

**L I F C**

LABORATOIRE D'INFORMATIQUE DE L'UNIVERSITE DE FRANCHE-COMTE

EA 4269

---

*An exhaustive comparison framework for distributed  
shape differentiation in a MEMS sensor actuator array*

Kahina Boutoustous — Eugen Dedu — Julien Bourgeois

---

**Rapport de Recherche no RR 2011-????**

THÈME 4 – Juillet 2008





## **An exhaustive comparison framework for distributed shape differentiation in a MEMS sensor actuator array**

Kahina Boutoustous , Eugen Dedu , Julien Bourgeois

Thème 4  
OMNI

Juillet 2008

**Abstract:** The Smart Surface<sup>1</sup> project aims at designing an integrated micro-manipulator based on an array of micromodules connected in a 2D array network. Each micromodule has a sensor, an actuator and a processing unit. One of the aims of the processing unit is to recognize the shape of the part that is put on top of the smart surface. This recognition or more precisely this differentiation is done through a distributed algorithm that we call a criterion. The aim of this article is to present the ECO framework, which is able to test exhaustively the efficiency of different differentiation criteria, in terms of differentiation efficiency, memory and processing power needed. The tests show that ECO is of great help for choosing the best criteria to implement inside our smart surface.

**Key-words:** shape differentiation, distributed computing, MEMS



# **Une plate-forme de comparaison exhaustive pour différencier des formes en distribué sur une matrice de capteurs actionneurs MEMS**

**Résumé :**

**Mots-clés :**



## 1 Introduction

During an assembly process, it is necessary to feed assembly line workstations with well-oriented and well-positioned parts. These parts are often jumbled and they need to be sorted and conveyed to the right workstation. To do so, the operations to be performed on parts are the following: identifying, sorting, orienting, positioning, feeding, and assembling. Among the most promising solutions to perform these tasks on microparts, is the combination of micro-electro mechanical systems (MEMS) in order to form an actuator arrays. However, if a single microactuator is not powerful enough to move a micropart, several microactuators working cooperatively might very well do it. A MEMS sensor/actuator arrays with embedded intelligence is referred as a smart surface.

The objective of the Smart Surface project is to design such an integrated MEMS system which will be able to identify, to sort, to orient and position microparts. This article deals only with the identification part of the process: A micropart is put on the Smart Surface which have to recognize the part shape and give the proper orders to the control system to move it on the right place. In fact, recognition is not the proper term. Given a set of part, the Smart Surface have to *differentiate* all the parts within the set. As the processing power of the Smart Surface is embedded in very limited space, this differentiation process has to be optimized both in term of memory used and processing power needed. The differentiation is made by a distributed program loaded in the Smart Surface. For the rest of the paper we call this program a differentiation *criterion*. The aim of the Exhaustive COmparison (ECO) framework which is presented in this article is to test exhaustively, i.e. for all kinds of possible part shapes, a set of criteria to choose the most adapted ones. The main condition for choosing a criterion is that it must be able to differentiate all the possible parts, that is what we call total differentiation. The other two remaining conditions are: using the less memory and using the less computing power.

The rest of the paper is organized as follows. Section 2 details the Smart Surface project. Section 3 presents the ECO framework, while the tests are performed on section 4. Some related works to shape representation are presented section 5 and they are followed by the conclusion and presentation of future works.

## 2 The Smart Surface Project

There have been numerous projects of MEMS actuator arrays in the past and more precisely in the 1990's. These pioneer researches have developed different types of MEMS actuator arrays, based on actuators either pneumatic [11, 5], servoed roller wheels, magnetic or thermobimorph and electrostatic. Some of these preliminary studies use a sensorless manipulation scheme based on the Goldberg's algorithm [7] for parallel jaw grippers. The jaw grippers are obtained with MEMS actuator arrays by creating opposite field forces which then can orient and move the parts. Bohringer et al. [1] have proposed a concept called "programmable force field" which is an extension of the Goldberg's algorithm. This manipulation scheme which is well-adapted for jaw grippers has shown some limitations when adapted to MEMS actuator arrays. For instance, the absence of a command law can lead to uncertain behaviours or MEMS actuator arrays has to be programmed for each different kind of parts. More recent research has been conducted in order to include sensors and to add intelligence to MEMS actuator arrays but it either fails to develop it at a micro-scale or to be fully integrated [6].

