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

EA 4269

Simulation to help calibration of a MEMS sensor network

Kahina Boutoustous — Eugen Dedu — Julien Bourgeois

Rapport de Recherche no RR 2011–????

THÈME 4 – 2010

Simulation to help calibration of a MEMS sensor network

Kahina Boutoustous , Eugen Dedu , Julien Bourgeois

Thème 4
OMNI

2010

Abstract: The Smart Surface[1] project aims at designing an integrated micro-manipulator based on an array of micromodules connected with a 2D array topology network. Each micromodule comprises a sensor, an actuator and a processing unit. One of the aims of the processing unit is to differentiate the shape of the part that is put on top of the Smart Surface. From a set of shapes this differentiation is done through a distributed algorithm that we call a criterion. The article presents Sensor Network Calibrator (SNC), a calibrator which allows to parametrize the Smart Surface and to determine the necessary number of sensors required by our Smart Surface. The tests will show that SNC is of great importance for choosing the number of sensors, and therefore to determine the size of the sensors grid.

Key-words: shape differentiation, distributed computing, MEMS

Laboratoire d'Informatique de l'Université de Franche-Comté,
Antenne de Montbéliard — UFR STGI,
Pôle universitaire du Pays de Montbéliard,
25200 Montbéliard Cedex (France)
Téléphone : +33 (0)3 81 99 47 43 — Télécopie +33 (0)3 81 99 47 91

Simulation pour aider à la calibration d'un réseau de capteurs MEMS

Résumé :

Mots-clés :

Laboratoire d'Informatique de l'Université de Franche-Comté,
Antenne de Montbéliard — UFR STGI,
Pôle universitaire du Pays de Montbéliard,
25200 Montbéliard Cedex (France)
Téléphone : +33 (0)3 81 99 47 43 — Télécopie +33 (0)3 81 99 47 91

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 array 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 among several models 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 parts, the Smart Surface will 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.

The aim of the Sensor Network Calibrator SNC which is presented in this article is to parametrize our Smart Surface, i.e. to find the size of the sensors grid necessary to differentiate our models. For that, a Smart Surface platform consisting of a Smart Surface prototype with a camera positioned above it is used. The rest of the paper is organized as follows. Section 2 details the Smart Surface project. Section 3 presents the SNC, while the tests are performed on section 4. Some related works to shape representation are presented in section 5 and they are followed by conclusions 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 [20, 10], servoed roller wheels [16, 17], magnetic [14] or thermobimoph and electrostatic [24]. Some of these preliminary studies use a sensorless manipulation scheme based on Goldberg's algorithm [12] 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. [2] have proposed a concept called ”programmable force field” which is an extension of 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 [19] or MEMS actuator arrays have 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 [1] or to be fully integrated [11].

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 (e.g. 4 × 4).

Figure 1 shows the Smart Surface. The rectangular holes seen on the front-side are the air nozzles. Air-flow comes through a micro-valve in the back-side of the device and then passes through the nozzle. The advantage of this solution is that the micro-actuators, the most fragile part of the surface, are protected. The circle holes are used by the micro-sensors to detect the presence or not of the part on the surface.

The strength of our project is the multidisciplinary collaboration between 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 differentiation of the part and communication infrastructure.

3 The SNC Calibrator

One of the parameters of the Smart Surface is its number of sensors. Due to space restriction, this number has to be chosen as wisely as possible: if it is overestimated, the design of the SS will be impossible, if it is underestimated, the part differentiation will also be impossible. It is therefore a crucial parameter that has to be set.

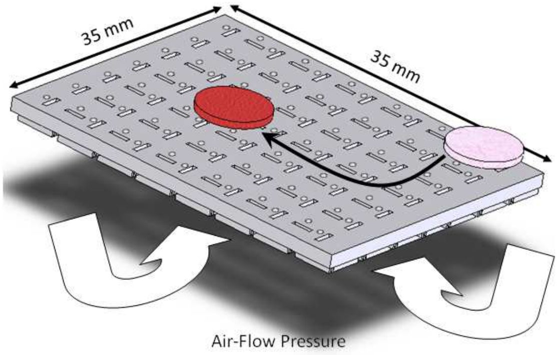


Figure 1: An overview of our Smart Surface.

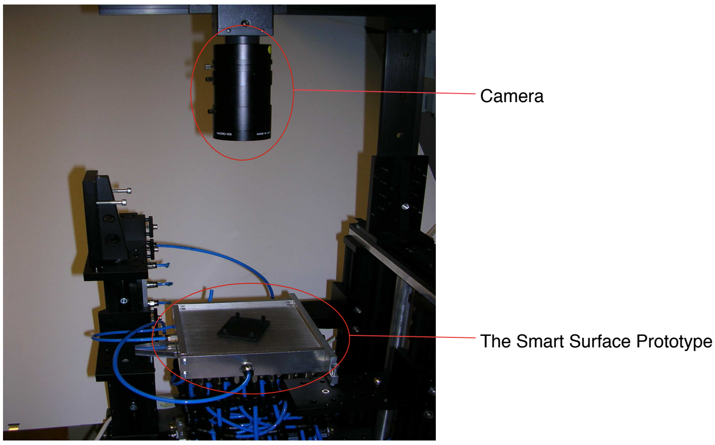


Figure 2: Overview of the Smart Surface platform.

