

## Multi-Objects Interpretation

Jean-Philippe Tarel

INRIA

Domaine de Voluceau, Rocquencourt,

B.P. 105, 78153 Le Chesnay Cedex, France.

<http://www-syntim.inria.fr/~tarel/>

E-mail: [Jean-Philippe.Tarel@inria.fr](mailto:Jean-Philippe.Tarel@inria.fr)

### Abstract

*We describe a general-purpose method for the accurate and robust interpretation of a data set of  $p$ -dimensional points by several deformable prototypes. This method is based on the fusion of two algorithms: a Generalization of the Iterative Closest Point (GICP) to different types of deformations for registration purposes, and a fuzzy clustering algorithm (FCM).*

*Our method always converges monotonically to the nearest local minimum of a mean-square distance metric, and experiments show that the convergence is fast during the first few iterations. Therefore, we propose a scheme for choosing the initial solution to converge to an "interesting" local minimum. The method presented is very generic and can be applied:*

- *to shapes or objects in a  $p$ -dimensional space,*
- *to many shape patterns such as polyhedra, quadrics, splines,*
- *to many possible shape deformations such as rigid displacements, similitudes, affine and homographic transforms.*

*Consequently, our method has important applications in registration with an ideal model prior to shape inspection, i.e. to interpret 2D or 3D sensed data obtained from calibrated or uncalibrated sensors. Experimental results illustrate some capabilities of our method.*

### 1. Introduction

Model-based interpretation of output from 2D edge detectors and 3D reconstruction algorithms has received growing attention over recent years. In particular, data matching and registration relative to a reference model, produced either interactively or automatically is of crucial importance

in computer vision. Recent examples can be found in such varied topics as 3D medical imaging, aerial site observation, and range images.

We distinguish two major approaches to 3D registration: matching in progress approaches such as hypothesis and verification [6] or Iterative Closest Point methods (ICP) [1], and accumulation approaches [11] such as generalized Hough transforms. These methods perform only one shape model fitting. Few studies based on superquadrics have generalized registration to several shapes [7, 4, 10].

For interpretation of 2D data, as opposed to 3D data, popular approaches in unsupervised pattern recognition are based on clustering. Thus, patterns are only particular shapes such as lines and ellipses but several patterns can be taken into account. Specifically, fuzzy clustering approaches derived from Fuzzy c-Mean Method (FCM) [2], and known for their practical efficiency in situations where a great deal of uncertainty exists, have proved capable of carrying out geometric clustering [3, 9].

But 2D and 3D problems are the same, so we formalize and unify the description of this key problem by taking advantage of two approaches proposed in the computer vision and pattern recognition context. Thus, in this paper, we present:

- a Generalization of the ICP algorithm (GICP) to transforms such as similitude and homography in section 2,
- and in section 4, a generalization of the GICP algorithm to take into account several patterns by using the clustering approach which is summarized in section 3.

Unifying FCM and ICP is very powerful and generic, and allows a large class of algorithms to be generated that handle a great number of patterns in a  $p$ -dimensional space. Thus, as an illustration, we present in section 5 results obtained using the method which explains a data points set as the overlapping of simple geometric shapes when these shapes undergo a simple geometric transform.

## 2. Generalized Iterative Closest Point method

Besl [1] has introduced an interesting 3D rigid registration algorithm named Iterative Closest Point (ICP) where matching is implicitly done in an iterative least square minimization of the distance between data and model. The convergence of the ICP algorithm to a local minimum is demonstrated, when data are exhaustive and the transform between data and model is a rigid displacement. But a natural generalization for rigid transforms to nonrigid ones is to substitute rigid displacement by another type of transform  $F$  such as a similitude, an affine or homographic transform.

Given a data set of points, we suppose that we know the shape modeling this set, and the type of the deformation between data and model. For instance, to interpret a data set which is the sampling of a parallelogram, we can choose the unit square, in its centered reference system, as the model shape which undergo affine deformations. The aim of the GICP is to obtain the best transform which superposes data and model shape.

In this paper, we focus on five types of transforms  $F$ , but numerous extensions can be developed such as snakes, splines and physical deformable models. The five chosen types of transform are:

- translation:  $F(X) = X + T$ ,
- rigid displacement, or a rotation and a translation:  $F(X) = RX + T$ ,
- similitude, or a rigid displacement and a scale factor  $s$ :  $F(X) = sRX + T$ ,
- scales transform, or a rigid displacement and a scale factor on each axis:  $F(X) = RLX + T$  with  $L$  a diagonal matrix,
- affine transform:  $F(X) = AX + T$ ,
- homographic transform.

