

Short Papers

Data Mining of Printed-Circuit Board Defects

Andrew Kusiak and Christian Kurasek

Abstract—This paper discusses an industrial case study in which data mining has been applied to solve a quality engineering problem in electronics assembly. During the assembly process, solder balls occur underneath some components of printed circuit boards. The goal is to identify the cause of solder defects in a circuit board using a data mining approach. Statistical process control and design of experiment approaches did not provide conclusive results. The paper discusses features considered in the study, data collected, and the data mining solution approach to identify causes of quality faults in an industrial application.

Index Terms—Autonomous systems, data mining, machine learning, manufacturing fault detection, PCB assembly, quality engineering, rough set theory.

I. INTRODUCTION

The problem considered in this paper deals with the occurrence of solder-ball defects underneath electronic components assembled as printed circuit boards (PCBs). These defects may contribute to unforeseen failures of electronic systems containing the affected PCBs. Due to the production faults, random samples of finished circuit boards are collected and x-rayed in order to check for the presence of solder-ball defects. If a defect is found on a board, the entire assembly batch is x-rayed to ensure no defective board leaves the factory. Of course, the x-ray process becomes a significant component of the circuit cost. Approximately 4.5% of all circuit boards manufactured are found to contain solder-ball defects. For details of electronic assembly process, the reader may refer to [1]–[4].

The research reported in the paper is based on the developments in machine learning and data mining (e.g., [5], [6], and [7]). Some of the best-known learning algorithms are as follows.

- *ID3*: Induction decision tree is a supervised learning algorithm [8].
- *AQ15*: Inductive learning system generates decision rules, where the conditional part is a logical formula [9]. Domain knowledge is used to generate new attributes that are not present in the input data.
- *Naïve-Bayes*: A simple induction algorithm that computes conditional probabilities of the classes. Given the instance, it selects the class with the highest posterior probability [10].
- *OODG*: Oblivious read-once decision graph induction algorithm for building oblivious decision graphs, using a bottom-up approach [11].
- *Lazy decision trees*: An algorithm for building the best decision tree for every test instance [12].
- *C4.5*: The decision-tree induction algorithm [13].
- *CN2*: The direct rule induction algorithm [14]. This algorithm combines the best features of both *ID3* and *AQ*, where it uses

pruning techniques similar to the techniques used in *ID3* and similar to the conditional rules used in *AQ*.

- *IB*: The instance-based learning algorithm [15].
- *OCI*: The oblique decision-tree algorithm [16].
- *T2*: The two-level error-minimizing decision tree algorithm [17].
- *LEERS*: Learning from examples using rough sets system [18].

Examples of other algorithms and developments in learning and data mining can be found in [19]–[22]. For a survey of important applications of machine learning, see [23].

A. Why Data Mining?

Regression analysis and neural networks are two tools that could potentially be applied to solve the quality engineering problem considered in this paper. Both share some commonality with the rough set theory approach proposed by Pawlak [24], which is used in this paper. However, there are fundamental differences between the two approaches and data mining. First, both neural network and regression models involve approximation errors, while the rough set theory approach is able to accurately capture relationships between input features and the decision. Neural network and regression models are “population based” as a single model is formed for the entire population (training data set), while the rough set theory approach follows an “individual (data object)-based” paradigm. The two “population-based” tools determine features that are common to the population. The rough set theory approach identifies unique features of an object and sees whether they are shared with other objects. It is obvious that the two paradigms differ and in general the set of features derived by any of the two paradigms is different. In addition, each of the two “population-based” methods uses a fixed set of features. In the approach advocated in the paper, the same set of features applies to a group of objects rather than the entire population. Finally, the model (rules) created by the data mining approach used in this paper is explicit.

One of the greatest advantages of data mining is that data needed for analysis can be collected during normal operation of the process being studied. This contrasts with other approaches such as the design of experiment (DOE) approach, where costly experimentation is essential. Another advantage of data mining is that the data set used for extracting decision rules does not have to be complete.

Most of the rule extraction algorithms developed to date fall into the following two classes:

- 1) decision tree algorithms, e.g., *ID3* [8] and *C4.5* [13];
- 2) decision rule algorithms, e.g., *AQ15* [9] and *LEERS* [18].

