ESANN 2010 proceedings, European Symposium on Artificial Neural Networks - Computational Intelligence and Machine Learning. Bruges (Belgium), 28-30 April 2010, d-side publi., ISBN 2-930307-10-2.

Oriented Bounding Box Computation Using Particle Swarm Optimization*

Pierre B. Borckmans, P.-A. Absil

Department of Mathematical Engineering, Université catholique de Louvain, B-1348 Louvain-la-Neuve, Belgium (http://www.inma.ucl.ac.be/~absil/)

Abstract. The problem of finding the optimal oriented bounding box (OBB) for a given set of points in $\mathbb{R}^3$, yet simple to state, is computationally challenging. Existing state-of-the-art methods dealing with this problem are either exact but slow, or fast but very approximative and unreliable. We propose a method based on Particle Swarm Optimization (PSO) to approximate solutions both effectively and accurately. The original PSO algorithm is modified so as to search for optimal solutions over the rotation group $SO(3)$. Particles are defined as 3D rotation matrices and operations are expressed over $SO(3)$ using matrix products, exponentials and logarithms. The symmetry of the problem is also exploited. Numerical experiments show that the proposed algorithm outperforms existing methods, often by far.

1 Introduction

The Oriented Bounding Box problem (OBB) can be stated as follows: given a set of n points in $\mathbb{R}^3$, find the minimal-volume oriented parallelepiped enclosing all the points. This question arises in many practical applications and notably in vision problems. In collision detection for example, intersections are preferably checked using bounding volumes, such as boxes, spheres or ellipsoids, since it is computationally more efficient than with complex 3D shapes (convex hull, . . .). Oriented bounding boxes are a common choice because of the simplicity of the intersection test.

The best existing methods for solving the OBB problem can be sorted into two categories: exact methods and approximations. In the first category, the state-of-the-art method was proposed by O'Rourke [1]. Based upon the 2D rotating calipers technique, this 3D adaptation is exact but hard to implement and it runs with a complexity of $\mathcal{O}(n^3)$, which is quite inefficient for large data sets.

In the second category, some heuristics have been proposed using either principal component analysis (PCA) or brute-force approaches. PCA based methods can be implemented easily and produce an approximation very quickly. However, their precision is very sensitive to the data point distribution and can result in

*This paper presents research results of the Belgian Network DYSCO (Dynamical Systems, Control, and Optimization, funded by the Interuniversity Attraction Poles Programme, initiated by the Belgian State, Science Policy Office). The scientific responsibility rests with its authors. This work is supported by a FRIA (Fonds pour la formation à la Recherche dans l'Industrie et dans l'Agriculture) fellowship and by "Communauté française de Belgique - Actions de Recherche Concertées".

far from optimal volumes [2]. Brute-force methods are usually based on a sampling of the search space; they are usually inefficient because they do not exploit the structure of the problem. For a more detailed state-of-the-art review, the reader can refer to [3].

The idea presented in this article is to exploit a reformulation of the OBB problem as an optimization over the rotation group

$$SO(3) = \{R \in \mathbb{R}^{3\times3} : R^T R = I, \det(R) = 1\}.$$

Finding the minimal volume oriented bounding box can be expressed as follows:

$$\min_{R\in SO(3)} f(R) = \mathrm{V}_{\mathrm{AABB}}(RX) = (x'_{max} - x'_{min})(y'_{max} - y'_{min})(z'_{max} - z'_{min})$$

where $R \in SO(3)$ is the rotation matrix, $X \in \mathbb{R}^{3\times n}$ denotes the set of points and $\mathrm{V}_{\mathrm{AABB}}$ is the volume of the so-called axis-aligned bounding box (AABB). For given R and X, the volume of the AABB is simply obtained by rotating the set of points X by R: $X' = RX = (x', y', z')$ and computing the product of the span along each rotated direction. As can be observed with a 2D example in figure 1, the function $f(R)$ is only C^0 and presents multiple minima. The non-differentiable and multimodal aspects of $f(R)$ make it a good candidate for derivative-free optimization methods, especially global methods [4].

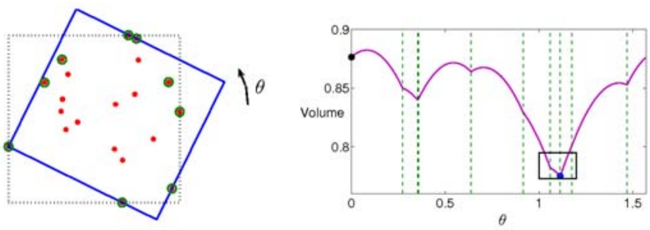

Fig. 1: 2D Oriented bounding box: the volume f(R) is not differentiable and presents multiple minima.

In this paper, we propose to adapt Particle Swarm Optimization (PSO) to this new formulation of the OBB problem. Indeed, PSO seems to be a good solver candidate since it is clearly suited for non-differentiable, multimodal problems and because it exhibits intrinsic qualities to find a good trade-off between exploration and exploitation of the search domain. In order to achieve this adaptation, the PSO algorithm has to be modified to fit the $SO(3)$ search space. Some properties of this manifold are used to keep the proposed method as close as possible to the original algorithm.