For particular shapes: the unit square, the unit circle and the unit isosceles triangle in its centered reference system, it is interesting to bear in mind the class of shapes generated when these simple shapes undergo one of the previous transforms. With the unit square, all squares, rectangles, parallelograms or quadrilaterals can be interpreted by choosing the adequate transform, whereas with the unit circle, it is the set of all circles, ellipses, or conics. Figure 1 shows the different possibilities in detail. Note that a set of points, lines or triangles, implicit or parametric curve/surface can be used as the model of the shape.

### 2.1. GICP algorithm

We now describe how to find  $F$  with the GICP algorithm, when the data is a set  $\{X_i\}_{i=1,\dots,n}$  of  $n$  given points:

	Trans.	Displa.	Simil.	Affine	Homo.
centered unit triangle 	unit oriented triangles 	unit isosceles triangles 	isosceles triangles 	triang. 	triang. 
centered unit circle 	unit circles 	unit circles 	circles 	ellipses 	conics 
centered unit square 	unit oriented squares 	unit squares 	squares 	parall. 	quadril. 

Figure 1. shapes generated by geometric transforms on three particular objects.

- **step 0**: Choose the model shape  $S$ .  $F$  is initialized not too far from an interesting solution.
- **step g**: Compute the closest point  $X'_i$  from the model shape  $S$  of each data point  $F(X_i)$ .
- **step g'**: Compute transform  $F$  which minimizes the mean-square distance between sets  $\{X'_i\}$  and  $\{F(X_i)\}$ .
- **step g''**: If the change in mean-square error below a preset threshold specifying the desired precision, stop; else go to **step g**.

The Generalized Iterative Closest Point algorithm needs to be able to compute analytically the best least square solution for each transform type (**step g'**), when two sets of  $n$  matched points ( $\{X_i\}$  and  $\{X'_i\}$ ) are given. For each of the transforms discussed here, an algorithm for finding the best solution in **step g'** exists. If  $F$  is an affine transform, this evaluation is just a linear least square problem, but for other types of transforms, the solution is not so obvious (see [8] for the displacement case, for example).

The mean-square error minimized by the GICP is:

$$e_{GICP}(F, X'_i) = \sum_{i=1}^n d^2(F(X_i), X'_i) \quad \text{with } X'_i \in S \quad (1)$$

where  $d(X, X')$  is the Euclidean distance between the two points  $X$  and  $X'$ .

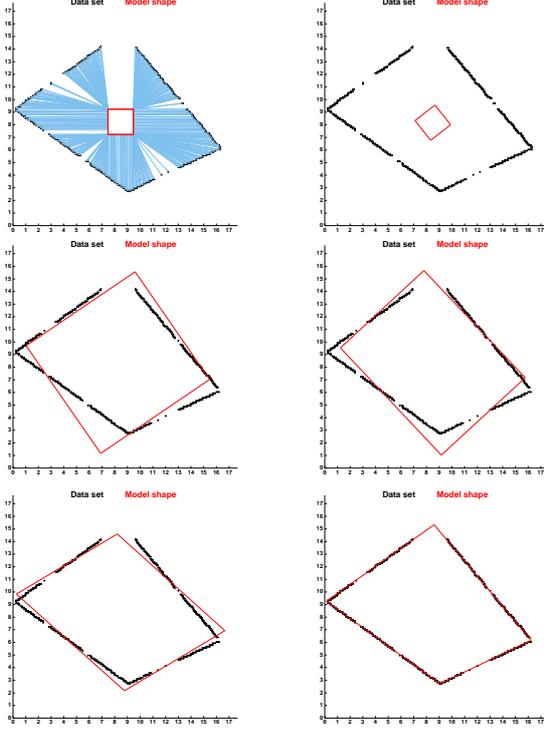
### 2.2. Convergence

*The Generalized Iterative Closest Point algorithm always converges monotonically to a local minimum with respect to the mean-square objective function (1).*

This convergence theorem of the GICP algorithm is exactly the same as the one proposed in [1], in a more generic framework, since its demonstration does not require transform  $F$  to be a rigid displacement.