In this paper, the rough set approach [24] will be used to detect causes of the PCB quality problem. One of the reasons for using the rough set approach over the decision tree approach is the belief that the former approach is more suitable for the problem considered in this paper. The example presented next illustrates the results produced by the two classes of learning algorithms.

Example: Consider the training data set in Fig. 1 containing eight objects, each described by four features F1–F4 and a known decision (outcome) *D*.

The rules shown in Fig. 2 are derived from the data in Fig. 1 with the decision tree algorithm [13]. The objects supporting each decision rule are listed behind each rule.

Manuscript received July 11, 2000; revised February 21, 2001. This paper was recommended for publication by Associate Editor Y. Narahari and Editor N. Viswanadham upon evaluation of the reviewers' comments.

The authors are with the Intelligent Systems Laboratory, Department of Industrial Engineering, University of Iowa, Iowa City, IA 52242-1527 USA (e-mail: andrew-kusiak@uiowa.edu).

Publisher Item Identifier S 1042-296X(01)04816-9.

No.	F1	F2	F3	F4	D
1	1.02	Red	2.98	High	2
2	2.03	Black	1.04	Low	1
3	0.99	Blue	3.04	High	2
4	2.03	Blue	3.11	High	2
5	0.03	Orange	0.96	Low	1
6	0.04	Blue	1.04	Medium	1
7	0.99	Orange	1.04	Medium	2
8	1.02	Red	0.94	Low	1

Fig. 1. Training data set.

Rule 1. IF F4 = High THEN D = 2; [1, 3, 4]
 Rule 2. IF F4 = Medium AND F2 = Blue THEN D = 1; [6]
 Rule 3. IF F4 = Medium AND F2 = Orange THEN D = 2; [7]
 Rule 4. IF F4 = Low THEN D = 1; [2, 5, 8]

Fig. 2. Rule set derived by the C4.5 algorithm.

No.	F1	F2	F3	F4	D	Rule
1	1.02	Red	2.98	High	2	Rule 1
2	2.03	Black	1.04	Low	1	Rule 4
3	0.99	Blue	3.04	High	2	Rule 1
4	2.03	Blue	3.11	High	2	Rule 1
5	0.03	Orange	0.96	Low	1	Rule 4
6	0.04	Blue	1.04	Medium	1	Rule 2
7	0.99	Orange	1.04	Medium	2	Rule 3
8	1.02	Red	0.94	Low	1	Rule 4

Fig. 3. Patterns corresponding to rules in Fig. 2.

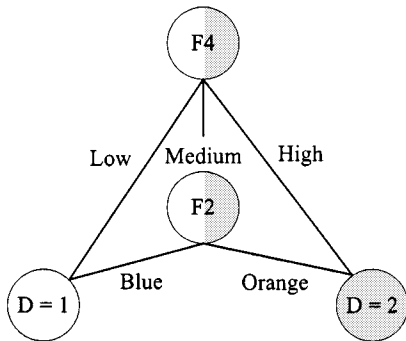


Fig. 4. Decision tree representing the rules in Fig. 2.

Rule 1. IF F4 = Low THEN D = 1; [2, 5, 8]
 Rule 2. IF F1 = 0.04 THEN D = 1; [6]
 Rule 3. IF F4 = High THEN D = 2; [1, 3, 4]
 Rule 4. IF F1 = 0.99 THEN D = 2; [3, 7]

Fig. 5. Rule set derived by the rough set algorithm.

The elements of the data set included in the rules in Fig. 2 create a pattern in the matrix in Fig. 3. All feature values of column F4 are involved in the rules, which is characteristic of tree type algorithms.

The patterns in Fig. 3 are reinforced with the decision tree in Fig. 4.

Another set of rules extracted with a rough set algorithm and the corresponding patterns are shown in Figs. 5 and 6. In this case, feature F4 is only partially involved in the rules. In addition, object 3 appears in two rules, 3 and 4.

The decision rules of Fig. 5 are represented as the tree in Fig. 7.

No.	F1	F2	F3	F4	D	Rule
1	1.02	Red	2.98	High	2	Rule 3
2	2.03	Black	1.04	Low	1	Rule 1
3	0.99	Blue	3.04	High	2	Rule 3, 4
4	2.03	Blue	3.11	High	2	Rule 3
5	0.03	Orange	0.96	Low	1	Rule 1
6	0.04	Blue	1.04	Medium	1	Rule 2
7	0.99	Orange	1.04	Medium	2	Rule 4
8	1.02	Red	0.94	Low	1	Rule 1

Fig. 6. Patterns corresponding to the rules in Fig. 5.

