A configuration assistant for versatile vision-based inspection systems

Olivier Hüsser, Heinz Hügli

Institute of Microtechnology, University of Neuchâtel, Switzerland

ABSTRACT

Nowadays, vision-based inspection systems are present in many stages of the industrial manufacturing process. Their versatility, which permits to accommodate a broad range of inspection requirements, is however limited by the time consuming system setup performed at each production change. This work aims at providing a configuration assistant that helps to speed up this system setup, considering the peculiarities of industrial vision systems. The pursued principle, which is to maximize the discriminating power of the features involved in the inspection decision, leads to an optimization problem based on a high dimensional objective function. Several objective functions based on various metrics are proposed, their optimization being performed with the help of various search heuristics such as genetic methods and simulated annealing methods. The experimental results obtained with an industrial inspection system are presented, considering the particular case of the visual inspection of markings found on top of molded integrated circuits. These results show the effectiveness of the presented objective functions and search methods, and validate the configuration assistant as well.

Keywords: Visual inspection, vision systems, automatic setup, system configuration, heuristic search

1. INTRODUCTION

In the automatic inspection field, advanced visual inspection systems perform a large and ever growing number of controls. These systems, however, require some tuning in order to perform efficiently on a specific task. Given the increasing complexity of the system, manual tuning becomes a major problem of an operator using such a system. Means for automatic tuning are required.

Automatic optimization tasks in the vision field are numerous. Some example tasks recently investigated are: contour tracking with *snakes*[1]; pattern recognition in association with *markov random field* representations[2]; image restoration[3]; geometric transformation estimation by *iterative closest point* methods[4]. Among the few papers related to visual inspection systems and their optimization, the paper by Daniel and West[5] describes a method for modeling shape variations of forged devices from samples, and its application to optimize the production process. In the present paper, the focus is set on the optimization of a visual inspection system in order for it to minimize its error rate.

A visual inspection system consists mainly of two parts : an image analysis and a decision. The image of the device to be inspected, X, is first analyzed with help of image analysis tools that produce a set of measurements **M**. This measurement vector is then classified by the decision module giving the final decision ω. Both steps may involve several parameters **P** that influence the output (decision) of the system.

In order to tune the system, human operators execute the following supervised learning procedure. First, a certain number of good and bad representatives is drafted among the devices to be inspected. Then the automatic inspection is performed and, depending on the system's decision, some parameters might be modified. Inspections and modifications are performed until experimental decisions match the a priori good/bad partition.

This setup process is not easy and can be time consuming. Furthermore it is approximate, in the sense that no hint on the confidence of decisions is available. In order to remedy these shortcomings, we developed a configuration assistant that is presented in this paper. The configuration assistant is designed to perform the parametric tuning automatically. The pursued idea is not only to enhance the decision module, as it is often the case, but also to maximize the discriminating power of the measurements.

The performance of the visual inspection must be formulated in a way that permits a comparison between the expected and experimental results. Somehow, this comparison must be quantifiable and the resulting quantity be pertinent. We call this quantity the **objective function F**. This scalar function over the parameters of the system should trustfully represent the similarity of the results, must be reliable and informative altogether. Thus, in order to find the best configuration for the system the idea is to use the objective function to measure its performance and to use search heuristics for exploring the parametric space spanned by **P**. Practically, a "good enough" solution is sufficient.

This paper focuses on the design of a configuration assistant for an inspection system whose decision module consists of a so-called *and-classifier*. This type of decision is widespread in the industrial inspection field where checked devices have to satisfy multiple mandatory constraints.

The paper is organized as follows. Next section describes the inspection system. We present in section 3 some ways to define the objective function appropriate for an *and-classifier* and present the methods used to perform the supervised learning. Section 4 presents the particular case of the configuration assistant applied to IC marking inspection. Results obtained with real data are presented in section 5.

2. INSPECTION SYSTEM

Industrial visual inspection systems determine whether a produced device satisfies some requirements. Most of time, all of these requirements have to be satisfied in order for a device to be labeled as good. These requirements are said to be mandatory: a device must satisfy the first requirement *and* the second *and* … to be accepted. Hence the name of *and-classifier* used here to designate the decision stage.
Given the definition of figure 1 the requirements are formulated with respect to some measurements **M** and usually take the form of thresholds T_i. A convenient inspection system, has normalized measurements with high values for good devices. The decision rule of such a classifier is written as:

$$\omega = \begin{cases} \text{accepted} & \text{if } \mathrm{M_i} > T_i \ \forall\, \mathrm{i}, \mathrm{i} = 1,2,\ldots,\mathrm{n} \\ \text{rejected} & \text{otherwise} \end{cases}$$

