scch
software competence center
hagenberg

FUZZY.LOGIC.LAB.LINZ

Advances in Knowledge-Based Technologies

Proceedings of the
Master and PhD Seminar
Winter term 2006/07, part 1

Softwarepark Hagenberg
SCCH, Room 0/10
November 15, 2006

Software Competence Center Hagenberg
Hauptstrasse 99
A-4232 Hagenberg
Tel. +43 7236 3343 800
Fax +43 7236 3343 888
www.scch.at

Fuzzy Logic Laboratorium Linz
Hauptstrasse 99
A-4232 Hagenberg
Tel. +43 7236 3343 431
Fax +43 7236 3343 434
www.flll.jku.at

Program

9:00–10:00 Session 1 ***(Chair: Roland Richter)***

9:00	Bernhard Moser: *Compactness of admissible transformations of fuzzy partitions*
9:30	Xiaowei Zhou: *Evolving Fuzzy Modelling from Real-time data streams*

10:00 Coffee Break

10:15–12:15 Session 2 ***(Chair: Bernhard Moser)***

10:15	Leila Muresan: *Analysis of large microarray images*
10:45	Frank Bauer: *An Alternative Approach to Parallel MRI*
11:15	Thomas Hoch: *Stochastic resonance and energy efficient information processing in single neurons*
11:45	Thomas Natschläger: *Large-scale simulation of neuronal micro-circuits*

12:15–13:30 Lunch

13:30–15:00 Session 3 ***(Chair: Roland Richter)***

13:30	Carlos González Morcillo: *A Multi-Agent Approach to 3D Rendering Optimization*
14:00	Volkmar Wieser: *High performance surface inspection method for thin-film sensors*
14:30	Edwin Lughofer: *Object Recognition in Deviation Images for Fault Detection*

Technical Report
SCCH–TR–0635

On the Compactness of Admissible Transformations of Fuzzy Partitions

Bernhard Moser
Software Competence Center Hagenberg
A-4232 Hagenberg, Austria
bernhard.moser@scch.at

Abstract — This paper analyzes regularization constraints in terms of fuzzy logical concepts for transforming families of fuzzy sets in order to guarantee stable and linguistically interpretable solutions of fine tuning fuzzy systems. For this purpose, transformations for fuzzy sets are introduced which are constructed by means of the compositional rule of inference. It is investigated, under which conditions on these transformations various characteristics of fuzzy partitions and fuzzy sets like redundancy, convexity and topological aspects keep invariant. However, further restrictions like the compactness of the set of admissible transformations are required in order to guarantee stable solutions. It is pointed out that compactness in the uniform sense can be characterized in terms of purely fuzzy logical concepts that is by means of fuzzy equivalence relations.

Key words — *Fuzzy partition, compositional rule of inference, fine tuning, pseudo-inverse, regularization, T-equivalence, sequential compactness*

Kompetenzzentren-Programm

1 Introduction

Fine tuning fuzzy rule bases can be looked at as a sequence of transformations of the fuzzy partitions involved. In order to avoid the destruction of the meaning of the original fuzzy sets the goal is to define admissible transformations, which preserve certain properties of fuzzy sets like convexity or the ordering between the fuzzy sets. Analogously, the question is of interest which characteristics of a fuzzy partition should keep invariant. For instance, if the original family $\{A_1, \ldots, A_n\}$ of fuzzy subsets of $\mathcal{X}$ is a Ruspini partition, is this also true for the transformed family $\{B_1, \ldots, B_n\}$, that is, does $\sum_i A_i = 1$ imply $\sum_i B_i = 1$? As a fundamental concept in this paper we consider the redundancy of a family of fuzzy subsets introduced in [28]. The redundancy reflects, to which degree one fuzzy set (as a member of the family considered) is contained in the union of the others. Thereby, the redundancy of a family of Cantorian sets satisfying the covering and the disjointness conditions of a partition becomes zero. Redundancy is therefore a criterion for classifying families of fuzzy sets as fuzzy partitions. A high degree of redundancy tells us, that the family of fuzzy sets considered does not match the intuitive notion of a fuzzy partition very well. For this reason, we postulate that tuning of fuzzy rule bases must not increase the degree of redundancy of a transformed fuzzy partition. In this paper transformations of fuzzy sets are proposed which do not change the degree of redundancy, when applied to a family of fuzzy sets. However, it turns out that redundancy is a rather weak criterion which cannot prevent the ill-posedness of fine tuning a fuzzy system in general. It will be shown that the extensionality with respect to a fuzzy equivalence relation is a further necesaary criterion which prohibits that an originally proper fuzzy set mutates to a Cantorian crisp set.

After recalling concepts for families of fuzzy subsets aiming at extending the notion of a partition in the classical sense to fuzzy sets in Section 2. The goal pursued is to impose reasonable constraints on the set of admissible transformations in order to preserve various invariants which are expected from a fuzzy partition. Such invariants are convexity and upper semi-continuity of the members, the redundancy and the so-called **FP**-property as relational constraint between the members of the family of fuzzy subsets under consideration.

Finally, the starting point of Section 3 is an example from approximation theory which shows that the criteria introduced so far are not sufficient to assure a numerically stable solution. It is shown that the transformed fuzzy sets necessarily have to meet a compatibility condition with respect to a fuzzy equivalence relation which is shown to be a topological condition on the set of transformation functions in terms of compactness.

2 Disjointness and covering criteria for fuzzy partitions

In literature one can find various notions of a fuzzy partition, see, e.g., [11,14,25,28,32] The motivation for the various concepts usually depend on the application context like clustering, fuzzy control or modeling input-output relations with fuzzy systems in order to get a linguistically interpretable system rather than a numerical representation of the relations, see, e.g., [1–5,7,13,15–18, 22,26]. However, a new aspect comes in when analyzing the stability conditions under which the tuning process or – if executed automatically – the optimization and approximation process leads to a well defined uniquely solvable problem as analyzed by [8,9].

In this paper, we focus on the transforming process of fuzzy partitions as studied in [9] in the context of fuzzy control from the point of view of approximation theory. As a main result of this investigations it comes out that without regularizing the optimization or approximation process the problem to fit automatically an optimal fuzzy partition is usually not well-posed. Therefore, we study sets of fuzzy partitions rather than a single fuzzy partition and investigate structural conditions on this set which are required from the point of view of stability analysis. As a central property we will concentrate on the so-called **FP**-property introduced by Höhle [23] and Kruse et al. [29] which is closely related to T-equivalence relations.

2.1 FP-property

In what follows, we introduce and recall the basic notions in the framework of T-equivalence relations which are needed later on. By definition a triangular norm, briefly t-norm, $T : [0,1]\times[0,1] \to [0,1]$ is a commutative, associative, monotone operation with 1 as neutral element (see [19–21,27] for further details). If there is a number $a \in]0,1[$ such that $T(a,a) < a$ then T is called Archimedean. Continuous Archimedean t-norms play a special role as they can be characterized by generating functions, see [30]. More precisely, T is a continuous Archimedean t-norm if, and only if, there is a continuous, strictly decreasing function (additive generator) $f : [0,1] \to [0,\infty]$ with $f(1) = 0$ such that for $x, y \in [0,1]$

$$T(x,y) = f^{-1}(\min\{f(x) + f(y), f(0)\}). \tag{1}$$

If $f(0) = \infty$ the Archimedean t-norm is called *strict* otherwise *non-strict*.

A many valued model of an implication is provided by the so-called residuum given by

$$\overrightarrow{T}(a,b) = \sup\{c \in [0,1] | T(a,c) \le b\} \tag{2}$$

where T is a left-continuous t-norm. For details, e.g., see [20,27]. Consequently, the operator

$$\overleftrightarrow{T}(a,b) = \min\left\{\overrightarrow{T}(a,b), \overrightarrow{T}(b,a)\right\} \tag{3}$$

models a biimplication. As was proven in [31] only non-strict Archimedean t-norms induce continuous biimplications, see Appendix 5.1:

Proposition 1. *For a continuous t-norm T the residual biimplication $\overleftrightarrow{T}$ is continuous if, and only if, T is non-strict Archimedean.*

In some sense, T-equivalence relations are a generalization of biimplications. By definition a function $E : X^2 \longrightarrow [0,1]$ is called a *T-equivalence relation* with respect to the t-norm T if it is reflexive, symmetric and T-transitive that is

$$T(E(x,y), E(y,z)) \le E(x,z). \tag{4}$$

Given a T-equivalence relation which models to which extend the elements of the universe of discourse cannot be distinguished, it is natural to require from a fuzzy subsets A to be compatible with E in the sense that "If $x \in A$ and x cannot be distingished from y then $y \in A$", that is

$$\sup_{x\in\mathcal{X}} T(A(x), E(x,y)) \le A(y). \tag{5}$$

If A satisfies the inequality (5), the fuzzy subset A is said to be *extensional* with respect to E. We want to point out two theorems concerning T-equivalence relation which are of interest later on:

Theorem 2. *Let T be a continuous t-norm. Further, let $\{A_i\}_{i\in I}$ be a family fo fuzzy subsets of $\mathcal{X}$. Then, the relation $E : \mathcal{X} \times \mathcal{X} \to [0,1]$ given by*

$$E(x,y) = \inf_{i_I}\{\overleftrightarrow{T}(A_i(x), A_i(y))\}$$

is the biggest T-equivalence relation for which all A_i are extensional.

Theorem 3. *Let T be a continuous t-norm. Further, let $\{A_i\}_{i\in I}$ be a family fo fuzzy subsets of $\mathcal{X}$ and $x_{\in}\mathcal{X}$ such that $A_i(x_i) = 1$ for all $i \in I$. Then, following two assertions are equivalent*

- *(i) There is a T-equivalence $E : \mathcal{X} \times \mathcal{X} \to [0,1]$ such that $A_i(.) = E(x_{,}.)$ for all $i \in I$*
- *(ii) for all $i, j \in I$ there holds*

$$\sup_{x\in X}\{T(A_i(x), A_j(x))\} \leq \inf_{x\in\mathcal{X}}\{\overleftrightarrow{T}(a,b)(A_i(x), A_j(x))\} \tag{6}$$

Inequality (6) can be interpreted in the sense that if A_i and A_j overlap then they are equal. For further details on T-equivalence relations we refer to [6, 10, 12, 23, 24, 26, 27]

2.2 T-Redundancy

In [28] we have introduced the concept of T-redundancy as a fundamental criterion for classifying families of fuzzy subsets over a universe $\mathcal{X}$ as fuzzy partitions. The T-redundancy measures the degree, to which some element of a family of fuzzy subsets is included in the union of the others, where the set operations for inclusion and union are related to a left-continuous t-norm T. Since the logical counterpart of the subset relation is the implication, we use a many-valued implication, that is, the residuum

$$\overrightarrow{T}(a,b) = \sup\{c \in [0,1] |\, T(a,c) \leq b\}, \tag{7}$$

to describe the mentioned concept of redundancy:

Definition 4. Let $\Gamma = \{A_1, \ldots, A_n\}$ be a family of fuzzy subsets of the universe $\mathcal{X}$; T a left-continuous t-norm and S_T the induced t-conorm, that is, $S_T(a,b) = 1 - T(1-a, 1-b)$, then

$$\mathbf{Red}_T(\Gamma) = \max_{i=1}^{n}[\inf_{x\in\mathcal{X}} \overrightarrow{T}(A_i(x), S_{T,i\neq j}(A_j(x)))]$$

is called the T-redundancy of the family Γ.

It is quite natural to interpret the value $Red_T(\Gamma)$ as the quasi-truth value of the statement "some A_i is a subset of the union of the others". Obviously, redundancy is a phenomenon for families of proper fuzzy subsets of $\mathcal{X}$, i.e., $Red_T(\Gamma) = 0$ for partitions of $\mathcal{X}$ consisting of crisp subsets only. As an example, we present the explicit formula for the Łukasiewicz t-norm T_L of an arbitrary family $\Gamma = \{A_1, \ldots, A_n\}$:

$$\mathbf{Red}_{T_L}(\Gamma) = \max_{i=1}^{n}[\inf_{x\in\mathcal{X}}\{\min(1 - A_i(x) + \sum_{k\neq i} A_k(x), 1)\}].$$

The next proposition characterizes the extremal case, when the redundancy equals one. The proof can be found in [28] and in the Appendix, see 5.2.

Proposition 5. *Let $\Gamma = \{A_1, \ldots, A_n\}$ be a family of fuzzy subsets of $\mathcal{X}$ and T be an arbitrary left-continuous t-norm. Then, there holds* $\mathbf{Red}_T(\Gamma) = 1$ *if, and only if, for some $i \in \{1, \ldots, n\}$ the fuzzy subset A_i is entirely included in the union of the others, i.e., $A_i(x) \leq S_{Tj\neq i}(A_j(x))$ for all $x \in \mathcal{X}$.*

3 Compactness of admissible transformations in the framework of T-equivalence relations

As pointed out in the last sections, the transformations $\mathcal{T} : \mathcal{F}(I) \to \mathcal{F}(I)$, $\mathcal{T}(A) = R_t^{(c)} \bullet A$ induced by the compositional rule of inference as consider with strictly monotone functions t keep characteristic properties of fuzzy sets and fuzzy partitions like convexity, normality, continuity aspects and the T-redundancy invariant. For strictly monotonic functions t we have

$$\mathcal{T}(A) = R_t^{(c)} \bullet A = A \circ t^{-1}.$$

Note, that $g = t^{-1}$ is a (not necessarily strict) monotonic function. Any monotonically increasing function g induces a strictly increasing transformation t such that $R_t^{(c)} \bullet A = A \circ g$. Therefore, from now on we consider admissible transformations as induced by monotonically increasing functions g via the usual composition of functions, i.e.,

$$\mathcal{T}(A) = A \circ g.$$

Nethertheless, analysis from the point of view of approximation theory shows that these conditions do not prevent for example data-driven constructions of fuzzy controllers from being ill-posed [8, 9].

Generally, let denote

$$\mathcal{L}(\Gamma, \vec{\alpha}) : \mathcal{G} \times \mathcal{P} \to \mathbb{R}_0^+$$

a continuous loss function over the product space of a set $\mathcal{G}$ of admissible fuzzy partitions Γ, and a set $\mathcal{P}$ of admissible parameters $\vec{\alpha}$. Then, as it is well known from functional analysis the minimization problem has a solution if the product space $\mathcal{G} \times \mathcal{P}$ is compact in the uniform sense. For example, consider $\mathcal{L}$ to be a least squares problem with a regularization term as studied in [9] in order to find the optimal fuzzy partition and the parameters of a Sugeno controller to approximate given input-output data. In this Section, we, therefore, concentrate on the compactness property of a set of fuzzy partitions and how it can be interpreted in terms of fuzzy logical concepts. The following theorem gives an answer to this question.

Theorem 6. *Let $\{A_1, \ldots, A_n\}$ be a family of normal, convex and continuous fuzzy subsets of a compact set $\mathcal{X} \subset \mathbb{R}^d$, $d \in \mathbb{N}$ satisfying the* **FP**-*property (6) with respect to the non-strict Archimedean t-norm T. further, let $(\Gamma, \|.\|_\infty)$ be a linear normed subspace of continuous and monotonically increasing functions $g : \mathcal{X} \to \mathcal{X}$ endowed with the supremum norm $\|.\|_\infty$ and containing the identity* **id**, $\mathbf{id}(x) = x$. *Then, the following two assertions are equivalent*

(i) $\mathcal{A}_i = \{A_i \circ g \mid g \in \Gamma\}$ is sequentially compact with respect to $\|.\|_\infty$ for all $i \in \{1, \ldots, n\}$

(ii) there is a continuous T-equivalence relation E such that for any $g \in \Gamma$ the transformed fuzzy sets $A_1 \circ g, \ldots, A_n \circ g$ are extensional with respect to E .

Proof. Let us start by assuming (i). Theorem 3 and Proposition 1 the **FP**-property (6) and the continuity assumption of A_i, $i \in \{1, \ldots, n\}$, guarantee that

$$E(x, y) = \inf_{1 \leq i \leq n} \overset{\leftrightarrow}{T}(A_i(x), A_i(y)) \tag{8}$$

is a continuous T-equivalence relation for which all fuzzy sets A_i, $i \in \{1, \ldots, n\}$, are extensional.

Next, we show that the relation $\tilde{E} : \mathcal{X} \times \mathcal{X} \to [0, 1]$ defined by

$$\tilde{E}(x, y) = \inf_{g \in \Gamma} E(g(x), g(y)) \tag{9}$$

is a continuous T-equivalence relation for which all $A_i \circ g$, $i \in \{1, \ldots, n\}$, are extensional. The relation (9) is indeed a T-equivalence relation, because yb construction it is reflexive and symmetric. The T-transitivity can be seen as follows: Choose arbitrary elements $x, y, z \in \mathcal{X}$. From the sequential compactness of $\mathcal{A}_i$ and the continuity of E it follows that there exists a $g_0 \in \Gamma$ such that

$$\tilde{E}(x, z) = E(g_0(x), g_0(z)) = \inf_{g \in \Gamma} E(g(x), g(z)).$$

Now, the monotonicity of the t-norm T leads to

$$\begin{aligned}
T(\tilde{E}(x, y), \tilde{E}(y, z)) &= T(\inf_{g \in \Gamma} E(g(x), g(y)), \inf_{g \in \Gamma} E(g(y), g(z))) \\
&\leq T(E(g_0(x), g_0(y)), E(g_0(y), g_0(z))) \\
&\leq E(g_0(x), g_0(z)) \\
&= \tilde{E}(x, z)
\end{aligned}$$

which proves the T-transitivity of $\tilde{E}$. Since the fuzzy subsets A_i, $i \in \{1, \ldots, n\}$, are extensional with respect to E we obtain for an arbitrarily chosen $g \in \Gamma$

$$\begin{aligned}
T((A_i \circ g)(x), \tilde{E}(x, y)) &= \sup_{x \in \mathcal{X}} T(A_i(g(x)), \inf_{\tilde{g} \in \Gamma} E(\tilde{g}(x), \tilde{g}(y))) \\
&\leq \sup_{x \in \mathcal{X}} T(A_i(g(x)), E(g(x), g(y))) \\
&\leq (A_i \circ g)(y)
\end{aligned}$$

which demonstrates that all transformed fuzzy subsets $A_i \circ g$, $g \in \Gamma$, are extensional with respect to $\tilde{E}$. In order to prove that $\tilde{E}$ is continuous consider $(x_0, y_0) \in \mathcal{X} \times \mathcal{X}$ and a sequence $(x_n, y_n)_{n \in \mathbb{N}}$, $(x_n, y_n) \in \mathcal{X} \times \mathcal{X}$, that converges to (x_0, y_0). Due to the sequential compactness assumption of $\mathcal{A}_i$ (respectively the set) for each pair (x_n, y_n) there is a transformation $g_n \in \Gamma$ such that

$$\tilde{E}(x_n, y_n) = E(g_n(x_n), g_n(y_n))$$

As the sequence $(g_n)_n$ converges uniformly to a continuous tranformation $g_0 \in \Gamma$ for any $\varepsilon > 0$ for sufficiently large n we obtain

$$\begin{aligned}
|E(g_n(x_n), g_n(y_n)) - E(g_0(x_0), g_0(y_0))| &\leq |E(g_n(x_n), g_n(y_n)) - E(g_0(x_n), g_0(y_n))| \\
&\quad + |E(g_0(x_n), g_0(y_n)) - E(g_0(x_0), g_0(y_0))| \\
&\leq \varepsilon
\end{aligned}$$

which proves the continuity of $\tilde{E}$.

Now, consider the converse. Assume that assertion (ii) of Theorem 6 holds true. In order to prove that $\mathcal{A}_i$ is sequentially compact, we, firstly, show that sequences $(a_n)_{n\in\mathbb{N}} \subseteq \mathcal{A}_i$ are equicontinuous, that is, for all $\varepsilon > 0$ and all elements $x \in \mathcal{X}$ there is a $\delta > 0$ such that for sufficiently large indeces n the condition $|\tilde{x}-x| < \delta$ implies $|a_n(\tilde{x})-a_n(x)| < \varepsilon$. As the functions of $\mathcal{A}_i$ are globally bounded Arzela-Ascoli's theorem will assure the uniform convergence of the sequence under cosnideration. On details on equicontinuity and Arzela-Ascoli's theorem see, e.g., [33]. Suppose the contrary then there is $\varepsilon > 0$ an element $x_0 \in \mathcal{X}$ and a sequence $(x_n)_{n\in\mathbb{N}}$, $x_n \in \mathcal{X}$ with $\lim_n x_n = x_0$ such that

$$|a_n(x_0) - a_n(x_n)| \geq \varepsilon \tag{10}$$

Now, by the continuity of the t-norm T and, hence, its uniform continuity it follows that for $\varepsilon/2 > 0$ there is a $\delta > 0$ such that $|a_1 - a_2| < \delta$ and $|b_1 - b_2| < \delta$, $(a_1, b_1), (a_2, b_2) \in [0,1] \times [0,1]$ implies $|T(a_1, b_1) - T(a_2, b_2)| < \varepsilon$. As $\lim_n x_n = x_0$ and the continuity assumption of E for sufficiently large indeces n we have $1 - E(x_0, x_n) < \delta$ and, consequently, we obtain

$$\begin{aligned} a_n(x_0) - \frac{\varepsilon}{2} &\leq T(a_n(x_0), E(x_0), x_n) \qquad (11) \\ a_n(x_n) - \frac{\varepsilon}{2} &\leq T(a_n(x_n), E(x_0), x_n). \end{aligned}$$

From the extensionality assumption it follows

$$\begin{aligned} T(a_n(x_0), E(x_0), x_n) &\leq a_n(x_0) \qquad (12) \\ T(a_n(x_n), E(x_0), x_n) &\leq a_n(x_n), \end{aligned}$$

and finally, combining the inequalities (11) and (12), we obtain

$$|a_n(x_0) - a_n(x_n)| \leq \frac{\varepsilon}{2}$$

which contradicts the assumption (10) that $\mathcal{A}_i$ is not equicontinuous. □

Proposition 7. *If Γ is a set of admissible transformations, then the fuzzy relation (9) is the biggest T-equivalence relation for which all transformed fuzzy partitions are extensional.*

Proof. The proof follows immediately from Theorem 2. □

For example, the set Γ_L of Lipschitz continuous functions g, i.e., $|g(x) - g(y)| \leq L|x - y|$ with the Lipschitz constant L is compact. Let us consider the Łukasiweicz t-norm $T_L(a,b) = \max\{a+b-1, 0\}$ then the fuzzy relation $E_L(x,y) = 1-\min\{|x-y|/L, 1\}$ satisfies the condition (ii) of Theorem 6.

4 Conclusion

In this paper we started with various invariants for fuzzy sets and fuzzy partitions like convexity and T-redundancy. After looking for more general transformations based on the compositional rule of inference it turned out that under such restrictions it suffices to restrict to transformations obtained by the usual composition of functions. In order to guarantee the existence of solutions of optimizing a loss function of fuzzy partitions and other parameters the compactness condition was studied more thoroughly. It turned out that the compactness in the uniform sense of the set of admissibly transformed fuzzy partitions can be characterised also in terms of purely fuzzy logical concepts that is by means of T-equivalence relations.

5 Appendix

5.1 Proof of Proposition 1

(i) Let us assume that T is continuous but not Archimedean and let $\overset{\leftrightarrow}{T}$ be continuous. Then, by definition there is a number a, $0 < a < 1$, with $T(a,a) = a$. Consequently, we get

$$\overset{\leftrightarrow}{T}(a,0) = \min\{\vec{T}(a,0), \underbrace{\vec{T}(0,a)}_{1}\} = \vec{T}(a,0) < a < 1 = \overset{\leftrightarrow}{T}(a,a)$$

The continuity assumption entails the existens of a number b satisfying

$$\overset{\leftrightarrow}{T}(a,b) = a, \qquad 0 < b < a. \tag{13}$$

Applying the T-transitivity of $\overset{\leftrightarrow}{T}$, conditon (13) leads to a contradiction as

$$a = T(\underbrace{\overset{\leftrightarrow}{T}(1,a)}_{a}, \underbrace{\overset{\leftrightarrow}{T}(a,b)}_{a}) \leq \overset{\leftrightarrow}{T}(1,b) = b.$$

This shows that the continuity of $\overset{\leftrightarrow}{T}(a,b)$ implies the validity of the Archimedean property of T.