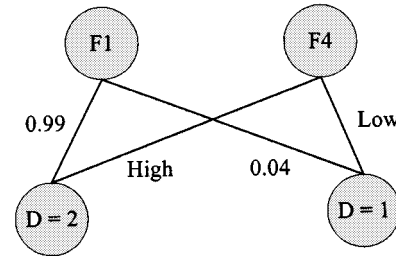


Fig. 7. Tree representation of the decision rules from Fig. 6.

The patterns in Figs. 3 and 6 and the corresponding trees in Figs. 4 and 7 point to differences between the two types of learning algorithms, the decision tree and the decision rule algorithms. The rule tree has flat shape as opposed to a decision tree that spans over more levels.

The fact that in the solution provided by the rough set algorithm none of the features appear in all rules makes this algorithm suitable for the application considered in this paper as the extracted rules are more "individualized."

II. DEFINITION OF FEATURES

The first step in developing a solution approach was to understand the assembly process and to identify all features that could potentially cause PCB defects and could be measured.

The assembly process begins with the blank boards, which are manufactured by an independent manufacturer. In this case, the boards themselves are immediately ruled out as a source of the solder-ball defects as it is assumed the surfaces and composition of all boards are homogeneous and defect-free coming from the manufacturer. In some applications where mining would be performed at higher level of granularity, the boards themselves would need to be described with features.

Solder is applied to the PCBs through either an automated or manual process where the solder paste is applied through a stencil specific to the assembly being produced. With the hand-pasted solder, as with any manual process, there is always the possibility of human error. The stencils are to allow solder paste to encounter the PCB only on top of the designated sites, or pads, where the electronic components will be placed. Naturally, for different PCBs there are different stencils; different both in pattern and composition. There are several unique features present in this stage of the manufacturing process, which makes it a critical stage for the data collection.

The stencils themselves are the source of five unique features: stencil composition (copper or steel), the stencil thickness (in millimeters), the justification of the stencil pattern (left, right, or centered), whether or not vias (small holes in stencil present for alignment purposes) are present, and whether or not the stencil was used for a two-sided assembly (components on both sides of a PCB) or a one-sided assembly (components on only one side of a circuit board). Also found to be important features to consider as potential sources of defects were the age

of the solder paste and the composition of the solder paste. However, the nature of the features made them impractical and near impossible to account for.

Also associated with the stencils and the solder paste application are two additional features. Depending on the type of stencil (i.e., presence of vias, component patterns, etc.), a vacuum may or may not be turned on to secure the PCB in place while the solder paste is applied. This feature was selected as an important feature because an uneven application of solder paste due to PCB movement during the application process can lead to the creation of solder-ball defects. Solder-ball defects may also be created because of contamination present in the solder paste. One source of contamination is solder paste that has been exposed to air for an extended period and consequently hardens and self-contaminates.

A second and most common source of solder ball contamination is when solder residue accumulates on the stencil and is transferred onto the PCB during the pasting process. To account for this, the frequency of stencil cleanings for each assembly was incorporated in the data collection.

The next step in the production process is for the electronic components to be placed onto the PCBs. Placement of the components is accomplished either by hand or by an automated computer-controlled machine that rapidly and highly accurately places each component in the designated position on the PCB. Machine-applied components are of little concern as a potential cause of solder-ball defects because the placement process is highly accurate thanks to advanced computer optical recognition technology. Each component's position is double-checked by software that receives data from a camera monitoring every placement. Not only is this a highly accurate process, but it takes place in a sealed and controlled environment. The only thing that ever is exposed to the solder paste or the PCB at this stage is the components themselves. With hand-applied components human error is possible, making it a potential cause of defects.

The assembly process directly following the placement of components was immediately recognized as the most probable source of defects. Immediately after the components are placed on the solder and before the solder is hardened in a furnace, the PCBs are moved down a conveyor belt where they are picked up by a human operator, inspected, and placed on a cart for transport to the oven. At this stage, it has been observed that the movement along the conveyor belt and the jarring effects of a human operator handling the PCB occasionally cause a component to slide out of place on the solder paste. When a component is found to have slid out of place, a human operator must manually reposition that component back in to the appropriate position. It is possible that the movement of components cause a smearing of the solder paste, which in turn creates a condition where the solder paste is not evenly distributed underneath the components. Upon initial inspection of the assembly process, the movement of components was suspected to be the primary source of the solder-ball defects.