The objective of the Smart Surface project is to design a distributed and integrated micro-manipulator based on an array of micro-modules in order to realize an *automated positioning and conveying surface*. Each micro-module will be composed of a micro-actuator, a micro-sensor and a processing unit. The cooperation of these micro-modules thanks to an integrated network will allow to recognize the parts and to control micro-actuators in order to move and position accurately the parts on the smart surface. The parts are small, they cover a few numbers of micro-modules.

The strength of our project is the multidisciplinary collaboration among six labs specialized in their field and more than twenty researchers. We are responsible for the information management inside the smart surface, i.e. distributed part differentiation and communication infrastructure.

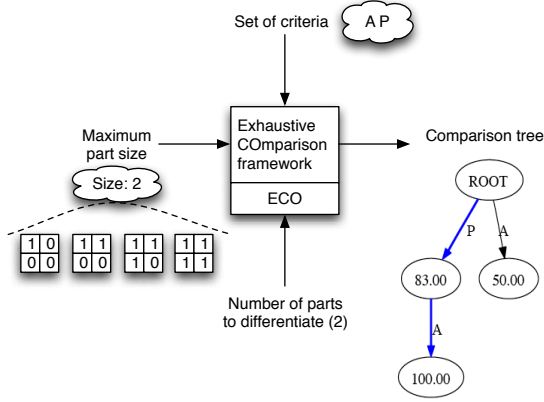


Figure 1: Overview of the framework.

### 3 The ECO framework

Before implementing the part differentiation algorithms on the Smart Surface, we are interested to find out criteria allowing high differentiation rates.

This section presents a framework for criteria comparison in differentiating parts, based on an exhaustive part generation. The framework is presented in figure 1. It receives as input a set of criteria, the maximum part size (a square) and the *number* of parts to differentiate. The framework exhaustively generates all the appropriate parts. It generates several comparison trees: differentiation tree, cost tree. An example of question which the framework answers to is: What criteria differentiate best three random parts not greater than  $3 \times 3$ ?

The constraints of the framework which will be relaxed in future works are: parts can be rotated only at  $90^\circ$ ; no error in sensors and communications; we work on *family of parts*. We define a family of parts all the *ideal* parts which have the same image (discrete representation) on the surface. For example, the typographic letter L and L (with and sans serifs) have the same image on the surface, because the serifs are much smaller than the sensors.

The parts on the Smart Surface are supposed to be represented by square matrices of size 3 or 4. In order to find criteria reaching 100% differentiation, all possible parts of size  $P \times P$  with  $P = 3$  and  $P = 4$  are considered. This set of parts is used to generate groups of parts. These

Table 1: Information for groups of three parts.

Max part size	Number of parts generated	Number of unique parts ( $T$ )	Number of groups
$3 \times 3$	512	35	$C_{35}^3 = 6545$
$4 \times 4$	65536	1280	$C_{1280}^3 = 348706560$

groups are used to test the criteria or combinations of criteria which reach total differentiation. Our method has five steps:

- (1) First, all the parts of size  $P \times P$  are generated.
- (2) Afterwards, the resulting set of parts is reduced by eliminating translations,  $90^\circ$  rotations and mirrors (see tab. 1), as detailed below.
- (3) Afterwards, all the combinations of  $n$  parts from the previously generated parts are generated (see tab. 1).
- (4) Afterwards, all the combinations of  $CC_i$  criteria are generated. For example, if  $C_i = \{A, S, P\}$  is the set of criteria, the generated combinations are  $CC_i = \{\{A\}, \{S\}, \{P\}, \{AS\}, \{AP\}, \{SP\}, \{ASP\}\}$ . This means that all the criteria are combined in order to differentiate the parts. Several criteria have been tested, presented in section 4.
- (5) Finally, the differentiation phase, detailed below.