(ii) Consequently, applying Ling's theorem, see equation (1), we get the representation $T(a,b) = f^{-1}(\min\{f(a)+f(b), f(0)\})$ where f is continuous and, for the induced residuum

$$\overset{\leftrightarrow}{T}(a,b) = \begin{cases} f^{-1}(|f(a)-f(b)|) & \text{if } a,b > 0 \\ 1 & \text{if } a = b = 0 \\ 0 & \text{else.} \end{cases} \tag{14}$$

The case $f(0) = \infty$ induces $\overset{\leftrightarrow}{T}(0, \frac{1}{n}) = 0$ whereas $\overset{\leftrightarrow}{T}(0,0) = 1$ showing thatfor strict Archimedean t-norms the induced residual biimplication cannot be continous. For the case $f(0) < \infty$ the representation (14) reduces to $\overset{\leftrightarrow}{T}(a,b) = f^{-1}(|f(a)-f(b)|)$ which proves its continuity.

5.2 Proof of Proposition 5

Let $\Gamma = \{A_1, \ldots, A_n\}$ be a family of fuzzy subsets of $\mathcal{X}$ and T be an arbitrary left-continuous t-norm. Then, there holds $\textbf{Red}_T(\Gamma) = 1$ if, and only if, for some $i \in \{1,\ldots,n\}$ the fuzzy subset A_i is entirely included in the union of the others, i.e., $A_i(x) \leq S_{Tj\neq i}(A_j(x))$ for all $x \in \mathcal{X}$.

To start with, let us assume that there is some fuzzy subset that is contained in the union of the others, i.e., $A_i(x) \leq S_{T,i\neq j}(A_j(x))$ for all $x \in \mathcal{X}$ for some i. Due to Definition 7 we have $\vec{T}(a,b) = 1$ whenever $a \leq b$. From this it follows, that

$$1 \leq \inf_x \vec{T}(A_i(x), S_{T,i\neq j}(A_j(x))) \leq \textbf{Red}_T(\Gamma).$$

The other way round, let us assume that $\textbf{Red}_T(\Gamma) < 1$. This means that for each $i \in \{1,\ldots,n\}$ there exists an element $x_i \in \mathcal{X}$ sucht that

$$\vec{T}(A_i(x_i), S_{T,i\neq j}(A_j(x_i))) < 1,$$

which implies

$$A_i(x_i) > S_{T,i\neq j}(A_j(x_i)) \geq \max_{i\neq j} A_j(x_i)$$

showing that there is no member of the family Γ that is contained in the union of the others.

Acknowledgements

Bernhard Moser gratefully acknowledges partial support by the Austrian Government, the State of Upper Austria, and the Johannes Kepler University Linz in the framework of the K*plus* Competence Center Program. Furthermore, special thanks go to Felix Kossak for careful proof-reading.

References

[1] A. Baraldi and P. Blonda. A survey of fuzzy clustering algorithms for pattern recognition — Part I. *IEEE Trans. Syst. Man Cybern. B*, 29(6):778–785, 1999.

[2] A. Baraldi and P. Blonda. A survey of fuzzy clustering algorithms for pattern recognition — Part II. *IEEE Trans. Syst. Man Cybern. B*, 29(6):786–801, 1999.

[3] J. C. Bezdek. *Pattern Recognition with Fuzzy Objective Function Algorithms*. Plenum Press, New York, 1981.

[4] J. C. Bezdek, J. Keller, R. Krishnapuram, and N. K. Pal. *Fuzzy Models and Algorithms for Pattern Recognition and Image Processing*, volume 4 of *The Handbooks of Fuzzy Sets*. Kluwer Academic Publishers, Boston, 1999.

[5] U. Bodenhofer. A generalized approach to fuzzy orderings and its applicability in fuzzy control. Technical report, Dept. of Telecommunications and Telematics, Technical University of Budapest, July 1998.

[6] U. Bodenhofer. A note on approximate equality versus the Poincaré paradox. *Fuzzy Sets and Systems*, 133(2):155–160, 2003.

[7] F. Bolata and A. Nowé. From fuzzy linguistic specifications to fuzzy controllers using evolution strategies. In *Proc. 4th IEEE Int. Conf. on Fuzzy Systems*, volume III, pages 1089–1094, Yokohama, 1995.

[8] M. Burger, J. Haslinger, and U. Bodenhofer. Tuning of fuzzy systems as an ill-posed problem. In M. Anile, V. Capasso, and A. Greco, editors, *Progress in Industrial Mathematics at ECMI 2000*, volume 1 of *Mathematics in Industry*, pages 493–498. Springer, 2002.

[9] M. Burger, J. Haslinger, U. Bodenhofer, and H. W. Engl. Regularized data-driven construction of fuzzy controllers. *J. Inverse Ill-Posed Probl.*, 10(4):319–344, 2002.

[10] B. De Baets and R. Mesiar. Pseudo-metrics and T-equivalences. *J. Fuzzy Math.*, 5(2):471–481, 1997.

[11] B. De Baets and R. Mesiar. T-partitions. *Fuzzy Sets and Systems*, 97:211–223, 1998.

[12] B. De Baets and R. Mesiar. Metrics and T-equalities. *J. Math. Anal. Appl.*, 267:331–347, 2002.

[13] M. De Cock, U. Bodenhofer, and E. E. Kerre. Modelling linguistic expressions using fuzzy relations. In *Proc. 6th Int. Conf. on Soft Computing*, pages 353–360, Iizuka, October 2000.

[14] M. Demirci. On many-valued partitions and many-valued equivalence relations. *Internat. J. Uncertain. Fuzziness Knowledge-Based Systems*, 11(2):235–253, 2003.

[15] M. Drobics. machine learning framework for *Mathematica* — creating understandable computational models from data. In *Proc. of the 2003 Mathematica Developer Conf.*, Champaign, IL, 2003. Wolfram Research Inc.

[16] M. Drobics. Choosing the best predicates for data-driven fuzzy modeling. In *Proc. 13th IEEE Int. Conf. on Fuzzy Systems*, pages 245–249, Budapest, July 2004.

[17] M. Drobics and U. Bodenhofer. Fuzzy modeling with decision trees. In *Proc. 2002 IEEE Int. Conf. on Systems, Man and Cybernetics*, volume 4, Hammamet, Tunisia, October 2002.

[18] M. Drobics, U. Bodenhofer, and W. Winiwarter. Mining clusters and corresponding interpretable descriptions — a three-stage approach. *Expert Systems*, 19(4):224–234, 2002.

[19] D. Dubois and H. Prade. A review of fuzzy set aggregation connectives. *Inform. Sci.*, 36:85–121, 1985.

[20] S. Gottwald. Fuzzy set theory with t-norms and Φ-operators. In A. Di Nola and A. G. S. Ventre, editors, *The Mathematics of Fuzzy Systems*, volume 88 of *Interdisciplinary Systems Research*, pages 143–195. Verlag TÜV Rheinland, Köln, 1986.

[21] S. Gottwald. *Fuzzy Sets and Fuzzy Logic*. Vieweg, Braunschweig, 1993.

[22] D. E. Gustafson and W. C. Kessel. Fuzzy clustering with a fuzzy covariance matrix. In *Proc. IEEE Int. Conf. on Decision and Control*, pages 761–766, San Diego, CA, 1979.

[23] U. Höhle. Fuzzy equalities and indistinguishability. In *Proc. 1st European Congress on Fuzzy and Intelligent Technologies*, volume 1, pages 358–363, Aachen, 1993.

[24] U. Höhle. The Poincaré paradox and non-classical logics. In D. Dubois, E. P. Klement, and H. Prade, editors, *Fuzzy Sets, Logics and Reasoning about Knowledge*, volume 15 of *Applied Logic Series*, pages 7–16. Kluwer Academic Publishers, Dordrecht, 1999.

[25] F. Höppner and F. Klawonn. Improved fuzzy partitions for fuzzy regression models. *Internat. J. Approx. Reason.*, 32:85–102, 2003.

[26] F. Klawonn. Mamdani's model in the view of equality relations. In *Proc. 1st European Congress on Fuzzy and Intelligent Technologies*, volume 1, pages 364–369, Aachen, 1993.

[27] E. P. Klement, R. Mesiar, and E. Pap. *Triangular Norms*, volume 8 of *Trends in Logic*. Kluwer Academic Publishers, Dordrecht, 2000.

[28] E. P. Klement and B. Moser. On the t-redundancy of fuzzy partitions. *Fuzzy Sets and Systems*, 85:195–201, 1997.

[29] R. Kruse, J. Gebhardt, and F. Klawonn. *Foundations of Fuzzy Systems*. John Wiley & Sons, New York, 1994.

[30] C. H. Ling. Representation of associative functions. *Publ. Math. Debrecen*, 12:189–212, 1965.

[31] B. Moser. *A New Approach for Representing Control Surfaces by Fuzzy Rule Bases*. PhD thesis, Johannes Kepler Universität Linz, October 1995.

[32] S. V. Ovchinnikov. Similarity relations, fuzzy partitions, and fuzzy orderings. *Fuzzy Sets and Systems*, 40(1):107–126, 1991.

[33] E. R. Phillips. *An Introduction to Analysis and Integration Theory*. Dover Publications, New York, 1984.

Evolving Neuro-Fuzzy Modelling from Real-time Data Stream and Applications

Xiaowei Zhou
Infolab21, Lancaster University,
Lancaster, LA2 0PF, UK
x.zhou3@lancaster.ac.uk

Modelling based on Neuro-Fuzzy systems that has an evolving structure is an newly introduced area. Different to the existing adaptive systems, so called evolving fuzzy systems has not only the ability of automatically tuning the parameters in the model, but also the ability of updating the structure of the rule-base online and unsupervised. Based on the parameter-free online clustering tool eClustering [P.Angelov, 2002], the studies on the Takagi-Sugeno fuzzy rule based online evolving solutions for a series of modelling problems such as classification, prediction, and controlling have been started and shows good potential.[P.Angelov, D.Feliv, 2002-2004] The structure of the fuzzy rule-base, namely the number and the antecedent part of the rules, is generated from the real-time data stream, fully unsupervised. Consequent part of the fuzzy rule can be automatically learned with or without supervised information. The proposed online evolving systems have the advantage of working fully automatically; parameters-free; computationally efficient (one-pass, no history data to be memorized or processed) and fully online work mode. All these features of evolving systems enable the solutions works for real-times automatic applications.

Researches have also been carried out in different application directions. There are already industrial orientated solutions and applications in Oil refinery industry [Jose J.Macias, P.Angelov, X.Zhou, 2006], Robotics [X.Zhou, P.Angelov, 2005-2006], World Wide Web [Tony, P.Angelov, X.Zhou 2006], Mobile Communications [E.Jones, P.Angelov, C.Xydeas,2006] and so on.

Technical Report
FLLL–TR–0210

Analysis of large microarray images

Leila Muresan
Fuzzy Logic Laboratorium Linz-Hagenberg
e-mail `leila.muresan@jku.at`

Institut für Wissensbasierte Mathematische Systeme

Fuzzy Logic Laboratorium Linz

Johannes Kepler Universität Linz
Altenberger Str. 69
A-4040 Linz
Austria

Softwarepark Hagenberg
Hauptstrasse 99
A-4232 Hagenberg
Austria

Introduction

The new technology of ultra-sensitive microarray scans [4] generates a need for image processing algorithms that can cope with the challenges of the novel technique. This work presents two approaches to the full analysis of such images, however the emphasis is on the microarray gridding step.

A short comparison of the two approaches is presented. An outline of future work concludes the report.

1 Microarray technology

Analyzing microarray images offers a high-throughput method to obtain biological information. Schematically, the experiment consists of laying spots of probes (reporters) on a glass, silicon or plastic slide in a regular pattern forming a $2D$ grid. The slide with the affixed spots is called microarray or chip and it represents a dictionary of probes, in wich an entry (DNA, oligonucleotide, antibody etc.) is identified by its position in the grid.

For the sake of statistical soundness, each probe on the chip is replicated a certain number of times (subject to a trade-off between cost and reliability of results).

Next, a sample is marked with a fluorescent tag and washed over the chip. In the case of two different samples (e.g. one from normal cells and the other for cancerous cells),the samples are marked with distinguishable tags, emitting at different wavelengths (typically red and green), then mixed and finally washed over the microarray.

The binding of the sample to the probes gives a measure of interaction intensity (hybridization) with the respective probe. This hybridization is measured by scanning the microarray, usually with a confocal laser microscope and observe the fluorescent intensity for each spot. This last step represents the image processing task involved in the microarray experiments. Statistical analysis of the differences between the two scans corresponding to the two samples, is used to gain understanding on the role the reporters play in the problem at hand.

Some of the best known software that perform the image processing and statistical analysis steps are GenePix, ScanAlyze, Maia, Spot.

The classical microarray techniques are missing important factors in the analysis: the intensity of the spot image is due not only to the amount of the bound sample, but also to the sample itself (e.g. the number of fluorophores that bind to a single molecule may vary with the length of the molecule). The binding process itself is not uniform (e.g. unspecific binding, consisting of fluorophores that are not attached to sample molecules, is lower inside the probe spots than between the spots). The varying illumination can further distort the inference process.

The novel technique described in [4] ensures that the resolution of the scanning is tremendously increased, to one pixel imaged in the classical way correspond 400 pixels with the new technique. At this resolution it is possible to detect and count single molecules, whose appearance is a Gaussian blur, the point spread function of the optical system applied to a point source. However if the density of the molecules in a spot is too high, single molecule counting algorithms fail and the classical average intensity -based methods have to be used.

The new technology brings important advantages, like reducing the amount of sample needed and more refinement in information. However there are several challenges of the new method related with the huge amount of data that has to be processed, the scarcity of the highly expressed spots, the high variance of the Poisson processes modelling the counting of molecules as opposed to the Gaussian statistics etc.

2 Grid alignment

The first step in microarray image analysis is grid alignment or grid registration (gridding). It consist of finding the location of a regular two-dimensional spot array in the image. The number of spots in each row (N) and in each column (M) is assumed to be known, the interspot distance is assumed normal. Small variations of the interspot distance due to the spotting procedure may occur. Sometime, the spots are groupped in several sub-blocks. However, the problem of identifying sub-blocks will not be discussed here.

If the microarray contains many bright spots (representing highly expressed genes, for instance) it is possible to allow for variability in the location and shape of the spots, to increase accuracy without losing interpretability (the probes could still be correctly identified according to their position).

For very low sample concentrations this is however not the case and since it is necessary to analyze also the spots that cannot be differentiated from the background based on their intensity, we shall search for grids with spot centers on a fixed, regular grid and circular spots with fixed radius. We shall perform the search based on the information offered by the bright, well-expressed

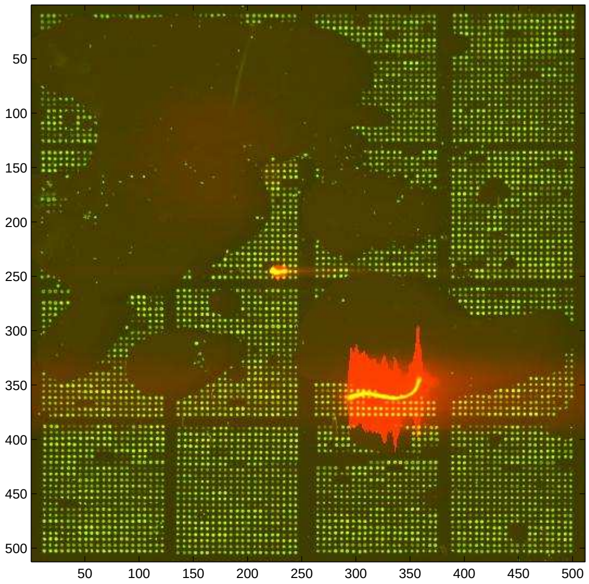


Figure 1: Microarray (Maia test image)

spots.

Finding the position of the grid in the scan consists of determining the appropriate scaling, rotation and translation of the set $\{(\frac{i}{N}, \frac{j}{M}) \mid i = 1, \ldots, N, j = 1, \ldots, M\}$, such that it best fits the bright spots in the scanned image.

Since analyzing in memory two images of 22GB each is practically impossible, two approaches are considered to perform the gridding. The first one is to downsample the image and apply an adapted version of a classical gridding algorithm. After the grid is available, the peaks in each spot are counted via the a trous wavelet method described in [7, 6].

In the majority of cases gridding is achieved by the following four steps: image pre-processing for spot amplification, rotation correction, inter-spot distance estimation and finally the construction and adjustment of the grid. The order does not necessarily has to be the one given above.

The main difficulty in gridding for the ultra-sensitive scans is the low number of bright spotrs, due to the low concentration of the samples.

2.1 Identification of spot locations

There is a plethora of methods to find the location and dimensions of bright spots, based on which the orthogonal grid is built. Among these we mention the method of Angulo and Serra based on mathematical morphology [1], clustering of the pixels into foreground, background and artifact pixels, watershed based algorithms etc. (Some of these approaches are used to adjust the spots adter an initial grid was found).

For segmentation we apply a fast and simple local thresholding method. Since we know the total number of spots, we can chose a window size in which one can assume that one has approximately one spot compute it's mean and variance and threshold accordingly. (The background is assumed Gaussian distributed). Small or non-circular features resulting after thresholding are removed. The combination of the two images (red and green) further improves the result. The binarization step is replaced in [**?**] by OMT and in [2] by convolution with a matched filter.

The purpose of all these approaches is the amplification of the signal at the spot locations, which consequently also improves the results of the rotation estimation via the subsequent Radon transform.

2.2 Rotation estimation

In order to determine the rotation of the grid pattern, the Radon transform is applied to the binarized image:

$$\mathcal{R}[B](\theta, s) = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} B(x, y)\delta(x\cos\theta + y\sin\theta - s)\, dx\, dy$$

The rotation is assumed to be small, in the range of ± 5 degrees. This is due to the fact that on one hand the Radon transform is computationally expensive, on the other hand rotations have to be determined with high accuracy. The error in the estimation of the rotation angle $\Delta\alpha$ has to fulfill: $\Delta\alpha \approx \sin\Delta\alpha < \frac{H}{n_H \cdot W}$, where H and W are the height and width of the downscaled image, and n_H is the number of spots in a column.

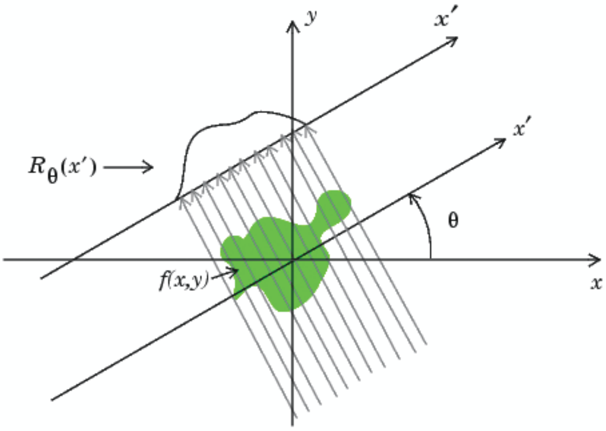


Figure 2: Illustration of the Radon transform (Matlab)

To select the rotation angle, we select the angle α for which the Shannon entropy of $\mathcal{R}[B](\alpha,\cdot)$ is minimum :

$$\alpha_0 = \operatorname*{argmin}_{\alpha} \left\{ -\sum_i \mathcal{R}[B](\alpha,i)^2 \cdot \log\left(\mathcal{R}[B](\alpha,i)^2\right) \right\}$$

Alternatively, one can minimize (taking into account horizontal and vertical alignments):

$$\left\{ -\sum_i \mathcal{R}[B](\alpha,i)^2 \cdot \log\left(\mathcal{R}[B](\alpha,i)^2\right) - \sum_i \mathcal{R}[B](\alpha+\frac{\pi}{2},i)^2 \cdot \log\left(\mathcal{R}[B](\alpha+\frac{\pi}{2},i)^2\right) \right\}$$

Finally, the image B is corrected with the computed rotation $-\alpha_0$.

2.3 Grid construction

In order to determine the horizontal and vertical interspot distances, D_H and D_V, we compute the position of the global maximum in the periodograms of the projection of the rotated binary image B on the two axes, x respective y.

Using the same projected data, the position of the grid is determined by computing the location of local maxima values (high values close to local maxima), at distance D_H apart from each other for the vertical projection, and D_V for the horizontal one. Results are better if a small deviation from these distances is allowed (fig. 3).

The quality of the gridding results is measured as $\sum_i D(g_i, c_i)$, where $D(g_i, c_i)$ represents the distance between c_i, the centroid of a blob in the binary image B and the closest grid point g_i.

Other approaches to grid construction include the k-nearest neighbor based approach from [5], or the one based on partial Voronoi diagrams [3].

3 Comparisons of analysis techniques

As an alternative analysis approach we counted first the peaks in small (200 × 200 pixel) non-overlapping patches over the whole scan, and produced an image of dimensions $\frac{N}{200} \times \frac{M}{200}$, where each pixel $I_{count}(x, y)$ contains the number of peaks found in the respective patch. The same a trous wavelet method was used for peak counting as in the downsampling method.

Although more costly (the area analyzed via the wavelet method is roughly four times bigger), this approach offers insights in the performance of the new technique.

The images obtained from counting molecules (I_{count}) as described above were compared with the images (I_{mean}) of the same scans, having the same dimensions, in wich each pixel represents the mean value of the 200×200 pixels patch.