The reason why the algorithm converges is that the two steps **step g** and **step g'** of the GICP algorithm reduce the

mean distance  $e_{GICP}$  between corresponding points at each iteration. Because  $X_i^!$  is the closest point to  $F(X_i)$ , **step g** reduces individually each term of sum (1), and thus also the average distance. **step g'** generically reduces the average distance, because it is a least square minimization. Consequently, as the lower bound of  $e_{GICP}$  is 0, the algorithm must converge monotonically to a local minimum value with respect to  $e_{GICP}$ .



**Figure 2. GICP Registration with respectively, the oriented unit square, unit square, square, rectangle, parallelogram and quadrilateral. On the above left graph, matches between model shape and data are drawn.**

Figure 2 shows the GICP algorithm results where  $F$  is, respectively, a translation, a rigid displacement, a similitude, a scale transform, an affine transform and a homographic transform. In this example, the model shape is the unit square in its canonical reference system. Note that the data can be truncated without setting the algorithm a problem.

### 3. Clustering Method

The ICP approach is limited by the fact that only one shape model can be used at a time. The key idea is then to add a clustering framework to the GICP algorithm to obtain a multi-shapes interpretation method. But before describing this new method in detail (section 4), a clustering

algorithm named Fuzzy c-Means (FCM) and its advantages are presented.

Many clustering methods are available, but we choose FCM because it uses fuzzy memberships of a data point which sum to one across classes. Thus, ambiguous points do not bias the solution as is the case for hard methods. For our problem, this property is very important in order to avoid trivial solutions where the model shape is reduced to a point or degenerated, and to improve convergence speed.

#### 3.1. Fuzzy Clustering Method

The FCM algorithm, introduced by Bezdek [2], and its derivatives have been very successfully used in many applications such as pattern recognition and image segmentation. Fuzzy clustering is a generalization of partitioning methods in that the clusters are not sub-sets but fuzzy sub-sets of the data set of  $n$  points in  $\mathbf{R}^p$ . That is, each data point  $X_i$  has a membership  $u_{ik}$  between 0 and 1 which is a degree of sharing for each cluster  $k$ . It is assumed that the memberships  $u_{ik}$  are computed so that their sum for each point is 1. This constraint forces the cluster method to explain the whole data set by  $c$  fuzzy sub-sets. Cluster prototype is the center of gravity  $T_k$  of the cluster. The clustering algorithm minimizes the following functional:

$$e_{FCM}(T_k, u_{ik}) = \sum_{i=1}^n \sum_{k=1}^c (u_{ik})^m d^2(X_i, T_k) \quad (2)$$

with  $\sum_{k=1}^c u_{ik} = 1$ .

The FCM algorithm is summarized as follows (the convergence to a local minimum of this algorithm is demonstrated in [2] when  $m > 1$ ):

- **step 0**: Fix the number of clusters  $c$ , and the fuzzy exponent  $m$ . Initialize randomly the location of the cluster centers  $T_k$ .
- **step f**: Generate the new partition using the following equation of the fuzzy memberships:

$$u_{ik} = \frac{1}{\sum_{j=1}^c \left( \frac{d(X_i, T_k)}{d(X_i, T_j)} \right)^{\frac{2}{m-1}}} \quad (3)$$

- **step f'**: Compute new cluster centers  $T_k$  using the following equation:

$$T_k = \frac{\sum_{i=1}^n (u_{ik})^m X_i}{\sum_{i=1}^n (u_{ik})^m} \quad (4)$$

- **step f''**: If the distance partition is stable, stop; else go to **step f**.

### 3.2. Extensions

In the FCM algorithm, the cluster prototype is only a point. It seems very interesting for computer vision and pattern recognition applications to have a more generic prototype, such as a geometrical shape which may be a line, a triangle, a rectangle or an ellipse. FCM derivatives exist which allow some particular model shapes to be considered: lines and hyper-planes [2], hyper-ellipsoidal shells [3], spherical shells [9] by changing the prototype definition. But these methods are specific only to a particular shape. By introducing the ICP approach, we overcome this limitation and propose in the next section a generic algorithm to cluster geometric shapes which can be applied to a great number of shapes only by defining a procedure to find the closest point on the shape to a given point.