**Part generation** A part on the  $P \times P$  square may be represented as a binary matrix. In a  $P \times P$  square, there are  $2^{P \times P}$  parts. However, many of them are not *connex*, i.e. in fact there are two parts instead of one. The connexity checking is done with a research in depth. Afterwards, its *mask* is generated. The mask is a matrix generated from the initial matrix where the first columns and first lines with only 0s are removed. This step removes translated identical parts. During the next step, masks are rotated  $90^\circ$ ,  $180^\circ$  and  $270^\circ$ , each mask is mirrored. After each 2 by 2 comparison, identical masks are removed such that only one mask of same type remains. The parts remaining after this process are unique compared to translation, rotation of multiple of  $90^\circ$  and mirroring. Let  $T$  be the total number of unique parts.



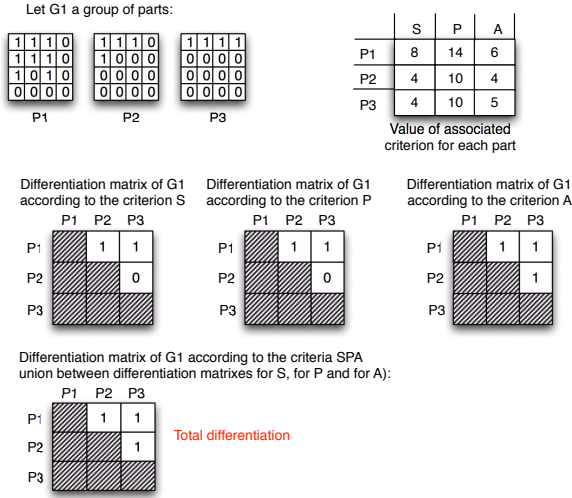


Figure 2: An example of a group differentiation according to a combination of criteria.

**Part differentiation** For each criterion  $C_j$  and each group  $G_i$  of parts a differentiation matrix  $D$  is generated, with  $D(i, j) = 1$ , if the values of the criterion between the two parts  $i$  and  $j$  are different, otherwise it is 0.

$$D_{G_i, C_j}(k, l) = 1, \forall k, l \in P \quad (1)$$

$$\Leftrightarrow C_j \text{ differentiates all } P \in G_i$$

In the case of a combination of several criteria  $CC_j$  the union of the differentiation matrices is computed. If the matrix  $D$  contains only 1 values, the parts are said to be differentiated according to this combination of criteria.

$$D_{G_i, CC_j} = \cup_{k \in CC_j} D_{G_i, C_k} \quad (2)$$

$$CC_j \subset \{C_1, C_2, \dots, C_m\}$$

The matrix  $D$  is upper triangular. A differentiation is said to be total if the matrices are differentiated according to *all* possible groups. Fig. 2 is an example of the computation of a differentiation matrix with  $G_1 = \{P_1, P_2, P_3\}$  for the combination of criteria  $CC_1 = \{ASP\}$ .

The major challenge is to find out a combination of criteria which leads to a total differentiation, i.e. for any group of parts, a differentiation of the parts using one combination of our criteria is always achieved.

The following algorithm details this process: Let  $n$  be the number of parts which are needed to be differentiated. The framework generates all the groups of  $n$  parts. If  $T$  is the total number of unique parts, there will be  $C_T^n$  groups.

- 1: **for** each  $CC_i =$  subset of  $\{C_1, C_2, \dots, C_m\}$  **do**
- 2:     **for** each group  $G_i$  subset of  $n$  elements in  $P$  **do**
- 3:         **if**  $CC_i$  is a criterion **then**
- 4:             build the differentiation matrix  $D_{G_i, CC_i}$
- 5:         **else**
- 6:              $\{CC_i$  is a combination of criteria $\}$
- 7:             **for** each  $C_j$  in  $CC_i$  **do**
- 8:                 build the differentiation matrix  $D_{G_i, C_j} =$
- 9:                      $\cup D_{G_i, C_j}$
- 10:             **end for**
- 11:             compute the differentiation rate  $t$
- 12:         **end for**
- 13:         compute the average  $ta$  of all differentiation rates  $t$
- 14:     **end for**
- 15: build comparison tree

The usefulness of criteria is presented as a simplified tree (path  $XY$  is the same as  $YX$ ) called *comparison tree* (see fig. 4 in section 4). Each node has a value expressed as percentage of differentiation using all the criteria of the path from the root of the tree. Finally, a cost (execution time, memory used etc.) is associated to each branch.

