering our practical previous experiences we believe the results are sufficient enough for this method to be implemented into the real automatic visual inspection system. There are some other studies on using SVM for visual inspection applications like [4] describing a SVM based method for self-inspection of integrated circuit pattern defects. However, those are different in terms of focus and evaluation methods.

The rest of the paper is organized as follows: Section2 describes SVM and why it is suitable for this problem, Section 3 describes the problem in detail, Section 4 describes our learning based approach to solve the problem, Section 5 describes our experimental setup and results in detail, in Section 6 we discuss advantages and disadvantages of this method, and finally Section 7 concludes the work and states the future work.

2. SVM

SVM a new technique for data classification is earning a large popularity among the researchers in various fields for its easiness to use and accuracy.

The SVM was chosen for the following reasons:

- The number of parameters to adjust is small.
- The decision regions are defined by carefully and automatically selected members of the training data. (These are the support vectors). This means that the reason for a decision can be given by

reference to the actual sample data which contributed most to a decision. With a neural network this cannot be done; the neural net has a black box nature. A customer might well query the “reason” for an inspection decision. It is very effective to be able to respond with examples which they have originally provided.
- It has the ability to find the essential out of a large number of parameters. Thus, the choice of parameters is not as relatively important as in some other algorithms.

A classification task usually involves with training and testing data sets. Each data set concludes one class label (target value) and several features (attributes). The goal of the SVM is to create a model using training data sets and use this model to predict the class label of a given test data set, which has only features. SVM maps the feature vectors of training data into a higher dimension space and it finds a linear separating hyper plane with a maximal margin in this higher dimension space.

SVM is easy to use: 1. Transform data to the format of an SVM software, 2. Simple scaling, 3. Select the kernel for learning, 4. Cross-validate and find best values for penalty parameter and kernel parameters, 5. Use those best parameters and train whole data, 6. Perform testing . For our experiments we used the easy to use LIBSVM [3] SVM library.

3. Statement of Problem

Contamination, which is a common defect type in IC package industry, appears as a shift in color in images taken from color CCD cameras. The appearance, colors of the pads (wire bonding areas and solder land areas) vary with the machine lighting and IC package materials. Figure 1 shows sample images for this situation where the pad colors are different. (A), (B) are of the same product while (C), (D) belongs to a different product.

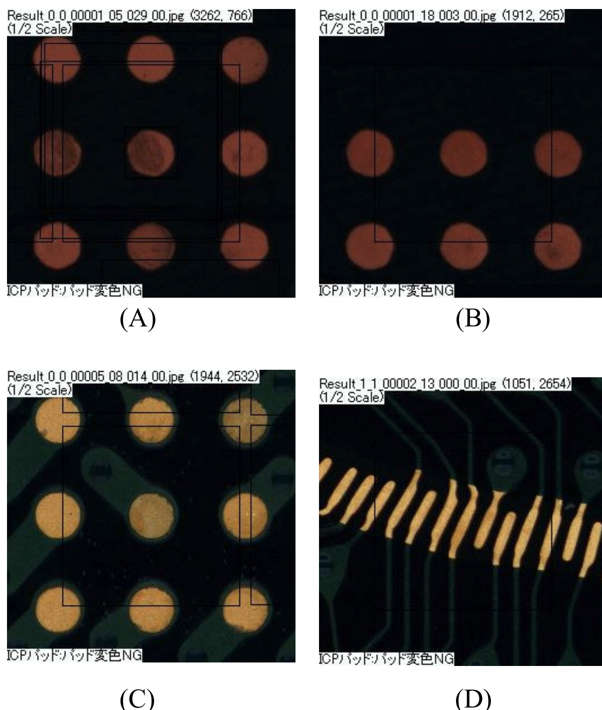

Figure 1. Defect candidates for contamination

```
For each pad calculate the mean values for R,G, B
  mr= Median (R)
  mg=Median (G)
  mb=Median (B)
Calculate the mean values for all pads
  mmr=Median (mr1, mr2,…,mrN)
  mmg= Median (mg1, mg2,…,mgN)
  mmb=Median(mb1,mb2,..,mbN)
For each pad judge for NG
  mrb = mr / mb
  mrg = mr / mg
  mrgb = mr / (mg + mb)
  mmrb = mmr / mmb
  mmrg = mmr / mmg
  mmrgb = mmr / (mmg + mmb)
  ma = sqrt(mr*mr+mg*mg+mb*mb);
  mma = sqrt(mmr*mmr+mmg*mmg+mmb*mmb);

  If any of the following are true output as defect
  mmrb / mrb < T0
  mmrg / mrg < T1
  mmrgb / mrgb < T2
  mr / mmr < T3
  mg / mmg < T4
  mb / mmb < T5
  ma / mma < T6
```