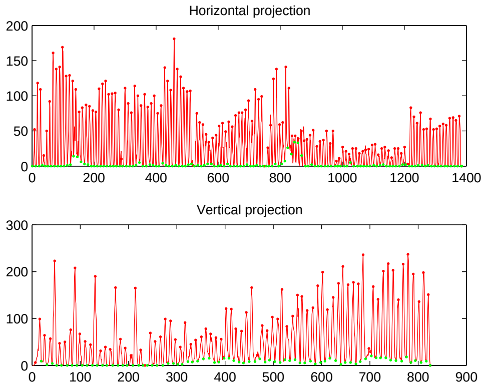


Figure 3: Detection of local maxima (red) and local minima (green) in the horizontal and vertical projection data

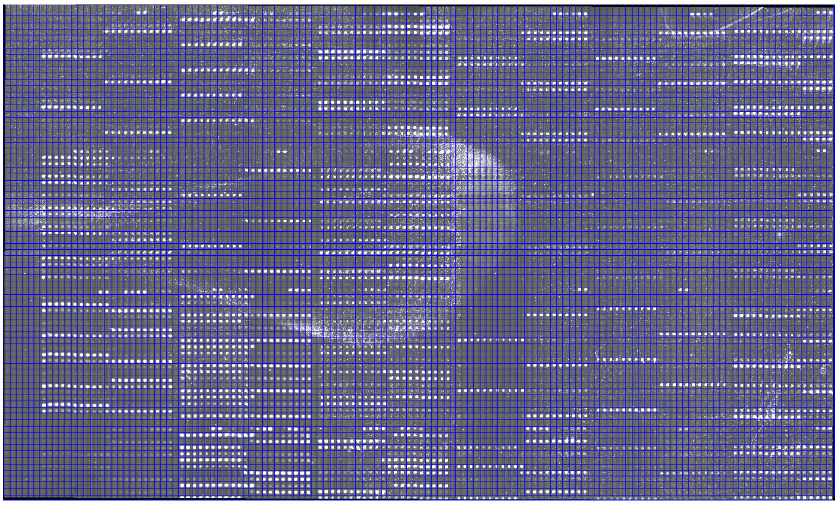
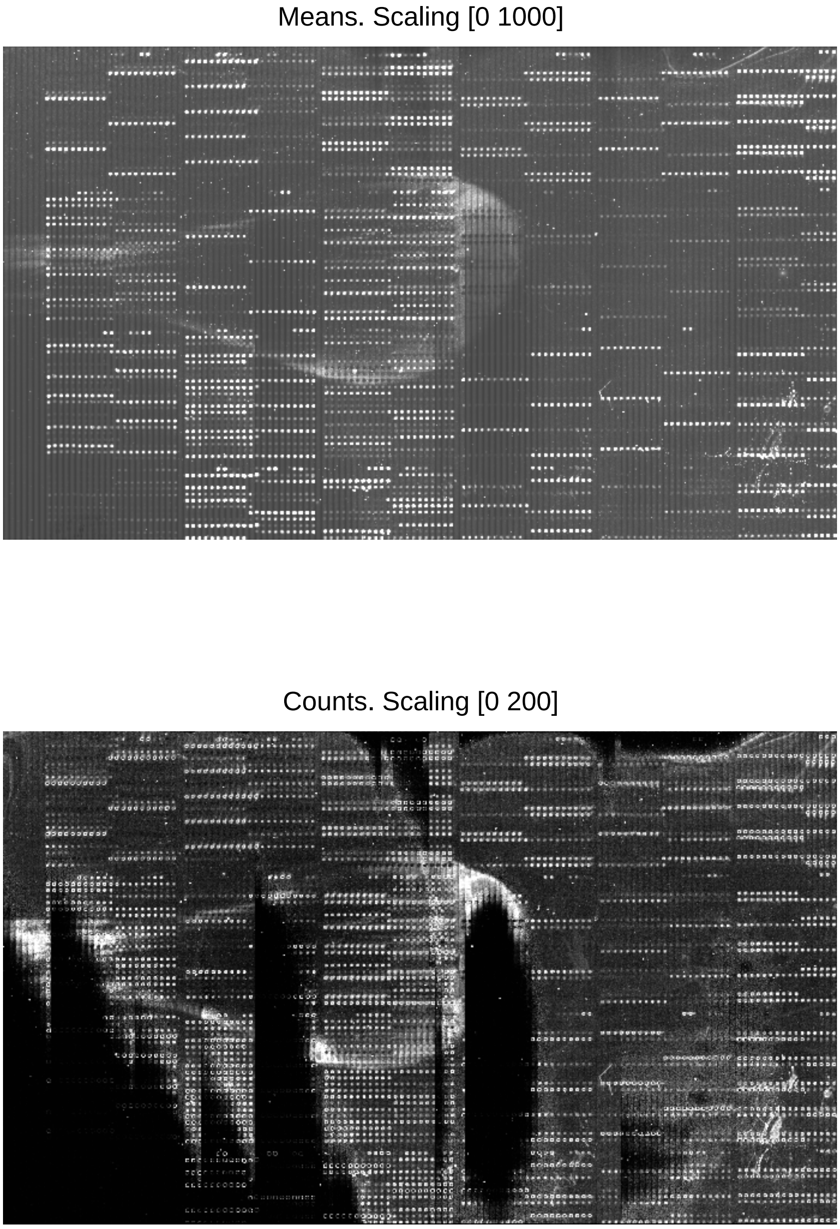


Figure 4: Result of grid alignment. The image intensities are scaled for better visibilty.



Means. Scaling [0 1000]

Counts. Scaling [0 200]

Figure 5: Comparison of the two results for the same microarray: intensity means and peak counts in patches of 200×200 pixels. The images are scaled as indicated for better visualization.

In figure 5 one can observe the two kind of results for the same microarray scan, one obtained by counting molecules and the other by computing the mean pixel intensities. Both images are scaled for better visbility. A major concern for the ultra-sensitive approach is illustrated in the counting image: if the object (the chip) is not in focus the counting of the peaks cannot be performed properly and one has to relly only on mean pixel data. (Defocused regions are the black parts in the lower left corner and in the center of the count results for the microarray image in figure 5.)

Figure 6 represents a relative difference of the results computed as:

$$\frac{N_{count}(x,y) - N_{mean}(x,y)}{N_{mean}(x,y)},$$

where for each patch (x, y), $N_{count}(x, y)$ and $N_{mean}(x, y)$ represent the values of $I_{count}(x, y)$ and $I_{mean}(x, y)$, respectivelly, normalized to $[0, 1]$.

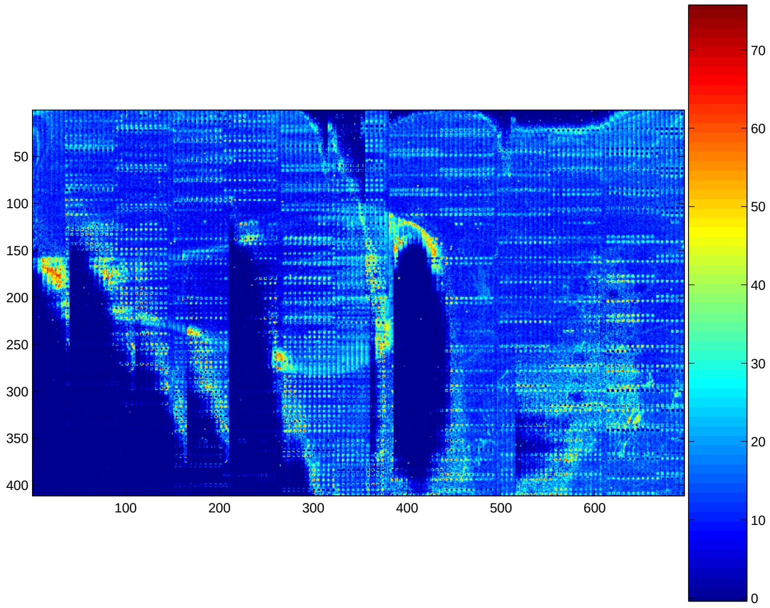


Figure 6: Percentage of change between the two methods of analysis (counting and mean). See text for details.

After conveniently thresholding the difference image ($T = 20\%$), one computes the typical range for which the new technique brings an improvement over the classical one (the results can be expressed both in counts and in mean values).

4 Conclusion and future work

Two approaches to analysis of large microarray scans were presented, which complement each other. Their results are illustrated on real data. However further testing of the methods is necessary, especially for the cases in which subsequences of consecutive bright spots are not available. Furthermore since the analysis of a microarray scan takes six to eight hours, paralellization of the algorithms might improve considerably the speed. The parallelization of the presnted methods is straightforward.

References

[1] Jesus Angulo and Jean Serra. Automatic analysis of DNA microarray images using mathematical morphology. *Bioinformatics*, 19(5):553–562, 2003.

[2] N. Brändle, H. Bischof, and H. Lapp. Robust DNA microarray image analysis. *Machine Vision and Applications*, 15:11–28, 2003.

[3] V.L. Galinsky. Automatic registration of microarray images. i. rectangular grid. *Bioinformatics*, 19(14):1824–1831, 2003.

[4] Jan Hesse, Jaroslaw Jacak, Maria Kasper, Gerhard Regl, Thomas Eichberger, Martina Winklmayr, Fritz Aberger, Max Sonnleitner, Robert Schlapak, Stefan Howorka, Leila Muresan, Anna-Maria Frischauf, and Gerhard J. Schütz. RNA expression profiling at the single molecule level. *Genome Research*, 16:1041–1045, 2006.

[5] Ho-Youl Jung and Hwan-Gue Cho. An automatic block and spot indexing with k-nearest neighbors graph for microarray image analysis. *Bioinformatics*, 18(2):141–151, 2002.

[6] L. Muresan, B. Heise, J. Kybic, and E.P. Klement. Quantitative analysis of microarray images. In *IEEE International Conference on Image Processing, ICIP 2005.*, pages II– 1274–1277, 2005.

[7] J.-C. Olivo-Marin. Extraction of spots in biological images using multiscale products. *Pattern Recognition*, 35:1989–1996, 2002.

An Alternative Approach to Parallel MRI

Frank Bauer[1], Thorsten Hohage[2], Stephan Kannengiesser[3]

[1]: Fuzzy Logic Laboratorium Linz-Hagenberg
University of Linz
Softwarepark Hagenberg
Hauptstraße 99
4232 Hagenberg
Austria
Email: frank.bauer@jku.at

[2]: Institute for Numerical and Applied Mathematics
University of Göttingen
Lotzestr. 16-18
37083 Göttingen
Germany
Email: hohage@math.uni-goettingen.de

[3]: Siemens Medical Solutions
Karl-Schall-Str. 6
91052 Erlangen
Germany
Email: stephan.kannengiesser@siemens.com

Abstract

Magnetic Resonance Imaging with parallel data acquisition requires algorithms for reconstructing the patient's image from a small number of measured k-space lines.

In contrast to well-known algorithms like SENSE and GRAPPA and its flavors we consider the problem as a non-linear inverse problem. Fast computation algorithms for the necessary Fréchet derivative and reconstruction algorithms are given.

For a motivation we will repeat the introduction and numerics section of the submitted article "An Alternative Approach to the Image Reconstruction for Parallel Data Acquisition in MRI" by Frank Bauer and Stephan Kannengiesser (Preprint `ftp://ftp.num.math.uni-goettingen.de/pub/preprints/bauer/main1.pdf`) on the next page. Additionally some new material (together with Thorsten Hohage) which concerns an even faster computation of the Fréchet derivatives will be presented in the talk.

1 Introduction

Magnetic resonance imaging (MR imaging, MRI) routinely relies on critical sampling in the three-dimensional spatial frequency space (Fourier domain, k-space) for spatial encoding. A fairly recent development allows to replace some of the time-consuming sequential steps of phase encoding by switched magnetic gradient fields, so-called k-space lines, by parallel acquisition with an array of detector elements with a manifold of non-uniform spatial sensitivities. This partially parallel imaging (PPI) has been proposed in many forms and colorful acronyms such as SMASH [1], SENSE [2], GRAPPA [3], SPACE-RIP [4], etc. In short, PPI allows to reconstruct images from k-space data sets with limited support.

All the known methods have in common that they work in two distinct steps: a calibration step, in which a separately acquired data set, or part of the undersampled data set is used to extract information about or related to the sensitivity characteristics of the detector array, and a reconstruction step, in which the undersampled data set is either completed, or an artifact-free image is synthesized from the undersampled data and the sensitivity information.

A popular concept in PPI is called autocalibration, and works by acquisition of a regularly undersampled k-space grid plus a few k-space lines near the center of k-space, i.e., in the low spatial frequency range.

It should be noted that, although the sensitivity characteristics of the array detector elements are unknown and vary with the patient specific detector placement and choice of diagnosis protocol, the noise characteristics of the detector array can easily be measured with high accuracy.

There is today no method available which performs the image reconstruction in a single process which takes into account both the undersampled imaging data and the extra imaging data or the extra calibration data. Also, there is today no optimality criterion for the quality of the image reconstruction, except for those methods which assume perfect knowledge of the spatial sensitivity profiles of the array detector elements, which is of course not achievable in practice.

In this paper we will show how one can reconstruct both the image and the sensitivity out of the data simultaneously. To this end we will reinterpret this problem as a non-linear inverse problem, see e.g. [5, 6] and the references therein. These have received great attention in the recent times in all areas of non-destructive testing. A specific property of inverse problems is their intrinsic instability which is also an explanation for many effects observed in the practice of PPI. The advantage of the methods which we will introduce here is that we can actually guarantee the optimal order of accuracy in our solutions. In general and without additional information it is not possible to beat these methods apart from a constant factor.

The paper is organized as follows. First we describe the specific model we use and introduce notation. In the second part we will give an explicit calculation of the Fréchet derivative which is an essential part of the later considerations. We will give some considerations to reduce the number of necessary operations. Afterwards we will present different possibilities to solve the non-linear inverse problem including some explicit algorithmic parts. The last sections presents a short numerical experiment using the presented algorithms.

2 Numerics

2.1 Sensitivities

A major ingredient for the numerical treatment is the right initial guess for the sensitivities. This decomposes again into two parts. One is estimating this quantity out of real data and the other is finding an appropriate but small basis system $\{\mathcal{B}_n\}_{n\in\{1,\dots,\mathcal{B}_{\text{num}}\}}$ to represent the sensitivities.

Considering the basis system there are several possibilities. A standard approach is using a small area of the k-space (typically about 0.1%) to model the sensitivities. This works considerably well if one uses a very smooth initial guess.

2.2 Simulation

We intensively tested the method with simulated data. However in terms of reliability we have to keep in mind that the initial "measurement" and the later simulations of measurements needed for the solution procedure were performed in the same way.

We observed that the method worked reliably in this setting, however a good knowledge of the sensitivities was of great advantage. For small noise levels reconstructions could be (in terms of reconstruction error) more than ten times better than GRAPPA which basically tells that the new method seems intrinsically to be much less biased.

Depending on the size and the noise level reconstruction times were considerably long. However, this is just the case for very small noise levels, the more noise the faster the method.

In all our applications we had the impression that IRGN with CG yielded much better results in a shorter time than Landweber iteration. However, due to the speed of one step of the Landweber iteration and the big improvements in the first steps it might still be worthwhile to consider this method to go over the data shortly if time is an issue.

2.3 Application

For the application we used real data acquired on a Siemens MAGNETOM Trio a Tim System clinical whole-body scanner operating at a magnetic flux density of 3.0 Tesla. A full three-dimensional gradient echo data set of the head of a healthy volunteer was acquired using the system's 12-channel head matrix coil. The field of view was 240mm in all three spatial directions with the frequency encoding in head-feet direction. The resolution was 256 samples in all three spatial directions.

In order to do fast prototyping we used Matlab® 7 where we reconstructed an image with resolution 256×256. We restricted our attention to one slice with these dimensions. For the reconstruction we used 32 of the 256 lines in the middle and an acceleration factor of 3, i.e. additionally every third line.

As comparison method and initial guess we used GRAPPA. In figure 1 we see GRAPPA, the new method and a reference picture using all possible data beside each other. As usual we applied a manual contrast enhancing step for all of these images.

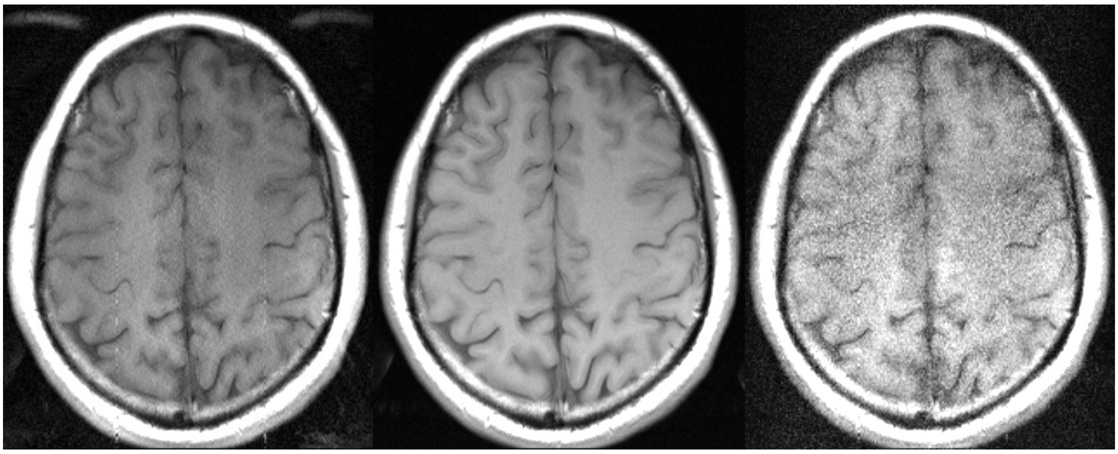


Figure 1: l: New Method m: using all data (reference) r: GRAPPA

2.4 Discussion

From a visual inspection of the results it seems that the new method yields results which are superior to the initial GRAPPA image. An important point is that the GRAPPA reconstruction is medically seen not usable, whereas the new method provides an image which can be used for this purpose. Hence we can alternatively consider this new method as an add-on which can improve images considerably in times where the full processor power of the MRI-machines is not needed.

The proposed method integrates for the first time the sensitivity calibration step and the image reconstruction step in PPI into a single processing step. Therefore, it has a high potential of being superior to all existing methods, especially since the inherent information about the receiver sensitivities which is contained in the undersampled imaging data themselves is also exploited. As an interesting side remark, this method could therefore even work if no additional calibration data are acquired.

Through its general mathematical framework, this method also appears flexible enough to be able to incorporate other a-priori information about the unknown image than sensitivity information of the array detectors. Many methods exploiting structural a-priori information have been proposed in the past for MRI, ranging from real-valuedness over limited support, to model-based approaches with reference images. Being able to incorporate any of these additional a-priori constraints would make the method even more powerful.

Acknowledgements

The support for the first author of the Graduiertenkolleg 1023 of the University of Göttingen is gratefully acknowledged.

References

[1] Sodickson D, Manning W. Simultaneous acquisition of spatial harmonics (SMASH): fast imaging with radiofrequency coil arrays. Magn Reson Med. 1997 Oct;38(4):591–603.

[2] Pruessmann K, Weiger M, Scheidegger M, Boesiger P. SENSE: sensitivity encoding for fast MRI. Magn Reson Med. 1999 Nov;42(5):952–962.

[3] Griswold M, Jakob P, Heidemann R, Nittka M, Jellus V, Wang J, et al. Generalized autocalibrating partially parallel acquisitions (GRAPPA). Magn Reson Med. 2002 Jun;47(6):1202–1210.

[4] Kyriakos W, Panych L, Kacher D, Westin C, Bao S, Mulkern R, et al. Sensitivity profiles from an array of coils for encoding and reconstruction in parallel (SPACE RIP). Magn Reson Med. 2000 Aug;44(2):301–308.

[5] Engl H, Hanke M, Neubauer A. Regularization of Inverse Problems. Dordrecht, Boston, London: Kluwer Academic Publisher; 1996.

[6] Bauer F, Hohage T. A Lepskij-type stopping rule for regularized Newton methods. Inverse Problems. 2005;21:1975–1991.

Stochastic resonance and energy efficient information processing in single neurons

Thomas Hoch
Software Competence Center Hagenberg, Hagenberg, Austria
thomas.hoch@scch.at

November 10, 2006

Abstract

Recently, it has been shown that synaptic background activity can facilitate the processing of weak input signals in cortical neurons, as for example in a stochastic resonance setting. Stochastic resonance, however, was often claimed to play no significant role for neural information processing since the brain is highly adaptive, and could easily change neural properties to improve information processing beyond that of a stochastic resonance scenario. Energy consumption, on the other hand, has been suggested by many researchers to constrain information processing. Energy efficient codes favor low firing rates and subthreshold input distributions, which suggest that stochastic resonance may be a useful mechanism for low cost information transmission. Using a single leaky integrate-and-fire neuron we show that the inclusion of the metabolic cost for information transmission indeed favors subthreshold input distributions and that noise can improve information transmission.

1 Introduction

The fundamental principle by which the brain processes information about a stimulus is still unknown. An apparent signature of cortical neurons is the high variability of their spiking activity (Stevens and Zador, 1998; Shadlen and Newsome, 1998). It is widely accepted that the irregularity in the neural response arises from strong fluctuations of the membrane potential. To what extent the fluctuations of the membrane potential might be considered unwanted noise, which may reduce the information processing capabilities of a neuron, or play an important role in neural information processing, remains an open question. Recent experimental and theoretical studies have shown ways how noise may facilitate information processing in neural systems. For instance, it has been shown that noise improves the speed with which a population responds to transient inputs (Silberberg et al., 2004), modulates the responsiveness (gain) of cortical neurons to a driving input current (Chance et al., 2002), or even allows for a transmission of weak (subthreshold) signals, as for example in a stochastic resonance setting (Wiesenfeld and Moss, 1995; Russell et al., 1999).

Stochastic resonance is a phenomenon in which the transmission of a signal - measured for example by the mutual information between input and output -

becomes optimal for a certain noise level, which depends on the properties of the distribution of the input signal. Stochastic resonance is fundamental in many physical, chemical, and biological processes [for a general review on stochastic resonance see Gammaitoni et al. (1998)] and occures if a system is non-linear (e.g. bistable) and the driving input consists of a weak (subthreshold) signal and a stochastic force (noise). Neural systems are highly nonlinear and subjected to background activity, which might act as a noise source. Unsurprisingly, it was recognized early that stochastic resonance may also play a major role in neural information processing. Over the last two decades, stochastic resonance in neural systems has been investigated, both theoretically (Wiesenfeld and Moss, 1995; Rudolph and Destexhe, 2001; Hoch et al., 2003) and experimentally (Russell et al., 1999; for a review see Moss et al., 2004). These studies concluded that their is strong evidence for stochastic resonance in the sensory and peripheral nervous system. Nevertheless, it is still unclear to what extend stochastic resonance plays a role in the central nervous system of higher animals.

Maximal information transmission, however, may not be the only goal neurons try to achieve. Neural activity is costly in metabolic terms, and energy consumption and dissipation becomes a concern, for example for the densely packed central nervous system of higher animals. Several researchers have suggested that the overall energy consumption constrains information transmission, and it has been argued that neurons try to achieve a balance between information transmission and energy consumption, leading to energy efficient codes (Levy and Baxter, 1996; Laughlin et al., 1998; Balasubramanian et al., 2001; Hoch et al., 2003).

In the following, we explore the complex relationship between information transmission, energy consumption and background activity. Using a leaky integrate-and-fire neuron, we show that noise can enhance information transmission substantially if the input signal is subthreshold. Nevertheless, information rates are much higher for supra-threshold signals and we show that the addition of noise always deteriorates information transmission for such signals. The transmission of signals in the supra-threshold regime, however, is also accompanied with high firing rates of the neurons involved. Thus the question arises as to why cortical cells favour low firing rates (Olshausen and Field, 2005). After including metabolic constraints we find that optimal information transmission in terms of information rate per unit cost occurs mainly in the subthreshold regime, where neurons can exploit the membrane potential fluctuations to maximize information transmission (stochastic resonance). Thus we conclude that in the central nervous system of higher animals, information is likely to be coded using low firing rates and populations of cells.

2 Materials and Methods

We consider a leaky integrate-and-fire (LIF) neuron. The membrane potential V of the leaky LIF neuron changes in time according to the following differential equation:

$$\tau_m \frac{dV(t)}{dt} = (E_L - V(t)) + \frac{I_s(t)}{g_L} + D\frac{dW(t)}{dt}, \tag{1}$$

where $\tau_m = 10\,ms$ is the membrane time constant, $g_L = 0.05\,\mu S/cm^2$ is the leak conductance of the membrane, $E_L = -70\,mV$ is the reversal potential, $I_s(t)$ is the external signal (an aperiodic Gaussian stimulus), and $dW(t)$ is the infinitesimal increment of a Wiener process, which describes the influence of the background activity on the membrane potential.

Equation (1) describes the subthreshold dynamics of the membrane potential V. For $I_s = 0$ and $D = 0$, the membrane potential V decreases exponentially towards E_L with time constant $\tau_m = \frac{C_m}{g_L}$. For $I_s > 0$, the injected current increases the membrane potential towards the spike threshold. If I_s is strong enough, it will drive the membrane potential V across the spike threshold V_{th} leading to a spike event. After a spike, the membrane potential is immediately reset to V_{reset} and usually clamped to this value for an absolute refractory period of $T_{ref} = 2\,ms$.

To characterize stochastic resonance, we estimated the information between the input signal and the neural response with the help of the linear reconstruction method. The information rate is obtained with the following equation (Borst and Theunissen, 1999):

$$I_{lin} \geq -\frac{1}{2\pi}\int_0^\infty log_2\left[1-\gamma^2(\omega)\right]\,d\omega, \tag{2}$$

where $\gamma^2(\omega) = \frac{|P_{SR}(\omega)|^2}{P_{SS}(\omega)P_{RR}(\omega)}$ is the coherence between the stimulus and the response, $P_{SS}(\omega)$ and $P_{RR}(\omega)$ are the power spectra of the stimulus and the spike train, and $P_{SR}(\omega)$ denotes the cross power spectrum between the stimulus and the spike train.