## 4 Tests

The aim of our work is to differentiate relatively small parts by finding a set of criteria. These parts are represented by square matrices of order 3 or 4. All criteria are tested in order to find combinations of criteria reaching total differentiation. Among these criteria, the fastest execution time and/or the lowest memory cost are selected.

**Description of the criteria** The differentiation criteria must be simple and easy to implement. For example, the first criterion,  $P$  (the perimeter), is the number of cell frontiers between “1” (pressed sensor) and “0” (unpressed sensor). The second criterion,  $S$  (the area), consists in counting all the “1” contained in a part. Here are the criteria used:

Contour-based criteria:  $P$ , the number of 1 having at least one neighbor at 0;  $A$ , the number of 1 having at least three neighbors to 0 and forming a right angle.

Region-based criteria:  $S$ , the number of 1 of the part;  $L$ , the maximum length between 1 of the part;  $N$ , the sum of the number of bits that change between two successive lines respectively columns;  $Z$ , the maximum length between all the 0 of part;  $D$ , the sum of 1 located on both diagonals;  $F$ , the sum of all Manhattan distances between 0;  $M$ , the sum of the number of bits that change;  $R$ , the sum of the number of V shape angles;  $I$ , the sum of the number of identical lines with the number of identical columns;  $T$ , the product of all Manhattan distances between 0;  $Y$ , the product of all Manhattan distances between 1;  $E$ , the product of the number of bits that change between each two successive lines with the number of bits that change between each two successive columns;  $K$ , the product of the number of bits that change from: the first line with the other lines, the last line with the other lines, the first column with the other columns, the last column with the other columns;  $C$ , the sum of the number of V shape angles.

**Criteria reaching total differentiation** Among all the combinations of criteria, only combinations reaching total differentiation are considered. The test show that the minimal combinations of criteria for matrices of size  $3 \times 3$  are:

$CC_i = \{\{TM\}, \{TK\}, \{YF\}, \{YM\}, \{YK\}, \{YE\}\}$  and for matrices of size  $4 \times 4$  are:

$CC_i = \{\{CFIDMRZ\}, \{CFILMRZ\}\}$ .

Fig. 3 presents the number of combinations of criteria reaching total differentiation function of the size of the combination. For  $3 \times 3$  matrices all combinations of size 2 are removed from the combinations of size 3. For example (T,M) and (Y,E) reach 100%, therefore combinations ATM and AYE have been removed because they provide no additional differentiation. It's the same for all combinations.

**Memory costs and execution times of the criteria reaching 100%** To sum up, for  $3 \times 3$  matrices six combinations of two criteria, among the criteria that we are defined, reach a total differentiation. However, it is obvious that the binary representation criterion, together with

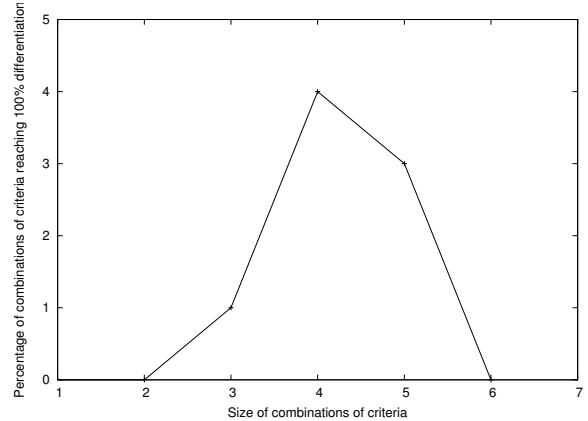


Figure 3: Size of combination reaching 100%.

the grid based method [14], is sufficient to differentiate the parts. Although it is very costly in memory because the whole matrix is saved, i.e. 9 bits,  $90^\circ$  rotation matrices and mirrors matrices must also be saved. This gives 72 bits. Fig. 4 shows an example of memory consumption for all combination of the criteria  $T, Y, F, M, K, X$ .

