

A Modified Particle Swarm Optimization Applied in Image Registration

M. Khalid Khan*, Jens Hedrich* Ingela Nyström*
*Centre for Image Analysis, Uppsala University, Sweden
Email: {khalid, jens.hedrich, ingela}@cb.uu.se

Abstract—We report a modified version of the particle swarm optimization (PSO) algorithm and its application to image registration. The modified version utilizes benefits from the Gaussian and the uniform distribution, when updating the velocity equation in the PSO algorithm. Which of the distributions that is selected depends on the direction of the *cognitive* and *social* components in the velocity equation. This direction checking and selection of the appropriate distribution provide the particles with an ability to jump out of local minima. The registration results achieved by this new version verifies the robustness and its ability to find a global minimum.

I. INTRODUCTION

Image registration is a fundamental problem in several image analysis and computer vision applications [1]. Figures 1 and 3 shows examples where registration is necessary prior to analysis. Image registration techniques are generally classified into two main classes: feature-based and area-based techniques. Feature-based techniques [1], [2] search for common features (edges, junction points, boundaries etc., of an object) between the reference and the floating image and provide a mathematical transformation which maps the floating image to the reference image. It is usually very difficult to find reliable feature correspondences between the floating and the reference image which limits the application of feature-based techniques. Image registration from area-based techniques can be thought of as a maximization problem often requiring an optimization algorithm to speed up the search process. It can be formulated as:

$$T^* = \arg \max_{\alpha} O(A, T_{\alpha}(B)), \quad (1)$$

where A and B are the reference and floating images, respectively. T_{α} is one possible transformation where α is some generalized index. O represents the objective function (similarity measure) and T^* defines the transformation which maximizes O over all possible values of α . We have chosen normalized cross-correlation as our objective function and particle swarm optimization (PSO), see, e.g., [3], as optimization algorithm. PSO belongs to the family of population-based optimization algorithms where each individual (particle) is a candidate solution. These particles move in the multi-dimensional search space to find the optimum within some constraints. Since its introduction during the last decade, many versions have been proposed. Most of these versions use uniform distribution to generate random vectors to update the velocity equation of the PSO. Uniform distribution



(a) Reference image



(b) Floating image

Fig. 1. The superimposed circular regions are used for registration.

provides good exploitation of the search space but at the cost of exploration. This is why uniform distribution in the PSO algorithm often gets trapped in a local minima. Some researchers have also reported promising results using Gaussian, Cauchy and exponential distributions [4], [5], [6]. Gaussian distribution often provides a good balance between exploration and exploitation of the search space. It has been reported that PSO with Gaussian distribution can also get stuck in local minima, while [6] provides a jump strategy to get out of such situations. We present a modified version of PSO (MPSO) which takes advantage both of the exploration abilities of the Gaussian distribution as well as of the exploitation abilities of the uniform distribution. This removes the need of a jump strategy mentioned in [6].

II. PARTICLE SWARM OPTIMIZATION

PSO represents all possible solutions of the optimization problem through particles. The movement of every particle is dependent on its past experience and that of its

neighboring particles. Mathematically, it is represented as:

$$x_i^{k+1} = x_i^k + v_i^{k+1}. \quad (2)$$

Here, x_i^k and x_i^{k+1} represent the past and the current position of the particle x_i , respectively, while v_i^{k+1} represents its velocity. Note that it is the velocity vector that is driving the particle in the search space. The velocity can be written as:

$$v_{ij}^{k+1} = \beta[v_{ij}^k + c_1 r_{1j} \{y_{ij}^k - x_{ij}^k\} + c_2 r_{2j} \{\bar{y}_{ij}^k - x_{ij}^k\}], \quad (3)$$