One final feature measured was the position in the assembly run of the boards from which data was collected (i.e., boards were the first five manufactured or last five manufactured). Data was collected for this feature for two reasons. The first reason ties back to the stencil cleaning; if the majority of the solder-ball defects were found to be present in boards produced near the end of the run and if the stencil was never cleaned, this evidence would strongly suggest the solder balls were being caused by a dirty stencil. The second reason is simply to capture a broader and more accurate sample of the conditions present during the assembly process. If a temporary, external, and unidentifiable feature were present at the beginning of the assembly run, it is possible that it will not be present toward the end of the run, or vice versa. In this case, the undetectable external features influence will easily be-

TABLE I
FEATURE SET

F1	Stencil composition (steel or brass)
F2	Stencil thickness (in mm)
F3	Stencil vias (present or not present)
F4	Side (components on bottom or both)
F5	Stencil center justification (Yes or No)
F6	Operator (operator name)
F7	Cleaning frequency (frequency of stencil cleanings)
F8	Vacuum (On or Off)
F9	Position (1 = first 5 PCBs, 2 = last 5 PCBs)
F10	Component application (M = Machine, H = Hand)
F11	Moved components (designators of components moved by hand)
F12	Hand placed components (designators of components placed by hand)
F13	Paste application (M = Machine, H = Hand)
F14	Designator of component with a solder ball

come observable as its effect will be seen in the PCBs taken from the first half of the run and not the PCBs taken from the second half.

If a solder-ball defect was discovered on a PCB, the designator of the components under which the defects were discovered was recorded. There are two different types of components that are placed on the PCBs. It was believed that specific types of components were more prone to defects than others. The first type of component is an SOIC. SOICs are low profile components that are placed directly on the PCB and are typically resistors, transistors, and other simple electronic components. The second type of component is a fine detail component. Fine detail components are typically integrated circuit chips that sit above the circuit board with a large number of pins connecting it to the PCB (this is where the term "fine detail" comes from).

The list of features used in this study is shown in Table I.

III. DATA COLLECTION

The most important consideration during the data collection process was the accuracy and completeness of data collected. Accuracy is crucial for obvious reasons; if any given feature were recorded incorrectly, a data set that does not correctly represent the actual manufacturing process might be created. Consequently, any conclusions or findings based on such data would be erroneous and trivial.

Completeness was a condition that was more difficult to fulfill. Completeness refers to ensuring every possible feature that could be accounted for had been acceptably recorded. There were two major obstacles to overcome when assuring the completeness of data collected: technical obstacles and human obstacles. Technical obstacles consisted of the challenge to identify and quantify a set of features that truthfully and wholly represented the manufacturing process and the environmental conditions present during the manufacturing process. To accomplish this, collaboration between management, machine operators, and outside consultants was required to guarantee every possible feature was considered. It is important to note that the machine operators identified some of the most important and previously overlooked features.

The human obstacles arose when it was realized that by recording the name of the operator on the assembly for which data was being collected, it was possible for defects to be tied back to the operator. If the operators realized this, the possibility that perhaps the operators would not record important information (i.e., movement of components) for fear of being associated with the defects was introduced. To neutralize this possibility, all raw data was kept confidential and used only for data mining purposes. Management never saw any data suggesting a

```
IF (Thickness = 6) AND (Vias = No) AND (Type_of_comp = SO_8C) THEN
(Solder_Ball = N); [518, 26.39%, 100.00%]
```

Fig. 8. Rule 1.

```
IF (Hand_Placed = Y) AND (Operator = W) AND Type_of_comp = SO_8C) AND
(Machine_or_Hand_Applied_Paste = M) AND (Vias = No) (THEN (Solder_Ball
= N); [159, 8.10%, 100.00%]
```

Fig. 9. Rule 2.

```
IF (Machine_or_Hand_Applied_Paste = M) AND (Hand_Placed = U11) AND
(Vac_on/off = OFF) AND (Type_of_comp = SO_8C) THEN (Solder_Ball = N);
[153, 7.79%, 100.00%]
```

