Particle Swarm Optimisation Based 3D Reconstruction of Sketched Line-Drawings

Ana PIQUER *, Pedro COMPANY *, Manuel CONTERO +

* *Departament de Tecnologia, Universitat Jaume I,*
Campus del Riu Sec. E-12071 Castellón. Spain
Ana.Piquer@uji.es , pcompany@tec.uji.es

+ *Dpto. de Expresión Gráfica en la Ingeniería.*
Universidad Politécnica de Valencia, Camino de Vera s/n. E-46022 Valencia. Spain
mcontero@degi.upv.es

Abstract. The purpose of this paper is to demonstrate the application of particle swarm optimisation to line drawings reconstruction. The paper's new contribution is the application of swarm intelligence in dealing with machine perception of sketch-based modelling interfaces. Traditional descent or gradient-based optimisation algorithms are not always practical in this context because of the severe numerical noise and ill-defined objective function of the optimisation-based reconstruction problem Our results point to particle swarm optimisation as a promising alternative.

Keywords. Machine Perception, Sketch Understanding, Line Drawings Reconstruction, Swarm Intelligence, Particle Swarm Optimisation.

Introduction

Particle Swarm Optimisation (PSO) is based on a simplified social model that is closely tied to *Swarming Theory* [1]. PSO belongs to the family of non-gradient based probabilistic search algorithms. These algorithms are generally easy to implement do not require continuity in response functions and are well suited for finding global or near global solutions. However, they exhibit potentially high computational cost.

On the other hand, sketching-based geometric modellers (see Fig. 1) are aimed to improve design support systems [2]. Optimisation has been used with some success in this sketch-based modelling approach [3], [4], but typical failure rates for 3D reconstruction by optimisation approaches (due to local minima) are still important. Consequently, one of the main challenges in optimisation-based 3D reconstruction is the mathematical formulation of *perceptual cues*, also called *artefacts*, or *regularities* and the other is the search for more efficient and robust "global" optimisation algorithms.

In this paper we include first a related work section, to justify the state of the art in both PSO and optimisation-based 3D reconstruction. Next a brief description of PSO algorithm is included to introduce the notation and to clarify the particular version of PSO we do employ in our sketching-based geometric modeller. The main contributions of the paper are on section "Optimisation Strategy and Parameters", where the approach followed to adapt PSO to a 3D reconstruction problem is presented. Discussion of tests done to validate the approach and some conclusions are included at the end.

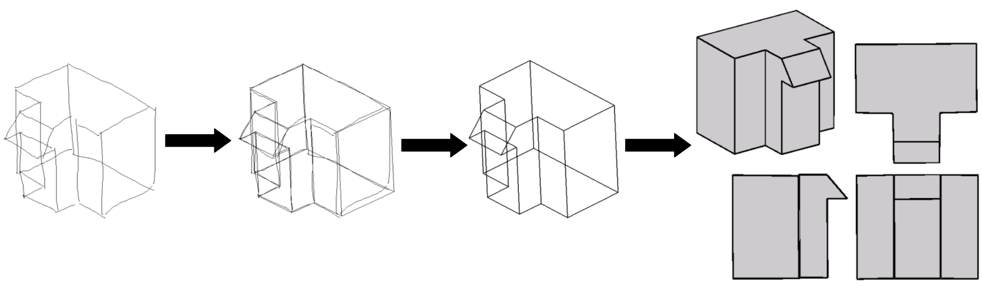


Figure 1. 3D reconstruction process from sketch to model through line-drawing

1. Related Work

Particle swarm optimisation is based on the universal behaviour of individuals, which can be summarized as: *evaluate*, *compare* and *imitate* [1]. In the numerical implementation of this simplified social model, the population is referred to as a swarm and each individual as a particle. The numerical implementation repeatedly updates the position of each particle to simulate the adaptation of the swarm to the environment. PSO was first introduced by Kennedy and Eberhart [5], [6]. Currently, they do exist standard textbooks on PSO, like [7], treating both the social and computational paradigms. Besides, interesting information can be easily obtained through a simple search on the Web, i.e. an extensive bibliography on PSO can be obtained from [8]. However, as far as we know, PSO has not yet been applied to solve geometric reconstruction problems.