where x_{ij}^k is the current location of the i th particle in dimension j at k th step. y_{ij}^k is the best position found by particle x_i since its initialization (*personal best*), while $\bar{y}_{ij}^k$ is the best position found by the neighborhood defined (using ring topology) of particle x_i since initialization (*local best*). The term $c_1 r_{1j} \{y_{ij}^k - x_{ij}^k\}$ is often called the *cognitive component* as it is responsible for attracting the particle back to its own best position. The term $c_2 r_{2j} \{\bar{y}_{ij}^k - x_{ij}^k\}$ is known as the *social component* as it quantifies the performance of particle x_i relative to its neighbors. r_1 and r_2 are random vectors between $[0, 1]$ drawn from a uniform distribution and are responsible for the stochastic nature of the algorithm. v_{ij}^k is the velocity of the particle x_i at k th step in dimension j and is often referred to as the momentum, because it prevents the particle from drastically changing its direction and provides a bias towards the previous direction of the particle. Here, β is a constriction coefficient introduced by [3] to control the velocity of particles and to ensure convergence. It can be represented as:

$$\beta = \frac{2\kappa}{|2 - \psi - \sqrt{\psi(\psi - 4)}|}, \quad (4)$$

where $\kappa \in [0, 1]$ and $\psi = c_1 + c_2$. A smaller value of κ results in fast convergence with local exploitation, while larger values results in slow convergence but with higher degree of exploration. κ is often set to 1 which is successful for most applications [3].

The remaining two parameters are often set to $c_1 = c_2 = 2.05$ [6], which scales the *cognitive* and *social components* equally. It is difficult to know a priori the individual weights of these two components, so it seems logical to weigh them equally.

III. MODIFIED PSO

As mentioned in the previous section, c_1 and c_2 are often set to 2.05. Convergence is reached for these values while for higher values, the algorithm often gets stuck in local minima. To improve the robustness of the PSO, a novel approach is to take different decisions based on the directions of the *cognitive* and the *social* components. In fact, most previous studies have considered positive random vectors either from the Gaussian or the uniform distribution, which is an important constraint for convergence. We present a simple modification which takes into consideration the directions of these components while updating the velocity. The modified algorithm is summarized in Algorithm 1, where f defines the objective function to be minimized. p_{vec}^i is a vector between the

Algorithm 1 Modified PSO as Minimization Problem

```

1: uniformly distribute  $n$  particles in the search space
2: repeat
3:   for each particle  $i = 1, \dots, n$  do
4:     % set the personal best position
5:     if  $f(x_i) < f(y_i)$  then
6:        $y_i = x_i$ ;
7:     end if
8:     % set the neighborhood best position
9:     if  $f(y_i) < f(\bar{y}_i)$  then
10:       $\bar{y}_i = y_i$ ;
11:    end if
12:  end for
13:  for each particle  $i = 1, \dots, n$  do
14:     $p_{vec}^i = y_i - x_i$ ;
15:     $n_{vec}^i = \bar{y}_i - x_i$ ;
16:    if  $\text{sign}(p_{vec}^i)$  is same as  $\text{sign}(n_{vec}^i)$  in all dimensions then
17:      update the velocity using Eq. 3 with  $N(0, 1)$ 
18:    else
19:      update the velocity using Eq. 3 with  $U(0, 1)$ 
20:    end if
21:    update the position using Eq. 2
22:  end for
23: until stopping condition is true

```

personal best position (y_i) of a particle x_i and its current location, while n_{vec}^i is a vector between the best position ($\bar{y}_i$) found by the neighborhood of x_i and its current location.

If these two vectors have the same sign in all dimensions, then we update the velocity by using Equation 3, but with Gaussian distribution having zero mean and unit variance. This update will not only provide the particles with the freedom to change their direction abruptly, it also provides the particles with a chance to jump faraway from its current location. For instance, in case of a 2-dimensional search space, if p_{vec}^i and n_{vec}^i has the same sign and if they lie in the first quadrant, then with the PSO, it will continue to explore a small portion of the first quadrant. See Figure 2(a). More precisely, all position updates for a particle will make the particle to converge to a point on a line that connects the *personal best* position of that particle and the *local best position* [7]. The red and green vectors correspond to p_{vec} and n_{vec} , respectively, while the blue vector corresponds to the velocity of the particle. Here, we assume that the particle is currently at the origin. The multiplication of random vectors drawn from a uniform distribution with p_{vec} and n_{vec} will result in any vector from the red and green rectangular regions, respectively. For the sake of simplicity, we have used $c_1 = c_2 = 1$ while drawing illustrations in Figure 2. For the MPSO, the particles can jump to any of the four quadrants and thus will provide more opportunities of exploration as shown in Figure 2(b). Here, for a better visual comparison, we have superimposed the geometrical representation of the MPSO on top of the PSO. The