Fig. 10. Rule 3.

```
IF (Type_of_comp = SO_14B) THEN (Solder_Ball = Y); [8, 8.99%, 100.00%]
IF (Type_of_comp = SO_20A) THEN (Solder_Ball = Y); [4, 4.49%, 100.00%]
IF (Type_of_comp = SO_16LA_B) THEN (Solder_Ball = Y); [6, 6.74%, 100.00%]
```

Fig. 11. Three rules identifying faulty PCBs.

trend where a specific operator was linked to solder-ball defects unless the findings were significant enough to suggest that the operator was indeed the source of the defects.

To precisely represent the standard manufacturing process, it was determined that a DOE approach was not only inappropriate, but also unnecessary. Instead, data was collected as the PCBs were manufactured in the standard operating procedure. For each unique batch of PCBs (kits), ten PCBs were selected from the kit and data was collected on only those ten items. The PCBs selected were the first five and last five for the reasons described earlier in this paper. A kit may consist of anywhere from 5–500 PCBs.

For every kit that data was collected from, a standardized data collection sheet was associated with the kit. Data was collected on the ten PCBs selected for data collection by the operators who were responsible for the PCBs at each stage along the production line. For instance, if a PCB came out of the component placement stage with misplaced components, the operator responsible for checking the placement of components would record the location and type of component that was out of place and move the component back in to place.

After the entire kit had been completed, the ten PCBs slated for data collection were x-rayed to check for the presence of solder-ball defects. If solder-ball defects were found to be present on a PCB, the locations and types of components the solder balls were discovered under were recorded.

Data was collected for 2052 PCBs out of which 89 PCBs contained defects. The data was collected over a three-month period.

IV. RESEARCH FINDINGS

The data was analyzed with the data mining algorithm thus producing three separate rule sets. The first set of rules defined the conditions under which solder-ball defects did not occur. The second rule set defined when solder-ball defects did occur. The third rule set provided an approximate rule where alternative outcomes occurred under the same set of conditions. Each of the three categories of rules is illustrated next.

The first rule describes the relationship between the thickness of the stencil, vias, the component type, and the presence of solder-ball defects (Fig. 8).

What this rule states is that when the condition is met no solder-ball defects are produced. This rule is significant because it represents 26.39% (518 boards) of the population of PCBs, and of those, 100% were solder-ball defect-free.

The second rule set represents a relationship that was entirely unexpected. The rule describes the relationship between hand-placed components, operator, the component type, machine-applied paste, no vias, and the absence of solder-ball defects (Fig. 9).

This rule signifies a relationship that was unexpected because it shows a correlation between defect-free PCBs and a rather long list of conditions. However, one must be careful not to plainly state that the components described by this rule can be ruled out as a source of defects since the rule is dependent upon the combination of various factors. This rule is significant since it represents 8.10% of all PCBs, of which there is a zero occurrence of solder-ball defects.

The third illustrative rule shown in Fig. 10 is similar to Rule 2 of Fig. 9. The relationship represented is that between machine-applied paste, hand-placed component, vacuum is in an off state and the absence of solder balls.

When the paste is machine applied, there are no hand-placed components and the PCB is in the last half of the kit, there is a zero occurrence of solder-ball defects. This rule applies to 7.79% of the PCBs.

Rules derived when looking at the data for PCBs with solder-ball defects would be more telling of where the problem lies, but with 89 defects out of a sample of 2052 it would be premature to derive final conclusions. While the rule sets are not by any means final, they do very strongly suggest the source of the problem is primarily the type of component. Data mining provided numerous rules defining the conditions under which solder-ball defects occurred.

Three illustrative rules are shown in Fig. 11.

These rules simply state that of 18(8 + 4 + 6) solder-ball defects found, 8.99% were associated with component S0_14B, 4.49% were found under component S0_20A, and 6.74% were associated with component S0_16LA_B.

In addition to the exact rules, approximate rules were extracted that associate conditions and the outcome with ambiguity. Examples of two such rules are shown in Fig. 12.