3 Results

3.1 Stochastic Resonance

Here we investigate the role of noise on the dynamics of subthreshold input signals. When a noisy current is injected into the neuron, stochastic resonance occurs and the transmission of subthreshold input signals is enhanced. Figure 1 shows the stochastic resonance like behavior of a leaky integrate-and-fire neuron for different bias currents. The Fig. shows the information rate plotted against the noise level. If the stimulus is totally subthreshold (blue curve), the input signal is to weak to evoke a response and, therefore, without noise no information can be transmitted. An increase of the noise level leads to a stronger fluctuating membrane potential, which occasionally initiates action potentials. In the beginning, information rate increases fast with increasing noise levels, because the stronger fluctuating membrane potential evokes more spikes, which allows the transmission of more information about the stimulus. At higher noise levels, however, the firing rates of the neuron are high, but the information that the action potential contains about the stimulus is reduced. Thus, the beneficial increase of the firing rate is counterbalanced by less informative action potentials leading to a decrease of the information rate at high noise levels, as Fig. 1 shows. Nevertheless, there exists an optimal amount of noise, which maximizes information transmission.

This optimal noise level depends on the statistic of the input signal and on the distance between the average membrane potential and the spike threshold,

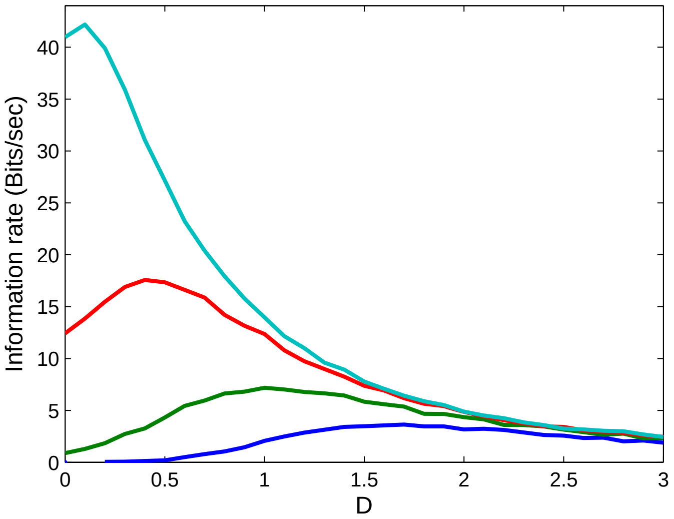


Figure 1: Dependency of the stochastic resonance phenomenon on the strength of the input signal, expressed by the bias current. The blue line represent a weak signal ($I_{bias} = 0.6\,nA$), where as the cyan line corresponds to stimulation with a strong signal ($I_{bias} = 1,1\,nA$). The green and the red line represent intermediate values of I_{bias}, e.g. $I_{bias} = 0.75\,nA$ and $I_{bias} = 0.9\,nA$, respectively. The standard deviation of the input signal was $0.1\,nA$.

as Fig. 1 indicates. Information transmission is greatly improved only in the subthreshold regime ($I_{bias} \leq 1\,nA$). At small values of I_{bias}, the distance to the spike threshold is increased and a stronger fluctuating membrane potential is needed in order to evoke a response. Thus, information transmission is optimal at higher noise levels. Conversely, for a given noise level information transmission is optimized, if the neuron lowers its threshold. At high noise levels ($D > 2$), however, the information rate only weakly depends on the distance to threshold.

3.2 Energy Efficient Information Transmission

Information transmission in the brain is metabolically expensive (Laughlin et al., 1998, Lennie, 2003). In particular, the generation of spikes consumes a huge amount of energy. If the cost of firing is high in comparison to the *housekeeping* cost, then it is advantageous for the brain to use energy efficient neural codes (Levy and Baxter, 1996; Laughlin et al., 1998; Balasubramanian et al., 2001; Hoch et al., 2003). Given a fixed amount of energy, one strategy to achieve energy efficient information transmission is to maximize the information rate per metabolic cost.

As a measure of metabolic efficiency we consider the ratio between the transmitted information I_R and the total metabolic cost E. We assume that the average metabolic cost E per unit time is a sum of a term proportional to the average rate $\bar{r}$ of the neuron and a term which contributes a fixed baseline cost b. The baseline cost represents the metabolic expense of maintaining a neuron.

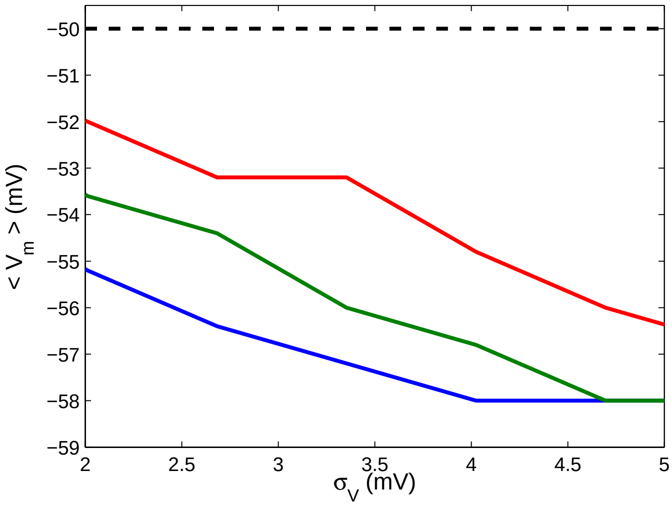


Figure 2: Average membrane potential, for which information rate per unit cost is maximal, plotted against the standard deviation of the membrane potential σ_V for different baseline costs($b = 1$: blue line; $b = 3$: green line; and $b = 10$: red line).

We set

$$E = c(b + \bar{r}), \tag{3}$$

where c is a proportionality constant, which can be interpreted as the average cost per spike. Both information rate and average cost change with the actual noise level.

To investigate the relationship between the average membrane potential, the noise level, and the information rate per unit cost I_E, we vary the bias current I_{bias}, the baseline cost b, and the noise level of the neuron. For each combination of noise level and baseline cost, we obtained the average membrane potential of the neuron from the bias current, which maximizes the information transmission per unit cost in order to determine the optimal encoding regime. Figure 2 shows the average membrane potential that optimizes information transmission, plotted against the standard deviation of the membrane potential σ_V for different baseline costs($b = 1$: blue line; $b = 3$: green line; and $b = 10$: red line). Interestingly, the Fig. reveals that the optimal distance between the average membrane potential and the threshold (black dashed curve) is about one to two σ_V, depending on the actual baseline cost. Thus, for less dominating baseline cost, the information transmission becomes maximal at an average membrane potential, which rests about one standard deviation of V_m below threshold.

4 Discussion

In this model study we examined how background activity affects the information transmission. We showed that the background activity can have a wide influence on the information transmission properties of a LIF neuron if the driving signal is mainly subthreshold. For such signals, the noise level has to be adjusted accurately in order to optimize stimulus encoding for changing intensities of the input (Hoch et al., 2003). On the other hand, a neuron could improve information transmission by making the stimulus more supra-threshold. In principle, this could be achieved either by lowering the spike threshold or by increasing the average membrane potential through a strengthening of the stimulus inducted synaptic drive. In the high noise regime, however, a reduction of the distance between the average membrane potential and the spike threshold was found to barely affect the information rate.

Recent studies have shown that neural systems prefer information transmission via many parallel low intensity channels (Laughlin et al., 1998; Balasubramanian et al., 2001; Hoch et al., 2003). Similar to that, our simulations of the LIF neuron have shown that after taking the cost of information transmission into account, optimal information transmission occurs in the subthreshold regime. This holds even for small noise levels, provided that the baseline costs are not too small compared to the rate dependent costs. Since application of noise is one way to allow for transmission of otherwise subthreshold signals, the strive for energy efficient codes may be a justification of stochastic resonance in neural systems.

5 References

Balasubramanian V, Kimber D, Berry MJ, 2nd (2001) Metabolically efficient information processing. Neural Comput. 13: 799-815.

Borst A, Theunissen FE (1999) Information theory and neural coding. Nat. Neurosci. 2: 947-957.

Chance FS, Abbott LF, Reyes AD (2002) Gain Modulation from Background Synaptic Input. Neuron 35: 773-782.

Gammaitoni L, Hänggi P, Jung P, Marchesoni F (1998) Stochastic resonance. Rev. Mod. Phys. 70: 223-287.

Hoch T, Wenning G, Obermayer K (2003) Optimal noise-aided signal transmission through populations of neurons. Phys. Rev. E 68: 011911.

Laughlin SB, de Ruyter van Steveninck RR, Anderson JC (1998) The metabolic cost of neural information. Nature Neurosci. 1: 36-41.

Lennie P (2003) The cost of cortical computation. Curr. Biol. 13: 493-497.

Levy WB, Baxter RA (1996) Energy efficient neural codes. Neural Comput. 8: 531-543.

Moss F, Ward LM, Sannita WG (2004) Stochastic resonance and sensory information processing: A tutorial and review of applications. Clinical Neurophysiol. 115: 267-281.

Olshausen BA, Field DJ (2005) How close are we to understand V1?. Neural Comput. 17: 1665-1699.

Rudolph M, Destexhe A (2001) Do neocortical pyramidal neurons display stochastic resonance? J. Comput. Neurosci. 11: 19-42.

Russell DF, Wilkens LA, Moss F (1999) Use of behavioural stochastic resonance by paddle fish for feeding. Nature 402: 291-294.

Silberberg G, Bethge M, Markram H, Pawelzik K, Tsodyks M (2004) Dynamics of population rate codes in ensembles of neocortical neurons. J. Neurophysiol. 91: 704-709.

Shadlen MN, Newsome WT (1998) The Variable Discharge of Cortical Neurons: Implications for Connectivity, Computation, and Information Coding. J. Neurosci. 18: 3870-3896.

Stevens CF, Zador AM (1998) Input synchrony and the irregular firing in cortical neurons. Nat. Neurosci. 1: 210-216.

Wiesenfeld K, Moss F (1995) Stochastic resonance and the benefits of noise: from ice ages to crayfish and SQUIDs. Nature 373: 33-36.

Large Scale Simulations of Neural Microcricuits with PCSIM

Thomas Natschläger

November 10, 2006

Abstract

In this contribution we will describe a tool for simulating models of neural circuits composed of simple point neurons which is called PCSIM and its parallel version PCSIM. In particular we will focus on the issue of highly distributed simulation of large neural models and discuss recent very promising results.

1 Introduction

The Parallel C*ircuit* SIM*ulator* PCSIM is a tool for simulating heterogeneous networks composed of (spike emitting) point neurons. PCSIM is intended to simulate networks containing a few neurons, up to networks with 10^6 to 10^7 neurons and on the order of 10^9 synapses. It was written to do modeling at the network level in order to analyze the computational effects which can not be observed at the single cell level. To study single cell computations in detail we give the advice to use simulators like GENESIS or NEURON.

Easy to use Python interface: The core of PCSIM is written in C++ which is controlled by means of Python. We have chosen Python since it provides very powerful graphics and analysis add ons and is a widely used programming language which is freely available. Hence it is not necessary to learn yet another script language to set up and run simulations with PCSIM.

Object oriented design: We adopted an object oriented design for PCSIM which is similar to the approaches taken in GENESIS and NEURON. That is there are objects (e.g. a `LifNeuron` object implements the standard leaky-integrate-and-fire model) which are interconnected by means of well defined signal channels. The creation of objects, the connection of objects and the setting of parameters of the objects is controlled at the level of Python whereas the actual simulation is done in the C++ core.

Fast C++ core: Since PCSIM is implemented in C++ and is not as general as e.g. GENESIS simulations are performed quite fast. We also implemented some ideas from event driven simulators which result in a considerable speedup (up to a factor of three for low firing rates; see the subsection about implementation aspects below).

Different levels of modeling: By providing different neuron models PCSIM allows to investigate networks at different levels of abstraction: sigmoidal neurons with analog output, linear and non-linear leaky-integrate-and-fire neurons and compartmental based (point) neurons with spiking output. A broad range of synaptic models is also available for both spiking and non-spiking neuron models: starting from simple static synapses ranging over synapses with short-term plasticity to synapse models which implement different models for long-term plasticity.

2 Built-in models

Neuron models: PCSIM provides two different classes of neurons: neurons with analog output and neurons with spiking output. Neurons with analog output are useful for analyzing population responses in larger circuits. For example PCSIM provides a sigmoidal neuron with leaky integration. However, there are much more different objects available to build models of spiking neurons:

- Standard (linear) leaky-integrate-and-fire neurons
- Non-linear leaky-integrate-and-fire neurons based on the models of Izhikevich
- Conductance based point neurons with and without a spike template. There are general conductance based neurons where the user can insert any number of available ion-channel models to build the neuron model. On the other hand there is a rich set of predefined point neurons available used in several studies.

Spiking Synapses: As for the neurons PCSIM also implements synapses which transmit analog values and spike transmitting synapses. Two types of synapses are implemented: static and dynamic synapses. While for static synapses the amplitude of each postsynaptic response (current of conductance change) is the same, the amplitude of an postsynaptic response in the case of a dynamic synapse depends on the spike train that it has seen so far, i.e. dynamic synapses implement a form of short term plasticity (depression, facilitation). For synapses transmitting spikes the time course of a postsynaptic response is modeled by $A \times \exp(-t/\tau_{syn})$, where τ_{syn} is the synaptic time constant and A is the synaptic strength which is constant for static synapses and given by the model described in (Markram et al., 1998) for dynamic synapses.

Note that static as well as dynamic synapses are available as current supplying or conductance based models.

Analog Synapses: For synapses transmitting analog values, such as the output of a sigmoidal neuron, static synapses are simply defined by their strength (weight), whereas for dynamic synapses we implemented a continuous version of the dynamic synapse model for spiking neurons (Tsodyks et al. 1998).

Synaptic plasticity: PCSIM also supports spike time dependent plasticity, STDP, applying a similar model as in (Song et al., 2000). STDP can be modeled most easily by making the assumption that each pre- and postsynaptic spike pair contributes to synaptic modification independently and in a similar manner. Depending on the time difference $\Delta t = t_{pre} - t_{post}$ between pre- and postsynaptic spike the absolute synaptic strength is changed by an amount $L(\Delta t)$. The typical shape for the function $L(\Delta t)$ as found for synapses in neocortex layer 5 (Markram et al., 1997) is implemented. Synaptic strengthening and weakening are subject to constraints so that the synaptic strength does not go below zero or above a certain maximum value. Furthermore additional variants as suggested in (Froemcke and Dan, 2002) and (Guetig et al. 2003) are also implemented.

3 Implementation aspects

Network input and output: There are two forms of inputs which can be supplied to the simulated neural microcircuit: spike trains and analog signals. To record the output of the simulated model special objects called `Recorder` are used. A recorder can be connected to any object to record any field of that object.

Simulation Strategy: PCSIM employees a clock based simulation strategy with a fixed simulation step width *dt*. Typically the exponential Euler integration method is used. A spike which occurs during a simulation time step is assumed to occur at the end of that time step. That implies that spikes can only occur at multiples of *dt*.

Efficient processing of spikes: In a typical simulation of a neural circuit based on simple neuron models the CPU time spent in advancing *all* the synapses may by larger then the time needed to integrate the neuron equations. However if one considers the fact that synapses are actually "idle" most of the time (at least in low firing rate scenarios) it makes sense to update during one time step only those synapses whose postsynaptic response is not zero, i.e. are active. PCSIM implements this idea by dividing synapses into a list of idle and a list of active synapses where only the latter is updated during a simulation time step. A synapse becomes active (i.e. is moved from the idle list to the active list) if a spike arrives. After its postsynaptic response has vanished the synapse becomes idle again (i.e. is moved back from the active list to the idle list). This trick can result in considerable speed up for low firing rate scenarios.

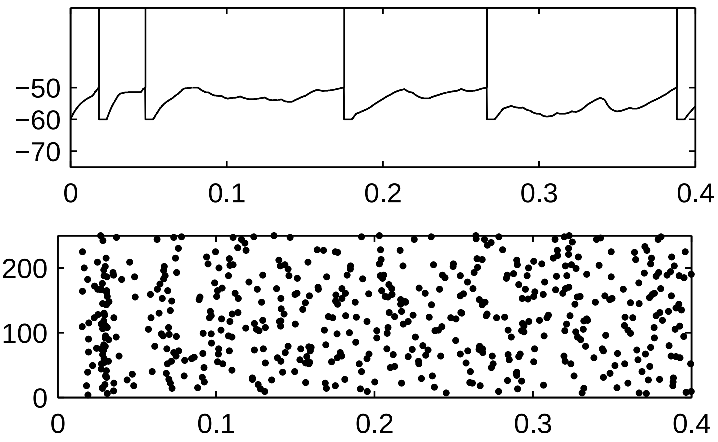


Figure 1: Results of PCSIM simulations of the benchmark 2. The top panel shows the voltage trace (in mV) of a selected neuron. The spikes superimposed as vertical lines. The lower panel shows the spike raster for randomly selected neurons for each of the three benchmarks.

Distributed simulations: PCSIM employees MPI and multi-thread technologies to achieve hight performance for distributed simulations. The main requirement (as for all parallel algorithms) is to keep the amount of inter process communication (and synchronization low). This can be achieve in the case of spiking communication by exploiting the fact that full spike exchange has only to take place after the time span of the minimal delay in the system. If that minimal delay is for example 10 times the simulation step with a supra linear speedup can be achieved (see Fig. 2).

4 PCSIM implementations of a benchmark simulations

Results of a PCSIM simulations of bechmark 2 (see appendix) are depicted in Figure 1. This figures were produced by the simulation scripts provided for each benchmark using Pythons's powerful graphics capabilities (see the file `make_figures.m`) and illustrate the sustained irregular activity described by Vogels and Abbott (2005) for such networks.

The current development version of PCSIM has been used to perform scalability tests based on the CUBA benchmark. The results are summarized in Figure 2. For the small 4000 neuron network the speedup for more than four machines vanishes while for the larger networks a more than expected speedup occurs up to six machines. This shows that PCSIM is scalable with regard to the problem size and the number of available machines. The development version of PCSIM together with the python script for the CUBA benchmark can be obtained from `http://sourceforge.net/projects/pcsim`.

5 Further information

The current development version of PCSIM can be obtained from
`http://sourceforge.net/projects/pcsim`.

Appendix: Benchmark simulations

In this appendix, we present a series of "benchmark" network simulations using both integrate-and-fire (IF) or Hodgkin-Huxley (HH) type neurons. They were chosen such that at least one of the benchmark can be implemented in the different simulators (the code corresponding to these implementations will be provided in the ModelDB database at http://senselab.med.yale.edu/senselab/modeldb).

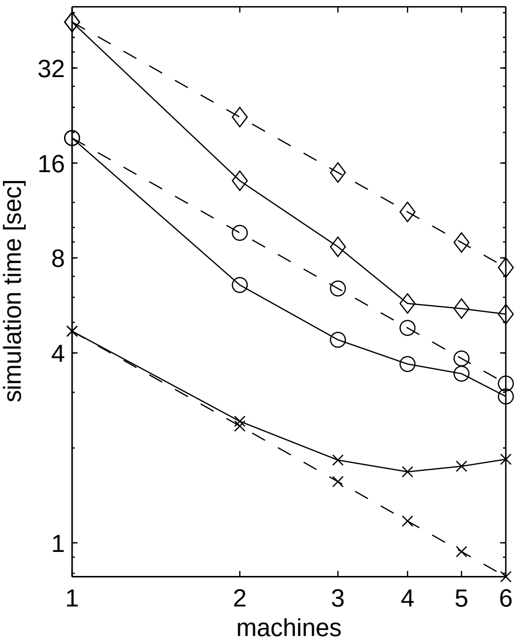


Figure 2: Performance of PCSIM. The time needed to simulate the CUBA network for one second of biological time (solid line) as well as the expected times (dashed line) are plotted against the number of machines (Intel Xeon, 3.4 Ghz, 2 Mb cache). The CUBA model was simulated for three different sizes: 4000 neurons and 3.2×10^5 synapses (stars), 10000 neurons and 2×10^6 synapses (circles), and 20000 neurons and 20×10^6 synapses (diamonds).

The models chosen were networks of excitatory and inhibitory neurons inspired from a recent study (Vogels and Abbott, 2005). This paper considered two types of networks of leaky IF neurons, one with current-based synaptic interactions (CUBA model), and another one with conductance-based synaptic interactions (CUBA model; see below). We also introduce here a HH-based version of the COBA model, as well as a fourth model consisting of IF neurons interacting through voltage deflections ("voltage-jump" synapses).

Network structure

Each model consisted of 4,000 IF neurons, which were separated into two populations of excitatory and inhibitory neurons, forming 80% and 20% of the neurons, respectively. All neurons were connected randomly using a connection probability of 2%.

Passive properties

The membrane equation of all models was given by:

$$C_m \, \frac{dV}{dt} \; = \; -g_L(V - E_L) \; + \; S(t) \; + \; G(t) \; , \tag{1}$$

where $C_m = 1\ \mu\mathrm{F/cm}^2$ is the specific capacitance, V is the membrane potential, $g_L = 5{\times}10^{-5}\ \mathrm{S/cm}^2$ is the leak conductance density and E_L = -60 mV is the leak reversal potential. Together with a cell area of 20,000 $\mu\mathrm{m}^2$, these parameters give a resting membrane time constant of 20 ms and an input resistance at rest of 100 MΩ. The function $S(t)$ represents the spiking mechanism and $G(t)$ stands for synaptic interactions (see below).

Spiking mechanisms

IF neurons

In addition to passive membrane properties, IF neurons had a firing threshold of -50 mV. Once the Vm reaches threshold, a spike is emitted and the membrane potential is reset to -60 mV and remains at that value for a refractory period of 5 ms.

Synaptic interactions

Conductance-based synapses

For conductance-based synaptic interactions, the membrane equation of neuron i was given by:

$$C_m \frac{dV_i}{dt} \;=\; -g_L(V_i - E_L) \;+\; S(t) \;-\; \sum_j g_{ji}(t)(V_i - E_j) \;, \tag{2}$$

where V_i is the membrane potential of neuron i, $g_{ji}(t)$ is the synaptic conductance of the synapse from neuron j to neuron i, and E_j is the reversal potential of that synapse. E_j was of 0 mV for excitatory synapses, or -80 mV for inhibitory synapses.

Synaptic interactions were implemented as follows: when a spike occurred in neuron j, the synaptic conductance g_{ji} was instantaneously incremented by a quantum value (6 nS and 67 nS for excitatory and inhibitory synapses, respectively) and decayed exponentially with a time constant of 5 ms and 10 ms for excitation and inhibition, respectively.

Current-based synapses

For implementing current-based synaptic interactions, the following equation was used:

$$C_m \frac{dV_i}{dt} \;=\; -g_L(V_i - E_L) \;+\; S(t) \;-\; \sum_j g_{ji}(t)(\bar{V} - E_j) \;, \tag{3}$$

where $\bar{V}$ = -60 mV is the mean membrane potential. The conductance quanta were of 0.27 nS and 4.5 nS for excitatory and inhibitory synapses, respectively. The other parameters are the same as for conductance-based interactions.

Benchmarks

Based on the above models, the following four benchmarks were implemented.

Benchmark 1: Conductance-based IF network. This benchmark consists of a network of IF neurons connected with conductance-based synapses, according to the parameters above. It is equivalent to the COBA model described in Vogels and Abbott (2005).

Benchmark 2: Current-based IF network. This second benchmark simulates a network of IF neurons connected with current-based synapses, which is equivalent to the CUBA model described in Vogels and Abbott (2005). It has the same parameters as above, except that every cell needs to be depolarized by about 10 mV, which was implemented by setting E_L = -49 mV (see Vogels and Abbott, 2005).

For all four benchmarks, the models simulate a self-sustained irregular state of activity, which is easy to identify: all cells fire irregularly and are characterized by important subthreshold voltage fluctuations. The neurons must be randomly stimulated during the first 50 ms in order to set the network in the active state.

References

[1] Froemke RC, Dan Y (2002). Spike-timing-dependent synaptic modification induced by natural spike trains. Nature, 416(3):433–438.

[2] Guetig R, Aharonov R, Rotter S, Sompolinsky H (2003). Learning input correlations through non-linear asymmetric hebbian plasticity. Journal of Neuroscience, 23:3697–3714.