3.1 Experimental platform

The Smart Surface is still in design phase and it is noticeably easier to construct a Smart Surface Prototype (SSP) at a greater scale than the microscale SS. So, until its final fabrication, a Smart Surface platform has been built, which consists of Smart Surface Prototype (SSP), with a camera positioned above it (see fig. 2).

The Smart Surface Prototype (SSP) is a 30x30 cm square surface with a 15x15 actuators array. The air flow wich comes through actuators continuously moves the part on SSP. A camera positioned above the SSP allows free discretization of the part.

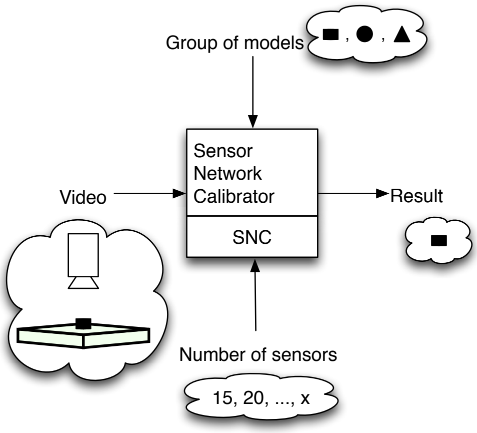


Figure 3: Overview of the calibrator.

The Smart Surface platform allows us to perform real experiments.

3.2 SNC overview

This section presents the calibrator that has been implemented to define the number of the sensors to be embedded into the Smart Surface. The calibrator presented in fig. 3 receives as input:

- the video from the camera which is positioned above the SSP; this camera shoots the SSP, as well as the parts on top of it;
- the group of models which is recognized by the SSP. A model is a part which is passed as a parameter;
- the sizes of the sensors grid to test.

The calibrator provides as output the result of the differentiation that is an answer to the following question: For each given size of sensors grid, what is the differentiation rate?

In our previous paper [3] we presented as future work the free rotation. In the current paper we present this free rotation. Another aspect of this paper is that it does not do exhaustive research, but only a few various models.

Fig. 4 presents the structure of our calibrator. The calibrator has two phases:

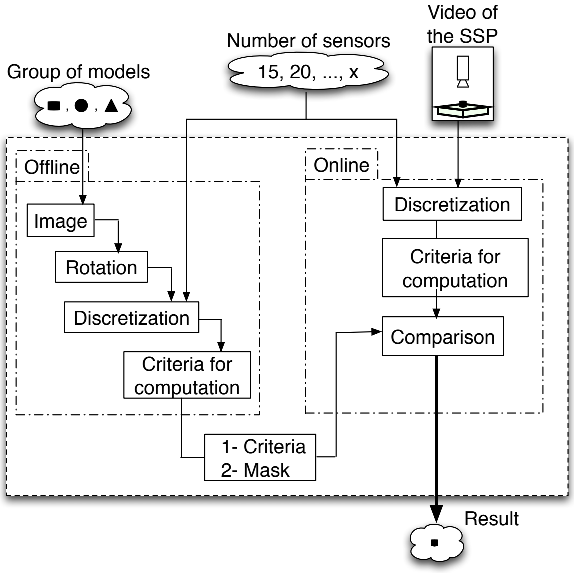


Figure 4: Global structure of our calibrator.

1. Offline: For each group of models of the SSP and each size of sensors grid, find the best combination of criteria (and their masks, see later) for the differentiation of the models between each other [3].

2. Online: An attempt to differentiate the part on the SSP is made using a differentiation algorithm, the various size sensors grid given as input and the results of the first offline phase.

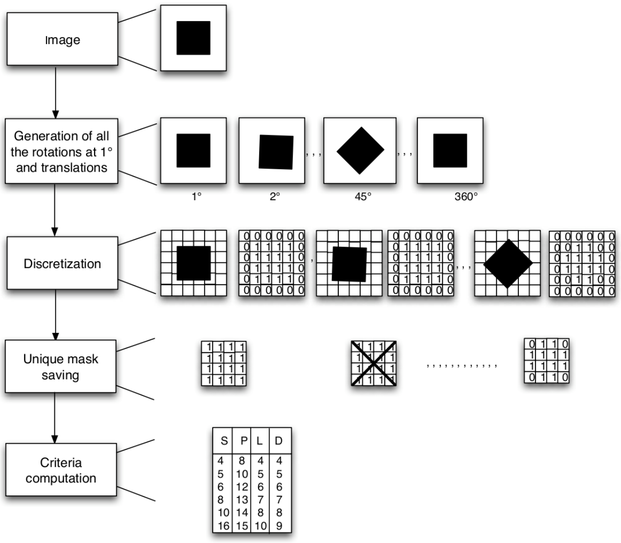
3.3 Offline phase

In this phase, detailed in fig. 5, for each group of models, all combinations of criteria to be differentiated among them are determined. This phase is subdivided into five sub-phases:

1. all rotations[1] of 1° are generated, $MRot_i$, from the image of the model M_i, given as input, with respect to the centre of the image, and all translations of $width(MRot)/10$ pixels are generated from the image $MRot_i$;

2. a grid corresponding to the positions of the sensors (middle point of the cell) is drawn on the images;

[1] The OpenCV library was used for rotations and translations.



S	P	L	D
4	8	4	4
5	10	5	5
6	12	6	6
8	13	7	7
10	14	8	8
16	15	10	9

Figure 5: Offline phase.