## 4. Unification of GICP and FCM algorithms

We can notice similarities and connections between FCM and GICP algorithms. First, the two methods minimize an objective function (1) and (2) by a two-step iterative algorithm. In particular **step f'** and **step g'** are the same computation: the geometric transform between shape and data. Moreover, the equation is the same with the GICP in the translation case, and the FCM in one cluster prototype case. Thus, the key idea is to add the clustering framework to the GICP algorithm to obtain a Multi-Objects Interpretation method (MOI).

### 4.1. Description of the MOI algorithm

The Multi-Object Algorithm without outliers is:

- **step 0**: Fix the number of clusters  $c$ , the exponent  $m$  ( $m = 1.5$  in all our tests) and the model shape  $S$ . Initialize the transforms  $F_k$  (see section 4.3).
- **step m**: Compute the new closest point  $X'_{ik}$  from the model shape  $S$  of each data point  $F_k(X_i)$ .
- **step m'**: Generate the new partition using the following equation of the fuzzy memberships:

$$u_{ik} = \frac{1}{\sum_{j=1}^c \left( \frac{d(F_k(X_i), X'_{ik})}{d(F_j(X_i), X'_{ij})} \right)^{\frac{2}{m-1}}} \quad (5)$$

- **step m''**: Compute new best  $F_k$  which minimizes the  $(u_{ik})^m$  weighted mean-square error between points  $F_k(X'_{ik})$  and  $X_i$ .
- **step m'''**: If the distance partition is stable, stop; else go to **step m**.

Consequently, the MOI algorithm minimizes the objective function below:

$$e_{MOI}(F_k, X'_{ik}, u_{ik}) = \sum_{k=1}^c \sum_{i=1}^n (u_{ik})^m d^2(F_k(X_i), X'_{ik}) \quad (6)$$

with  $\sum_{k=1}^c u_{ik} = 1$  and  $X'_{ik} \in S$ .

We can view the GICP algorithm as a particular case of the MOI algorithm, when the cluster is unique ( $u_{i1} = 1$ ). And we can interpret the FCM algorithm as another particular case, when the shape model  $S$  is reduced to a point and the transform  $F_k$  is a translation  $T_k$ .

### 4.2. Convergence

*The Multi-Object Interpretation algorithm always converges monotonically to a local minimum with respect to the mean-square objective function (6) ( $m > 1$ ).*

The outlines of the demonstration are analogous to those of GICP and FCM. The principle is that each step reduces the objective function (6). Of course this convergence is only to a local minimum. Thus, the question arises as to initializing the MOI algorithm to reach an interesting minimum.

### 4.3. Initialization

In the GICP and MOI approaches, an initial solution is assumed to be known because, the obtained solution is critically dependent in practice on the initial estimation. In particular, the less the transform is constrained, the nearer the initialization must be to the correct result.

Consequently, we propose a scheme for the initialization of  $F_k$  transforms:

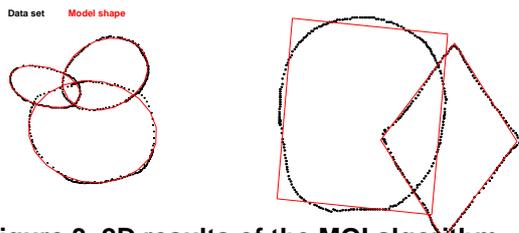
- Find the translation  $T_k$  of  $F_k$  by identification to solutions of the FCM algorithm.
- Use inertia matrices of previous fuzzy subsets to compute an initial rotation or an initial affine transform.

Of course, coarse-to-fine approach can be used, such as find the best displacements, then the best affine transforms, and finally the best homographic transforms (see figure 2).

We assume the data shapes not to be interlaced in this scheme. This approach is satisfactory in most of the configurations. But, as it is shown in figure 3, although the method is able to cope with certain particular and difficult configurations, when it fails further investigation is required with regard to initializing the MOI algorithm.