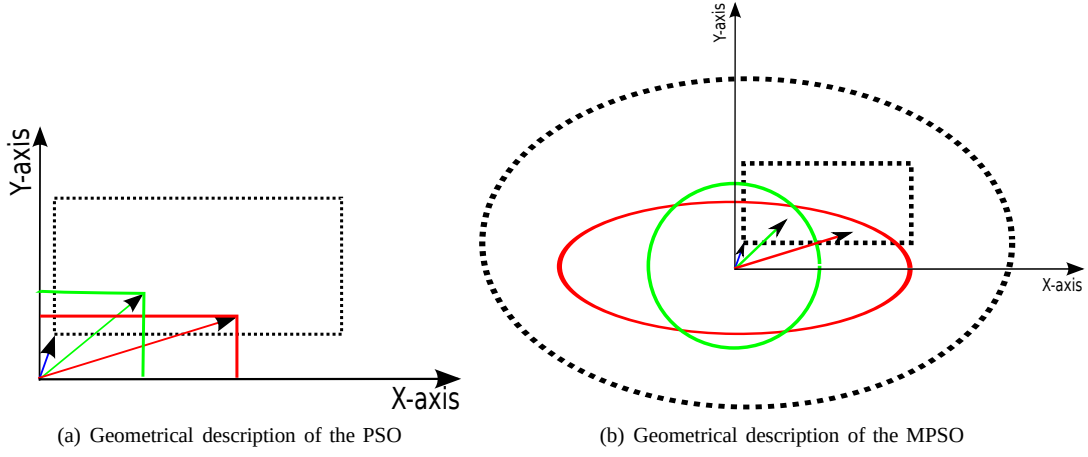


Fig. 2. PSO and MPSO will end up in the dotted rectangle and dotted ellipse, respectively.

multiplication of random vectors drawn from a Gaussian distribution with p_{vec} and n_{vec} will result in any vector from the red and green elliptical regions, respectively. Here, a particle will choose its new location within the region prescribed by the dotted ellipse. The large support provided by the MPSO will help the particle to explore, while the Gaussian nature will also provide it with an opportunity to exploit at the very same time. Large values for c_1 and c_2 provide the particles with a chance to jump anywhere in the search space. The bigger the values are for these parameters, the farther the particles can travel away from their current location. These properties improve the MPSO algorithm with a better chance to jump out of local minima. Figure 2 also illustrates how a particle move in the search space. For instance, in Figure 2(b), a multiplication of a random vector with p_{vec} and n_{vec} will only effect the variance of the Gaussian distribution, while an addition of the velocity vector will make the velocity vector as a centre (mean) of the Gaussian distribution.

For the case when p_{vec}^i has different sign from n_{vec}^i in any dimension, we update the velocity by using Equation 3 with random vectors from a uniform distribution in the range $[0, 1]$. This part of the algorithm is responsible for the exploitation of the search space.

IV. APPLYING MPSO IN IMAGE REGISTRATION

We have opted to test our proposed algorithm on two applications, registration of (A) retinal images and (B) pre-modern coins.

A. Registration of retinal images

Ophthalmologists are often interested in determining the movement of blood vessels which is considered a major symptom of glaucoma [8], [9]. It is also of great importance to determine the development of new blood vessels or bleeding in the optic disc which is considered as key information in early diagnosis of glaucoma.