3. the images are discretized in a matrix by affecting 1 if the sensor is covered by the part and 0 otherwise; in order to have negligible errors for the rotation, the resolution of the image should be much greater than the resolution of the SSP;

4. unique masks corresponding to the initial matrix without the rows and columns that contain only zeros are saved;

5. the values of each criterion are calculated for all masks of the model.

The following algorithm details these 5 sub-phases:

```
for each size of sensors grid $(n, n)$, with $n \subset 15, 20, 25, 30, ..., 50$ do
    for each $M_i \subset M_1, M_2, M_3, ..., M_{nbr\_models}$ do
        Acquire the image $Im$ of the model on the SSP
        for $d = 1°$ to $360°$ do
            Generate $ImRot$ by rotating the image $Im$ by $d$ degrees
            for each $y \subset 0, 10, 20, 30, ..., Size(ImRot)$ do
                for each $x \subset 0, 10, 20, 30, ..., Size(ImRot)$ do
                    Generate $ImTrans$ by translating the image $ImRot$ by $x$ steps on $Ox$ and $y$ steps on $Oy$
```

```
          Discretize the image $ImTrans$
          Generate and save the mask
          Calculate and save the value for each criterion
        end for
      end for
    end for
  end for
end for
```

Among all the calculated criteria, the best criterion is chosen according to the differentiation rate, the memory cost and the execution time [3]. Finally, this criterion and its masks are given as input to the online phase.

3.4 Online phase

This phase consists of differentiating the part on the SSP, which means determining to which model recognized by the SSP it is associated. This phase is divided into four sub-phases:

1. the camera positioned above the SSP takes a sequence of images of the SSP and the part on it;

2. each image obtained is discretized (the image has only black and white pixels) to each grid size given as input to SNC, and the mask of the binary representation of this part is extracted;

3. the values of all combinations of the criteria of the masks are calculated;

4. the values of the criteria are compared with the values of the criteria of the models, to differentiate the part on the SSP.

The algorithm below describes this phase:

```
Film the part on the SSP
Generate the result tree of criteria values of the models recognized by
  the SSP
for each image of the part on the SSP do
  Discretize the image
  Generate the mask
  Compute the criteria values
  Compare with the values of criteria of each model
  Provide the results of the differentiation
end for
```

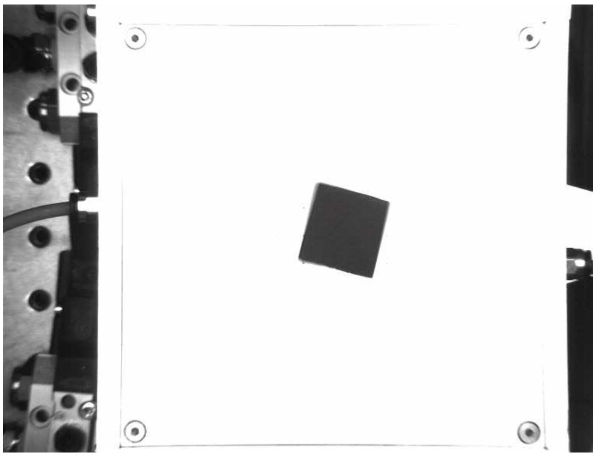


Figure 6: Picture of the SSP with a square on it.

4 Tests

The aim of our work is to parametrize the Smart Surface, i.e. finding the right size of the sensors grid to be used for differentiating [3] groups of given models (fig. 6 shows one of these models on the SSP). For this, several tests have been performed on the SSP. The differentiation is made by computing the different criteria and by applying a differentiation algorithm [3].

A set of four basic models have been chosen (see fig. 7(a)). Starting from these models all group of three models have been generated (see fig. 7(b)–(e)). Several sizes of grid sensors are used see below.

4.1 Offline phase

The offline algorithm is applied to each group of models. As already described, it consists of:

1. Generate all rotations of 1° and all translations of 10 pixels of the model images (size 550x550);

2. Discretize these images using the following sizes of sensors grid (15, 15), (20, 20), ..., (50, 50);

3. Save all unique binary representations of these images;

4. Compute criteria values for each size of sensors grid.

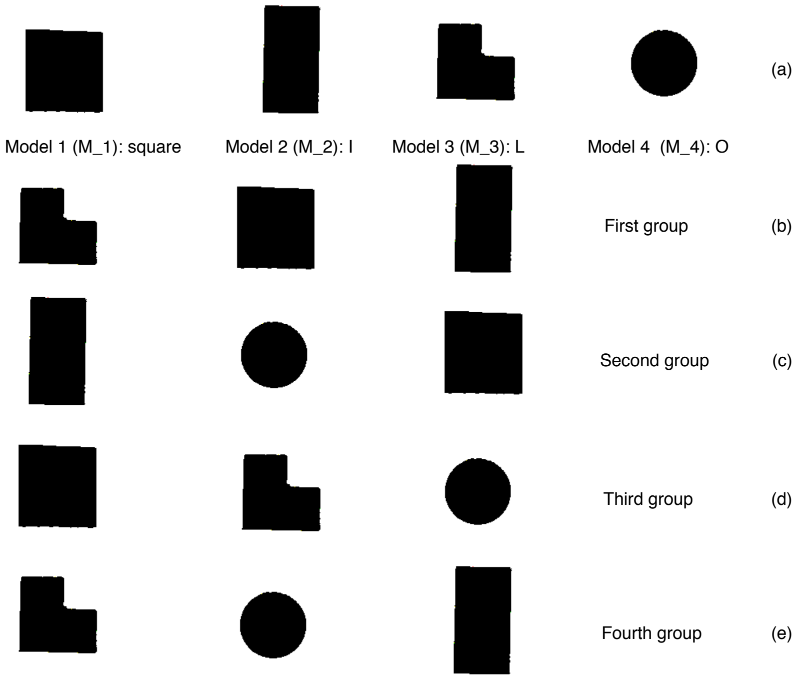
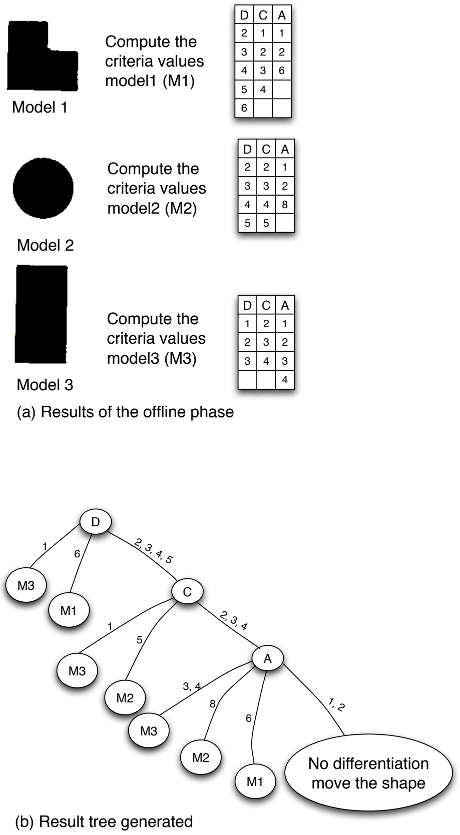


Figure 7: (a) Definition of our models, (b)–(e) all groups of three models.


Model 1
Compute the criteria values model1 (M1)
Model 2
Compute the criteria values model2 (M2)
Model 3
Compute the criteria values model3 (M3)
D
1
6
2, 3, 4, 5
M3
M1
C
1
5
2, 3, 4
M3
M2
A
3, 4
8
6
1, 2
M3
M2
M1
No differentiation move the shape

D	C	A
2	1	1
3	2	2
4	3	6
5	4	
6		

D	C	A
2	2	1
3	3	2
4	4	8
5	5	

D	C	A
1	2	1
2	3	2
3	4	3
		4

(a) Results of the offline phase

(b) Result tree generated

Figure 8: An example of the result tree of criteria values according to a combination of criteria and a set of models.

The offline phase consists of computing the values for differentiation criteria [3] calculated for the models M_1, M_2, M_3. Fig. 8(a) represents an example of results of the offline phase, with D, C, A being the differentiation criteria for the models L, O, I, the last group in fig. 7(e). Figure 8(b) presents the tree generated according with respect to the results shown in figure 8(a).

4.2 Online phase

The aim of this phase is to determine for each group of models, the size of a grid of sensors that allows differentiating our models. For this purpose, the following values must be computed for each group of models:

```
for each size of the sensors grid do
    for each $M_i \subset$ models of the group do
```

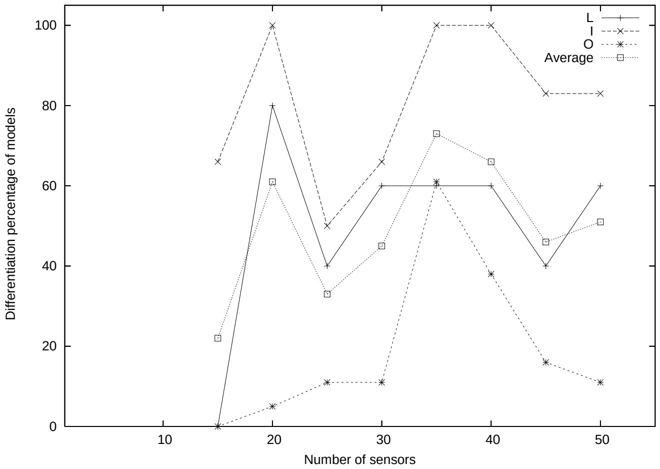


Figure 9: Differentiation percentage of models L, I, O, and their average.

```
        The differentiation rate for the $M_i$
    end for
end for
```

The algorithm of the online phase is applied to the group of models $\{O, L, I\}$ (see fig. 7(e)). The results are shown in fig. 9, representing the differentiation rate for the O, I and L belonging to group $\{O, L, I\}$.

The size of the sensors grid of the SSP must be determined such that it allows a better differentiation rate. For that, the average[2] differentiation rate of our models has been computed. Fig. 9 shows also the average differentiation rate of group $\{O, L, I\}$. It can be easily observed that with 35 sensors a higher differentiation rate average is obtained. With the same number of sensors (35) the following results are obtained:

- For the O, the highest differentiation rate is 61.11%.
- For the I, the highest differentiation rate is 100%.
- For the L, the differentiation rate is 60%.

Fig. 10, fig. 11 and fig. 12 contain the differentiation rate of each model belonging to each group of models: $\{square, I, L\}$, $\{square, L, O\}$ and $\{square, I, O\}$ respectively (see fig. 7(b), fig. 7(c), fig. 7(d)) as well as the average differentiation rate for each of them.