Figure 2. A simple algorithm, which we used to detect color shift defects

Figure 2 shows a simple algorithm, which can be used to detect the color shift. First for each pad we calculate median values of R, G and B. Then we calculate the median of median values for all the pads for each channel R, G and B. Finally take the fractional values as shown in Figure 2 and compare them with the input parameters (T0,..T6). Figure 1 images are color shift candidates output by the above algorithm. (A), (C), and (D) are real color shift defects while (B) is an overdetection. It is a tough task finding the correct set of parameters where (A), (C) and (D) get detected and (B) does not get detected, since there are seven input parameters. This process could be time consuming or even end up without finding a correct answer. Therefore a machine learning based method could be preferable.

We describe our novel machine learning based approach to solve this problem in the next section.

4. Outline of Method

We use a SVM classifier based approach to solve the problem described in Section 3. We selected LIBSVM described in Section 2 as the SVM tool kit for our approach.

Figure 3 describes the outline of the proposed solution. We can divide our approach into two main stages: training stage and machine judging stage. In the training stage we collect data as training data for the SVM. We use the algorithm described in Figure 2 to find defect candidates for the color shift. Feature data includes median data calculated in the algorithm described in Figure 2. In the Human Judgment process, for each color shift defect candidate a judgment is done based on the defect image.

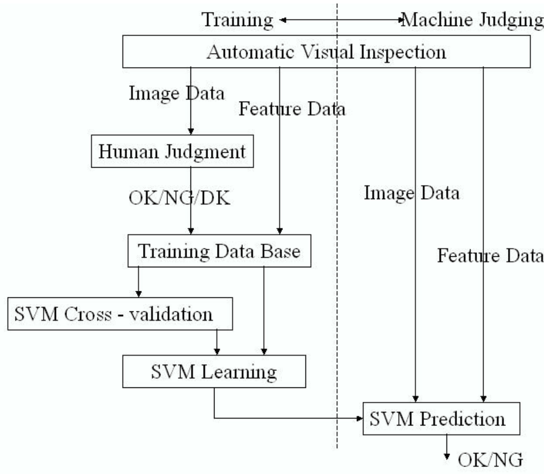

Figure 3. Outline of Method

There are three human judgments OK, NG and DK. OK represents not a color shift defect, NG represents a real color shift defect, DK represents difficult to judge candidates. Training data are labeled as OK, NG or DK. For training process we only use data labeled as OK or NG. Feature data are scaled before applied to the SVM.

SVM Cross-validation process can prevent the overfitting problem of the classifier. Here we evaluate the learning parameters and find the best parameters for the penalty parameter (C) and kernel parameters (gamma). Cross-validation is performed over various combination s of C and gamma. Then "grid-search" is performed over the grid of (C, gamma) pairs. Actually this has two steps; first a coarse grid search and identify a better region in the grid, then perform a fine grid search over that area. Next the data is trained in the SVM Learning process using the best parameters for C and gamma.

In the machine judging stage, SVM Prediction process makes decisions on input data vectors using the classifier calculated during training over whole training the data set. And it only makes two decisions: OK or NG.

5. Experiment Details

Here we describe our experimental setup and results analysis in detail. For the evaluation we used 100 training data (color shift defect candidates) output by our simulation programs, which uses the algorithm in Figure 2. Next we scaled these training data and input them to the SVM program and using the cross-validation method we calculated the best values for penalty parameter(C) and kernel parameter (γ). We plot graphs showing the grid of (C,γ). Then we train the whole training data set with these best parameters of (C,γ). Finally we test data sets with this classifier and record the classification accuracy.

5.1 Experimental Setup

During the Automated Visual Inspection we set the input parameters of color shift algorithm described in Figure 2 as $T0=T1=T2=T3=T4=T5=T6=0.8$. Then the color shift defect candidates were used as training data for the SVM. We can define the training data as (x_i, y_i), where x_i represents feature data vector and y_i the class label. During the human judge process we set the class label as,

y_i = -1 if judgment is NG
y_i = 0 if judgment is DK
y_i = 1 if judgment is OK

During the inspection median values calculated by the algorithm described in Figure2 are stored with defect candidates, and later they will be used to form the feature vectors:

$mrb = mr / mb$
$mrg = mr / mg$
$mrgb = mr / (mg + mb)$
$mmrb = mmr / mmb$
$mmrg = mmr / mmg$
$mmrgb = mmr / (mmg + mmb)$
$x_i = \langle mmrb/mrb, mmrg/mrg, mmrgb/mrgb \rangle$

We use C-SVM classification and radius based function (RBF) kernel with degree set to 2. We used n-fold cross validation to find the best parameter pair (C,γ). In this evaluation n-fold was set to 10. We used exponentially growing sequences of (C,γ) and make a grid.

$$C = \gamma = 2^{-20}, 2^{-19}, \ldots, 2^{20}$$

Then we perform a coarse grid search and find the best combination for (C,γ) and it is used for the learning of training data set. The penalty weights for classes are not set for this evaluation. Then finally we test the same training data set using the SVM prediction. Next sub session will describe our experimental results.