Fig. 4 shows that the combination  $TM$  reaches total differentiation with 37 bits, less than the binary representation. Execution times necessary for each criterion are measured. Fig. 5 presents the scatter of points of memory cost function of execution time of criteria. There are several combinations of criteria that reach total differentiation with lower execution time and memory cost than binary representation.

## 5 Related work

**Contour-based approaches** Fourier descriptors: The image is defined by a 1D function called *shape signature*, which represents a compact representation of the image [9]. Afterwards, a Fourier transform is applied [4, 3]. It results in coefficients called Fourier descriptors. These descriptors represent the shape of the object in the frequency domain.

Freeman coding [8]: it consists in browsing the borders of shape with elementary moves and coding the movement.

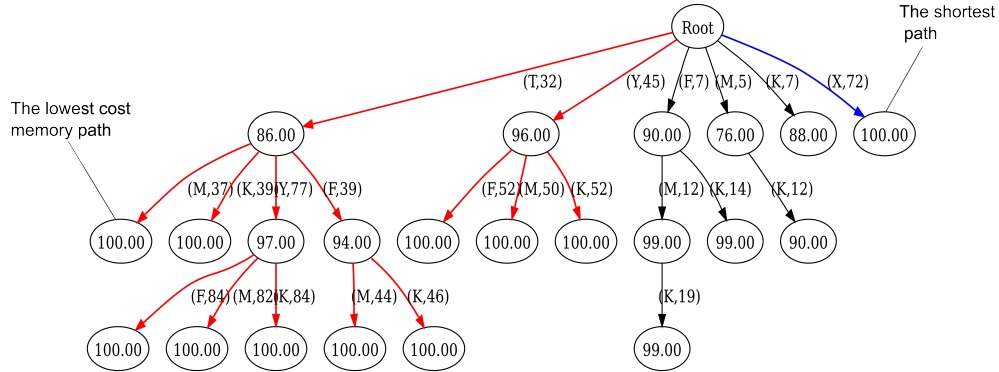


Figure 4: Memory cost.

These methods are widely used for big pictures where the outline of the image differs noticeably from the inside of the images (parts). In our study these methods are not very interesting given that we are working on tiny images where the contour is equal or nearly equal to the surface.

**Region-based approaches** Grid based: a fixed-length grid of cells on the image is drawn [13]. Going along the grid from top to bottom and from left to right, each cell wholly or partly covered by the form is affected with the value 1, and others cells with 0 [14]. This produces a binary number, which is the representation of our shape. The difference between two parts is given by an XOR between their binary representations. This binary representation is very sensitive to rotation, translation and dilatation, that is it requires a prestandardization.

**Invariant moments:** A set of seven descriptors are used [12, 10, 2], called Husont invariants, computed by normalizing central moments of order three. The invariant moments are widely used in three dimension models or large images that need to be compacted. It is not very useful to apply this method in our case because the images are very small.

## 6 Conclusions and future works

In this article we presented an exhaustive framework allowing to identify the criteria reaching a total differentia-

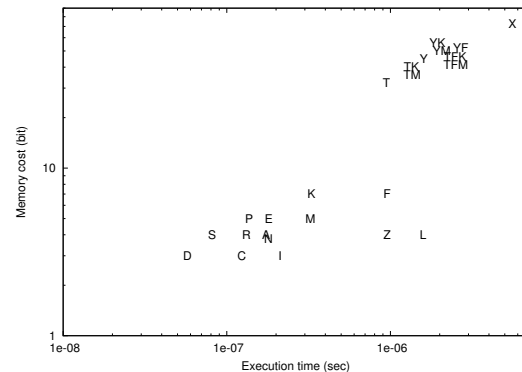


Figure 5: Memory cost according to execution times of criteria.