[2] We consider that the models have the same probability (33%) to be put on the SSP, otherwise weights should be used to compute the average.

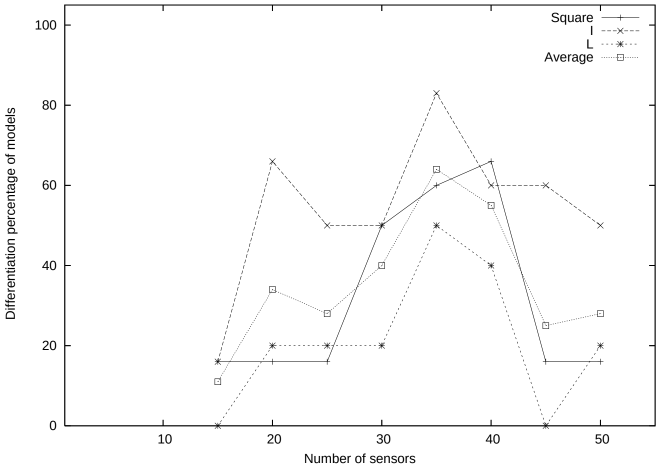


Figure 10: Differentiation percentage of models $square, I, L$, and their average.

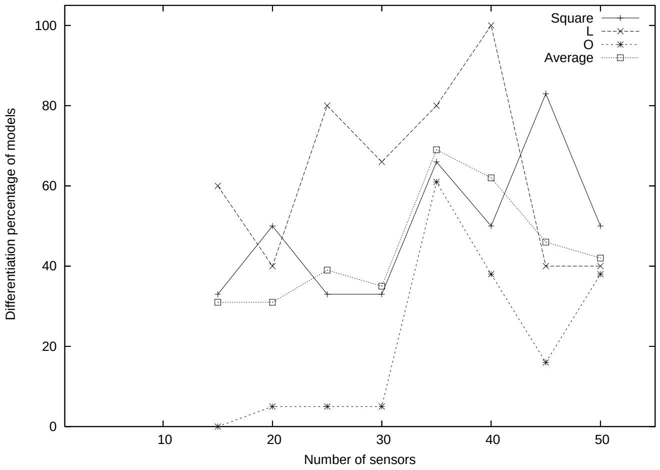


Figure 11: Differentiation percentage of models $square, L, O$, and their average.

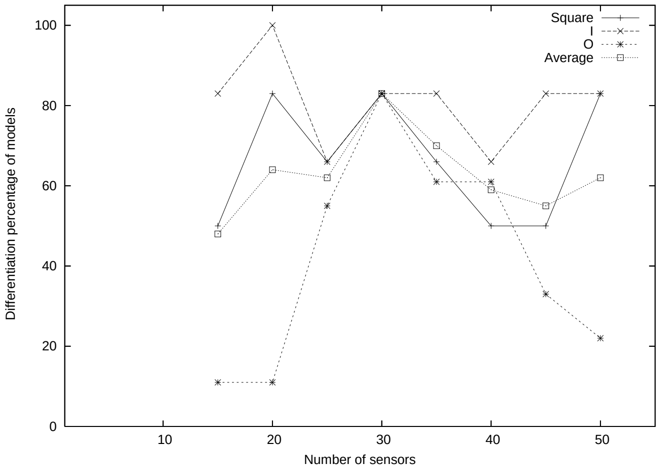


Figure 12: Differentiation percentage of models $square, I, O$, and their average.

Taking into account the values from fig. 10, fig. 11 and fig. 12 we can state that a sensors grid of $(35, 35)$ is an appropriate parameter for the SSP because a high differentiation rate average is obtained.

5 Related work

Several methods of shape representation exist in the literature. They are divided into two categories: *contour-based* methods and *region-based* methods. *Contour-based* methods are widely used. But generally, for complex images, the contour is not enough to describe the image content, therefore *region-based* methods are better.

5.1 Contour-based approaches

In the contour-based approach, the pixels of the contour are considered. The most used contour-based descriptor is Fourier descriptor.

The most used contour-based descriptor is Fourier descriptor [8, 29, 26].

5.1.1 Fourier descriptors

In this approach, the image is defined by one-dimensional function called *shape signature*, which is nothing else than a compact representation of the image [15]. Then *Fourier* transform is applied on this signature.

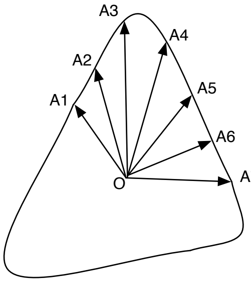


Figure 13: Centroid distance approach.

Shape signatures Many methods to calculate the signature have been developed. The three most common shape signatures are described, which are: centroid distance, chordLength signature and area function.

Centroid distance A centroid distance function uses the distance from the centroid (x_0, y_0) of the shape to the boundary points (distance between two consecutive points is equal, as shown in fig. 13) [26, 27]. Shape Signature centroid distance is defined by:

$$Z(t) = [x(t) - x_0] + i[y(t) - y_0] \tag{1}$$

where (x_0, y_0) are the coordinates of the centroid of the shape defined by:

$$x_0 = \frac{1}{N} \sum_{t=0}^{N-1} x(t), y_0 = \frac{1}{N} \sum_{t=0}^{N-1} y(t) \tag{2}$$

and $x(t), y(t)$ are the coordinates of point t, and N the number of points of the shape.