In the field of geometrical reconstruction, the most recent contributions can be classified as algebraic or optimisation based. In the algebraic approach, the emphasis is putted in testing the correctness of line drawings and solving a system of algebraic equations that represent the conditions, constraints or requirements of the model (see [9]-[13]).

Optimisation-based 3D reconstruction was first introduced by Marill [3], and was significantly improved later by several authors (see [14]-[22]). In this approach, the (x_j, y_j) coordinates of every junction in the drawing are made equal to (x_j, y_j) coordinates of the corresponding vertex in the 3D model, and the aim is to *inflate* the drawing, in other words, to find a set of z coordinates of all its vertices that represents a *psychologically plausible* 3D model (see Fig.2).

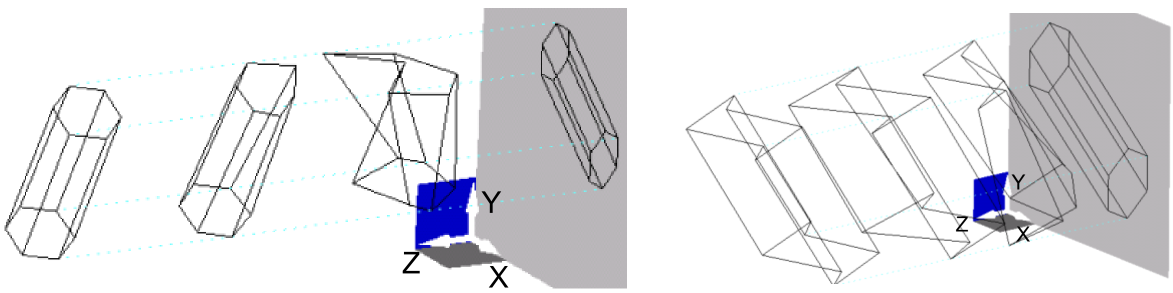


Figure 2. Plausible (left) and unplausible model (right) after inflating the same line-drawing.

Inflation is performed by *optimising* an objective function:

$$F(z)= \Sigma\, \alpha_j\, R_j(z) \qquad (1)$$

Where, α_j is the j-th weighting coefficient that normalise regularities' ranges and sensitivities. Regularities ($R_j(z)$) are perceptual cues of the line drawing that must correspond to some properties in the searched model and are expressed in terms of the independent variables z. Regularities are formulated to be equal to zero for a complete compliance of the condition. Regularities have been studied by some authors ([3],[4],[15],[21]) and dissertations about their formulation and their importance in the reconstruction process can be found in [22] and [23].

Optimisation-based reconstruction is prone to local minima. For instance, the input line drawing becomes a local minimum when edge parallelism, or face planarity regularities are used, since they are trivially accomplished in the line drawing. The strategies developed to avoid this trivial optimum lead to "tentative" initial models [22] (models that tend to drive optimisation near the global optimum). However, global-search optimisation still seems to be a good alternative to tentative models; at least in those cases where tentative models cannot be obtained at all, or have a high computational cost. In this context, Brown and Wang's [18] proposed (although apparently did not implement) the use of standard methods for minimizing local minima difficulties. Later, it has been said that the introduction of Simulated Annealing algorithms, which were supposed to be able to find the global minimum, resulted in similar failure rates to those of local optimisation algorithms [19]. Nevertheless, as far as we know, no other promising families of global search algorithms haven been tested in this context.

2. Particle Swarm Optimisation Algorithm

The basic process followed by PSO can be outlined as follows:

1. Create an initial swarm of N particles, with a random distribution and random velocities.
2. Calculate a velocity vector for each particle, using the knowledge gained by the swarm and the particle's memory.
3. Update the position of each particle, in terms of its velocity vector and previous position (see equation (3)).
4. Go to Step 2 and repeat until convergence.