The last two numbers behind each rule indicate the number of PCBs supporting each of the two alternative outcomes. The presence of approximate rules indicates that the feature set considered in this study was not sufficient and additional features need to be defined.

```

IF (Type = Steel) AND (Vias = Yes) AND (Centered = No) AND (Moved_Comp
= U1) AND (Hand_Placed = U11) THEN (Solder_Ball = Y) OR (Solder_Ball =
N); [42, 33.60%, 100.00%] [10, 32]
IF (Operator = D) AND (Moved_Comp = Q3) THEN (Solder_Ball = Y) OR
(Solder_Ball = N); [4, 3.20%, 100.00%] [1, 3]
    
```

Fig. 12. Two approximate rules.

TABLE II
ABSOLUTE CLASSIFICATION ACCURACY

D	Y	N	None
Y	42	47	0
N	1	1962	0

TABLE III
CLASSIFICATION ACCURACY FOR THE DATA SET.

	Correct	Incorrect	None
Y	52.52 +/- 26.60	47.48 +/- 26.60	0.00 +/- 0.00
N	99.95 +/- 0.15	0.05 +/- 0.15	0.00 +/- 0.00
Average	97.66 +/- 1.23	2.34 +/- 1.23	0.00 +/- 0.00

A. Validation Results

The quality of predictions with the rules extracted from a data set is usually evaluated by a cross-validation scheme [24]. The *k*-fold (here *k* = 10) cross-validation scheme is often recommended. In this scheme, a training data set is partitioned into *k* = 10 folds (subsets) and one at a time fold of objects (PCBs) is removed from the training data set and the rules are extracted from the set of *k* - 1 = 9 folds. The decision Solder_Ball is predicted for each object in the single fold based on the rules extracted from the *k* - 1 fold constituting a training data set. This process is repeated *k* = 10 times.

The testing of the data set with 2052 objects has produced the accuracy results presented in Tables II and III.

The letter D in Table II denotes the decision Solder_Ball. The diagonal numbers in Table II represent the number of outcomes Solder_Ball = Y or N that have been correctly predicted. In this case, 42 of the 89 (= 42 + 47) decisions Solder_Ball = Y have been correctly predicted along with 1962 decisions Solder_Ball = N. The numbers off the diagonal indicate the incorrectly predicted decisions. For the decision Solder_Ball = N (row N in Table II) one object of 1993 in the N category was incorrectly classified as Y category and for decision Solder_Ball = Y the number of misclassified objects is 47. These results are very encouraging as they indicate that the rules recognize all but one PCBs with no solder balls and almost half of PCBs with solder balls.

Table III reports the percentage of objects (known in data mining as classification accuracy) that have been classified correctly, incorrectly, or fall into the None category for the three decision values Solder_Ball = Y or N.

The last row in Table III includes the average classification accuracy for the rules extracted from each of the *k* - 1 = 9 folds and tested for the objects in each of the ten test folds.

B. Future Research

The approach discussed in this paper is a component of more general knowledge life cycle outlined in Fig. 13.

This knowledge life cycle is a process involving the following four phases:

- knowledge extraction;
- decision-making;

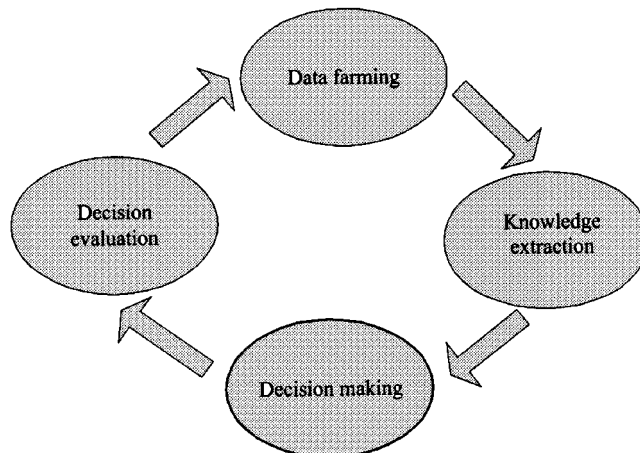


Fig. 13. Knowledge life cycle.

- decision evaluation;
- data farming.

In this paper, only the knowledge extraction phase was discussed. However, in general there are three additional phases in the knowledge life cycle. The extracted knowledge is used to make decisions, e.g., with the algorithms presented in [25]. The quality and robustness of the generated decisions need to be evaluated. The quest for high quality and robust rules may lead to data farming, which ultimately determines the viability of the data mining approach.