ChordLength signature The problem of the centroid approach is that it requires a point of reference which cannot apply for non-convex shape. Therefore [28] proposed a ChordLength signature, $r^*(t)$ derived from shape boundary without using any reference point. For each boundary point P, its $r^*(t)$ is the distance between P (see fig. 14) and another point of the boundary P' such as PP' is perpendicular to the tangent vector at P for each point P of the boundary. If there are several candidate points for P (in our example P_1 is a candidate points for P) the middle of PP_1 is researched and P_1 will be eliminated because P_2 is outside the shape. If P_2 is still within the shape, the middle of PP_2 and the middle of P_1P_2 are checked and this process is repeated until a middle point is found outside the shape.

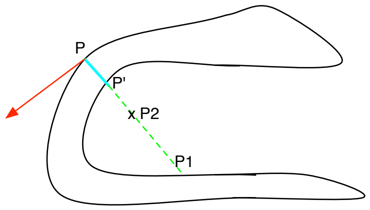


Figure 14: Example of application of the ChordLength signature method on a form that is not convex.

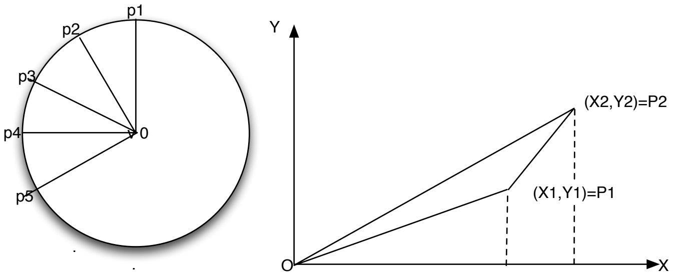


Figure 15: Area function for a circle shaped image.

Area function This function is defined by the surface of the shape. For instance, the triangle formed by P_1 and P_2 defines the Area function (see fig. 15), using this equation:

$$A(t) = \frac{1}{2}|x_1(t)y_2(t) - x_2(t)y_1(t)| \tag{3}$$

Fourier transform Once the shape signature $u(t)$ has been calculated, a Fourier transform is applied [7, 6], given by:

$$x_n = \frac{1}{n}\sum_{t=0}^{N-1} u(t)exp(-j2\pi nt/N) \tag{4}$$

where $n = 0, 1, \ldots, N - 1$.

The coefficients x_n are usually called Fourier descriptors of the shape. These descriptors represent the shape of the object in the frequency domain. The lower frequency descriptors contain information about global shape features of the shape. Although the number of coefficients generated from the transform is usually large, a subset of the coefficients is enough to capture the overall features of the shape. The very high frequency information describes

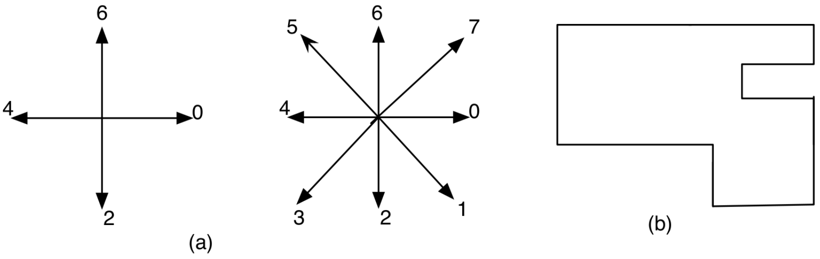


Figure 16: Sample code Freeman according 4 and 8 directions of move.

the fine details of the shape, it is not so helpful in shape discrimination, therefore, they can be ignored. The Fourier transform is invariant against translation, scale, rotation and their starting point.

5.1.2 Freeman code

Freeman coding consists in browsing through elementary moves the borders of the shape from a starting point and coding the movement (see fig. 16(a)) [13, 4].

For example in fig. 16(b), the highest left point is chosen as a starting point. From that point the code obtained is : 6460020242000. It is sensitive to rotation because Freeman code depends on the starting point. To reduce this dependence, the number formed has to be the minimal. Freeman code fig. 16(b) is: 0006460020242. The Freeman code is invariable to translation. It is also invariable to a rotation of 90° for the 4-connectivity and 45° for the 8-connectivity [9, 25].

The two methods presented, Fourier descriptor and Freeman code, are very effective and widely used for big pictures, where the outline of the image differs markedly 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 contour is equal or near equal to the surface.

5.1.3 Skeletons

In this method, the skeleton of a form X is defined by all the centers of the maximal circles that are strictly contained in X [4, 13] (see fig. 17(a)). In other words, a point s belongs to the skeleton $Sq(X)$ if and only if s is equidistant to two distinct contour points (see fig. 17(b)). It is clear that the skeleton depends entirely on the relative distance to the contour and thus the contour regularity. The skeleton has not always good properties. In fact

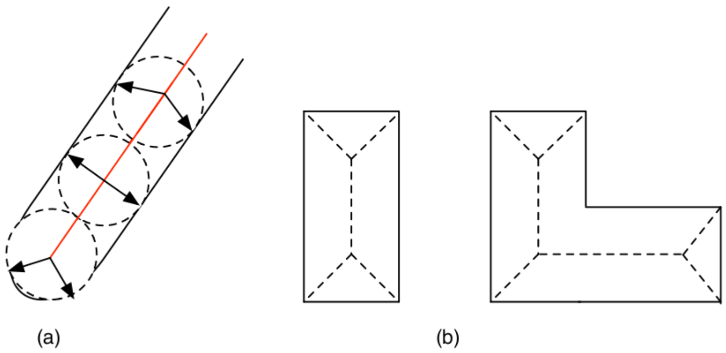


Figure 17: Definition of skeleton and example of skeletons of form.