We used a data set of color (RGB) retinal images obtained by the Dept. of Neurosciences, Ophthalmology at Uppsala University (UU) Hospital. Over a period of two years the retinas of 44 patients were imaged regularly

(usually with four months interval) with a fundus camera to monitor the progress of glaucoma. We tested the MPSO on 100 pairs of these images. All images are of size 1024×1360 pixels. We use only the green channel of the RGB image since it provides the highest contrast between vessels and background [10]. We have a 6-dimensional search space represented as a vector (x, y, s, rx, ry, rz) , where x, y represent the horizontal and vertical translation, respectively, while s denotes the scale. rx, ry and rz represent the rotation along x -, y - and z -axis, respectively. The rotations are caused by the positioning of the camera, tilting of the head and through ocular torsion. The registration can be performed between either the full images or within sub-regions. The movement of vessels makes it illogical to perform registration between full images. Using a sub-region which is least effected by vessel movement will present a true picture of the vessel movement. The vessels inside the optic disc which lie close to the origin move with time and often get overexposed during imaging. As mentioned in [2], [9], the end of the vessel get detached from the surface of the retina due to loss of the nerve fibers, which leaves us to use the area around the border of the optic disc, exemplified in Figure 1. The ring around the optic disc was determined manually for all images by choosing a center point and a radius. To check the robustness of the MPSO, c_1 was set equal to c_2 and c_1 was allowed to vary from 2.05 to 30.05 in steps of 2. A total of 30 particles were initialized in the search space. The algorithm was run 30 times for each value of c_1 and c_2 on the pair of images shown in Figure 1. From the total of 450 test runs the MPSO algorithm was successful 441 times in correctly registering the images. Here, we have used a hard threshold (200 iterations) as a stopping criteria. Using the same setting, PSO was only successful 119 times in correctly registering the images. PSO with Gaussian distribution was successful 409 times with the same parameters.

The higher the value of c_1 and c_2 are set, the more difficult it gets for the PSO with uniform distribution to get out of local minima. The MPSO is not affected by

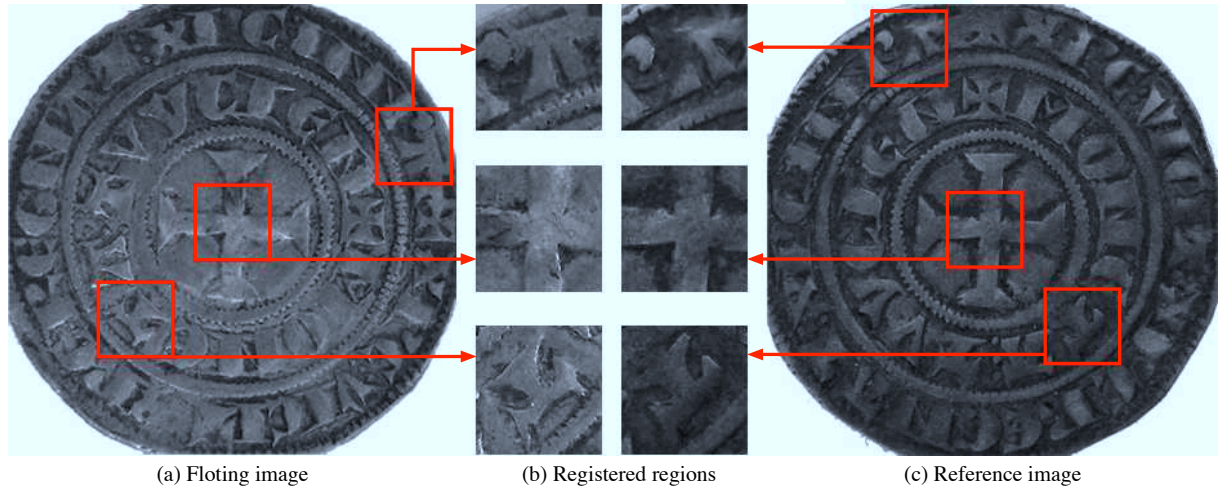


Fig. 3. The floating image (a) and the reference image (b) are from the same coin die, but different in wear and corrosion. Furthermore the floating image is rotated by 90° against the reference image. (c) shows selected regions after the registration process.

increasing the values of c_1 and c_2 . We also tested our algorithm to register 100 pairs of retinal images and it was successful in all 100 cases. The results presented in this manuscript are still preliminary but they clearly depict the robustness of the MPSO.