In our implementation of particle swarm optimisation, the *position* of the i-th particle is represented as:

$$\boldsymbol{z}^i = (z^i_1, z^i_2, \ldots, z^i_m) \qquad / \; i = 1,2,\ldots, N \qquad (2)$$

Where N is the number of particles and m is the number of vertices in the drawing. The position of each particle $\boldsymbol{z}^i$ at iteration $k+1$ changes in function of its position in the previous iteration ($\boldsymbol{z}^i_k$), a vector called *velocity* ($\boldsymbol{v}^i = (v^i_1, v^i_2, \ldots, v^i_m)$) and a time step value (Δt):

$$\boldsymbol{z}^i_{k+1} = \boldsymbol{z}^i_k + \boldsymbol{v}^i_{k+1}\,\Delta t \qquad (3)$$

A unit time step ($\Delta t = 1$) is the usual criterion, and so is done throughout the present work.

Velocity can be formulated in different ways but a formulation widely used and accepted in the literature is shown in (4):

$$\boldsymbol{v}^i_{k+1} = w\,\boldsymbol{v}^i_k + c_1\,r_1\,(\boldsymbol{p}^i - \boldsymbol{z}^i_k) + c_2\,r_2\,(\boldsymbol{p}^g - \boldsymbol{z}^i_k) \qquad (4)$$

New velocity of every particle is determined according to its previous velocity ($\boldsymbol{v}^i_k$), the distance of its current position ($\boldsymbol{z}^i_k$) from its own best position, and the distance of its current position from the group's best position. With this formulation previous velocity effect is controlled by a weight factor (w), called *inertia*; r_1 and r_2 are random numbers with values between 0 and 1; c_1 and c_2 are factors of confidence in the position of the particle and in the position of the swarm, $\boldsymbol{p}^i$ is the best position of particle i so far and $\boldsymbol{p}^g$ is the best position in the swarm along the process until now:

$$\boldsymbol{p}^i = (p^i_1, p^i_2, \ldots, p^i_m) \qquad / \; i = 1,2,\ldots, N \qquad \text{and} \qquad \boldsymbol{p}^g = (p^g_1, p^g_2, \ldots, p^g_m) \qquad (5)$$

In order to apply PSO for reconstructing 3D model from line drawings, specific parameters and initial values have to be set.

3. Optimisation strategy and parameters

To solve the step 1 in the PSO process described above, the *initial swarm* is created with all particles randomly distributed throughout the design space, each with a random initial velocity vector:

$$z^i_0 = r_1\,2\,e\,\Delta xy - e\,\Delta xy \qquad / \qquad i=1,2,\ldots,m \qquad (6)$$

$$v^i_0 = r_2\,2\,e\,\Delta xy - e\,\Delta xy \qquad / \qquad i=1,2,\ldots,m \qquad (7)$$

$$\text{with} \qquad \Delta xy = \max\left(|x_{max} - x_{min}|,\ |y_{max} - y_{min}|\right) \qquad (8)$$

Where r_1 and r_2 are random numbers between 0 and 1, and $x_{max} = max(x_1, x_2,\ldots,x_m)$, $x_{min} = min(x_1, x_2,\ldots,x_m)$, $y_{max} = max(y_1, y_2,\ldots,y_m)$ and $y_{min} = min(y_1, y_2,\ldots,y_m)$. In other words, we recognise benefits of a proportional model [22], and, hence, the design space range of $\boldsymbol{z}$ coordinates is made proportional to the line-drawing dimensions. The value of e is fixed to 0.5, to place initial position of particles between [$-\Delta xy$, Δxy].

It must be highlighted that random initial swarm effectively prevents the appearance of the trivial optimum associated with, for instance, line parallelism and face-planarity regularities.

We consider the *convergence*, cited in the 4th step, is achieved when the first of the next three *stopping criterions* is fulfilled:

- Maximum *number of iterations* is achieved.
- Costs of objective function are monitored, and the process is finished whenever a value less or equal to a threshold (usually 0.01) is obtained.
- The maximum change in the objective function for a specified number of consecutive iterations is less than a predefined allowable change (i.e. if any improvement exists for 20 consecutive iterations).

It has been described that a relation between number of particles and number of iterations exists*:* in most cases, increasing number of particles allows to reduce the number of iteration required. In our earlier experiments we reached the same conclusion. Moreover, we realised that the number of particles should increase in parallel to the drawing's complexity. Hence, we did adopt the criteria of automatically assigning the number of particles N as three to ten times the number of vertices in the line drawing, and simultaneously giving an appropriated convergence criterion that make to reduce the number of iterations when the optimum is reached.