[3] Markram H, Lubke, J, Frotscher, M, Sakmann, B (1997). Regulation of synaptic efficacy by coincidence of postsynaptic APs and EPSPs. Science, 275:213–215.

[4] Markram H, Wang Y, Tsodyks M (1998). Differential signaling via the same axon of neocortical pyramidal neurons. Proc. Natl. Acad. Sci., 95:5323–5328.

[5] Natschläger T, Markram H, Maass W (2003). Computer models and analysis tools for neural microcircuits. In Kötter, R, editor, Neuroscience Databases. A Practical Guide, chapter 9, pages 123–138. Kluwer Academic Publishers (Boston).

[6] Song S, Miller K, Abbott LF (2000). Competitive Hebbian learning through spike-timing-dependent synaptic plasticity. Nature Neurosci., 3:919–926.

[7] Tsodyks M, Pawelzik K, Markram H (1998). Neural networks with dynamic synapses. Neural Computation, 10:821–835.

A Multi-Agent Approach to 3D Rendering Optimization

Paper ID: #142

ABSTRACT

Physically based rendering is the process of generating a 2D image from the abstract description of a 3D Scene. Despite the development of various new techniques and algorithms, the computational requirements of generating photo-realistic images still do not allow to render in real time. Moreover, the configuration of good render quality parameters is very difficult and often too complex to be done by non-expert users. This paper describes a novel approach called MAgArRO (standing for "*Multi Agent AppRoach to Rendering Optimization*") which utilizes principles and techniques known from the field of multi-agent systems to optimize the rendering process. Experimental results are presented which show the benefits of MAgArRO-based rendering optimization.

Categories and Subject Descriptors

I.2.11 [**Distributed Artificial Intelligence**]: Multiagent systems; I.3.7 [**I.3.7 Three-Dimensional Graphics and Realism**]: Raytracing

General Terms

Rendering, Raytracing, Optimization, Multiagent System

Keywords

Multiagent Systems :: Cooperative distributed problem solving :: Task and resource allocation ** Tools and Techniques :: Computational complexity

1. INTRODUCTION

The process of constructing a 3D image comprises several phases such as modelling, setting materials and textures, placing the virtual light sources, and finally rendering. Rendering algorithms take a description of geometry, materials, textures, light sources and virtual cameras as input and produce an image or a sequence of images (in the case of an animation) as output. There are different rendering algorithms – ranging from simple and fast to more complex and accurate ones – which simulate the light behavior in a precise way. Such methods are normally classified in two main categories, namely, local and global illumination algorithms. High-quality photo-realistic rendering of complex scenes is one of the key goals and challenges of computer graphics. Unfortunately this process is computationally intensive and may require a huge amount of time in some cases (especially, global illumination algorithms), and the generation of a single high quality image may take several hours up to several days, even on fast computers. As pointed out by Kajiya [10], all rendering algorithms aim to model the light behavior over various types of surfaces and try to solve the so-called rendering equation (which forms the mathematical basis for all rendering algorithms). Because of the huge amount of time it requires, *the rendering phase is often considered to be a crucial bottleneck in photorealistic projects.* In addition, the selection of the input parameters and variable values of the scene (number of samples per light, depth limit in ray tracing, etc.) is very complex. Typically a user of a 3D rendering engine tends to "over-optimize", that is, to choose very high values which do not affect the perceptual quality of the resulting image but further increase the required rendering time considerably.

This paper describes a novel optimization approach called MAgArRO based on principles, techniques and concepts known from the area of multi-agent systems. Specifically, MAgArRO is based on design principles of the FIPA standards [1], employs adaptation and auctioning, and utilizes expert knowledge. The key advantages of this approach are robustness, flexibility, scalability, decentralized control (autonomy of the involved agents), and the capacity to optimize locally.

The paper is structured as follows. First, Section 2 overviews the state of the art and the current main research lines in rendering optimization. Thereby the focus is on the most promising issues related to parallel and distributed rendering. This section also surveys approaches which aim at applying Artificial Intelligence methods to rendering optimization and, more specifically, it points to related work on rendering based on multi-agent technology. Next, Section 3 describes MAgArRO in detail. Next, Section 4 shows empirical results obtained for different numbers of agents and input variables. Next, Section 5 offers thoughts and suggestions on promising future work. Finally, Section 6 concludes the article with more general considerations.

2. RELATED WORK

There are a lot of rendering methods and algorithms with different characteristics and properties (e.g., [16, 10, 15]). Common to these algorithms is that different levels of realism of the rendering are always related in one way or an-

AAMAS 2007 Honolulu, Hawaii USA

other to the complexity and computation time required. A key problem in realistic computer graphics thus is the time required for rendering due to the computational complexity of the related algorithms. Chalmers et al. [7] expose various research lines in the rendering optimization issues.

Optimizations via Hardware. Some researchers use programmable GPUs (Graphics Processing Units) as massively parallel, powerful streaming processors than run specialized portions of code of a raytracer [5]. Other approaches are based on special-purpose hardware architectures which are designed to achieve maximum performance in a specific task [14]. These hardware-based approaches are very effective and even the costs are low if manufactured in large scale. The main problem is the lack of generality: the algorithms need to be designed specifically for each hardware architecture. Against that, MAgArRO works at a very high level of abstraction and runs on almost any existing rendering engine without changes.

Optimizations using parallel/distributed computing. If we divide the problem into a number of smaller problems (each of which is solved on a separate processor), the time required to solve the full problem may be reduced significantly. In order to have all processing elements fully utilized, a task scheduling strategy must be chosen. This task constitutes the elemental unit of computation within the parallel implementation [7], and its output is the application of the algorithm to a specified data item. There are many related approaches such as [8] which use Grid systems for rendering over the Internet. The main advantage of the approaches based on parallel (or distributed) computing is the highly efficient use of existent machines. One of the key problems, however, is to achieve effective load balancing and node management. Compared to these parallel-computing approaches, MAgArRO uses dynamical in combination with non-centralized load balancing, and this makes MAgArRO more efficient especially when the number of nodes (thus the coordination overhead) increases.

Distributed Multi Agent Optimizations. The inherent distribution of Multi Agent systems and their properties of intelligent interaction allow for an alternative view of rendering optimization. The work presented by Rangel-Kuoppa et al. [11] uses a JADE-based implementation of a multi-agent platform to distribute interactive rendering tasks (rasterization) across a network. The distribution of the tasks is realized in a centralized client-server style (the agents send the results of the rasterization of objects to a centralized server). Although this work employs the multi-agent metaphor, essentially it does not make use of multi-agent technology itself. In fact, the use of the JADE framework is only for the purpose of realizing communication between nodes, but this communication is not knowledge-driven and no "agent-typical" mechanism such as learning and negotiation is used.

The work in stroke-based rendering (a special method of Non Realistic Rendering) proposed by Schlechtweg et al. [12] makes use of a multi agent system for rendering artistic styles such as stippling and hatching. The environment of the agents consist of a source image and a collection of buffers. Each agent represents one stroke and execute its painting function in the environment.

2.1 Comparison to MAgArRO

MAgArRO, the multi-agent approach to rendering proposed in this paper, significantly differs from all related work on rendering in its unique combination of the following key features:

- **Decentralized control.** MAgArRO realizes rendering in a decentralized way through a group of agents coordinated by a manager, where the group can be formed dynamically and most services can be easily replicated. (As regards decentralized control, MAgArRO follows the principle of volunteer computing [6].)

- **Higher level of abstraction.** While other approaches typically realize parallel optimization at a low level of abstraction that is specific to a particular rendering method, MAgArRO works with *any* rendering method. All that is required by MAgArRO are the input parameters of the render engine to be used.

- **Use of expert knowledge.** MAgArRO employs Fuzzy Set Rules and their descriptive power [4] in order to enable easy modelling of expert knowledge about rendering and the rendering process.

- **Local optimization.** Each agent involved in rendering can employ different models of expert knowledge. In this way, and by using a fine-grained decomposition approach, MAgArRO allows for local optimization of each rendering (sub-)task to be done.

3. THE MAGARRO APPROACH

MAgArRO is a system which gets a 3D Scene as input and produces a resulting 2D image. From the point of view of the user the system works in the same way as local render engines do, but the rendering in fact is made by different agents spread over the Internet.

MAgArRO uses the ICE middleware [2]. The location service IceGrid is used to indicate in which computers the services reside. Glacier2 is used to solve the difficulties related with hostile network environments, making available agents connected through a router and a firewall.

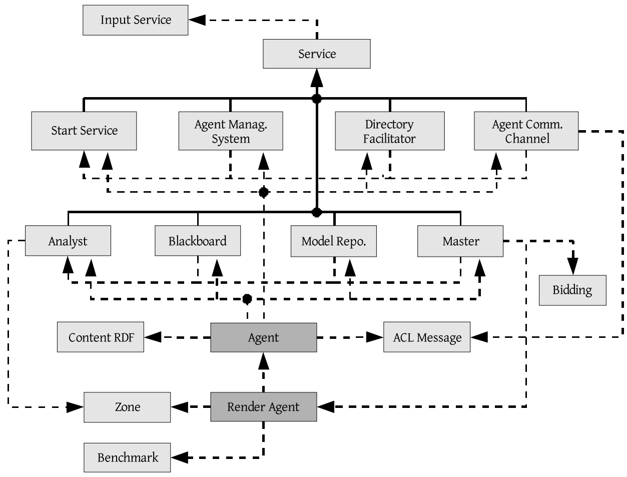


Figure 1: Magarro general Class diagram.

The general architecture of our system is based on the design principles of FIPA standard [wFip]. In figure 1 the general class diagram for the architecture is shown. There are some top-level general services (*Start Service*, *Agent Management System*, *Directory Facilitator* and *Agent Communication Channel*) available to all involved agents. On start-up, an agent is provided with a root service which describes or points to all the services available in the environment.

3.1 Architectural Overview

In MAgArRO a *StartService* offering two operations is available: an operation called *getServiceRoot* to obtain directly a description of all basic services, and another operation called *supplyBasicService* that enables the registration of a new basic service in the system. In accordance with FIPA, a basic service is defined by a unique Service Identify (string in our case), a list (one or more) of Transport Addresses and a description of Service Type.

The *Agent Management System* (AMS) is a general service which manages the events that occurs on the platform. This service also includes a naming service for *White Pages* which allow agents to find one another. The basic functionality of the AMS is to register, to modify a subscription, to unregister agents, and to search for agents.

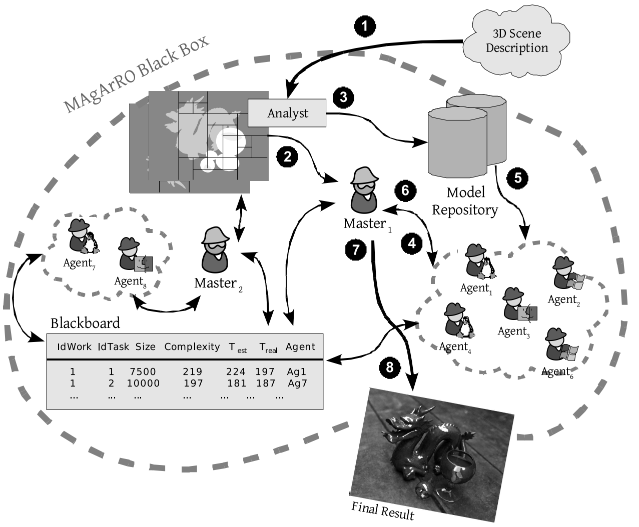
A basic service called *Directory Facilitator* (DF) provides *Yellow Pages* for the agents. As suggested by the FIPA standard, the operations of this service are related to the services provided by an agent, the interaction protocols, the ontologies, the content languages used, the maximum live time of registration and visibility of the agent description in DF.

Finally, MAgArRO includes a basic service is called *Agent Communication Channel* which receives and sends messages between agents. In fact, the only functionality needed is the ability to receive messages because the send operation is implemented in the agent. In accordance to FIPA standard, the data structure of each message is composed of two parts: the content of the message and the envelope (with information about the receiver and the sender of the message). The Agent Communication Language used in MAgArRO is based on XML and uses DTD as specified in the FIPA standard.

Each agent must implement a basic set of standard operations to be able to run in the MAgArRO environment. This operations are suspend, terminate, resume and receive a message.

In addition to the basic FIPA services described above, MAgArRO includes specific services related with Rendering Optimization are exposed. A service called *Analyst* studies the scene in order to enable the division of the rendering task. A blackboard is used to represent some aspects of the common environment of the agents. The environmental models processed by the agents are managed by the *Model Repository Service*. Finally, a manager service called (*Master*) handles dynamic groups of agents which cooperate by fulfilling subtasks. The Figure 2 illustrates this.

Figure 2 also illustrates the basic workflow in MAgArRO (the circled numbers in this figure represent the following steps). **1** – The first step is the subscription of the agents to the system. This subscription can be done at any moment; the available agents are managed dynamically. When the system receives a new file to be rendered, it is delivered to the Analyst service. **2** – The Analyst analyzes the scene, making some partitions of the work and extracting a set of tasks. **3** – The Master is notified about the new scene which is sent to the Model Repository. **4** – Some of the agents available at this moment are managed by the Master and notified about the new scene. **5** – Each agent obtains the 3D model from the repository and an auction is started. **6** – The (sub-)tasks are executed by the agents and the results are sent to the Manager. **7** – The final result is composed by the Manager using the output of the tasks previously done. **8** – The Manager sends the rendered image to the user. Some of the more important steps of this work flow will be studied in the next sections.



IdWork	IdTask	Size	Complexity	T_{est}	T_{real}	Agent
1	1	7500	219	224	197	Ag1
1	2	10000	197	181	187	Ag7
...	...	...	...	...	...	...

Figure 2: General workflow and main architectural roles.

3.2 Agent Subscription

As is shown in figure 1, a *Render Agent* is a specialization of a standard *Agent*, so all the functionality and requirements related with FIPA are inherited in its implementation. There are two actions that could be done by an agent to add a subscription or to unsubscribe in a group of rendering. This operations are related with one of the Masters of the architecture. The subscribe operation requires two parameters; the name of the agent and the *proxy* that represents the agent as a client. Using this proxy the Master could execute remote operations in the agent side. The agent could remove the association with one manager by means of unsubscribe operation.

The first time one Agent subscribes to the system, it runs a benchmark to obtain an initial estimation (a first approach) about its computing capabilities. This initial value will be modified in rendering time in order to obtain a more precise predictions.

3.3 Analysis of the Scene based on Importance Maps

MAgArRO employs the idea to estimate the complexity of the different tasks in order to achieve load-balanced partitining. Complexity analysis is done by the Analyst agent prior to (and independent of) all other rendering steps.

At the beginning, the Analyst makes a fast rasterization of the scene using an importance function to obtain a grey scale image. In this image (called *Importance Map*) the dark zones represents less complex areas and the white zones the more complex areas. In our current implementation a simple function is used (only takes in account the recursion levels in mirror and transparent surfaces). As it is shown in Figure 3 , the glass is more complex that the dragon because it has mirror and transparency with a greater number of ray interactions than other objects. The table is less complex because it does not have any of these properties. More advanced importance functions could be used in this grey scale image generation, using perception-based rendering algorithms (based on visual attention processes) to construct the importance map [13].

Once the importance map is generated, a partition is con-

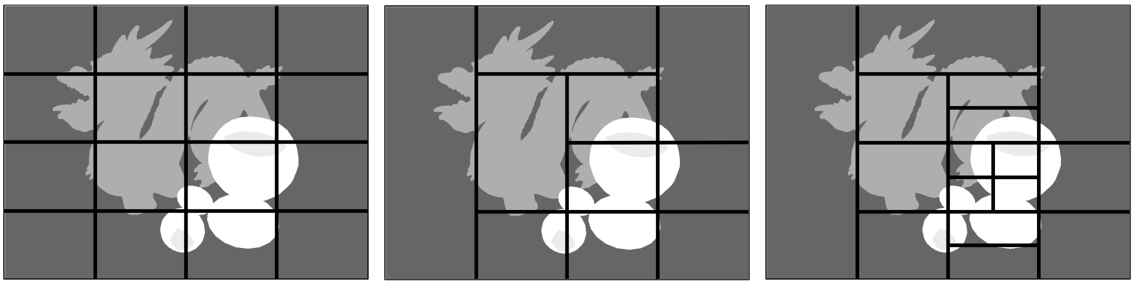


Figure 3: Importance maps. *Left:* Blind partitioning (First Level). *Center:* Join zones with similar complexity (Second Level). *Right:* Balancing complexity/size ratio (Third Level).

structed to obtain a final set of tasks. These partitions are formed hierarchically at different levels, where at each level the partitioning results obtained at the previous level are used. At the first level, the partition is made taking care of the minimum size and the maximum complexity of each zone. With this two parameters, the *Analyst* makes a recursive division of the zones (see Figure 3). At the second level, neighbor zones with similar complexity are joined. Finally, at the third level the *Analyst* tries to obtain a balanced division with almost the same complexity/size ratio of each zone. The idea behind this division is to obtain tasks which require roughly the same rendering time. As will be shown in Section 4, the quality of this partitioning is highly correlated to final rendering time.

3.4 Rendering Process

Once the scene is available in the *Model Repository*, the *Master* assigns agents to the individual tasks identified by the Analyst. These agents, in turn, apply in parallel a technique called profiling in order to get a more accurate estimation of the complexity of each task.[1] Specifically, the agents make a low resolution render (around 5% of the final number of rays) of each task and announce in the *Blackboard* the estimated time required to do the final rendering on the blackboard.

3.4.1 Blackboard service

The blackboard used by the agents to share their knowledge about the rendering task is realized as a service offering a set of read and write operations. The basic blackboard data structure (as shown in Figure 2) has 7 fields labelled as follows. *IdWork* is a unique identifier for each scene, *IdZone* is a unique identifier for each task of each work, *Size* is the number of pixels of each task (width x height), *Complexity* is the estimated complexity of this task, calculated by means of the importance map, T_{est} is the Estimated Time calculated by means of profiling technique by the agents, T_{real} is the real time required to finish a task (only have a value when the task is finished), and *Agent* is the name of the agent that is rendering this task.

3.4.2 Simple Adaptation

As said above, the estimated time is represented in the same way by all agents. More precisely, each agent has an internal variable that represents its relative computational power (V_{cp}). For example, assume that $Vcp = 1$ is chosen as a reference value. If a task T_A requires 5 minutes to be done by a particular agent A that has $V_{cp} = 0.5$, it annotates on the blackboard that the time required to do T_A is 10 minutes (e.g. because this agent is running in a very fast machine). On the other hand, if another agent B that is running on a very slow machine ($V_{cp} = 2$) estimates that the task T_B will require 2 minutes of processing time on its machine and announces this information on the blackboard, then agent A can infer from this that T_B is 30 seconds on its machine.

During run time, each agent adapts the value of its variable V_{cp} to obtain a more accurate estimation of the required processing time as follows. Whenever there is a difference between the estimated time T_{est} and the actual completion time T of a task, then an agent updates its internal variable V_{cp} according to

$$V_{cp} = (1 - k) \times V_{cp} + k \times (T - T_{est}) \qquad (1)$$

where k a constant. Small values of k assure a smooth adaptation. (k is set to 0.1 in the experiments reported below.)

3.4.3 Auctioning

In each moment of the execution of the system, all agents that are idle take part in an auction for available tasks. All agents try to obtain more complex tasks first. If two or more agents bid for the same task, the Master assign it to one of them taking in account two factors:

The number of credits of this agent. This parameter represents the success and failures of the agent in previous tasks. An agent have success in one task if the task is finished in a time less or equal than Test. In other case, the agent have a failure. The amount of credits added or subtracted to the agent are proportional to the difference of time with Test.

The historical behavior. This sequence represents the latests list of success and failures made by the agent. This is used to ponder the more recent activity of the agent in order to assign new tasks.

3.4.4 Using Expert Knowledge

When a task is assigned to an agent, a fuzzy rule set is used in order to model the expert knowledge and optimize the rendering parameters for this task. The choose of fuzzy rule sets as the method for modelling the expert knowledge is because the facility of description and extension. The output parameters (the consequent part of the rules) are configured to decrease the time required to finish the render and minimize the lose of quality. Each agent could model different expert knowledge with a different set of fuzzy rules. In the following part, we will study the rule sets made for

[1] Profiling is a technique with traces a small number of rays in a global illumination solution and uses the time taken to compute this few rays to make a prediction over the whole computation time.

Pathtracing rendering method. The output parameters of the rules are:

- **Recursion Level** [*Rl*], defined over the linguistic variables [17] {VS, S, N, B, VB}[2]. This parameter defines the global recursion level in raytracing (number of light bounces).

- **Light Samples** [*Ls*], defined over the linguistic variables {VS, S, N, B, VB}. This parameter defines the number of samples per light in the scene. The biggest, the more quality in the scene and the higher rendering time.

- **Interpolation Band Size** [*Ibs*], defined over the linguistic variables {VS, S, N, B, VB}. This parameter defines the size of the interpolation band in pixels, and it is used in the final composition of the image (as we will see in the next section).

The previous parameters have a strong dependency with the rendering method, in this case Pathtracing. The next parameters we will study are the antecedents of the rules, and could be used with other rendering methods but changing the description of the rules.

- **Complexity** [*C*], defined over the linguistic variables {VS, S, N, B, VB}. This parameter represents the complexity/size ratio of the task.

- **Neighbor Difference** [*Nd*], defined over the linguistic variables {VS, S, N, B, VB}. This parameter represents the difference of complexity of the current task in relation with its neighbor tasks.

- **Size** [*S*], defined over the linguistic variables {S, N, B}. This parameter represents the size of the task in pixels (calculated as width x height).

- **Optimization Level** [*Op*], defined over the linguistic variables {VS, S, N, B, VB}. This parameter is selected by the user, and determines the level of optimization (more or less aggressive with initial parameters indicated by the user).

The definition of the fuzzy sets of input variables is made dynamically; the intervals of this sets are calculated in runtime. For example, in a high complex scene, the definition of VS should be higher that in a simple scene the corresponding definition of VB. The partition of this variables is made by linear distribution. The same occurs with other parameters like Size and Neighbor difference. In the case of the Pathtracing method, the rule set is defined as follows (only a two rules of 28 are presented). These rules have been written by an expert in PahtTracing.

R_1**: If** C is {B,VB} $\wedge$ S is B,N $\wedge$ Op is VB **then** Ls is VS $\wedge$ Rl is VS $R_2$2**: If** Nd is VB **then** Ibs is VB, etc.

The output variables have their own fuzzy sets. We use trapezoidal functions as is shown in Figure4.

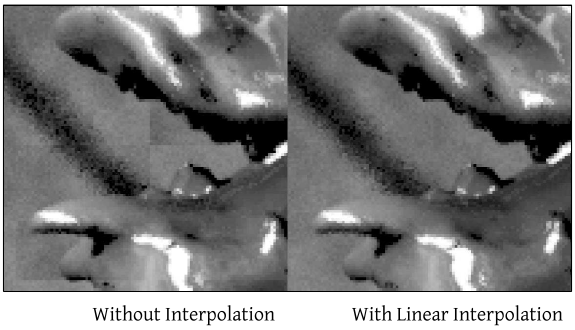


Figure 5: The artefacts appears without interpolation between tasks. This problem is solved with linear interpolation.

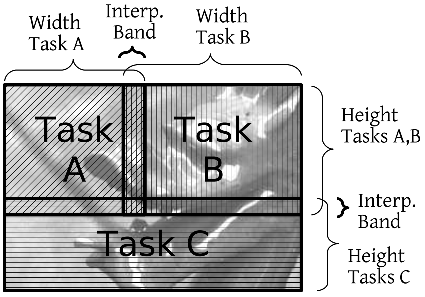


Figure 6: Diagram of task decomposition and interpolation band situation.