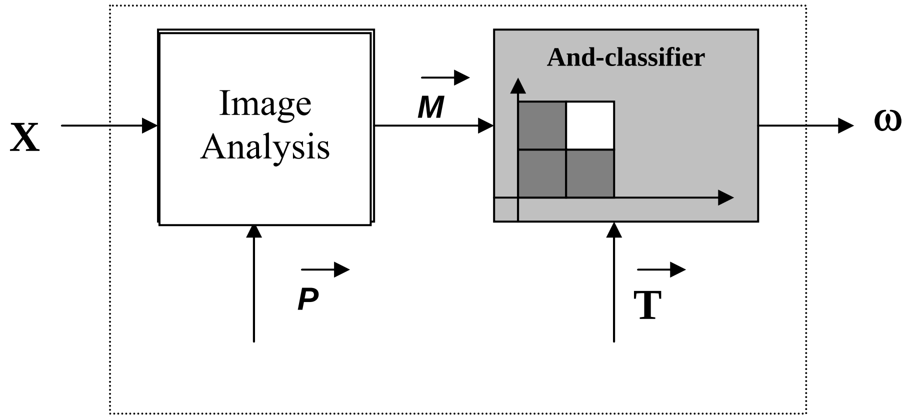

Figure 1: Structure of an inspection process

Given a set of devices represented by their digital image X={X_1, X_2, …, X_n }, and given an a priori classification of this set of devices by a human operator ω^*(X)={ $\omega^*(X_1)$, $\omega^*(X_2)$, …, $\omega^*(X_n)$} the learning set is defined as Ω_L={X, ω^*(X)}. The learning set is the union of two subsets: the good and the bad device's subsets. They are denoted here by Ω_x (good) and Ω_o (bad).
Given the learning set, the goal is to find parameters **P** and **T** such that expected inspection performance on future objects is optimized. Next section presents objective functions as a measure for the expected performance as well as search methods for finding the optimum.

3. OBJECTIVE FUNCTIONS AND SEARCH METHODS

This section presents objective functions F used to measure the quality of the parameter's configuration in a system whose decision stage is an *and-classifier*. It then introduces the search methods used to find the maximum of F, i.e. to find the best parametric configuration according to F.

3.1. Objective Functions

To optimize the expected system's performance in a general context, the representative learning set will be used. The goal is then to optimize the system's performance over this learning set. An obvious way to evaluate this performance is the

recognition rate, R, expressed as the fraction of correct decisions. It is normalized and it takes its maximum value if and only if all devices of the learning set are correctly classified. Although R represents the aptitude of the system on the learning set, it is not a good function for practical use. First because it is quantified on a small number of levels (=Card{Ω_L}) and is thus a rather rough function of the configuration. Furthermore it gives no hint on the confidence in the decision taken by the system. So the point is to replace a function of the learning set's partition F(ω) by a function of the learning set's measurements F({M}$_{\Omega L}$).
The function should evaluate the relevance of the measurements to make a decision. It could then be used with the help of parameters as a means to maximize the measurement's discriminating power. Interesting functions are the ones that are related to the distance between measurements and decision boundaries, greater distances being better.

3.1.1 Objective function in 1-dimensional space

Figure 2 displays for a 1-dimensional *and-classifier* the measurements M of a learning set of devices. The "X" symbols are the measurements of the Ω_x subset. Conversely, the "o" symbols are the measurements of the Ω_o subset. The vertical line denotes a fixed threshold T that has been preset. The objective function F is defined here as the minimum distance between a measurement M_i and the threshold T conditional to an absolutely correct classification:

$$F = \min_{X \in \Omega_L} \left| M_X - T \right| \cdot (R == 1).$$

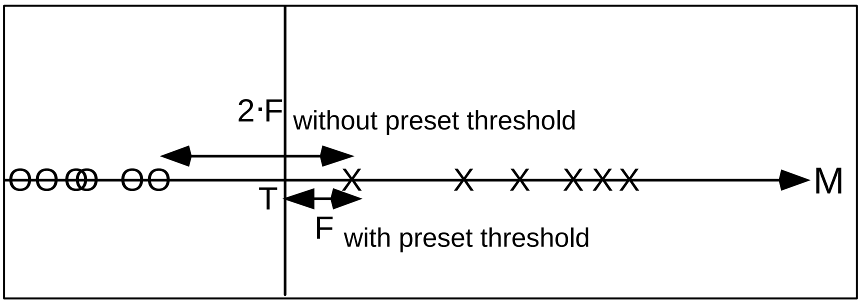

Figure 2: objective function for 1-dimensional measurement space

If the threshold is variable it can be optimized. The best position is at mid-distance of the closest pair of bad/good measurements. According to it, the objective function takes the form

$$F' = \frac{1}{2} \cdot \left(\min_{M_X \in \Omega_X} (M_X) - \max_{M_O \in \Omega_O} (M_O) \right) \cdot (R == 1).$$

3.1.2 Objective function in n-dimensional space

The objective function for multidimensional *and-classifiers* relies on the same concepts as for the 1-dimensional case. Some definitions must be given in order to define its variants.
Given the **parametric space** P spanned by the parameters to be set, we define the **acceptance subspace** A of an *and-classifier*. Acceptance space is the subspace of P where the inspection system considers the measurements as belonging to an acceptable device. In the complementary **rejection subspace** the system rejects the devices: $\overline{A} = P - A$. The partition in $\overline{A}$ and A is defined by the n thresholds.