In spite of the above general configuration, there are three problem dependent parameters in equation (4): the "inertia" of the particle's velocity (w), and c_1 and c_2, that are called *trust parameters*, *learning factors* or *attraction coefficients.* The values of the later usually range from [0, 4], and in many studies both are equal to 2, i.e. [5]. Trelea [24] did use a single trust parameter instead of two separate ones, and gave 1.7 as the optimum. Values of $c_1=c_2=1.5$ were adopted since they provide good behaviour in our examples.

The *inertia* weight has characteristics similar to the temperature parameter in the simulated annealing optimization. Trelea [24] found [-1 , 1] as the valid range and 0.73 as the best inertia value. It is generally argued that large inertia values facilitate global searches, which are best during first steps of the optimisation process and helps to fasten the whole process. On the other hand, small values facilitate local searches, favouring in this way the final refinement of the output model.

Finally, it was considered that it is more or less usual to limit the velocity in PSO algorithms: the *maximum velocity* value determines maximum change allowed of a particle for each iteration and we fix this range in:

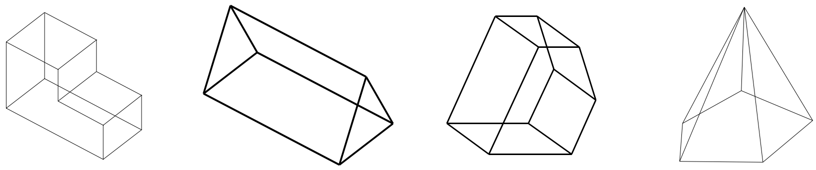
$$- d \cdot \Delta xy < \mathbf{v}^{i}_{k+1} < d \cdot \Delta xy \qquad (9)$$

We limited the search region in order to preserve proportionality. That is a criteria very important in 3D reconstruction, since it outcomes from the fact that human observers assume that the object has been represented by choosing a "general" point of view that clearly enhances the actual proportions [22]. And after the observation of the behaviour of some examples we determine d=0.25.

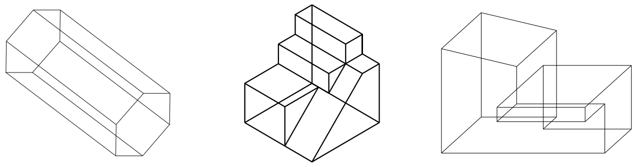
In addition to adjusting PSO, the reconstruction process must be adjusted too. In this sense, the sum of regularities in (1) does not always contain the whole set of regularities, but just the "appropriate" subset of them [22-23]. The common criterion is to select such subset depending on the shape to reconstruct. For instance, symmetry alone gives good results when planes of symmetry exist, and corner-orthogonality gives good result for "rectangular trihedral polyhedra" (example #1 in fig. 3).

4. Results

In our reconstruction engine, called REFER, the calculations and management of data are implemented in C++, GUI is a calligraphic interface implemented on the Wintab API under Windows 2000, and 3D visualisation is implemented in OpenGL. Some examples, extracted from the current literature on optimisation-based 3D reconstruction, were tested to validate this approach (see Fig.3). Tests were done on a Pentium III based personal computer and with graphical interaction active (screen was refreshed during execution), since we were checking for reliability and robustness, and we were not mainly concerned on efficiency.



#1 **#2** **#3** **#4**



#5 **#6** **#7**

Figure 3. Line drawings tested in this article.

In order the ensure repeatability of the experiments, we did use axonometric line-drawings instead of sketches, but this means no loose of generality since it has been said elsewhere that those drawings can be obtained from sketches by pre-processing steps [22] (see Fig.1).

To avoid the lack of repeatability inherent to PSO, tests were done by repeatedly executing the algorithm with the same configuration parameters. In a first scan 10 executions were made, and up to 30 executions were accomplished when required to "refine" the results in the neighbourhood of best values. Results were considered good when F(**z**) was close to zero *and* also the shape was *perceived* as the most plausible one.