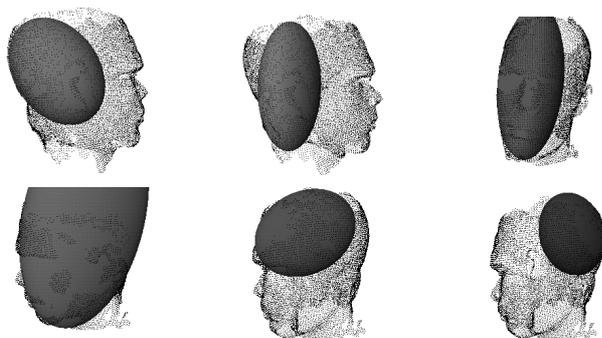
## 5. Results

In figure 3(a), point sets are synthetic, but noisy and partial. The first set is 3 intersecting ellipses. The second is



**Figure 3. 2D results of the MOI algorithm.**

a parallelogram intersecting an ellipse. The scheme of the MOI initialization does not always give a good initialization because the data shapes are interlaced, nevertheless in this case it is sufficient and the MOI algorithm succeeds. Figure 3(b) shows that the MOI algorithm can produce interesting solutions although the model shapes do not match the data, thus it is relatively robust. This property allows the proposed method to be used to segment data with heterogeneous patterns. In this case, the criterion used is the mean distance between model shapes and data. Figure 4



**Figure 4. 3D human head is interpreted as the superposition of 15 ellipsoids by the MOI algorithm.**

shows the application of the MOI algorithm in the simple case where the model shape is the unit sphere and the transform is affine. Data points are from a human head, courtesy of P.Fua [5]. Bottom up and down, sides of the skull and forehead are well recognized. Of course, our method also serves as a framework for image segmentation in planar, quadratic or cubic patches.

## 6. Conclusion

First, we propose a generalization of the ICP method to several important transforms such as similitudes and homographic transforms. But, as only one object can be taken into account by the Generalized Iterative Point algorithm, we propose a new method for Multi Object Interpretation of a point data set. This method takes advantage of fuzzy

clustering algorithms and the weighted GICP algorithm to produce robust results from noisy data. The MOI algorithm is described in a simple manner. Nevertheless, it is very powerful and generic. Moreover, the MOI algorithm is easily controlled by a small number of parameters: the most important of them is the number  $c$  of clusters.

Finally, we would like to point out that the proposed approach where clustering and implicit registration are mixed is a promising research topic. We believe that the approach can be successfully applied with snakes, splines and physical deformable models, etc. The extension to 2D functions allows image segmentation algorithms to be produced very directly. Although generalization to p-D projections of q-D shapes may raise problems of stability, it is nevertheless an interesting subject with important applications in multi-object tracking.

## Acknowledgments

The author is grateful to Nozha Boujemaa for pointing out important references related to the paper.

## References

- [1] P. Besl and N. McKay. A method for registration of 3D shapes. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 14(2):239–256, 1992.
- [2] J. Bezdek. *Pattern Recognition with Fuzzy Objective Function Algorithms*. Plenum Press, 1981.
- [3] R. Dave and K. Bhaswan. Adaptive fuzzy c-shells clustering and detection of ellipses. *IEEE Trans. Neural Networks*, 3(5):643–662, 1992.
- [4] F. Ferrie, J. Lagarde, and P. Whaite. Darboux frames, snake, and super-quadratics: Geometry from the bottom-up. In *Proceedings of Workshop on Interpretation of 3D Scenes*, pages 170–176, Austin, TX, 1989.
- [5] P. Fua and Y. Leclerc. Registration without correspondences. In *Conference on Computer Vision and Pattern Recognition, (Seattle, WA, June 21-23, 1994)*, pages 121–128. IEEE Computer Society Press, 1994.
- [6] W. Grimson and T. Lozano-Perez. Localizing overlapping parts by searching the interpretation tree. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 9(4):469–482, 1987.
- [7] A. Gupta and R. Bajcsy. Volumetric segmentation of range images of 3D objects using superquadric models. *CVGIP - Image Understanding*, 58(3):302–326, 1993.
- [8] K. Kanatani. Analysis of 3D rotation fitting. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 16(5):543–549, 1994.
- [9] R. Krishnapuram, O. Nasraoui, and H. Frigui. The fuzzy c-spherical shells algorithm: A new approach. *IEEE Trans. Neural Networks*, 3(5):663–670, 1992.
- [10] A. Pentland. Recognition by parts. In *First International Conference on Computer Vision, (London, England, June 8–11, 1987)*, pages 612–620. IEEE Computer Society Press, 1987.
- [11] G. Stockman. Object recognition and localization via pose clustering. *Computer Vision, Graphics and Image Processing*, 40(2):361–387, 1987.