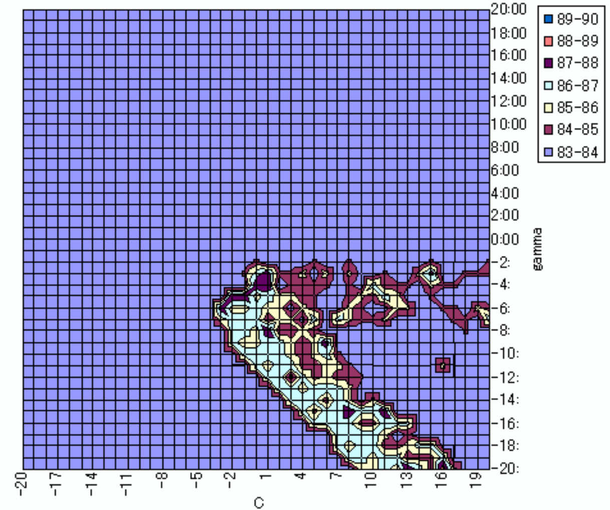

Figure 4. (C,γ) grid search using cross-validation

5.2 Results Analysis

Figure 4 shows the (C,γ) grid search results of cross-validation of the (C,γ) sequence described in 5.1. According to the graph it is clear that best values for (C,γ) lies in the region,

$$(2^{-3} < C < 2^{1}) \cap (2^{-6} < \lambda < 2^{-1})$$

with the cross-validation rate over 87%. We used (0.25, 0.03125) for (C,γ) and performed training of the data sets. Then when we test the same training data set using

this classifier and the classification accuracy was 91%.

By performing a finer grid search over the neighborhood of the coarse grid search best candidate we could have a better cross-validation percentage. We plan to fulfill this at the next step of this evaluation. For training we could also consider adding weights to change the penalty for some classes, if the input training data is unbalanced. Although we tested the same data sets, which were used as training data, use of different data sets would have been realistic.

6. Discussion

This method is supposed to be robust against the changes in materials of IC packages and lighting conditions. Although the experiments done in Section 5 involve only one product it can be easily extended to support multiple products from different materials under different lighting conditions. When there is a new product we can add training data to the database. In the case of a false machine judgment it is possible to investigate the training data candidate causing the judgment, is an advantage of using the SVM.

However unfortunately the lack of enough data prevents doing an intensive evaluation to prove it currently. The experimental data used here was data sets belonging to the same IC package material taken under same machine lighting, due the lack of supply. In the near future we plan to collect sufficient data covering variety of IC package materials and do a full evaluation of this method and comparison with the human expert accuracy.

Setting the feature space for the SVM is another task. According to the results shown in Section 5, the feature space described in this paper seems sufficient for this product.

7. Conclusion and Future Work

Identification of color shift defects of metallic pads in IC packages is a difficult task, considering the fact that pad colors various from product to product depending on the package material and Automatic Visual Inspection system lightening. We presented a SVM based learning method to solve this problem. Our method takes color shift defect candidates from the existing algorithm as training data to the SVM. We create feature vectors using median values and we use n-fold cross validation to find the best parameters for training.

This is an ongoing project and we plan to develop the current system to perform fine grid searching during the n-fold cross validation to maximize the classifier. Furthermore we are planning a full evaluation of this method using Automated Visual Inspection machines and experienced operators.

References

[1] Madhav Moganti, Fikret Ercal, Chihan H. Dagli, and Shou Tsunekawa: “Automatic PCB Inspection Algorithms: A Survery,” *Computer Vision and Image Understanding: CVIU*, vol. 63, no. 2, pp. 287-313, 1996.

[2] N. Cristianini and J. Shawe-Taylor: “AN INTRODUCTION TO SUPPORT VECTOR MACHINES (and other kernel-based learning methods),” *Cambridge University Pres, ISBN: 0 521 78019 5, 2000.*

[3] Chih-Chung Chang and Chih-Jen Lin: “LIBSVM : a library for support vector machines,” 2001. Software available at http://www.csie.ntu.edu.tw/~cjlin/libsv

[4] Hanying Feng, Jun Ye, and R. Fabian Pease: “Self inspection of integrated circuits pattern defects using support vector machines,” *Journal of Vacuum Science & Technology B: Microelectronics and Nanometer Structures* November 2005 ,Volume 23, Issue 6, pp. 3085-3089