As an example, table 1 summarizes results of varying w in the seven examples presented in figure 3. The remaining parameters were fixed to their best values: N=10·m, c_1=c_2=1.5, d=0.25, e=0.5. Those parameters were selected according to the reasons expressed in section 3. In each example, percentage of success and mean of iterations until a convergence criterion was reached are shown. The regularities employed in the reconstruction are presented too.

Table 1. Success rates versus *w* variation

	w=0.7		*w*=0.5		*w* =0.3		Regularities
	Success	Mean of iterations	Success	Mean of iterations	Success	Mean of iterations	
Example #1	100%	79	100%	40	100%	23	Total Symmetry
Example #2	100%	44	100%	37	100%	32	Total Symmetry
Example #3	98%	46	98%	40	100%	41	Total Symmetry
Example #4	100%	222	100%	128	100%	94	Total Symmetry
Example #5	100%	26	100%	24	100%	18	Total Symmetry
Example #6	93%	178	100%	221	100%	91	Corner Orthogonality and Parallelism
Example #7	38%	69	75%	84	56%	70	Corner Orthogonality and Parallelism

From our experiments we can conclude that:

- Number of particles larger that the ones proposed ($3 \cdot m \leq N \leq 10 \cdot m$) increase execution times without improving final models. And shorter values cut-off the process before optimum is achieved. N=10·m and a number maximum of iterations=1000 maximices the sucess ratio but execution time too. Nevertheless, the number of iterations never reaches to this value because of the convergence critera.
- Convergence criteria give good success rates and low processing times since they effectively cut-off the executions when optimum is achieved.
- Ranges of inertia propossed in the literature are appropriate in 3D reconstruction. But lower values of w makes the algorithm's behavior more conservative, because it produces shorter movements of particles in each iteration. That can become the swarm slow, but after testing many examples, results in most cases are even better, and the number of iterations employed do not increase. Even more, in many cases, number of iterations decreases with w.
- e parameter is responsible of the initial position of particles. A good initial distribution of particle can save time during the optimitation. A value of 0.5 allow to place particles in a space with dimension 2 times the dimension of original drawing. That seems a good value to obtain fastest convergence while guaranteeing the perceptual requisite of keeping proportions.

- Limits in velocity of each particle is determined by *d* and dimensions of original drawing. A small value of *d* can make the limitation too restrictive, but a value of 0.25 is giving good results without making to fall the algorithm in local minima for this reason.

Based in our experience working in 3D reconstruction with descent-based algorithm, computational times are higher for PSO in those problems where both algorithms work well. However, the PSO algorithm works better with severe numerical noise (i.e. complex objective functions), while descent algorithms are easily trapped in local minima (i.e. examples #7 and #4) or even in trivial optimum (without escaping from the drawing plane).

5. Conclusions

Applicability of swarm intelligence in dealing with optimisation-based 3D reconstruction problem has been studied. A valid PSO implementation has been presented as a novel method in the domain of the optimisation-based 3D reconstruction problem. The approach has been tested and usefulness of PSO algorithm in a 3D reconstruction environment has been proved.

PSO is an attractive alternative to descent or gradient-based algorithms, whether use of those regularities that have the trivial optimum is required (i.e., line parallelism and face planarity) or the objective function becomes too complex and prone to local minima. PSO has an acceptable success rate in avoiding local minima, even when using those regularities that are trivially accomplished in the input line drawing.

However, computational cost is too high and fine adjustment of parameters is required for every example, since general coefficients seem not to work fairly well in all cases. Hence, more study is required in those areas.

Acknowledgements

This work was partially supported by Universitat Jaume I (*Plan 2002 de promoció de la investigació a l'UJI*, Project P1-1B2002-08, entitled "From sketch to model: new user interfaces for CAD systems").

References

[1] J. Kennedy, R.C. Eberhart and Y. Shi, *Swarm Intelligence*. Morgan Kaufmann Publishers, San Francisco, CA, 2001.

[2] D.G. Ullman, "Toward the ideal mechanical engineering design support system." *Research in Engineering Design.* 13: 55-64, 2002.

[3] T. Marill, "Emulating the Human Interpretation of Line-Drawings as Three-Dimensional Objects." *Int. Journal of Computer Vision*, 6(2): 147-161, 1991.