The objective function should reflect how far the measurements are from a wrong classification, i.e. from the wrong, or opposite, subspace. The minimum distance is still the main concern and so, the following **objective function** F is defined based on above definition: *F is equal to the minimum distance, over the learning set, of a measurement to its opposite subspace*.
This can be either the distance of a good device to $\overline{A}$, or the one of a bad device to A:

$$F = \min \left[\min_{X_i \in \Omega o} \left\| X_i ; \overline{A} \right\| ; \min_{X_i \in \Omega_x} \left\| X_i ; A \right\| \right],$$

where "|| ||" denotes the minimum distance between a point and a subspace. "|| ||"=0 when the point belongs to the subspace.

The notion of distance (|| ||) is related to a metric. The distance functions take the following generic form for a L_n-metric: $\| \ \|_n = \left(\sum \Delta^n\right)^{\frac{1}{n}}$. In one-dimensional case all metrics are equivalent, and so the definition of F is unique. This is not the case in n-dimensional spaces where objective functions will take distinct forms depending on the chosen metric. Figure 3 shows in 2-dimensional space the equipotential lines of F defined with 3 different metrics: L_1, L_2, L_∞. It is visible that they only differ when multiple measurements compose the distance.

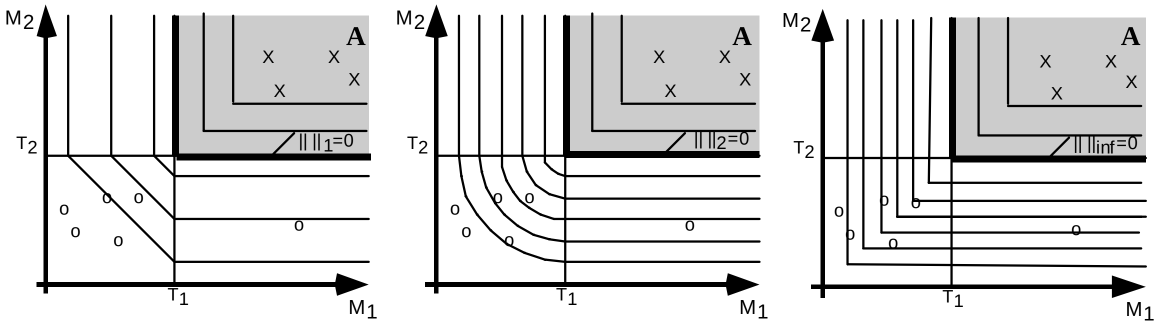

Figure3: Acceptance subspace & contour plot of objective functions based on L_1 , L_2 , L_∞

The choice of one particular metric system depends on high level considerations as it will be discussed in next section.

If the thresholds are variables they can be optimized according to the measurements of Ω_L. The resulting acceptance subspace must include all Ω_x measurements and exclude all Ω_o measurements. To satisfy this constraint and have a convenient way to calculate the objective function, we define here a minimum acceptance space A^*. Given the smallest measurements of the good subset $M_{ix}^{\min} = \min\limits_{X\in\Omega_x}\left[M_i(X)\right]$, A^* is the subspace whose points **P** satisfy $P_i \geq M_{ix}^{\min}\ \forall \mathrm{i}$. So, the F' objective function is defined in a similar way as in the 1-dimensional case as being equal to half of the minimum distance of a Ω_o measurement to A^*.

$$F' = \frac{1}{2}\cdot \min_{X_i\in\Omega o}\left\| X_i ; A^* \right\|$$

Figure 4 displays acceptance spaces for a bi-dimensional classifier. Left hand is an example with preset thresholds and right hand an example with minimum acceptance space.

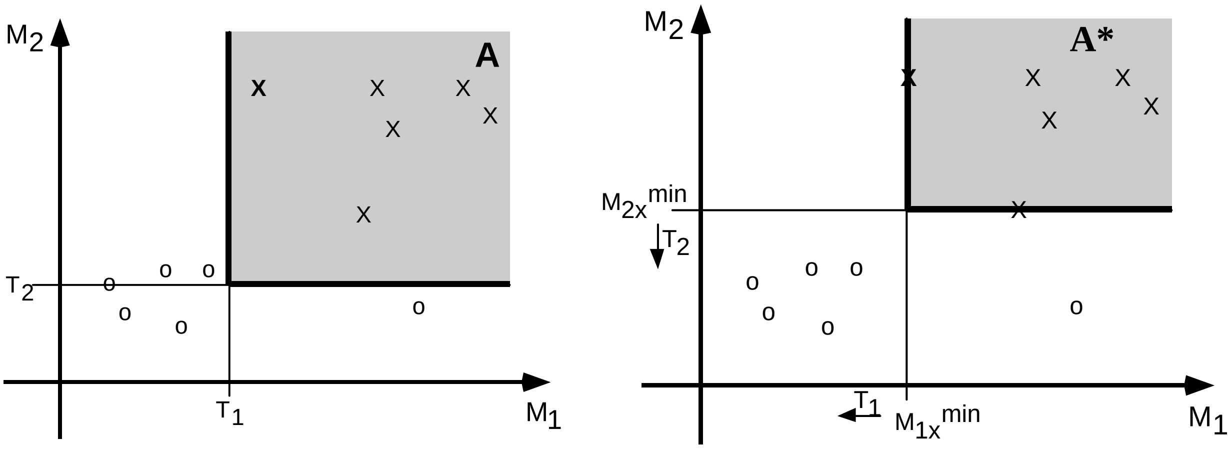

Figure 4: Acceptance subspaces with fix (left) or tunable thresholds (right)