B. Registration of coin images

The study of the production technique of coins and medals is of central importance to Numismatics. Most coins were minted with two coin dies, which get worn out varyingly fast and are finally replaced by some new dies. By comparing different coins, numismatists try to find out, e.g., which dies were used to mint a specific coin. As a result, so-called die chains can be created, which show how individual dies were combined with each other. These chains are very important to establish a relative chronology for coins that do neither state the coinlord's name nor the year of issue of coins. Additionally, die chains can be very helpful to estimate the amount of coins minted on the one hand, and consequently give information about the former general economic situation on the other hand [11], [12].

All coins are captured with a reproduction camera stand, same camera distance and lightning condition at the UU Coin Cabinet. We have four degrees of freedom for this application. Figure 3 shows a typical example of coin images with deformations and corrosion. For the registration, we have used the same parameters as in the retina application. MPSO has shown promising results, but there are some instances where it fails to register the coins. Currently, our emphasis is at evaluating different objective functions as cross correlation provides a weak similarity measure.

V. CONCLUSION

The freedom provided by the MPSO to choose any value for c_1 and c_2 is appealing as it characterizes the explanatory nature of the algorithm. The introduction of Gaussian distribution along with uniform distribution

has provided the right balance between exploitation and exploration. For convergence, the MPSO takes a few more iterations than PSO. We estimate that it is rather straightforward to modify it to bare bones PSO [6], which will be future work.

ACKNOWLEDGMENT

We are grateful to Prof. Albert Alm at the Dept. of Neurosciences, Ophthalmology, UU Hospital and Dr. Hendrik Mäkeler at the UU Coin Cabinet for sharing their experiences on the topic and Olle Gällmo at the Dept. of Information Technology at UU for introducing us to PSO.

REFERENCES

- [1] B. Zitová et al. Image registration methods: a survey. *Image and Vision Computing*, 21:977–1000, 2003.
- [2] W. Hart et al. Registering retinal images using automatically selected control point pairs. In *IEEE Int. Conf. on Image Processing (ICIP-94)*, 3:576–580, 1994.
- [3] M. Clerc et al. The particle swarm - explosion, stability, and convergence in a multidimensional complex space. *IEEE Trans. Evolutionary Comput.*, 6(1):58–73, 2002.
- [4] R. Krohling. Gaussian swarm: a novel particle swarm optimization algorithm. In *IEEE Conf. on Cybernetics and Intelligent Systems*, 1:372–376, 2004.
- [5] R. Krohling et al. PSO-E: Particle swarm with exponential distribution. In *IEEE Congress on Evolutionary Comput. (CEC'06)*, pp. 1428–1433, 2006.
- [6] R. Krohling et al. Bare bones particle swarm optimization with Gaussian or Cauchy jumps. In *IEEE Cong. on Evolutionary Comput. (CEC'09)*, pp. 3285–3291, 2009.
- [7] F. van den Bergh et al. A study of particle swarm optimization particle trajectories. *Information Sciences*, 176(8):937–971, 2006.
- [8] N. Ritter et al. Registration of stereo and temporal images of the retina. *IEEE Trans. Med. Imaging*, 18(5):404–418, 1999.
- [9] V. Miszalak. *Noninvasive Diagnostic Techniques in Ophthalmology*, Ch. Fundus Imaging and Diagnostic Screening for Public Health. Springer-Verlag, 1990.
- [10] J. Staal et al. Ridge-based vessel segmentation in color images of the retina. *IEEE Trans. Med. Imaging*, 23(4):501–509, 2004.
- [11] H. Mäkeler. Bemerkungen zu den Anfängen der Stempelvergleichenden Methode. In *Geldgeschichtliche Nachrichten 40 (2005)* 223, pp. 128–132, 2005.
- [12] G. F. Carter. Comparison of Methods for Calculating the Total Number of Dies from Die-Link Statistics. In *Statistics and Numismatics*, pp. 204–213, 1981.