[4] H. Lipson and M.A. Shpitalni, "New Interface of Conceptual Design Based on Object Reconstruction From a Single Freehand Sketch." *Ann CIRP*. 44(1): 133-136, 1995.

[5] J. Kennedy and R.C. Eberhart, "Particle Swarm Optimization," *Proc. of IEEE International Conference on Nerual Networks*, Perth, Australia, 1942–1948, 1995.

[6] R.C. Eberhart and J. Kennedy, "A New Optimizer Using Particles Swarm Theory," *6th Int. Symposium on Micro Machine and Human Science*, Japan, 39–43, 1995.

[7] E. Bonabeau, M. Dorigo and G. Theraulaz, *Swarm intelligence. From natural to artificial systems*. Oxford University Press. New York. 1999.

[8] X. Hu, PSO Bibliography. *http://web.ics.purdue.edu/~hux/biblioDate.shtml*. Last visited Nov. 26, 2003.

[9] K. Sugihara, *Machine interpretation of Line Drawings*. The MIT Press. 1986.

[10] A.K. Mackworth, "Interpreting pictures of polyhedral scenes" *Artificial Intelligence*, 4: 121-137, 1987.

[11] W. Wang and G.A. Grinstein, “Survey of 3D Solid Reconstruction from 2D Projection Line Drawings” *Computer Graphics Forum*, 12 (2): 137-158, 1993.

[12] I.J. Grimstead and R.R. Martin, “Creating solid models from single 2D sketches” *Proc 3rd Symposium on Solid Modeling and Applications*, *ACM Siggraph*: 323-337, 1995.

[13] L. Ros and F. Thomas, "Overcoming Superstrictness in Line Drawing Interpretation". *IEEE Trans. on Pattern Analysis and Machine Intelligence,* 24(4): 456-466, 2002.

[14] L. Baird and P. Wang, “3D object recognition using Gradient descent and the Universal 3-D array grammar.” *Proc. SPIE. Intelligent Robots and Computer Vision X*, 1607: 711-718, 1991.

[15] Y. Leclerc and M. Fischler, “An Optimization-Based Approach to the Interpretation of Single Line Drawings as 3D Wire Frames.” *International Journal of Computer Vision,* 9(2): 113-136, 1992.

[16] L. Baird. and P. Wang, “3D Object Perception Using Gradient Descent.” *Int. Journal of Mathematical Imaging and Vision (IJMIV)*, 5: 111-117, 1995.

[17] H. Lipson and M. Shpitalni, “Optimization-Based Reconstruction of a 3D Object from a Single Freehand Line Drawing.” *Computer-Aided Design,* 28(8): 651-663, 1996.

[18] E.W. Brown and P.S. Wang, “Three-dimensional object recovery from two-dimensional images: A new approach.” *Proc. SPIE. Intelligent Robots And Computer Vision XV*, 2904: 138-147, 1996.

[19] P. Company, J.M. Gomis and M. Contero, “Geometrical Reconstruction from Single Line Drawings Using Optimization-Based Approaches.” *WSCG’99,* 2: 361-368, 1999.

[20] H. Lipson and M. Shpitalni, “Correlation-Based Reconstruction of a 3D Object From a Single Freehand Sketch” *AAAI Spring Sym. Series- Sketch Understanding,* 2002.

[21] A. Piquer, R.R. Martin and P. Company, “Skewed Mirror Symmetry for Depth Estimation in 3D Line-Drawings”**.** *Lecture Notes in Computer Science*. Accepted for publication. July 2004.

[22] P. Company, M. Contero, J. Conesa and A. Piquer, “An optimisation-based reconstruction engine for 3D modelling by sketching.” *Computers & Graphics. Accepted for publication*. 2004.

[23] A. Piquer, “Percepción artificial de dibujos lineales”. *Tesis doctoral, Universitat Jaume I,* (in spanish) ISBN: 84-688-7061-7, 2003.

[24] I.C. Trelea, “The particle swarm optimization algorithm: convergence analysis and parameter selection”. *Information Processing Letters,* 85: 317–325, 2003.