Next table displays the formulae of 3 different objective functions in the cases of adjustable and fixed decision thresholds

F=	Fixed Thresholds	Configurable Thresholds
L_∞ metric (max)	$\min\left[\begin{array}{l}\min_{X_i\in\Omega_o}\left[\max_i\left[\max(0;T_i - M_i\ (X_i))\right]\right];\\ \min_{X_i\in\Omega_x}\left[\min_i\left[\max(0;M_i\ (X_i) - T_i)\right]\right]\end{array}\right]$	$0.5\cdot\min_{X_i\in\Omega_o}\left[\max_i\left[\max(0;M_{ix}^{\min} - M_i\ (X_i))\right]\right]$
L_1 metric (Manhattan)	$\min\left[\begin{array}{l}\min_{X_i\in\Omega_o}\left[\sum_i\left[\max(0;T_i - M_i\ (X_i))\right]\right];\\ \min_{X_i\in\Omega_x}\left[\min_i\left[\max(0;M_i\ (X_i) - T_i)\right]\right]\end{array}\right]$	$0.5\cdot\min_{X_i\in\Omega_o}\left[\sum_i\left[\max(0;M_{ix}^{\min} - M_i\ (X_i))\right]\right]$
L_2 metric (Euclidean)	$\min\left[\begin{array}{l}\min_{X_i\in\Omega_o}\left[\sqrt{\sum_i\left[\max(0;T_i - M_i\ (X_i))\right]^2}\right];\\ \min_{X_i\in\Omega_x}\left[\min_i\left[\max(0;M_i\ (X_i) - T_i)\right]\right]\end{array}\right]$	$0.5\cdot\min_{X_i\in\Omega_o}\left[\sqrt{\sum_i\left[\max(0;M_{ix}^{\min} - M_i\ (X_i))\right]^2}\right]$

3.2. Search Methods

Objective functions described before are defined over the measurements of the learning set. Measurements are depending on the parameters of the configuration. This dependency can take any form and is not restricted to be analytical. The number of parameters is not specified either. So, the goal is to maximize over the set of possible configurations an objective function F (**P**,**T**) of unknown shape. To do this we will use methods that are recognized as robust and efficient in such situations: *genetic algorithms* and *simulated annealing*. A third method called *enhanced random* is presented and considered.
One concern with the employed methods, is to avoid the replication of the problem to be solved. What cannot be tolerated, is a search method that would require some choices in order to function properly. It would just shift the parametric configuration problem to another level. Search methods must therefore be designed with versatility in mind in order for them to cope reasonably well with various tasks.

3.2.1 Genetic algorithm (GA)

For a general description of genetic algorithms we refer to existing documents [6, 7]. We will restrict ourselves to the description of the particular implementation of the *simple genetic algorithm* (SGA) used in this work. Its initial population is selected randomly over {**P**}. The evaluation, or fitness function, is the objective function previously discussed. The *roulette wheel* selection scheme is applied. A *non elitist* replacement scheme associated with a high mutation rate (P_{mut}=0.1) and an explorative *crossover* method prevent the GA's stagnation. We applied two crossover methods. First a standard multiple point crossover with a probability of 10% and more frequently (90%) a linear crossover scheme. The linear method creates two offspring located on the extended segment that connects both parents in {**P**} space. This process is illustrated in figure 5. The probability density is constant all along the segment.

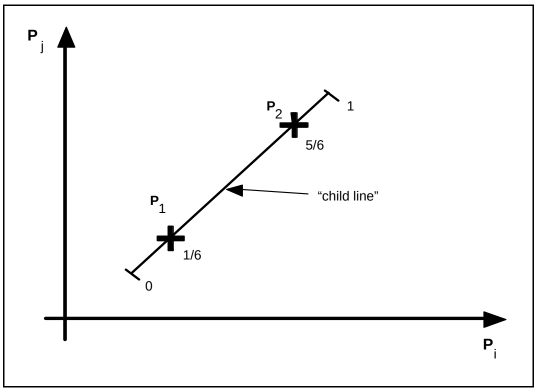

Figure 5: linear crossover process of the GA

Population size is constant and small: 20 individuals. This choice allows a large number of generations to take place until the time limit stopping criterion is reached.

3.2.2 Simulated annealing method (SA)

The name of this method comes from its relationship with physical systems and their equilibrium states. The configuration is considered here as a set of possible states for the inspection system. Associated with each state is a scalar value called *energy*. Here, the energy of a state **P** is $\varepsilon(\mathbf{P})=1-F(\mathbf{P})$. Transition from one state to another is given a probability that depends on their relative energy values and a parameter called *temperature*. So the algorithm performs at each step such a transition between two states (configurations). This process is called **stochastic relaxation** [3]. The chosen algorithm works with a binary temperature schedule, alternating T=0 and T=∞. Figure 6 shows the flow diagram of the SA method: Starting from a randomly selected configuration (T=∞), **gradient descent** (T=0) is performed until a local minimum is reached; then if the stopping criterion is not satisfied, the process starts over.