3.5 Final Result Composition

With the results generated by the different agents, the *Master* can compose the final image. This process is not made directly because slight differences between fragments can exist when obtained from distinct agents (this is due to the random component of Monte Carlo based methods such as Pathtracing). Figure 5 illustrates this. For that reason, it is necessary to smooth the joint between fragments which are neighbors using a lineal interpolation mask (Figure 6). In this way, a zone used to combine with other tasks is defined (and called Interpolation Band).

The size of the Interpolation Band is an output parameter of the rule set. This parameter should be bigger if the difference of quality between neighbor zones are important. To have better time results, this parameter should be maintained as small as possible because it is work repeated by more than one agent. This is specially important if the zone is very complex because the cost of rendering this interpolation band is also so high. The amount of time wasted in the interpolation band in the current architecture is between 2% and 5% of the final rendering time.

4. EXPERIMENTAL RESULTS

The results in this section have been generated with the implementation of MAgArRO which we have made available for download at [3] under GPL Free Software License.

In order to test the behavior of the system, 8 computers with the same hardware characteristics have been connected to the architecture. The computers are Pentium Intel Centrino 2 Ghz, 1GB RAM and Debian GNU/Linux. The rendering method used in all cases was Pathtracing (*Yafray* 0.0.9 render engine), 8 oversampling levels, 8 recursion lev-

[2] The notation used for the linguistic variables is typical in some works with Fuzzy Sets. This is the correspondence of the linguistic variables: VS is *Very Small*, S is *Small*, N is *Normal*, B is *Big* and finally VB is *Very Big*.

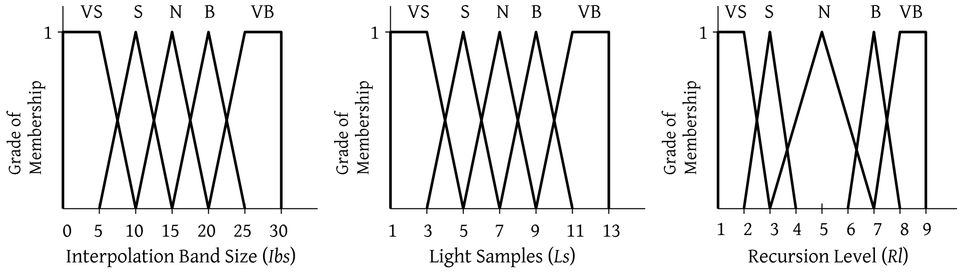


Figure 4: Definition of the output variables.

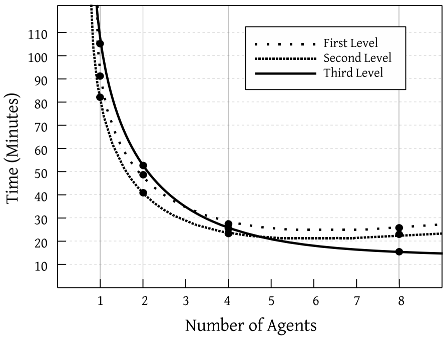


Figure 7: Different partitioning with (*Normal* optimization level).

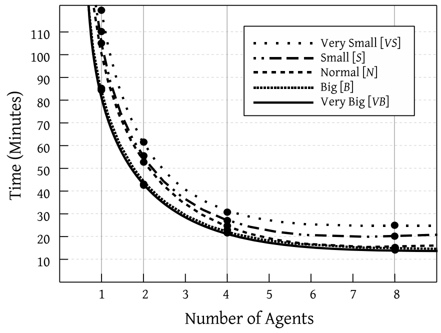


Figure 8: Different optimization levels (all with third level of partitioning).

els in global configuration of raytracing, 1024 Light samples by default. The scene to be rendered contain more than 100.000 faces, 5 levels of recursion in mirror surfaces and 6 levels in transparent surfaces (the glass). With this configuration, and making a rendering in one machine without any optimization, the rendering time was **121:17**[3].

First of all, we will study the importance of a good task division scheme. In table 1 the time required using different partitioning levels is shown. All of these times have been obtained using *N* (*Normal*) optimization level (Figure 7). Using a simple partitioning of first level, we could obtain good render time with few agents in comparison with third level of partitioning. The time in third level is bigger because we are making more partitions in the areas with more complexity (in the glasses). This higher number of partition requires interpolation bands and to repeat the rendering of some part of these areas in various tasks. This situation imply that some of the more complex work is made more than once, and for example, the rendering time with one agent in third level is *105* minutes and in first level near *93* minutes. However, when the number of agents grow, the performance of the system comes better because there are no too big differences of complexity between tasks. In partitioning of first and second level, there are complex tasks that slow down the finish of the whole rendering, and it doesn't matter the increase in the number of agents (time required with four or eight agents are essentially the same). The third partitioning level works better with a higher number of agents.

Table 1: Different partitioning with *Normal* optimization level.

Agents	1^{st} Level	2^{nd} Level	3^{rd} Level
1	92:46	82:36	105:02
2	47:21	41:13	52:41
4	26:23	23:33	26:32
8	26:25	23:31	16:52

Table 2: Third level of partitioning with different Number of Agents and level of optimization.

Agents	VS	S	N	B	VB
1	125:02	110:50	105:02	85:50	85:06
2	62:36	55:54	52:41	42:55	42:40
4	31:10	27:11	26:32	22:50	22:40
8	23:43	20:54	16:52	16:18	15:58

Table 2 shows the time required to render the scene using different levels of optimizations, always using a third level of partitioning (Figure 8). By simply using a *Small* level of optimization we obtain better results than making a render without optimizations. The time required with *Very Small* optimizations is greater than rendering the original scene because there are some times required in the communication and composing the results, that, in this case are greater than the time needed to render the scene only in one task with original values.

Excellent results are also obtained when only four agents are used. For instance, in the case of Normal level optimization, the time required to render the scene is just about 26 minutes (whereas the original rendering time is 120 minutes). Figure 9 shows the results of rendering obtained for different configurations.

As a final remark, note that optimization may result in different quality levels for different areas of the overall scene. This is because more aggressive optimization levels (i.e., Big or Very Big) may result in a lose of details. For example, in figure 9.e, the reflections on the glass are not so detailed as in Figure 9.a. Such quality problems can appear in different parts of the render (depending on the scene), and can be avoided by using more complex functions to develop the

[3] All times in this paper appears in the format *Minutes:Seconds*

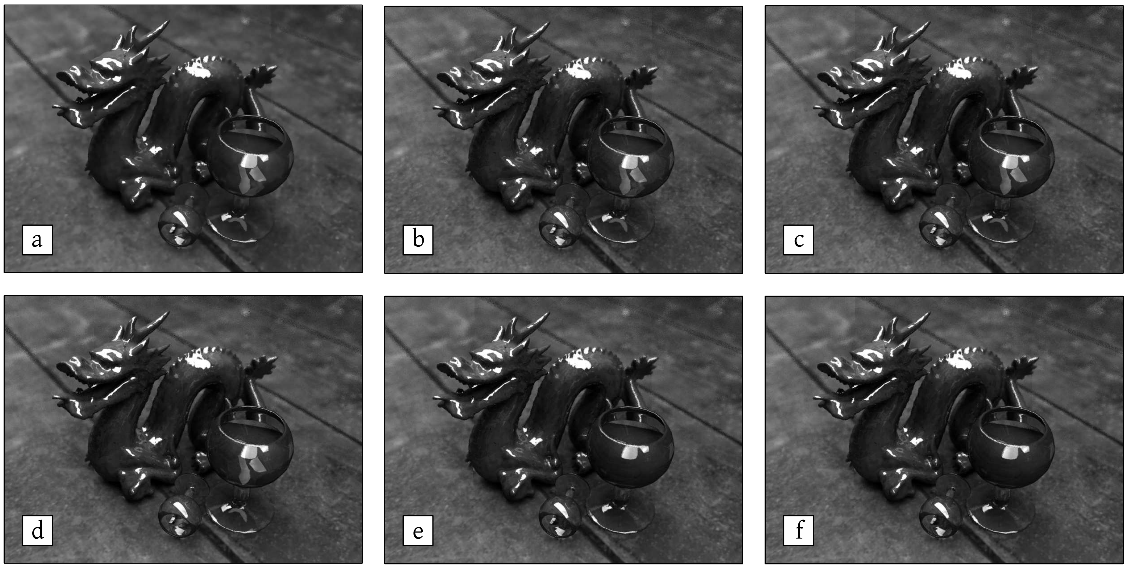


Figure 9: Result of the rendering using different optimization levels. (a) No optimization and render in one machine. (b) *Very Small* (c) *Small* (d) *Normal* (e) *Big* (f) *Very Big.*

Importance Map and/or less aggressive optimization methods.

5. DISCUSSION AND CONCLUSION

The media industry is demanding high fidelity images for their 3D scenes. The computational requirements of full global illumination are such that it is practically impossible to achieve this kind of rendering in reasonable time on a single computer. MAgArRO has been developed in response to this challenge.

The experimental results show that MAgArRO achieves excellent optimization results. In particular, MAgArRO achieves overall rendering times which are below the time required by one CPU divided by the number of agents. In addition to that, MAgArRO is a multi-agent approach which offers several desirable feature which together make it unique and of highest practical value. In particular:

- It is FIPA-compliant.
- It can be used on different hardware platforms and under different operating systems (including GNU/Linux, MacOSX, Windows, etc.).
- It enables importance-driven rendering through its use of importance maps.
- It employs effective auctioning and parameter adaptation, and it allows the application of expert knowledge in form of flexible fuzzy rules.
- It applies the principles of decentralized control and local optimization, and thus is scalable and very robust e.g. against hardware failures.

MAgArRO is pioneering in its application of multi-agent principles to 3D realistic rendering optimization. This open several new research avenues. Specifically, we think it is very promising to extend MAgArRO toward more sophisticated adaptation and machine learning techniques. Another promising avenue is to develop a Grid version of MAgArRO in which the agents can reside and run on different machines around the world. Such a Grid version could be a very effective answer to the challenge of handling rendering complexity.

In our current work, we concentrate on two research lines. First, the combination of different rendering techniques within the MAgArRO framework. Due to the high abstraction level of MAgArRO, in princple different render engines can be combined to jointly generate an image, using complex and realistic techniques only if needed. Second, we are exploring the possibilities to equip MAgArRO with agent-agent real-time coordination schemes which are more flexible than auctioning.

6. ACKNOWLEDGMENTS

This work has been funded by the Junta de Comunidades de Castilla-La Mancha under Research Project PBC06-0064

7. REFERENCES

[1] Foundation for Intelligent Physical Agents FIPA. http://www.fipa.com.

[2] ICE. Internet Communication Engine. http://www.zeroc.com.

[3] MAgArRO SVN Repository. http://code.google.com/p/masyro06/.

[4] *An introduction to fuzzy logic for practical applications.* Springer, 1998.

[5] *GPU Gems 2: Programming Techniques for High Performance Graphics and General-Purpose Computation*, chapter High-Quality Global Illumination Rendering using Rasterization. Addison-Wesley Professional, 2005.

[6] D. P. Anderson and G. Fedak. The computational and storage potential of volunteer computing. In *Sixth IEEE International Symposium on Cluster Computing and the Grid (CCGRID'06).*

[7] A. Chalmers, T. Davis, and E. Reinhard. AK Peters Ltd, July 2002.

[8] J. A. Fernandez-Sorribes, C. Gonzalez-Morcillo, and L. Jimenez-Linares. Grid architecture for distributed rendering. In *Proceedings of Ibero-American Symposium in Computer Graphics 2006.*

[9] R. Gillibrand, K. Debattista, and A. Chalmers. Cost prediction maps for global illumination. In *Proceedings of Theory and Practice of Computer Graphics 2005*, pages 97–104. Eurographics Association, June 2005.

[10] J. T. Kajiya. The rendering equation. *Computer Graphics*, 20(4):143–150, Aug. 1986. Proc. of SIGGRAPH'86.

[11] R. R. Kuoppa, C. A. Cruz, and D. Mould. Distributed 3d rendering system in a multi-agent platform. In *Proceedings of the Fourth Mexican International Conference on Computer Science (ENC'03).*

[12] S. Schlechtweg, T. Germer, and T. Strothotte. Renderbots – multiagent systems for direct image generation. In *Computer Graphics Forum.*

[13] V. Sundstedt, K. Debattista, P. Longhurst, A. Chalmers, and T. Troscianko. Visual attention for efficient high-fidelity graphics. In *Spring Conference on Computer Graphics (SCCG 2005).*

[14] J. S. Sven Woop and P. Slusallek. Rpu: A programmable ray processing unit for realtime ray tracing. In *Proceedings of ACM SIGGRAPH 2005 (to appear)*, July 2005.

[15] E. Veach and L. J. Guibas. Metropolis light transport. In *SIGGRAPH '97: Proceedings of the 24th annual conference on Computer graphics and interactive techniques*, pages 65–76, New York, NY, USA, 1997. ACM Press/Addison-Wesley Publishing Co.

[16] T. Whitted. An improved illumination model for shaded display. In *SIGGRAPH '79: Proceedings of the 6th annual conference on Computer graphics and interactive techniques*, page 14, New York, NY, USA, 1979. ACM Press.

[17] L. A. Zadeh. The concept of a linguistic variable and its applications to approximate reasoning. part i, ii, iii. *Information Science*, 1975.

Technical Report
SCCH–TR–xxxx

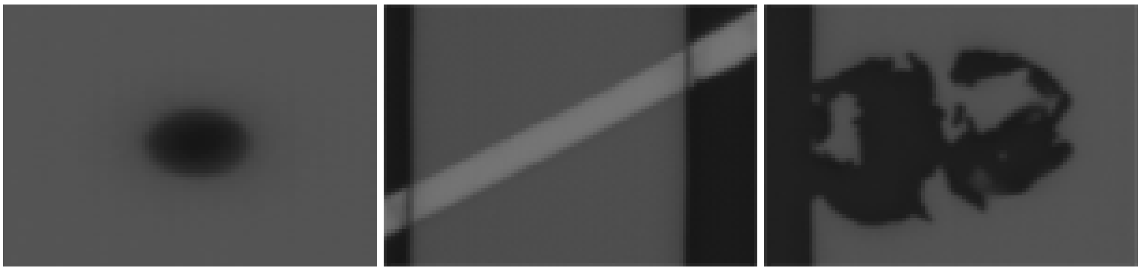
High Performance Surface Inspection Method for Thin-Film Sensors

Volkmar Wieser, Felix Kossak, Stefan Larndorfer and Bernhard Moser Software
Competence Center GmbH, 99 Hauptstrasse, Hagenberg, Austria 4232;
Software Competence Center Hagenberg
A-4232 Hagenberg, Austria
volkmar.wieser@scch.at

Abstract — Thin-film sensors for use in automotive or aeronautic applications must conform to very high quality standards. Due to defects that cannot be addressed by conventional electronic measurements, an accurate optical inspection is imperative to ensure long-term quality aspects of the produced thin-film sensor. In this particular case, resolutions of 1 μm per pixel are necessary to meet the required high quality standards. In this paper, a new method is proposed that solves the problem of handling local deformations due to production variabilities without involving the compensation of local image registration operations. The main idea of this method is based on a combination of efficient morphological preprocessing and a multi-step comparison strategy based on logical implication. The main advantage of this approach is that the neighborhood operations that care for the robustness of the image comparison can be computed in advance and stored in a modified master image. By virtue of this approach, no further neighborhood operations have to be carried out on the acquired test image during inspection time. As a result, the requirements of high-resolution inspection and high-performance throughput while accounting for local deformations are met very well by the implemented inspection system.

Key words — *Thin-film sensor, surface inspection, high-resolution inspection, template matching, high-performance inspection, quality assurance*

Kompetenzzentren-Programm



(a) pin-hole

(b) bright defect on brigth and dark background

(c) dark defect on bright background

Figure 1: Examples of defects which are able to occur on thin-film sensors

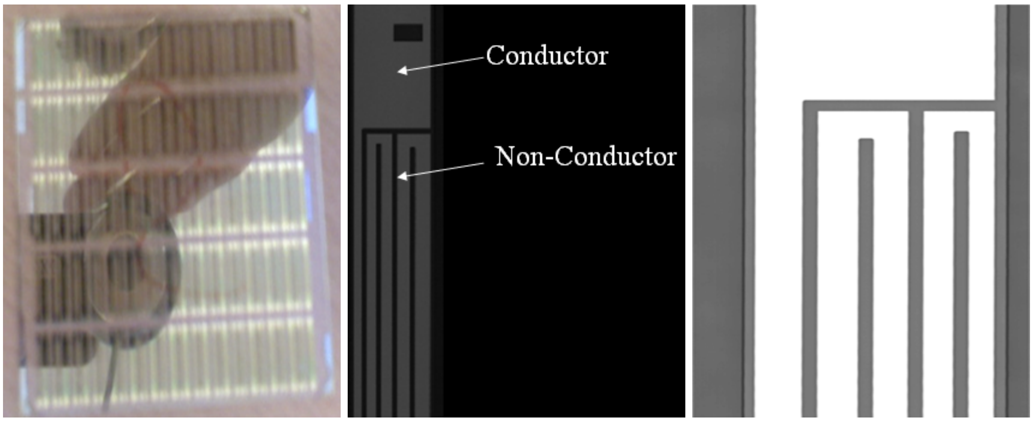
1 INTRODUCTION

This paper introduces a new optical method for checking the surface of thin film sensors. We compare this method with state-of-the-art methods of optical inspection [5] [2] [4] in terms of computational efficiency, robustness as well as simplicity of configuration.

Thin-film sensors for automotive or aeronautic systems must conform very high quality standards [3]. Due to defects that cannot be addressed by conventional electronic measurements, an accurate optical inspection is necessary to ensure long-term quality aspects of the produced thin-film sensor. In this particular case, resolutions of 1 μm per pixel are needful to meet the required high quality standards. Furthermore, it has to be guaranteed that defects are detected robustly with high reliability [6].

Particularly, the multi-layer coating of the sensor, which also consists of partly transparent coats, leads to a great variety of defects. The shape of the defects range from small pin-holes to scratch-like fine hair cracks which generally can appear as bright defects on dark background, dark defects on bright background, dark defects on dark background, or bright defects on bright background. Additionally, there are some global, small deviations in the appearance and local deformations of the geometric elements on the sensor due to production variabilities. For example, the thickness of the conductor paths slightly varies over time due to the limited accuracy in the production process. Figure 1 shows examples of different defects.

Therefore it is a main challenge for the inspection system to differentiate robustly between small deviations due to production variabilities and potentially small sensor defects. This Problem is basically an image registration problem as deviations to a master image have to be classified as deformations (due to variations in production process). The process of aligning is based on image registration techniques which may also compensate for local deformations. The computational time consumption of methods aiming at correcting such local deformations turns out to be crucial for the overall performance of the inspection system. When trying to achieve robustness, such methods usually lead to a higher overall complexity of the process, resulting in higher computational costs and increased configuration effort.



(a) Substrat

(b) Sensor

(c) Sensor with different non-conductors

Figure 2: Examples of substrate and sensors

1.1 Examples of Substrat and Thin-Film Sensors?

Figure 2(a) shows an example of a substrat. On each substrate are up to 300 sensors, where 2(b) is a schematic representation of one sensor. Each sensor has conductors and non-conductors (see figure 2(b)). In our case the non-conductor has two different grey values, which is better shown in 2(c). With increased brightness we can see, that there are two different grey intensities for non-conductors. On the one hand the grey values and on the other hand the black lines. This is important for the following calculation steps.

Summarizing we can define several properties of substrate and sensors.

- One substrate has 200 - 300 sensors
- Dimension of one substrate is about 51mm x 51mm
- Dimension of one sensor is max 1.2mm x 8mm
- Minimal width of conductor is 25μm
- Manuel inspection time takes 5 minutes per substrate
- Non-conductors have two different grey values

In the next sections we want to indroduce our proposed method and give some results.

2 The proposed method

Cross correlation is one of the most used correlation measure in template matching [5], but the matching process with high resolution images can be computationally very expensive. Equation 1 shows the complexity of cross correlation.

$$O(N * M * n^2) \tag{1}$$

Due to the fact, that our images have up to 1500pixel x 8000pixel, we need an efficient algorithm. In normal case, all image processing operations, will be done during the surface inspection. This is time-consuming and not efficient. Our approach is to do some preprocessing on a master image. This preprocessing has no influence to inspection time, because the calculation can be done before the surface inspection. We call this step "‘OFFLINE"’. Now we have fewer image processing operations during the inspection. We call operations during the inspection time "‘ONLINE"’. This separation allows us to get a fast system, which has a higher inspection rate than a human visual inspector. Therefore we can split our approach in two seperate steps:

1. *OFFLINE*: preprocessing and generation of two template images with binarization and erosion.
2. *ONLINE*: compare template image and test image with a negated logical implication.

After the splitting we have in the relevant Online inspection a lower complexity, which is shown in equation 2:

$$O(N * M * 2) \tag{2}$$

After the reduction of the complexity we get speedup advantage which example is shown in equation 3:

$$Speedup = \frac{n^2}{2} = 12.5 \text{ for } n = 5; \tag{3}$$

In the next subsections we want to elaborate preprocessing of a master image and test image which we need for template matching.

2.1 Preprocessing of master images

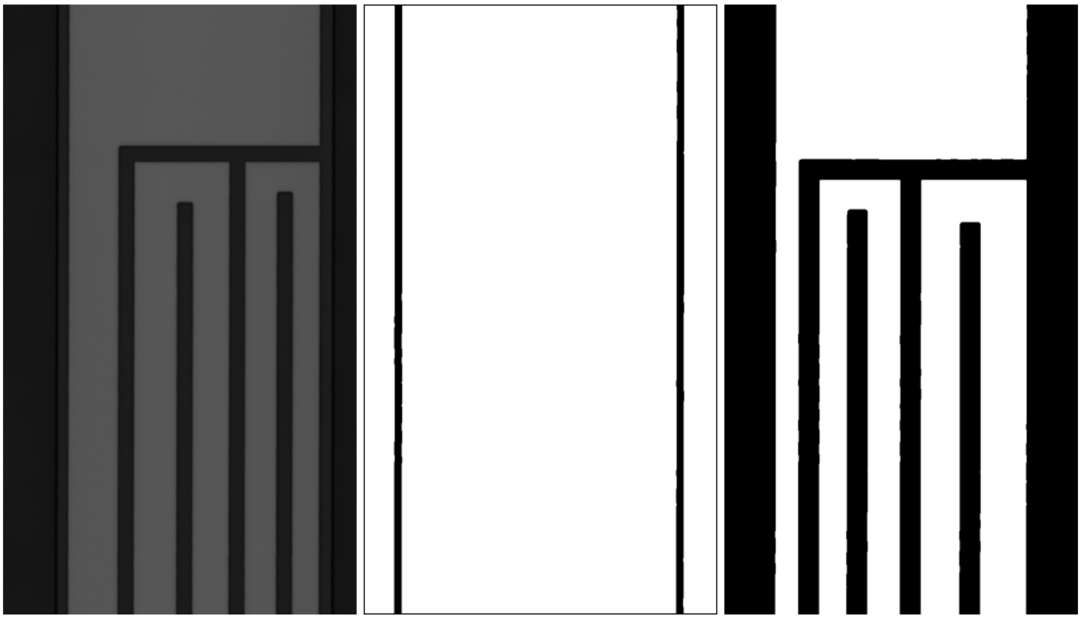
First of all, a master image is an optimal image without any defects. For a easier comparison we want to binarize the master image. Figure 2 shows that the non-conductors have two different grey values, therefor we have to generate two binarized images with the threshold of the grey value intensity of the non-conductors.

After that, the non conductors in figure 3(b) and figure 3(c) will be enhanced by an erosion, which is defined in equation (4). The enlargement of the non conductors effectuates that production variabilities have no influence on the defect detection. The larger the production variabilities the larger must be the structure element.

$$I \ominus S = \{(x, y)|(x, y) + S \subseteq I\} \tag{4}$$

Where I is the image and S is the structure element.

All calculation we have done up to now are offline calculations and have no influence to inspection time. In the next section we preprocesses the test images, which are online calculations.