V. CONCLUSION

In this paper, an important application of data mining in electronic assembly was discussed. The recent progress in data mining, including the rough set theory, provides opportunities for solving many industrial problems, e.g., equipment and process failure diagnosis. The knowledge extracted with data mining algorithms can be integrated with expert systems and web-based application software.

The rules derived from the data set considered in this paper provide a robust indication of where to narrow, and thereby make more effective, further investigations in to the cause of the solder-ball defects. They are invaluable in narrowing down the scope of the future investigation.

It is clear from the data that the defect occurrence is closely tied to component type. Perhaps the strongest conclusion that can be drawn from the data is a prediction that a significant portion of defects will be discovered to form underneath specific components. These conclusions should be considered as hypotheses of what may cause the defects and a further study of the manufacturing process should be undertaken to prove or disprove their validity. The future study will involve definition of additional features and further collection of data.

REFERENCES

[1] K. Brindley and M. Judd, *Soldering in Electronics Assembly*. Oxford, U.K.: Butterworth-Heinemann, 1999.
 [2] R. J. Rowland and P. Belangia, *Applied Surface Mount Assembly: A Guide to Surface Mount Materials and Processes*. New York: Wiley, 1992.

- [3] J. A. Smith and F. B. Whitehall, *Optimizing Quality in Electronics Assembly: A Heretical Approach*. New York: McGraw-Hill, 1996.
- [4] A. Kusiak, *Computational Intelligence in Design and Manufacturing*. New York: Wiley, 2000.
- [5] M. J. A. Berry and G. Linoff, *Data Mining Techniques: For Marketing, Sales, and Customer Support*. New York: Wiley, 1997.
- [6] R. Groth, *Data Mining: A Hands-on Approach for Business Professionals*. Englewood Cliffs, NJ: Prentice-Hall, 1998.
- [7] A. Kusiak, "Decomposition in data mining: An industrial case study," *IEEE Trans. Electron. Packag. Manufact.*, vol. 23, pp. 345–353, Oct. 2000.
- [8] J. R. Quinlan, "Induction of decision trees," *Machine Learning*, vol. 1, no. 1, pp. 81–106, 1986.
- [9] R. S. Michalski, I. Mozetic, J. Hong, and N. Lavrac, "The multi-purpose incremental learning system AQ15 and its testing application to three medical domains," in *Proc. 5th Nat'l. Conf. Artificial Intelligence*. Palo Alto, CA: AAAI Press, 1986, pp. 1041–1045.
- [10] P. Domingos and M. Pazzani, "Beyond independence: Conditions for the optimality of the simple Bayesian classifier," in *Proc. 13th Int. Conf. Machine Learning*, 1996, pp. 105–112.
- [11] R. Kohavi, "Wrappers for Performance Enhancement and Oblivious Decision Graphs," Ph.D. dissertation, Comput. Sci. Depart., Stanford Univ., Stanford, CA, 1995.
- [12] J. Friedman, Y. Yun, and R. Kohavi, "Lazy decision trees," in *Proc. 13th Nat'l. Conf. Artificial Intelligence*, 1996.
- [13] J. R. Quinlan, *C4.5: Programs for Machine Learning*. Los Altos, CA: Morgan Kaufmann, 1993.
- [14] P. Clark and R. Boswell, "The CN2 induction algorithm," *Machine Learning*, vol. 3, no. 4, pp. 261–283, 1989.
- [15] D. W. Aha, "Tolerating noisy, irrelevant and novel attributes in instance-based learning algorithms," *Int. J. Man-Machine Studies*, vol. 36, no. 2, pp. 267–287, 1992.
- [16] S. K. Murthy and S. Salzberg, "A system for the induction of oblique decision trees," *J. Artif. Intell. Res.*, vol. 2, no. 1, pp. 1–33, 1994.
- [17] P. Auer, R. Holte, and W. Maass, "Theory and application of agnostic PAC-learning with small decision trees," in *Proc. 8th Eur. Conf. Machine Learning, ECML-95*, A. Prieditis and S. Russell, Eds., 1995.
- [18] J. W. Grzymala-Busse, "A new version of the rule induction system LERS," *Fundamenta Informaticae*, vol. 31, pp. 27–39, 1997.
- [19] T. Y. Lin and N. Cercone, Eds., *Rough Sets and Data Mining*. Boston, MA: Kluwer, 2000.
- [20] J. G. Carbonell, Ed., *Machine Learning: Paradigms and Methods*. Cambridge, MA: MIT Press, 1990.
- [21] R. S. Michalski, I. Bratko, and M. Kubat, Eds., *Machine Learning and Data Mining*. New York: Wiley, 1998.
- [22] T. Mitchell, *Machine Learning*. New York: McGraw-Hill, 1997.
- [23] P. Langley and H. A. Simon, "Applications of machine learning and rule induction," *Commun. ACM*, vol. 38, no. 11, pp. 55–64, 1995.
- [24] Z. Pawlak, "Rough sets," *Int. J. Inform. Comput. Sci.*, vol. 11, no. 5, pp. 341–356, 1982.
- [25] M. Stone, "Cross-validated choice and assessment of statistical predictions," *J. Royal Statist. Soc.*, vol. 36, pp. 11–147, 1974.
- [26] A. Kusiak, J. A. Kern, K. H. Kernstine, and T. L. Tseng, "Autonomous decision-making: A data mining approach," *IEEE Trans. Inform. Technol. Biomed.*, vol. 4, no. 4, pp. 274–284, 2000.