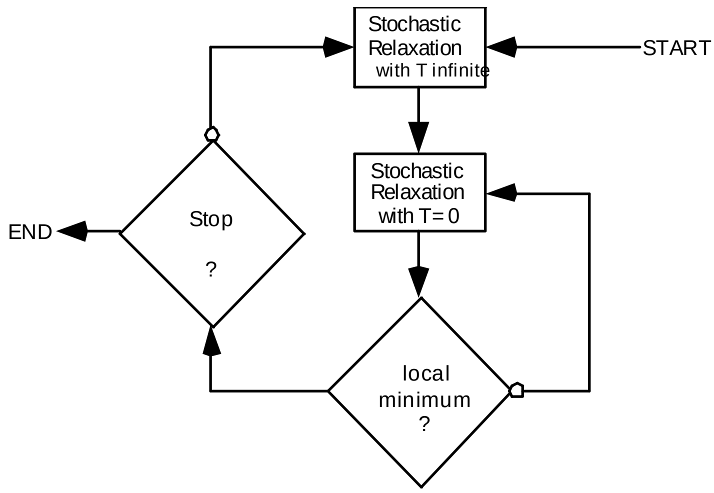

Figure 6:Temperature schedule of SA

3.2.3 Enhanced random method (ER)

This method combining a random selection scheme with a probability density estimation is expected to be versatile and to perform well on any kind of objective function.

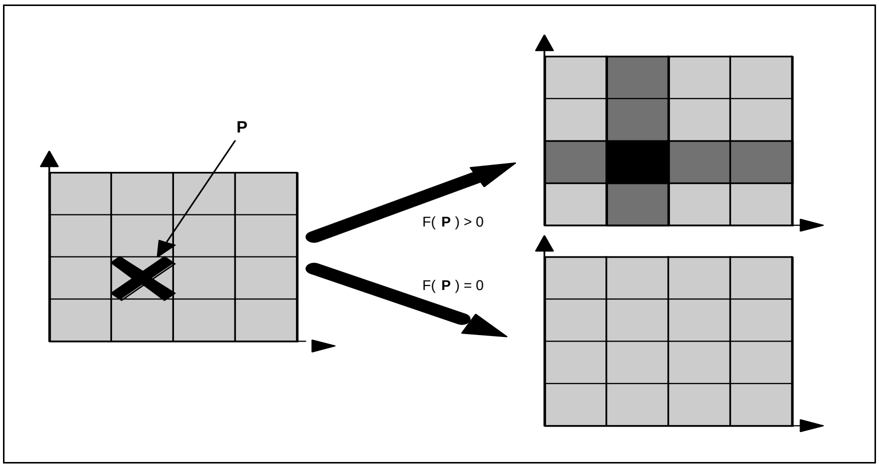

Figure 7: modification of $\rho(\mathbf{P})$

The principle of this algorithm is to use the past results to increase the probability of finding a good solution in the future. At the beginning, as no result is available, the first parametric vector **P** is chosen randomly over the whole parametric space of the permitted values specified by the user for the hidden configuration {**P**}. This random process is based on the assumption that all points $\mathbf{P}_i$ in {**P**} have the same probability to be a good solution and therefore to be chosen. In other words, the probability density ρ (**P**) is uniform over {**P**}. Then, to enhance the efficiency, the probability density is modified according to the results already obtained. This process is quite similar to the *Parzen windows*[8] used to approximate an unknown density function of a data set over a variable. The modification of ρ (**P**) is done for all single variables considered independently. This gives a star-shaped modification of ρ (**P**) with a maximum at center (see figure 7). Of course, a maximum limit has to be set for ρ (**P**) in order to prevent a possible automatic selection of good tested settings.

4. AN IC MARKING INSPECTION SYSTEM

The experimental part is performed on an industrial inspection system in the context of visual IC marking inspection. The inspection process checks the correctness of markings found on top of molded integrated circuit packages. These markings are either painted or laser burnt on the chip. The high throughput and limited precision of production have two consequences for this IC control. The chips markings will likely exhibit some variations, and the inspection system must be quick. An example of marking is shown in figure 8.

Figure 8: SOT device inspected in a production line

Decision to accept or reject a device for an eventual marking defect is based on two measurements : **contamination** M_1 and **missing character** M_2. These measurements are calculated relatively to a reference (golden unit). See figure 9. Given X and R the test and reference markings, given their intersection R∩X, we define the contamination pattern C=X–(R∩X) and the missing pattern M=R-(R∩X). With the associated surfaces S_X , S_R, S_C and S_M, the measurements are written:

$$M_1 = \frac{S_X - S_C}{S_X} \quad , \quad M_2 = \frac{S_R - S_M}{S_R}.$$

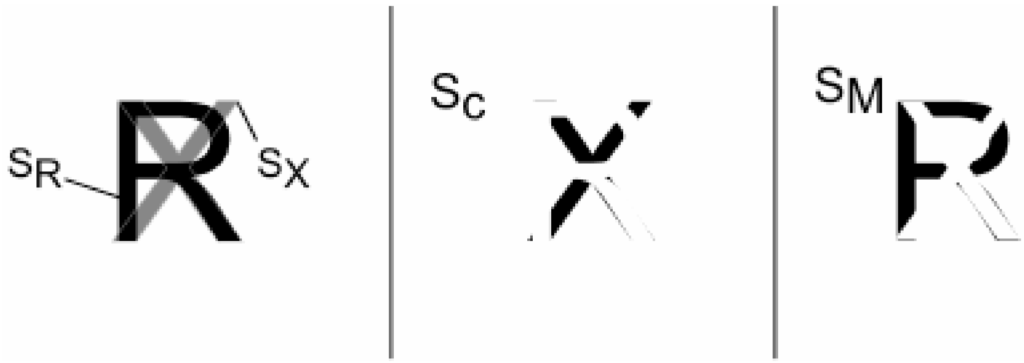

Figure 9: missing and contaminated areas ("R" is the reference)

Both measurements are normalized. *Contamination* represents the fraction of the inspected marking's surface that matches the reference. *Missing character* the fraction of the reference's surface that is not covered by inspected device. Of course, reference and test are optimally superposed during a preprocessing step.

These measurements are fast to compute and allow a high inspection throughput. They are versatile and apply to a broad spectrum of markings. Their disadvantage is that it is not obvious to set limit value for the acceptance of a marking. This led

us to consider measurements thresholds as belonging to the configuration of the system, and accordingly to set their values with the configuration assistant. A device's marking is rejected unless ($M_1 > T_1$) and ($M_2 > T_2$).
The configuration associated with these measurements has 6 parameters. With the two configurable decision thresholds, it makes an **8-dimensional** parametric space $\mathbf{P}=\{P_{1\text{-}6};T_{1\text{-}2}\}$.
Most of these parameters are concerned with morphological operations on the image of the inspected device. Of course, the dependence of the measurements on these parameters cannot be described analytically. Furthermore, the effects of one parameter depend on the other parameter's values. For example, P_1 and P_2 are offset values for the grayscale thresholds used when binarizing the reference and inspected images. The table below shows that the variation of P_1 and P_2 affects M_1 and M_2 in a way that is difficult to describe.

	S_X	S_R	S_M	S_C	M_1	M_2
P_1 ↗	-	↘	↘	↗	↘	?
P_2 ↗	↘	-	↗	↘	?	↘

↗ represents an increasing value and ↘ a decreasing value. ? denotes an unpredictable effect depending of the particular test image and reference. So, varying only these 2 parameters will already lead to totally unpredictable measurement changes. Here comes the configuration assistant!

We shall now select an objective function F for this task.
The decision thresholds are configurable so just a metric has to be chosen. The two metrics L_1 and L_∞ were considered. Experiments with the L_∞ metric led to disastrous results with only one measurement being still significant. Considering the contour plots of objective functions defined with L_1 and L_∞ metrics (fig.10) and the associated gradients, a possible explanation appears. If all measurements of Ω_o are located on the same side of the separation line S, the objective function based on L_∞ will increase the gap according to only one of the measurements, making probably the second one useless !
So, L_∞ was no more considered in this application and the chosen objective function was defined on the L_1 metric. See table of section 3.1 .

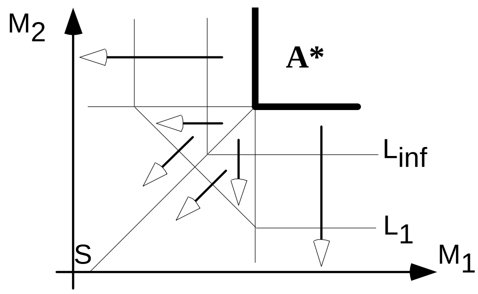

Figure 10:Gradients of the objective functions defined with L_1 and L_∞ metrics

5. EXPERIMENTS

Experiments were conducted with several data sets, varying number of configuration parameters. All the search methods were tested long enough to gather statistically significant experimental results.

5.1. Data

3 data sets are considered :

	"74X" data set	"SOT" data set	"Casablanca" data set
Origin	Lab	Lab	Production line
Type of defect	synthetic	synthetic	real
# of parameters	3+2	4+2	5+2
# of samples in Ω_x /Ω_o	10 / 10	21 / 36	20 / 20

In this table, "real" means effective defects found in production. Synthetic defects are voluntary defects such as: scratches, written over, false marking, white/black tape stripes pasted over the marking. The number of parameters refers to the parameters automatically adjusted by the configuration assistant. These parameters are given possible discrete values belonging to a range; typically 10-20 allowed values per parameter. The 2 additional parameters being the decision thresholds T_i.

Figure 11 shows two samples of bad devices.