tion among a set of criteria. Our tests on groups of 3 parts show that some combinations of two criteria for matrices of size  $3 \times 3$  reached a total differentiation. We have considered the memory cost and execution time of the criteria and combinations of criteria that achieve a total differentiation. We have made a comparative study of these results with the execution time and cost memory of the grid based method. We have deduced that some combinations of criteria reach a total differentiation with a smaller execution time and a lower cost memory than the grid based method.

One idea for future work is to allow a more flexible rotation (e.g. a  $5^\circ$  step-by-step rotation). Another idea is to develop distributed algorithms for criteria in order to

implement them in the Smart Surface and compare their execution time. Finally, we plan to implement the framework on a G80 GPU with CUDA, in order to speed up the comparison.

## References

- [1] K.-F. Böhringer, V. Bhatt, B. R. Donald, and K. Y. Goldberg. Algorithms for sensorless manipulation using a vibrating surface. *Algorithmica*, 26(3-4):389–429, 2000.
- [2] C. Chen. Improved moment invariants for shape discrimination. *Pattern Recognition*, 26(5):683–686, May 1993.
- [3] S. Derrode, M. Daoudi, and F. Ghorbel. Invariant content-based image retrieval using a complete set of Fourier-Mellin descriptors. In *ICMCS, Vol. 2*, pages 877–881, 1999.
- [4] R. Diaz de León and L. Sucar. Human silhouette recognition with Fourier descriptors. In *ICPR00*, pages 709–712, 2000.
- [5] H. Fujita. Group work of microactuators. In *International Advanced Robot Program Workshop on Micromachine Technologies and Systems*, pages 24–31, Tokyo, Japan, Oct 1993.
- [6] Y. Fukuta, Y.-A. Chapuis, Y. Mita, and H. Fujita. Design, fabrication and control of mems-based actuator arrays for air-flow distributed micromanipulation. *IEEE Journal of Micro-Electro-Mechanical Systems*, 15(4):912–926, Aug 2006.
- [7] K. Y. Goldberg. Orienting polygonal parts without sensors. *Algorithmica*, 10(2-4):210–225, 1993.
- [8] D. Lingrand. *Introduction au Traitement d’Images*. Vuibert, Paris, France, 2nd edition, Feb. 2008.
- [9] S. Loncaric. A survey of shape analysis techniques. *Pattern Recognition*, 31(8):983–1001, 1998.
- [10] M. Mercimek, K. Gulez, and T. Velimumcu. Real object recognition using moment invariants. *Sadhana*, 30(6):765–775, 2005.
- [11] K. Pister, R. Fearing, and R. Howe. A planar air levitated electrostatic actuator system. In *IEEE Workshop on Micro Electro Mechanical Systems*, pages 61–71, 1990.
- [12] R. Prokop and A. Reeves. A survey of moment-based techniques for unoccluded object representation and recognition. *GMIP*, 54(5):438–460, sep 1992.
- [13] A. Sajjanhar and G. Lu. A grid-based shape indexing and retrieval method. *Australian Computer Journal*, 29(4):131–140, 1997.
- [14] C. Shahabi and M. Safar. An experimental study of alternative shape-based image retrieval techniques. *Multimedia Tools Appl.*, 32(1):29–48, 2007.



**L I F C**

---

Laboratoire d'Informatique de l'université de Franche-Comté  
UFR Sciences et Techniques, 16, route de Gray - 25030 Besançon Cedex (France)

LIFC - Antenne de Belfort : IUT Belfort-Montbéliard, rue Engel Gros, BP 527 - 90016 Belfort Cedex (France)  
LIFC - Antenne de Montbéliard : UFR STGI, Pôle universitaire du Pays de Montbéliard - 25200 Montbéliard Cedex (France)

---

<http://lifc.univ-fcomte.fr>