(a) Original master image (M_I) (b) Master image with low threshold (M_D) (c) Master image with high threshold (M_L)

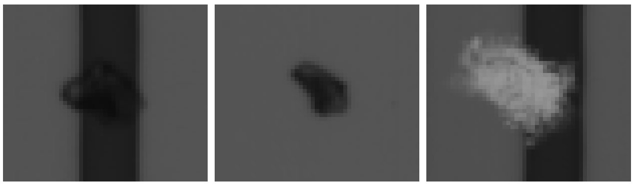
Figure 3: Master images

2.2 Preprocessing of test images

To make a binarization of the test images we have to define three classes of defects:

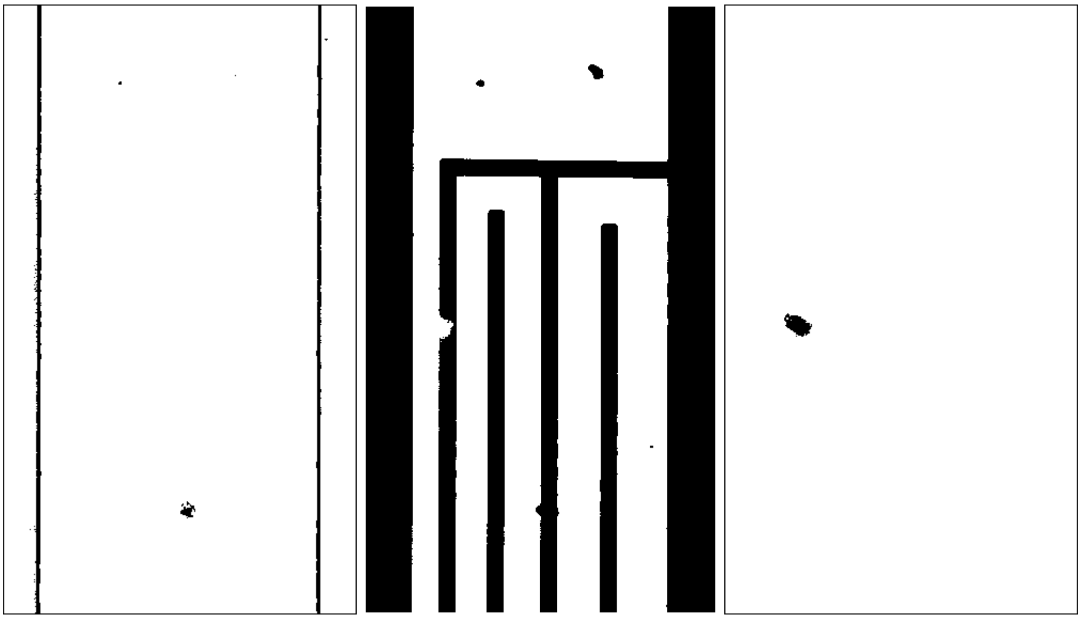
If we know the grey value intensity for each class, we generate for every class a binary image from test image. We map the image to $[0,1]$ in that way, that the defects are always black. Figure 5 gives an example about the generated test images, where t_x is the threshold.

Now we have finished all preprocessing steps and can make a simple logical comparison between master image and test image, which is shown in the next section.



(a) defects with low color intensity (dark) (b) defects with middle color intensity (grey) (c) defects with high color intensity (bright)

Figure 4: Defects with different color intensities



(a) Test image with t_{16} (b) Test image with t_{50} (c) Test image with t_{80}

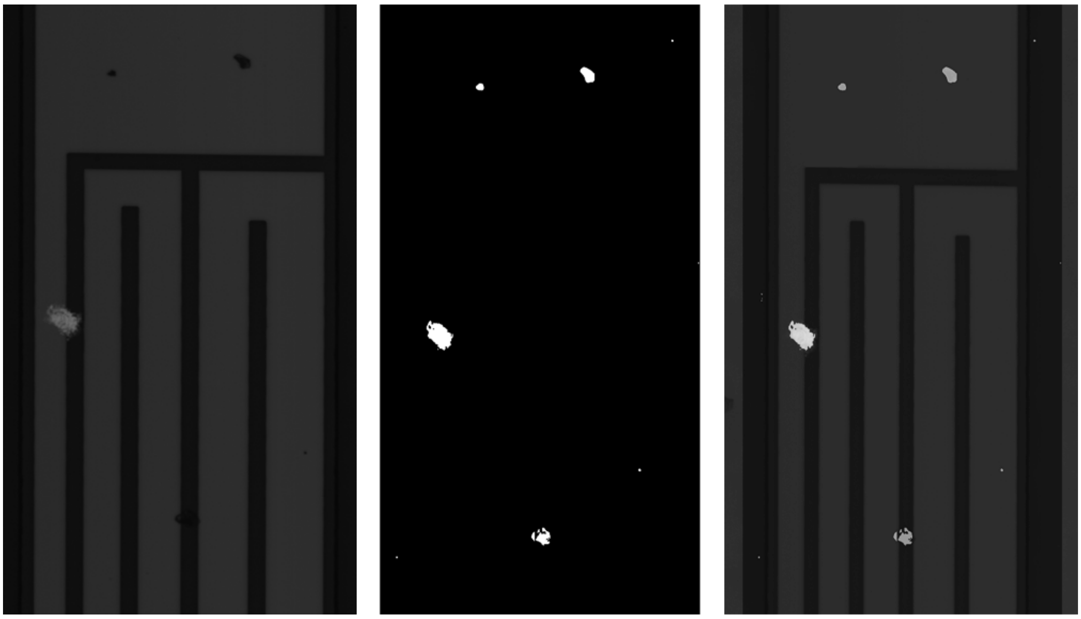
Figure 5: Generated test images with different thresholds

2.3 Comparison of master image and test image

Before we can start with the online inspection we must consider which types of defects can appear. For every founded case, we need a separat request. Due to the fact, that we have only grey values, we can define exemplarity four possible defects:

- Dark defect on bright background (D_{db})
- Bright defect on bright background (D_{bb})
- Dark defect on dark background (D_{dd})
- Bright defect on dark background (D_{bd})

Now we can decide by means of a negated logical implication if there is a defect in the image or not. Consider M(x,y) and T(x,y) is an intensity value in the corresponding image, then it turn out that (x,y) has to be classified as defect if M(x,y) = 1 and T(x,y) = 0.



(a) original Image (b) only defects (c) overlayed images

Figure 6: Result of defects detecting

$$\begin{array}{ccc} M_L & T_{50} & D_{db} \\ \hline 0 & 0 & 0 \\ 0 & 1 & 0 \\ 1 & 0 & 1 \\ 1 & 1 & 0 \end{array} \tag{5}$$

$$\begin{array}{ccc} M_D & T_{16} & D_{dd} \\ \hline 0 & 0 & 0 \\ 0 & 1 & 0 \\ 1 & 0 & 1 \\ 1 & 1 & 0 \end{array} \tag{7}$$

$$\begin{array}{ccc} M_L & T_{80} & D_{bb} \\ \hline 0 & 0 & 0 \\ 0 & 1 & 0 \\ 1 & 0 & 1 \\ 1 & 1 & 0 \end{array} \tag{6}$$

$$\begin{array}{ccc} M_D & T_{80} & D_{bd} \\ \hline 0 & 0 & 0 \\ 0 & 1 & 0 \\ 1 & 0 & 1 \\ 1 & 1 & 0 \end{array} \tag{8}$$

Where T_x is the test image with the corresponding threshold.

3 RESULTS

Figure 6 shows the result after the inspection. In this case, image 6(a) represents the original image with a various number of defects like pin-holes. The second image 6(b) displays only the detected defects. At this, white pixels marks the abnormalities between test image and master image. The last image 6(c) shows an overlay from images 6(a) and 6(b) for a better human comparison.

We used for the calculation a 3,2 Ghz machine with 2GB RAM and implemented the algorithm in C++, in additional with the framework "‘INTEL-IPP"’ [1]. Table 1 shows the difference of the calulation time between the cross corralation and our method.

Image dimension	Cross correlation	Our method
1500 x 8000	400 ms	13 ms

Table 1: Calculation time

References

[1] Intel integrated performance primitives
. http://www.intel.com/support/performancetools/libraries/ipp/index.htm.

[2] Chua-Chin Wang Chenn-Jung Huang, Chi-Feng Wu, editor. *Image Processing Techniques for Wafer Defect Cluster Identification*, volume 0740-7475/02. IEEE, 2002.

[3] Dominic F. Haigh Douglas W. Raymond, editor. *Why Automate Optical Inspection*, volume 1089-3539/97. IEEE, 1997.

[4] Cho The Gongyuan Qu, Sally L.Wood, editor. *Wafer Defect Detection Using Directional Morphological Gradient Techniques*, volume 686-703. Eurasip Journal on Applied Signal Processing, Hindawi Publishing Corporation, 2002.

[5] Martino Mola Luigi Di Stefano, Stefano Mattoccia, editor. *An Efficient Algorithm for Exhaustive Template Matching based on Normalized Cross Correlation*, volume 0-7695-1948-2/03. IEEE, 2003.

[6] Laura Peters. Defect detection for the 21st century. Semiconductor International, 1998.

Object Recognition in Deviation Images for Fault Detection - A Comparison of Methods

Edwin Lughofer, Roland Richter
Department of Knowledge-Based Mathematical Systems
Johannes Kepler University Linz
Alternbergerstrasse 69, A-4040 Linz
e-mail {edwin.lughofer,roland.richter}@jku.at

***Abstract*—In this application paper several algorithms are exploited for the purpose of object recognition in so-called *deviation images*. These images are obtained when calculating the deviation of a newly recorded image with its master. Usually, the master image represents a fault-free situation, specifying how an image should look like during the production process, which may deliver thousands or millions of duplications of the master image (for instance a print production process). Now, if clear deviations by pixel-wise comparisons are observed, it does not automatically mean that there is a fault in the production process at all. For instance, very small and tiny deviations triggering low grey levels as almost black pixels in the deviation images can be ignored as faults are not really visible in the original image. In this sense, it depends on the size, densities, shapes and outlooks of the objects in the deviation image formed by the deviation pixels, whether a faulty image is present or not. Hence, it is necessary to get a fully automatic correct extraction of objects in the deviation image, as otherwise the objects will not represent the current state of a recorded image correctly. Various algorithms and their combinations such as connected components with and without morphology, iterative prototype-based clustering and hierarchical clustering will be described in a red line throughout the paper. This should clearly demonstrate the applicability, feasibility and drawbacks of the approaches in case of different characteristics of objects in the deviation images. The paper is concluded with an evaluation of the described approaches by comparing the average number of extracted objects from a set of 20000 images, where the real number of objects are known. Furthermore, the evaluation includes a comparison of the methods within an image classification framework, which classifies images from a print production process into good ones (representing fault-free processes) or bad ones (representing faulty occasions). This is done based on a decision tree classifier trained from several image features which were extracted out of the recognized objects and should clearly underline the impact of object recognition methods on automated feature classification.**

***Index Terms*—automatic object recognition, deviation image, connected components, morphology, clustering, classification framework**

I. Motivation

In nowadays industrial systems, the recognition of objects in images plays a central role, whenever an automated pattern recognition or classification process is required. Examples of such image processing requirements are fingerprint identification, optical character recognition, DNA sequence identification, identification and transporting items with robots and so on. In this technical report a special emphasis is placed on fault detection in images, i.e. the detection of faulty artefacts in images. Hereby, it is not a matter about the origin of the fault artefacts (e.g. print/scan errors or production errors of items which are photographed), but it is more a matter how the faulty artefacts can be discriminated from other fault-free objects appearing in the images. One possible way, to solve this problem is to compare a newly recorded image with a (perfect) master image of the same item and to store the pixel-wise deviations into a grey-level image, where the intensity of the grey-levels reflect the degree of deviation. This referential method is one of the most common techniques of fault detection in image processing [3] [18]. Figure 1 shows an example of a faulty dot right beside the optical characters, the right image represents the deviation image from the master.

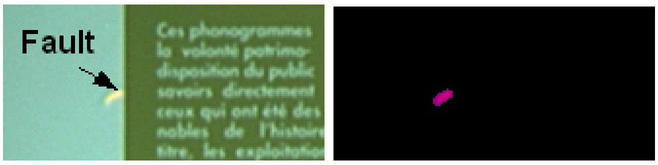


Fig. 1. Left Image: original image with a faulty dot; right image: the deviation image from (the fault-free) master

Now, someone may assume that if this 'deviation image' contains some pixels at all, it can be immediately classified as a bad image, i.e. an image showing an error from the production process. However, this is generally not true as some deviations simply are not visible in the original image or do not reflect an error. For visualizing this, the left image in Figure 2 shows the original image, where no error is on the image, while there is a clear deviation (light pixels) in the deviation image (right). This deviation stems from a small shift of the underlying item. Obviously, this deviation image is clearly different from that in Figure 1 regarding shape and compactness of the object it contains. For a successful discrimination between good and bad images, it is promising to identify the objects in the deviation images first, then to extract features from these objects and to classify the images based on these features. The classifier usually is constructed either from expert knowledge or from image training data sets. This leads us to the work-flow as described in Figure 3.

It should be quite obvious, that the accuracy of the classifier depends not only on the training samples and training method but also on the accuracy of the object recognition component,

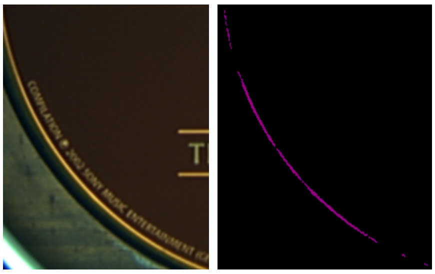


Fig. 2. Left Image: original (fault-free) image; right image: the deviation image from the (fault-free) master

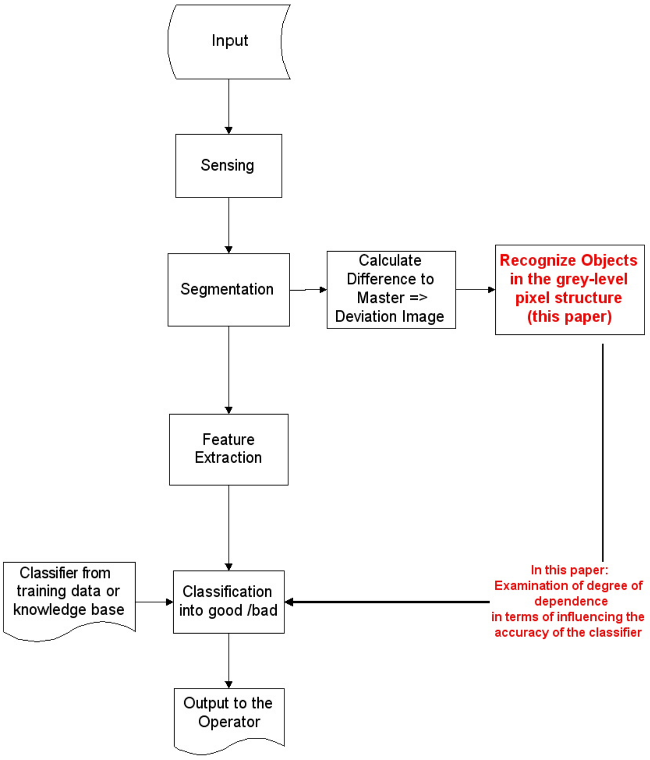
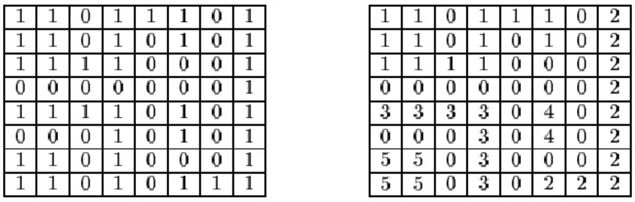

Input
Sensing
Segmentation
Calculate Difference to Master => Deviation Image
Recognize Objects in the grey-level pixel structure (this paper)
Feature Extraction
Classifier from training data or knowledge base
Classification into good /bad
In this paper: Examination of degree of dependence in terms of influencing the accuracy of the classifier
Output to the Operator

Fig. 3. Classification work-flow of images, taken from [4], extending the segmentation part with the idea of producing deviation images and recognizing objects therein

i.e. how correct the objects are extracted. This will be examined in Section III. The comparison includes the error between the number of extracted objects and the real (known) number of objects in a set of 20000 images as well as the impact of the various approaches on the accuracy of a (decision-tree based) classifier, the CART approach [16]. The tree is trained based on features matrices extracted from all of the objects in a set of images. The features were selected appropriately for the given application.

II. Description of the Object Recognition Approaches

In this section various approaches for object recognition in binary images are highlighted together with their drawbacks and advantages. The binary images are obtained by applying a unique threshold over the whole deviation image. As any pixel in the deviation image stems from a deviation to the master and hence is a potential candidate for a part of an error, the threshold is set to 0. This means that all non-black pixels are set to white and all others remain black. An object in a binary image can be now seen as a more or less compact data cloud of white pixels over a certain area.



1	1	0	1	1	1	0	1
1	1	0	1	0	1	0	1
1	1	1	1	0	0	0	1
0	0	0	0	0	0	0	1
1	1	1	1	0	1	0	1
0	0	0	1	0	1	0	1
1	1	0	1	0	0	0	1
1	1	0	1	0	1	1	1

a) binary image

1	1	0	1	1	1	0	2
1	1	0	1	0	1	0	2
1	1	1	1	0	0	0	2
0	0	0	0	0	0	0	2
3	3	3	3	0	4	0	2
0	0	0	3	0	4	0	2
5	5	0	3	0	0	0	2
5	5	0	3	0	2	2	2

b) connected components labeling

Fig. 4. Left Image: binary image (the white pixels as ones); right image: object labelling due to connected components

The approaches for object recognition described in this technical report include:

- Connected components: connecting neighbored pixels to one object
- Morphology with connected components: dilation or closure on the pixels first and then connected components
- Iterative recognition with prototype-based clustering: transforming the pixel space into a 2-dimensional data space and applying clustering methods, delivering cluster centers. An information which data samples belong to which cluster is required. In this paper a modified version of vector quantization as proposed in [17] will be used for generating the clusters.
- Hierarchical clustering: transforming the pixel space into a 2-dimensional data space and applying a clustering approach which connects those pixels to one object which lie near each other
- Extensions to hierarchical clustering: a.) overcoming noise in the images, b.) connecting pixels to objects by alternative criteria

A. Connected Components

Definition 1: A connected component of value p in an image is a set of pixels D, each having value p, such that every pair of pixels in this set are connected with respect to p [20].

In a binary image, this definition can be exploited in order to find objects through a sequence of foreground (white) pixels fulfilling Definition 1. In this sense, the assumption is that adjacent white pixels always belong to the same objects. In Figure 4 a simple example of four connected components is demonstrated, the original binary image is shown in the left image, whereas the right image shows the labelled image according to the connected components in the left image. Each label represents the object number to which the pixel belongs, a value of 0 means background. A detailed algorithm for finding the connected components can be found in [20],

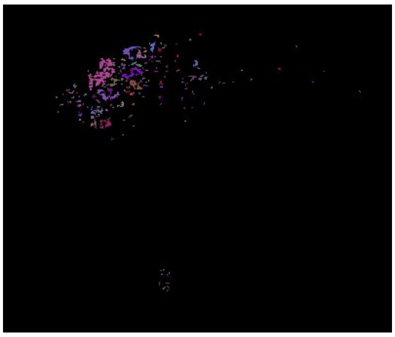


Fig. 5. Falsely detected objects (170 in sum, where there are only 2 distinct objects: bigger upper cloud and smaller lower one); different grey levels (colors) represent different objects

where two possible algorithms are described: one slim but more complex (recursive) one and an iterative algorithm where the whole image needs to be passed through the algorithm two times.

The drawback of connected components when searching for objects in binary deviation images lies immediately at hand when inspecting the deviation image in Figure 5. The upper (bigger) crowd of non-black pixels denotes one critical area and hence one object in the deviation image. However, some pixels are significantly spread over this local domain and not connected to the inner part of the object. Hence, the connected components algorithm recognizes in sum 170 different objects, presented with different grey levels (colors).

B. Morphology with Connected Components

One possibility to overcome the drawback mentioned in the previous section is to apply a closure on the objects in the image first and then to apply connected components on the closed objects. The closure of images is defined in the following way:

Definition 2: The closing of binary image B by structuring element S is denoted by $B \bullet S$ and defined by

$$B \bullet S = (B \oplus S) \ominus S \tag{1}$$

where $\oplus$ denotes the dilation and $\ominus$ the erosion of image B with respect to the structuring element S, see also [20], [19]. The closure possesses the favorable characteristics that holes between white pixels can be closed when choosing an appropriate structuring element with respect to the type of the pixel clouds present in the actual image. In Figure 6 an example for a closure obtained by a circular structuring element with a radius of 7 pixels ($\frac{1}{18}$th of the image size) is shown; the left image represents the original binary (deviation) image, the right image the corresponding closed image, where all the pixels near each other are joined into connected areas. Note that this is exactly what we want to have in order to be able to extract the correct objects with connected components afterwards.

This procedure has two drawbacks: the size and type of the structuring element has to be known in advance. For instance consider the right image in Figure 2, where only with an arc-type structuring element the pixels can be reasonably closed together. If applying the same structuring (disc-type) element as for the image in Figure 6, no pixels can be joined at all to form a continuous arc.

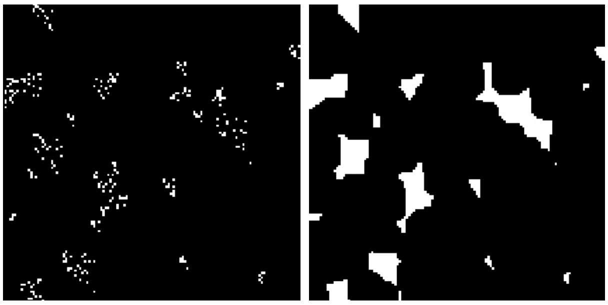


Fig. 6. Left: a binarized deviation image with objects represented by disjoint pixels; right: the closed image → the pixels are joint into connected areas

C. Recognition with Iterative Prototype-Based Clustering

An alternative to the approach described in the previous section is to perform a prototype-based clustering on the pixel data. In this sense, a cluster can be associated with one object. For doing so, first the image data needs to be transformed into a conventional numeric data used by conventional prototype-based clustering methods. This can be easily achieved by collecting the position of all white pixels in a data matrix, i.e. for instance a white pixel at position $(x_{im}, y_{im}) = (110, 58)$ in the image (where (1,1) is the upper left corner) would lead to an entry $(x_{data}, y_{data}) = (110, max_y - 58)$ in the data matrix. The y-coordinate should be transformed in this way, as the image y-coordinate increases from upper to lower, where for the numeric y-coordinate exactly the opposite is the case. Afterwards, a clustering method which delivers cluster centers as cluster prototypes is applied on the data matrix. Examples of such clustering methods are k-means [12], fuzzy c-means [13], Gustafsson-Kessel algorithm [8], vector quantization [6], SOMs [15] or subtractive clustering [2]. All cluster prototypes represent local density points in the data space and for all data points it is elicited to which cluster they belong (i.e. to which cluster they are nearest with respect to a certain distance measure). In this sense, the pixels are connected together in clusters (=objects). In fact, this approach has a quite similar effect as the approach by closure and connected components; however, the obtained cluster structure can be evaluated by a so-called cluster validation index [9]. The value of this index denotes the quality of the obtained clustering structure. In this way, iteratively obtained cluster partitions by varying the most essential parameter(s) can be compared against each other and the cluster structure with the best quality measure is then used for further processing. This triggers a fully self-automatic parameter tuning approach, which cannot be carried out in the case of the closure-connected components approach described in the previous section.