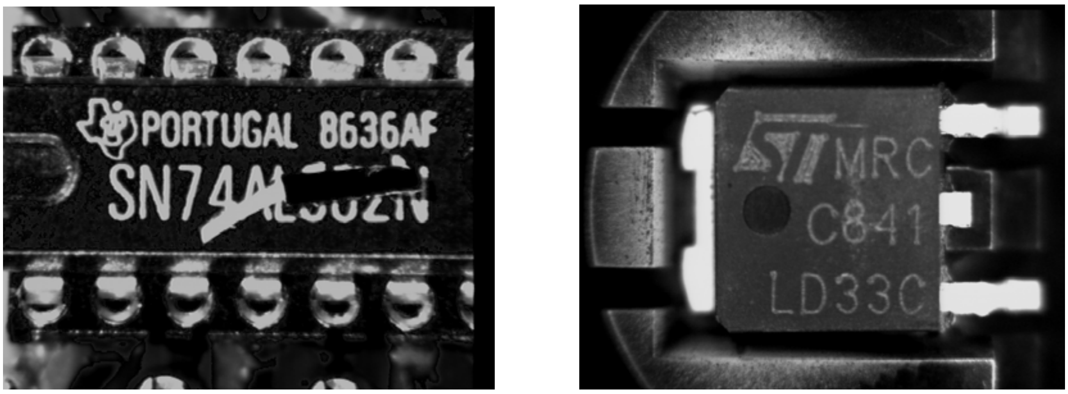

Figure 11: one synthetic (74x) and one real bad device

5.2. Example Objective Function

The objective function measured with the Casablanca data set is plotted in figure 12 with 2 varying parameters. It is typical of the functions that were encountered in this work: it exhibits the 3 following characteristics. First the function takes zero value over the greatest part of the space. Secondly, the places where it takes non-zero values mostly form some connected regions. Finally the objective function is rather "regular" over these interesting regions. Thus, it can be expected that intelligent search heuristics will take advantage of these regularities of the objective function.

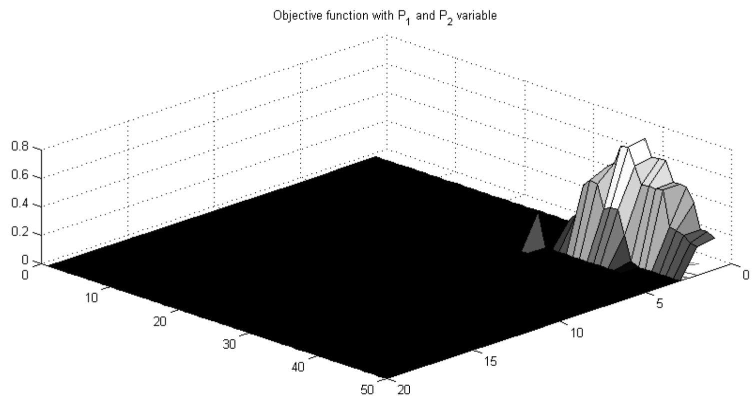

Figure 12: Objective function F for Casablanca data

5.3. Search Methods Comparison

First, an exhaustive search over the possible configurations was performed to gather the complete information about F(P). Of particular interest is the global maximum of F over {P} as it will be used as reference value to rate the search methods performances.

Then, the search algorithms were evaluated with this experimental objective function F(P): they search a setting P_{max} that maximizes F within a given number of trials. As the true maximum of F over {P} is known, the objective function can be normalized:

$$F' = \frac{F}{\max\left[F\left\{\vec{P}\right\}\right]}.$$

F' expresses how good relatively to the optimum, a proposed parametric setting is. **Objective score** S is defined here as the highest value of F' found at a given time of the system's configuration.
In order to compare the performance of the search methods, the mean objective score is plotted versus time (number of trials). A mean has to be used because of the intrinsic randomness of the used search methods, so only statistical results are significant. Figure 13 represents the objective score for the 3 search methods. Results of the pure random method are also displayed for comparison. The figure shows that intelligent search methods GA, SA outperform random selection. *Enhanced random* is a bit less efficient. This plot gives also an indication of the acceleration that a method brings for the configuration of the system. The configuration becomes fast nearly optimal. The speedup factor when compared with an exhaustive search of all possible configurations is important (about 100 for n=5). Typical speedup of SA or GA methods against random method are between 5-20 when approaching near optimal solutions.

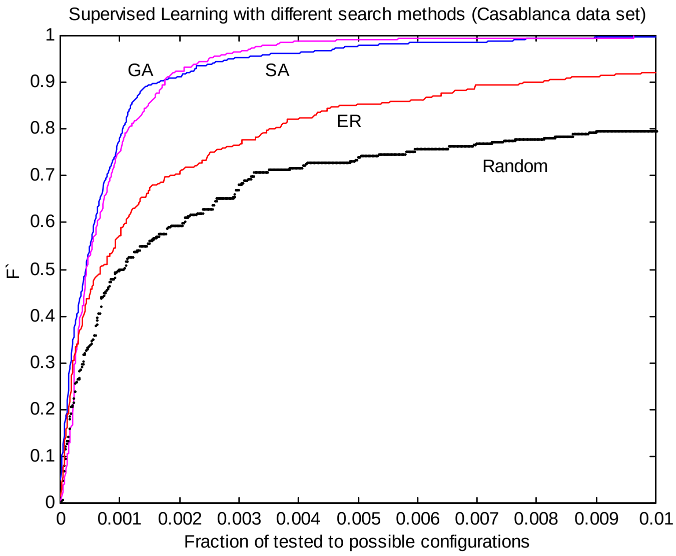

Figure 13: Mean objective scores of the search methods over 100 trials

The outstanding robustness of configurations obtained with the SA algorithm must be pointed out. This comes from its gradient descent scheme that increase its probability to propose settings whose neighborhoods are also good settings.