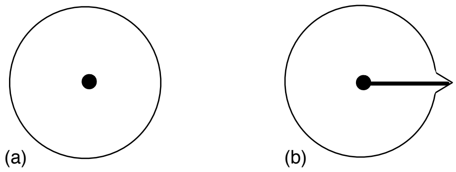


Figure 18: (a) Skeleton of a circle, (b) Skeleton of a slightly changed circle.

the skeleton of a circle which is slightly distorted on the outline is no longer a point but a straight line which connects the center and the asperity (see fig. 18).

This high noise sensitivity is the main problem with the skeleton. It is also very costly because it involves calculating the distance between every point in the object and the contour points.

To obtain a skeleton, a commonly used method is thinning repeatedly until convergence, i.e. when thinning no longer results in any change. Not to mention the noise sensitivity of the method of skeleton, this method is not very interesting for us because we are working on small pieces where thinning is useless.

5.2 Region-based approaches

In region-based methods, all the pixels within a shape are taken into account to obtain the shape representation. The most common region-based methods will be presented: grid based, invariant moments, skeletons and shape matrix.

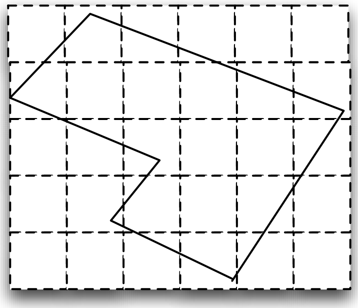


Figure 19: This map is a grid on our image.

5.2.1 Grid-based

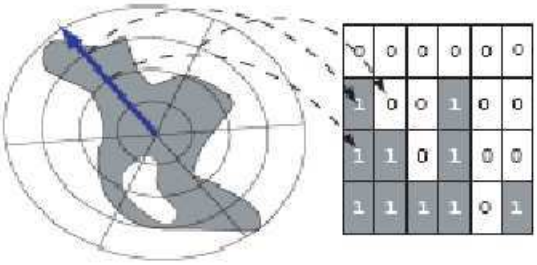
In this method [26, 23], a fixed-length grid of cells on the image is drawn, as shown in fig. 19. Going along our grid from top to bottom and from left to right, each cell that is wholly or partly covered by the form is affected with the value 1, and the other cells with 0 [23, 22]. This produces a binary number, which is the representation of our shape. For example in fig. 19 the binary representation is: 1111110101 010101010. The difference between two parts is given by a XOR between their binary representations. Such a binary representation is very sensitive to rotation, translation and dilatation, which requires a prestandardization [22].

5.2.2 Invariant moments

In this method [21, 18, 5], the invariant moments are used to represent the image. The $(p+q)$ order moment M of a density distribution function $f(x,y)$ of a two dimension image is defined by the following formula:

$$M_{pq} = \sum_{x=0}^{m} \sum_{y=0}^{m} x^p y^q f(x,y) \tag{5}$$

where: M_{pq}: the $(p+q)$-th moment of the image,
$m * m$: the size of the image,
$f(x,y)$: The function describing the image in each of its coordinates points.
H. Ming-Kuel proposed in 1963 a set of 7 descriptors called Husont invariants computed by normalizing central moments through order three. They are invariant to object scale, translation and orientation. These descriptors are used as input vectors for the classification method. There are several classification methods, among them neural networks are the most widely



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

Figure 20: Shape matrix.

used because of their fault tolerance, their ability of classification and their generalizability. The network architecture is as follows:

- The entries (the network receives seven input values).
- The hidden layer (the set of intermediate neurons).
- The output layer (the response of neurons according to entries).

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.

5.2.3 Shape matrix

This method (see fig. 20) is defined by placing a radial grid of size MN composed of M rings and N sectors at the center of a region, and then for each point of intersection between the radius and the circle of this grid, checking whether it belongs to the shape or not. It takes the value 1 if it is inside the shape, 0 otherwise. Focusing the grid on the center of the region makes the method robust against translations. For robustness against dilatation, the radius of the circle is defined as the largest radius of the circumcircle of the shape while the number of sectors and rings of the grid is fixed. For robustness against rotation a main direction must be established, which can be defined by the furthest point from the center of the shape.

This method applies to roughly compressed complex images. Intersection points between radiuses and circles are saved into a matrix which is a compact representation of the shape. Such a method is not very interesting given that we work on small images.

Method	Translation invariant	Rotation invariant	Dilatation invariant	Drawback
Centroid distance	yes	no	no	Applicable on closed shapes
Freeman code	yes	no	no	Very sensitive at dilatation
Grid-based	yes	yes	yes	Needs a standardization process
Invariant moments	yes	yes	yes	Very costly in time and calculation
Skeletons	yes	yes	no	Very sensitive to noise
Shape matrix	yes	yes	yes	Not very robust

Table 1: Comparison of criteria.