Figure 7 demonstrates a data set obtained from a binarized image and the clusters obtained with different parameter settings (from left to right) of the most essential (vigilance) parameter ρ when using a modified (incremental) version of

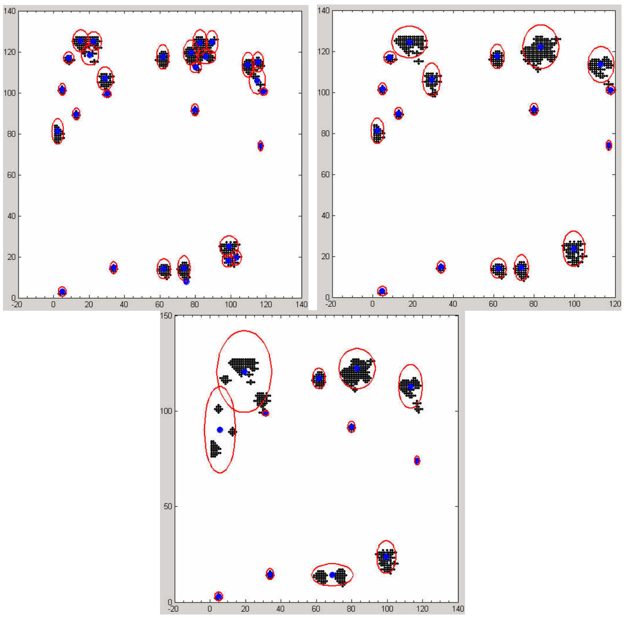


Fig. 7. Left to right: clustering obtained with varying vigilance parameter rho = 0.05, 0.1 and 0.2 → leads to 29, 17 and 12 found objects (whereas 18 objects are actually present in the binary image, the dark dots representing the deviation pixels to the master image)

vector quantization [17]. The corresponding quality measures are elicited by the Beringer-Hüllermeier index [1], which is an extension of the well-known Xie-Beni index [22]. This Beringer-Hüllermeier index correctly yields the best (i.e. the minimal) value for the clustering in the middle image, which delivers the most accurate object recognition (17 clusters = objects are found, while 18 objects are actually present in the image). It has to be mentioned that Beringer-Hüllermeier index could find the best parameter setting(s) on various images more often than Xie-Beni index.

The drawback of both procedures (i.e. closure-connected components and iterative prototype-based clustering) is that usually the object shapes are not known in advance. For instance consider the right image in Figure 2, where only with an arc-type structuring element the pixels can be reasonably closed together. If applying the same structuring (disc-type) element as for the image in Figure 6, no pixels can be joined at all to form a continuous arc. When applying prototype-based clustering the cluster center(s) would be totally miss-placed as the clusters do not have the form of circular or ellipsoidal data clouds.

D. Hierarchical Clustering

In order to omit this shortage, a version of agglomerative hierarchical clustering algorithm [11] is exploited, the so-called single linkage method. In this method, each cluster is represented by all the data points in the cluster. The similarity between two clusters is measured by the similarity of the closest pair of data points belonging to different clusters. If this similarity is high, the two clusters are joined together to one cluster. In this sense, the method can find clusters (=objects) of arbitrary shape and different size. It should be added that in our case of data points lying in the metric space the similarity between two points can be simply represented by the distance between the closest point with respect to a certain distance measure.

In this way, the mathematical formulation for hierarchical clustering becomes:

Algorithm 1: **Hierarchical Clustering**

```
1) Given: data set $\{x_1, ..., x_n\} \subseteq R^p$ and distance measure $d$
2) Initialization: $K_0 = n$, $\Gamma_0 = \{C_{0,1}, ..., C_{0,K_0}\} = \{\{x_1\}, ..., \{x_n\}\}$, $i = 0$, $thr = L$
3) While 1==1
    a) Determine $(a, b)$ such that $min\_dist = d(C_{i,a}, C_{i,b})$ is minimal
    b) if $min\_dist > thr$, break
    c) else $\Gamma_{i+1} = (\Gamma_i \setminus \{C_{i,a}, C_{i,b}\}) \cup \{C_{i,a} \cup C_{i,b}\}$
4) Output: sequence of clusterings $\Gamma_0, ..., \Gamma_i$ with $\Gamma_i$ the clustering structure which optimally partitions the data space with respect to distance measure $d$ and threshold $thr$.
```

Different distance measures d can be applied, the most usual one is the minimal distance between pairs of points, where one is taken from $C_{i,a}$, the other from $C_{i,b}$. The threshold thr is the essential parameter which determines, whether two clusters are close to each other or not.

In Figure 8 two images are presented, where the hierarchical clustering successfully extracts the correct number of clusters (=objects). This first image represent the same deviation image as in Figure 5, where connected components completely failed as extracted 170 objects in sum, whereas only 3 objects are present. In fact, the iterative prototype-based clustering approach could also find the correct number of objects for this image, but completely failed for the lower image, as extracted five clusters for the longer arc-type object.

E. Extensions to Hierarchical Clustering

The standard agglomerative hierarchical clustering method together with some extensions with respect to linkage strategies [11] has two major limitations:

1) It cannot deal with noise in the data
2) It does not exploit any knowledge about special characteristics of different clusters in one data set

The first limitation may cause a wrong merging direction of the clusters (e.g. when two clusters are merged, just because noisy points of both clusters lie very near each other). and can be solved by CURE [7]. The idea of CURE is to select just a few random points from each cluster, where the probability that a point is chosen depends on the size of the cluster, i.e. the number of data points belonging to a cluster.

The second limitation causes incorrect merging decisions of the algorithm, whenever two clusters lying near each other possess different shapes or a within-denisity/connectivity characteristics which is different to the between-density/connectivity characteristics. For instance consider the four clusters as shown in Figure 9.

The selection mechanism of Algorithm 1 will prefer merging clusters (a) and (b) over merging clusters (c) and (d), since

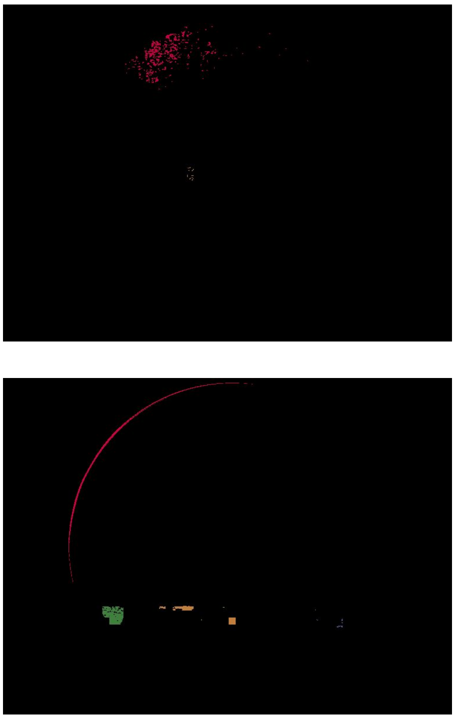


Fig. 8. Upper: hierarchical clustering finds correctly the two clusters (the bigger upper one and the smaller in the middle part of the image) for the image where the connected components failed completely (see Figure 5); lower: an image containing objects of different shapes, sizes and densities. In both images the different colors represent the different clusters found by hierarchical clustering

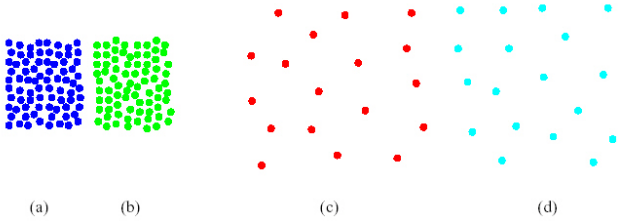


Fig. 9. Conventional hierarchical clustering algorithm prefers to merge (a) with (b) instead of (c) with (d) as correct choice

the minimum distances between the representative points of (a) and (b) will be smaller than those for clusters (c) and (d). But clusters (c) and (d) are better candidates for merging because the minimum distances between the boundary points of (c) and (d) are of the same order as the average of the minimum distances of any points within these clusters to other points. Another (practical) example of a wrong merging choice is given in Figure 10. There two clusters (representing the arc-type object and the strip-type object with the letters inside) are merged as lying near each other; however they possess a completely different shape and internal structure, so they can be assumed as two different objects.

An approach which gives a significant improvement of standard hierarchical clustering is the so-called CHAMELEON described in [14]. CHAMELEON operates on a sparse graph in which nodes represent data points and weighted edges represent similarities among the data points. The key feature of CHAMELEON's agglomerative hierarchical clustering algorithm is that it determines the pair of most similar clusters by taking into account both the (relative) inter-connectivity as well as the (relative) closeness of the clusters. In Figure 11 CURE as an extension of standard hierarchical clustering is compared with Chameleon, the data sets represent cluster with different densities and shapes and also include a significant noise level.

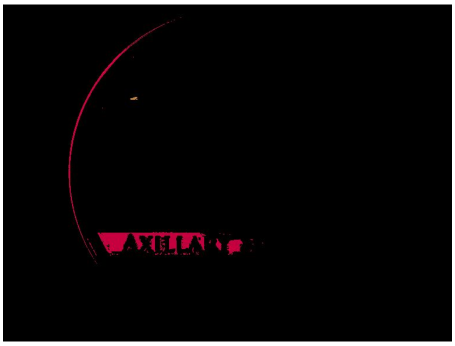


Fig. 10. An image where the conventional version of hierarchical clustering algorithm (Algorithm 1) fails (the arc-type and the strip-type object are joined together to one cluster)

III. Application Example and Results

In this section an application example is given (taken from the EU-Project DynaVis), which includes a complete self-automatic reconfigurable and adaptive fault detection framework for images, i.e. an online classification framework which classifies each image as good or bad and adapts/evolves the data-driven classifier upon operator's feedback → hybrid modelling. The whole framework is shown in Figure 12. Basically, the framework operates on two feature sets extracted from the binary deviation images obtained during the low level processing part: an object feature set and an aggregated feature set (in the framework denoted as 'adaptive feature aggregation'). The object feature set contains the features extracted from the single objects in the (training) set of images and the aggregated feature set contains the features extracted from the complete images. This means, an aggregated feature vector contains the information about the characteristics of an image when inspecting it as a whole. It is calculated by aggregating the feature information of the single objects. In this sense, it is possible to classify an image correctly as bad also in the case, when the single objects on the images are not distinct enough, that they can be classified as bad for their own. Furthermore, it may happen that only a rating of the operator(s) on image basis is available; in this case, training on the object feature vectors usually leads to undesirable miss-classification rates.

In Section III-B the impact of the different object recognition approaches on the accuracy of the classifier trained from the features extracted from the aggregated object information (17 features in sum listed in [5]) is demonstrated (as no labelling on object level was available). Examples of aggregated

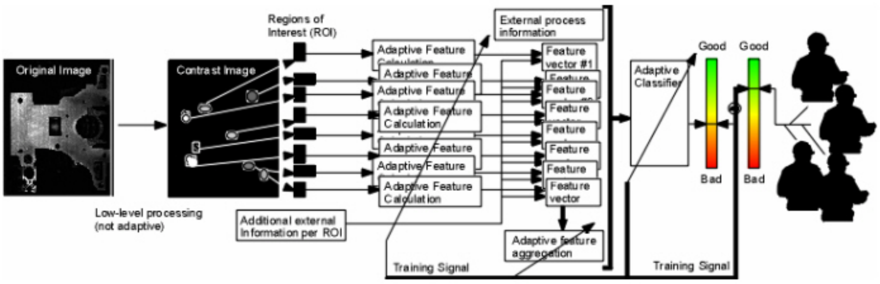

Regions of Interest (ROI)
Original Image
Contrast Image
Low-level processing (not adaptive)
Additional external information per ROI
Adaptive Feature Calculation
External process information
Feature vector #1
Feature vector
Adaptive Classifier
Good
Good
Bad
Bad
Adaptive feature aggregation
Training Signal
Training Signal

Fig. 12. Classification framework for classifying images into good and bad

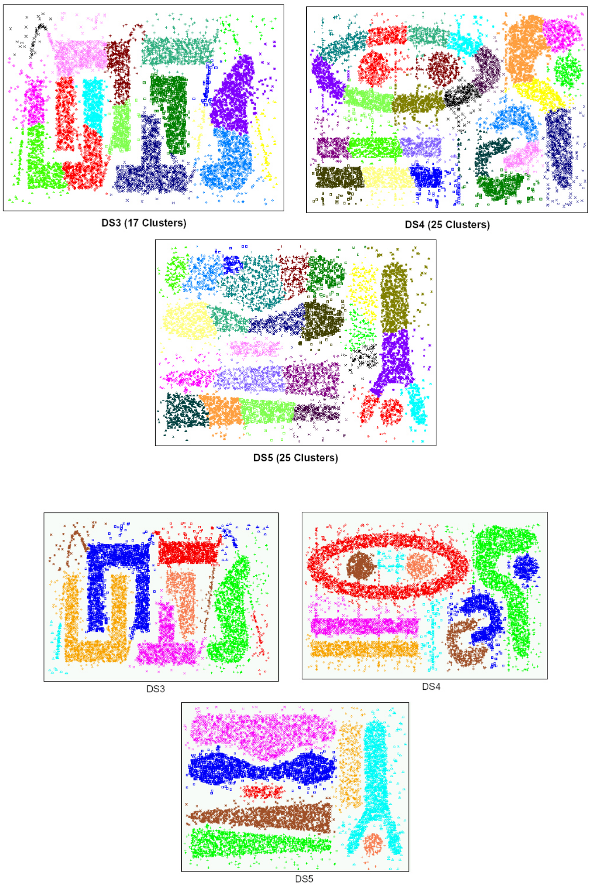


Fig. 11. Top row: clusters obtained with CURE, bottom row: clusters obtained with Chameleon for three different noisy data sets

features are 'the number of objects', 'the average intensity of objects' or the 'maximal density of objects with a certain radius'. For obtaining a valid comparison, the same method for classifier training was used for building up a decision tree classifier (CART [16]). In the subsequent section the different object recognition approaches are evaluated based on an image data set, where the number of objects in each image are a priori known. The extended hierarchical clustering approach could be not evaluated so far.

TABLE I
MAE BETWEEN REAL AND RECOGNIZED NUMBER OF OBJECTS OVER ALL 20000 IMAGES

Method	**MAE**
Connected Comp.	1.55
Closure and Connected Comp.	1.45
Iterative Prototype-based Clustering	1.09
Hierarchical Clustering	1.2

A. *Results on Artificial Data Set*

An artificial image data set of 20000 images was produced, which represents deviation images as they may appear naturally for real-recorded images. In Table I a comparison on the MAE between real (known) number of objects in the images and the number of objects recognized by different object recognition approaches is made. From this table it can be seen, that hierarchical clustering as best method produces an average deviation of approximately one object per image, where in average 7.74 objects are present in the images. This means that the percentual deviation is around 14%. For this data set hierarchical clustering performed not better than prototype-based clustering and not much better than closure with connected components as may be assumed. The reason for this result is that quite a lot of objects appear as more or less ellipsoidal and compact data clouds for which iterative prototype-based clustering and connected components with a disc-shaped structuring element produce quite reasonable results.

Now it is interesting to see, how this discrepancy in the recognized number of objects affects the accuracy of the decision tree classifier. For this task, a 10-fold cross-validation procedure [21] was performed on a CART decision tree classifier with optimal pruning strategy [16] based on all 20000 training images: therefore, this image set is split into 10 equal parts, where 9 out of 10 are joined together and fed into the decision tree building algorithm and the remaining one is used for deriving the miss-classification rate. This is done 10 times and the miss-classification rates on the small data sets are averaged for the overall miss-classification rate. This measure is a precise approximation of the generalized prediction error, see [10]. In Table II the miss-classification rates, i.e. the relative number of miss-classified samples, when applying the different object recognition approaches are compared.

TABLE II

EXPECTED MISS-CLASSIFICATION RATES WHEN TRAINING A DECISION TREE CLASSIFIER FROM DIFFERENT FEATURE MATRICES OBTAINED BY APPLYING DIFFERENT OBJECT RECOGNITION APPROACHES ON ARTIFICIAL DATA SET

Method	**Miss-Class Rate**
Connected Comp.	14.23%
Closure and Connected Comp.	13.60%
Iterative Prototype-based Clustering (VQ-INC-MOD)	12.89%
Hierarchical Clustering	13.07%
Re-Labelled Data	9.32%

Obviously, the impact of the object recognition approaches on the classifier's accuracy is present but quite low and almost in the same proportion as the impact on the number of recognized objects. The last row represents the case, when the extracted feature vectors are re-labelled according to a set of good/bad rules, due to which the artificial image data set was originally labelled (e.g. rules such as 'there is an object larger than 10 pixels and the object is in the middle of the image (at least 20 pixels away from the image sides)'). In this sense, the noise caused by the object recognition approaches (they are not able to extract the correct objects corresponding to the original labels — see Table I) is eliminated and this test case represents the pure bias of the classifier, i.e. the error which is only caused by the too less flexibility of the classifier for reproducing the original rules. Hence, we can conclude that an increase of 3% to 5% miss-classification rate is caused only due to an incorrect recognition of objects by the various object recognition approaches.

B. Results on Real-Recorded Images

Now, we examine the impact of the object recognition approaches on the classification accuracy of the classifier trained from the aggregated (object) features based on a labelled CD print data set containing 998 images. This data set includes several more complicated objects as data clouds with spread pixels, noise, long arc-type and letter objects (see also some images in Section II). The data set is divided into 81% good and 19% bad samples, so unequally distributed number of samples between the two classes. In Table III the miss-classification rate of the decision tree classifier with optimal pruning strategy (see Statistics-toolbox in MATLAB) is stated. This classifier was chosen as it performed best on this data set among ten other different methods (including SVMs, LDA, fuzzy classification etc.). This miss-classification rate is obtained again by applying 10-fold cross-validation [21] on the training data set (998 images) (see previous section).

From this table it can be realized that the chosen object recognition approach does have an impact on the classifier's accuracy. In fact, this impact is not extensive, however a decrease of 4% miss-classification rate (from approx. 15% to 11%) is always welcome and may also be a key point that a classifier is chosen for further processing in an industrial system (as for instance 11% may be accepted where 15% may be not accepted). It is quite obvious, that hierarchical clustering is the best performing method, as most of the images contain objects with arbitrary shapes, quite often appearing as long arcs. It should be also mentioned that in the case of a classifier training on object features (in the case when object labels are available), the discrepancy between approaches may get even higher, as then the objects should be extracted even more precise in order to fit to the corresponding labelling.

TABLE III

EXPECTED MISS-CLASSIFICATION RATES WHEN TRAINING A DECISION TREE CLASSIFIER FROM DIFFERENT FEATURE MATRICES OBTAINED BY APPLYING DIFFERENT OBJECT RECOGNITION APPROACHES ON CD-PRINT DATA SET

Method	**Miss-Classification Rate**
Connected Comp.	15.53%
Closure and Connected Comp.	13.82%
Iterative Prototype-based Clustering	14.21%
Hierarchical Clustering	11.58%

IV. CONCLUSION AND OUTLOOK

Four different methods for object recognition in binarized deviation images were described and their differences together with their advantages and drawbacks discussed. The discussion is systematically carried out on a step per step basis due to some example images. From this discussion someone may conclude which method is favorable for his/her deviation images, which are obtained by calculating the difference of newly recorded images to a pre-defined master. All the methods except the extended version of hierarchical clustering (Chameleon) are evaluated based on an artificial data set, where the true number of objects is known, and based on a real-recorded image set representing CD prints. While for the first data set morphology with connected components and iterative prototype-based clustering can compete with hierarchical clustering, for the second data set hierarchical clustering could clearly outperform the other methods. This is because of the circumstance that the images in these data set contain quite a lot objects with no pre-defined by more arbitrary shape, which is not the case for the first data set, where a lot of compact ellipsoidal data clouds are present in the images. In this sense, the selection of an appropriate object recognition approach may be an important issue, whenever new image data should be processed through a fault detection framework. The selection could be either carried out based on expert knowledge (by inspecting quite a lot of images and then decide which object recognition approach to use) or by trial-and-error search for the best performing one with the help of cross-validation. Another possibility would be to develop a quality measure which measures the quality of the partition of the image based on the white (foreground) pixels and can be applied for several object recognition approaches for comparison. For prototype-based clustering algorithms such measures exist (known as cluster validation indices), maybe some of these can be standardized for the other approaches discussed in this technical paper. Typically, if nothing is known about the shapes, size and densities of the clusters, hierarchical clustering together with its extensions should be always a hot choice. Future tests will include the exploitation of a more precise label information on single object basis. This will probably reduce the miss-classification rates of all approaches further, as then a training on object feature vectors is reasonably possible.

Acknowledgements

This work was supported by the European Commission (project Contract No. STRP016429, acronym DynaVis). This publication reflects only the author's view.

References

[1] J. Beringer and E. Hüllermeier. Online clustering of parallel data streams. *Data & Knowledge Engineering*, 58(2):180–204, 2006.

[2] S. Chiu. Fuzzy model identification based on cluster estimation. *Journal of Intelligent and Fuzzy Systems*, 2(3):267–278, 1994.

[3] Christian Demant, Bernd Streicher-Abel, and Peter Waszkewitz. *Industrial Image Processing: Visual Quality Control in Manufacturing*. Springer, 1999.

[4] R.O. Duda, P.E. Hart, and D.G. Stork. *Pattern Classification - Second Edition*. Wiley-Interscience, Southern Gate, Chichester, West Sussex PO 19 8SQ, England, 2000.

[5] C. Eitzinger, M. Gmainer, R. Richter, and E. Lughofer. List of reasonable features on deviation images. Documentation within DynaVis EU-Project, 2006.

[6] R.M. Gray. Vector quantization. *IEEE ASSP Magazine*, pages 4–29, 1984.

[7] S. Guha, R. Rastogi, and K. Shim. CURE: An efficient clustering algorithm for large databases. In *Proc. of 1998 ACM-SIGMOD Int. Conf. on Management of Data*, 1998.

[8] D. Gustafson and W. Kessel. Fuzzy clustering with a fuzzy covariance matrix. In *Proc. IEEE CDC*, pages 761–766, San Diego, CA, USA, 1979.

[9] M. Halkidi, Y. Batistakis, and M. Vazirgiannis. On clustering validation techniques. *Journal of Intelligent Information Systems*, 17(2/3):107–145, 2001.

[10] T. Hastie, R. Tibshirani, and J. Friedman. *The Elements of Statistical Learning: Data Mining, Inference and Prediction*. Springer Verlag, New York, Berlin, Heidelberg, Germany, 2001.

[11] A.K. Jain and R.C. Dubes. *Algorithms for Clustering Data*. Prentice Hall, 1988.

[12] T. Kanungo, D.M. Mount, N.S. Netanyahu, C.D. Piatko, R. Silverman, and A.Y.Wu. An efficient k-means clustering algorithm: Analysis and implementation. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 24(7):881–892, 2002.

[13] N. B. Karayiannis and J. C. Bezdek. An integrated approach to fuzzy learning vector quantization and fuzzy c-means clustering. *IEEE Trans. on Fuzzy Systems*, 5(4):622–628, 1997.

[14] G. Karypis, E. H. Han, and V. Kumar. Chameleon: Hierarchical clustering using dynamic modeling. *IEEE Computer*, 32(8):68–75, 1999.

[15] T. Kohonen. *Self-Organizing Maps*. Springer Verlag, Berlin Heidelberg, Germany, second extended edition, 1995.

[16] C.J. Stone R.A. Olshen L. Breiman, J. Friedman. *Classification and Regression Trees*. Chapman and Hall, Boca Raton, 1993.

[17] E. Lughofer and U. Bodenhofer. Incremental learning of fuzzy basis function networks with a modified version of vector quantization. In *Proceedings of IPMU 2006, volume 1*, pages 56–63, Paris, France, 2006.

[18] Madhav Moganti, Fikret Ercal, Cihan H. Dagli, and Shou Tsunekawa. Automatic PCB Inspection Algorithms: A Survey. *Computer Vision and Image Understanding*, 63(2):287–313, 1996.

[19] W.K. Pratt. *Digital Image Processing*. John Wiley and Sons Inc., 1991.

[20] G. Stockman and L. Shapiro. *Computer Vision*. Prentice Hall, 2001.

[21] M. Stone. Cross-validatory choice and assessment of statistical predictions. *Journal of the Royal Statistical Society*, 36:111–147, 1974.

[22] X. L. Xie and G. Beni. A validity measure for fuzzy clustering. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 13(48):841–847, 1991.