5.4. Evaluation of the configuration

Finally a check of the real quality of the found configuration is done by observing the measurements for the learning set.
The working configuration assistant will not of course make the exhaustive search, it will be given a time period to perform the setting and it must find the best configuration during this period. The search method used here, is *simulated annealing*. To test the reliability 100 trials were performed each with a time period of 5 minutes, corresponding to approximately 3000 marking inspections. These inspections represent only 0.15% of inspections required to perform an exhaustive search. According to the objective score, the median and worst settings are used on the learning set's devices. Left half of figure 14 shows measurements with median-setting and right half with worst-setting. Good and bad subsets are separated by a clear gap even with the worst setting.

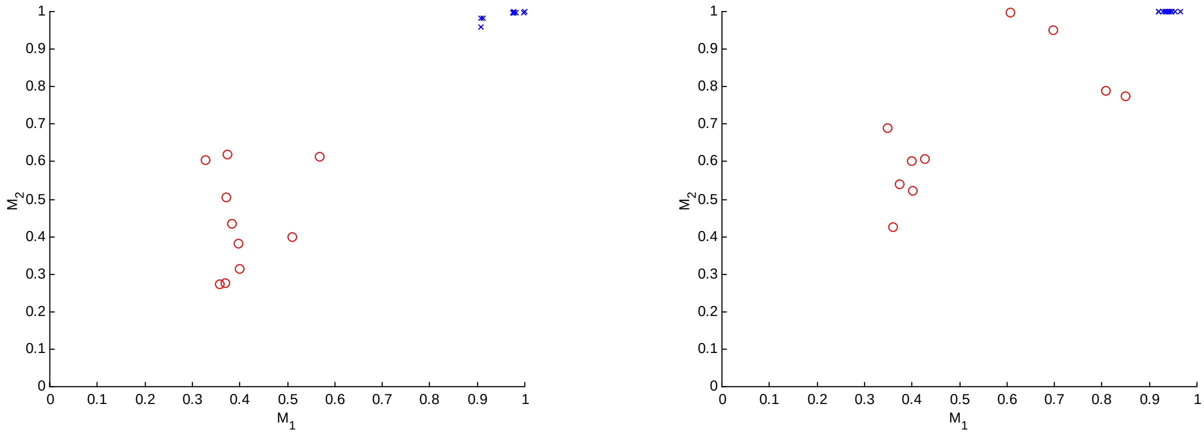

Figure 14: Measurements of the learning set with two different parameter's configuration

6. CONCLUSION

This paper presented a configuration assistant for industrial visual inspection systems. With the help of carefully chosen objective functions and robust search methods, automatization of *and-classifier* based systems was achieved. This automatization relieves the human operator of the complex setup of the inspection system and accelerates this process. The retained search methods, *genetic algorithms* and *simulated annealing*, showed to be typically much faster than a random search and drastically less time consuming than an exhaustive search.
Another advantage of the configuration assistant is the improvement in reliability. As the setup is performed with an objective function based on measurements instead of classification, a confidence value of the setup exists in the form of the objective score. This value leads to a more trustable setup and gives new information to the operator.

ACKNOWLEDGEMENTS

This research work was supported by the Swiss Commission for Technology and Innovation under project nr. 3460.1 . It was realized in collaboration with *Analytec SA*. Particular thanks go to Dr. Bernd Sommer, Dr Bernhard Schwab and Dr. Max Meyer.

REFERENCES

1. N.Ueda, K.Mase, *Tracking Moving Contours using Energy-minimizing Elastic Contour Models*, International Journal of Pattern Recognition and Artificial Intelligence, volume 9, no 3, pp 465-484, 1995
2. N.S.Friedland, A. Rosenfeld, *Compact Object Recognition using Energy-Function-based Optimization*, IEEE Transactions on Pattern Analysis & Machine Intelligence (PAMI), volume 14, no 7, pp770-777, 1992.
3. S.Geman, D.Geman, *Stochastic Relaxation, Gibbs Distribution, and the Bayesian Restoration of Images*, IEEE PAMI, November 1984
4. P.J. Besl and N.D. McKay, *A Method for Registration for 3-D Shapes*, Proceedings of IEEE Transactions on Pattern Analysis & Machine Intelligence (PAMI), volume 14, no 2, pp239-256, 1992.
5. B.T.Daniel, M.Cardew-Hall, *Geometric Variations : Analysis, Optimization and Control*, International Journal of Pattern Recognition and Artificial Intelligence, volume 11, no 1, pp 79-113, 1997
6. D.E.Goldberg, *Genetic Algorithms in Search, Optimization and Machine Learning*, Addislon Wesley, Reading Massachussets, 1989
7. M.Srinivas, *Genetic Algorithms : a survey*, IEEE Computer, June 1994
8. R.O.Duda, P.E.Hart, *Pattern Classification and Scene Analysis*, pp 88-95, John Wiley & Sons, Menlo Park, California, 1973