5.3 Comparison of criteria

Tab. 1 compares several methods presented above. Among the contour-based methods we see that in the Fourier transform method, the centroid the signature is the best among the signatures presented, is strong and preserves information. Among region-based methods the grid based method is very interesting in the case of very similar forms. The invariant moment based method is very time consuming as it is defined with both the borders of the form and its exterior. The skeleton method is very sensitive to noise.

6 Conclusions and future works

In this article we present the SNC calibrator which allows to parametrize our Smart Surface by determining the required number of sensors. Our tests performed on all groups of 3 out of 4 models show that a sensors grid of $(35, 35)$ is an appropriate parameter for the Smart Surface.

One of the ideas of our future work is to test more models with the SNC calibrator, and with a real Smart Surface when this will be operational. Also, to develop distributed algorithms for various criteria in order to implement them in the Smart Surface.

References

[1] D. Biegelsen, A. Berlin, P. Cheung, M. Fromherts, D. Goldberg, W. Jackson, B. Preas, J. Reich, and L. Swartz. Airjet paper mover. In *SPIE Int. Symposium on Micromachining and Microfabrication*, 2000.

[2] 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.

[3] K. Boutoustous, E. Dedu, and J. Bourgeois. An exhaustive comparison framework for distributed shape differentiation in a MEMS sensor actuator array. In *International Symposium on Parallel and Distributed Computing (ISPDC)*, pages 429–433, Krakow, Poland, July 2008.

[4] S. Bres, J.-M. Jolion, and F. Lebourgeois. *Traitement et analyse des images numériques*. Hermès, 2003.

[5] C. Chen. Improved moment invariants for shape discrimination. *Pattern Recognition*, 26(5):683–686, May 1993.

[6] 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.

[7] R. Diaz de Léon and L. Sucar. Human silhouette recognition with Fourier descriptors. In *ICPR00*, volume 3, pages 709–712, 2000.

[8] A. Folkers and H. Samet. Content-based image retrieval using Fourier descriptors on a logo database. In *ICPR02*, volume 3, pages 521–524, 2002.

[9] P.-E. Forssén and A. Moe. Contour descriptors for view-based object recognition. Technical Report LiTH-ISY-R-2706, Dept. EE, Linköping University, SE-581 83 Linköping, Sweden, Sept. 2005.

[10] H. Fujita. Group work of microactuators. In *International Advanced Robot Program Workshop on Micromachine Technologies and Systems*, pages 24–31, Tokyo, Japan, Oct. 1993.

[11] 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.

[12] K. Y. Goldberg. Orienting polygonal parts without sensors. *Algorithmica*, 10(2-4):210–225, 1993.

[13] D. Lingrand. *Introduction au Traitement d'Images.* Vuibert, Paris, France, 2nd edition, Feb. 2008.

[14] C. Liu, T. Tsao, P. Will, Y. Tai, and W. Liu. A micromachined permalloy magnetic actuator array for micro robotics assembly systems. In *The 8th International Conference on Solid-State Sensors and Actuators*, 1995.

[15] S. Loncaric. A survey of shape analysis techniques. *Pattern Recognition*, 31(8):983–1001, 1998.

[16] J. E. Luntz and W. Messner. A distributed control system for flexible materials handling. *IEEE Control Systems*, 17(1), Feb. 1997.

[17] J. E. Luntz, W. Messner, and H. Choset. Parcel manipulation and dynamics with a distributed actuator array: The virtual vehicle. In *IEEE Int. Conf. on Robotics and Automation (ICRA)*, pages 1541–1546, 1997.

[18] M. Mercimek, K. Gulez, and T. Velimumcu. Real object recognition using moment invariants. *Sadhana*, 30(6):765–775, 2005.

[19] G. C. Murilo and M. W. Peter. A general theory for positioning and orienting 2d polygonal or curved parts using intelligent motion surfaces. In *ICRA*, pages 856–862, 1998.

[20] 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.

[21] R. Prokop and A. Reeves. A survey of moment-based techniques for unoccluded object representation and recognition. *GMIP*, 54(5):438–460, Sept. 1992.

[22] A. Sajjanhar and G. Lu. A grid-based shape indexing and retrieval method. *Australian Computer Journal*, 29(4):131–140, 1997.

[23] C. Shahabi and M. Safar. An experimental study of alternative shape-based image retrieval techniques. *Multimedia Tools Appl.*, 32(1):29–48, 2007.

[24] J. Suh, S. Glander, R. Darling, C. Storment, and G. Kovacs. Combined organic thermal and electrostatic omnidirectional ciliary microactuator array for object positioning and inspection. In *Solid State Sensor and Actuator Workshop*, 1996.

[25] J. Sun and X. Wu. Shape retrieval based on the relativity of chain codes. In *MCAM07*, pages 76–84, 2007.

[26] D. Zhang and G. Lu. Content-based shape retrieval using different shape descriptors: A comparative study. In *ICME*, pages 1139–1142, 2001.

[27] D. Zhang and G. Lu. Shape-based image retrieval using generic Fourier descriptor. *SPIC*, 17(10):825–848, Nov. 2002.

[28] D. Zhang and G. Lu. Study and evaluation of different Fourier methods for image retrieval. *Image Vision Comput.*, 23(1):33–49, 2005.

[29] H. Zhang and E. Fiume. Shape matching of 3-D contours using normalized Fourier descriptors. In *SMI*, pages 261–268, 2002.

LIFC

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