Visibility-Based Pursuit-Evasion: The Case of Curved Environments

Steven M. LaValle and John E. Hinrichsen

Abstract—We consider the problem of visually searching for an unpredictable target that can move arbitrarily fast in a simply-connected two-dimensional curved environment. A complete algorithm is presented and is based on critical visibility events that occur because of inflections and bitangents on the environment boundary. By generalizing the notion of inflections and bitangents to polygonal and piecewise-smooth environments, the approach is considered as a step toward developing pursuit-evasion strategies that have little dependency on the representation of the environment.

Index Terms—Active sensing, computational geometry, mobile robotics, motion planning, visibility.

I. INTRODUCTION

Imagine entering a cave in complete darkness. You are given a lantern and asked to search for any people who might be moving about. Several questions might come to mind. Does a strategy even exist that guarantees I will find everyone? If not, then how many other searchers are needed before this task can be completed? Where should I move next? Can I keep from exploring the same places multiple times? This kind of scenario might apply to firepersons engaged in a rescue effort, law enforcement officials in a hostage situation, or soldiers attempting to secure a potentially hostile area. Since it is always preferable to place robots at risk instead of humans, we might like to determine whether successful searching strategies can be computed automatically for a mobile robot. Such strategies can also provide valuable advice to people as they plan for high-risk operations.

Plenty of applications exist that could benefit from visibility-based pursuit-evasion strategies. They can be embedded in surveillance systems that use mobile robotics with various types of sensors (motion, thermal, cameras, etc.). Small mobile robots with pursuit-evasion strategies can be used by special forces in high-risk military operations to systematically search a building in enemy territory before it is declared safe for entry. In scenarios that involve multiple robots that have little or no communication, a pursuit-evasion strategy could be used to help one robot locate others. One robot could even try to locate another that is malfunctioning. For remote presence applications, it would be valuable if a robot can locate automatically other robots and people using sensors. Beyond robotics, software tools can be developed that assist people in many applications that involve systematically searching or covering complicated environments. Relevant pursuit-evasion scenarios can be imagined in law enforcement, search-and-rescue, toxic cleanup, and in the architectural design of secure buildings. One limitation of our current work, however, is that the application must provide a complete representation of the environment.

Manuscript received August 10, 2000. This paper was recommended for publication by Associate Editor E. Pagello and Editor I. Walker upon evaluation of the reviewers' comments. This work was supported in part by the National Science Foundation under Grant IRI-9875304 (LaValle, CAREER Award). This paper was presented in part at the IEEE International Conference on Robotics and Automation, Detroit, MI, 1999.

S. M. LaValle is with the Department of Computer Science, Iowa State University, Ames, IA 50011 USA (e-mail: lavalle@iastate.edu).

J. E. Hinrichsen is with the Department of Mathematical Sciences, Carnegie Mellon University, Pittsburgh, PA 15213 USA (e-mail: john4@andrew.cmu.edu).

Publisher Item Identifier S 1042-296X(01)04808-X.