The next sections are organized as follows: section 2 presents the PSO algorithm basics and its adaptation to the special orthogonal group $SO(3)$, section 3

ESANN 2010 proceedings, European Symposium on Artificial Neural Networks - Computational Intelligence and Machine Learning. Bruges (Belgium), 28-30 April 2010, d-side publi., ISBN 2-930307-10-2.

presents some numerical results showing that the proposed method is faster and/or more accurate than existing methods, section 4 discusses possible enhancements of the method and other ideas that could be further investigated.

2 Adapting PSO to $SO(3)$

PSO is a stochastic population-based algorithm. Particles are points evolving in the search space, following simple rules, mimicking the behaviour of social groups. Points are initialized randomly in the search space and the driving force of the optimization process is given by the following update equations (iterated over k), for each particle (indexed by i):

$$\begin{cases} v_i(k+1) &= \underbrace{w(k)v_i(k)}_{inertia} + \underbrace{c\alpha_i(k)\left(y_i(k) - x_i(k)\right)}_{nostalgia} + \underbrace{s\beta_i(k)\left(\hat{y}(k) - x_i(k)\right)}_{social} \\ x_i(k+1) &= x_i(k) + v_i(k+1), \end{cases}$$

where x denotes position, v denotes velocity, y is the personal best position so far, $\hat{y}$ is the global best position of the swarm so far ; w is inertia coefficient (usually dynamic), c and s are adjustable coefficients, and α and β are random components. As can be seen, the behaviour of each particle is dictated by velocity increments composed of three simple components: inertia, cognition (nostalgic behaviour) and social behaviour. For more in-depth information about PSO, the reader is referred to [5].

In its original form, PSO is described for particles distributed in $\mathbb{R}^n$ so that $x \in \mathbb{R}^n$, $y \in \mathbb{R}^n$ and that the operations involved in the update equations (+,−,.) are the usual vectorial addition, difference and scaling. In order to adapt standard PSO to the $SO(3)$ search space, one must redefine x, v and the operations mentioned above. Since we are looking for the optimal rotation matrix, x must be an element of $SO(3)$ and so do y and $\hat{y}$. The velocity v is now an element of the tangent space $T_xSO(3)$ to $SO(3)$ centered at x. Since $SO(3)$ is a Lie group (see, *e.g.*, [6]), we have $v = x\tilde{v}$ for some $\tilde{v} \in \mathfrak{so}(3)$, where $\mathfrak{so}(3) = T_ISO(3)$ is the set of all skew-symmetric 3×3 matrices. Noting that the geodesic on $SO(3)$ with initial position $x \in SO(3)$ and initial velocity $x\tilde{v}$ is given by $x\exp(t\tilde{v})$ (see [6, §VII.8]), the position update in the second equation can be rewritten using matrix composition:

$$x_i(k+1) = x_i(k)\exp(\tilde{v}_i(k+1)).$$

Moreover, given x_1 and x_2 in $SO(3)$, the initial velocity $v = x_1\tilde{v}$ of the geodesic starting from x_1 that goes through x_2 at $t = 1$ is given by $\tilde{v} = \log(x_1^T x_2)$. Exploiting this expression for the nostalgia and social components of the velocity equation, one can write:

$$\tilde{v}_i(k+1) = w(k)\tilde{v}_i(k) + c\alpha_i(k)\log\left(x_i(k)^T y_i(k)\right) + s\beta_i(k)\log\left(x_i(k)^T\hat{y}(k)\right).$$

Note that the key idea behind PSO is preserved with this adaptation: the new rotation is built using a stochastically weighted combination of the old rotation (with inertia) and attraction towards both personal and global best rotations.

As figure 2 shows, the difference with classical PSO is that this combination now occurs in the tangent space centered at x_k, and is then retracted to the $SO(3)$ manifold.

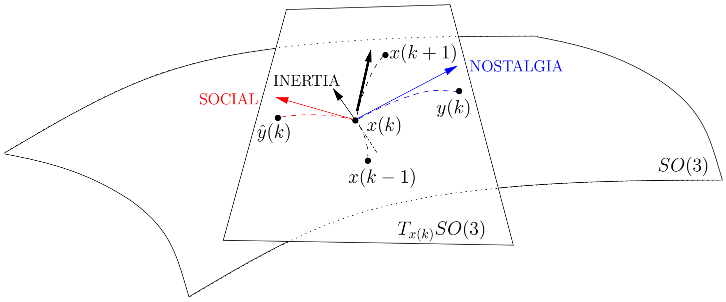

Fig. 2: Computation of the position update in the tangent space of $SO(3)$ centered at $x(k)$. The new iterate $x(k+1)$ is constructed using a combination of inertia, nostalgia and social contributions in $T_{x(k)}SO(3)$ and is then retracted to $SO(3)$.

Now that the update equations have been modified, the last step needed in order for PSO to be adapted to the $SO(3)$ manifold is to take the symmetry of the OBB problem into account. Considering the 2-dimensional problem first, a rotation matrix belongs to $SO(2)$ and can therefore be represented with a unique angle θ. Given such an orientation of a bounding rectangle, the corresponding area will clearly be the same for every further rotation of 90 degrees. This indicates that the search space should be truncated to the first quadrant only (matrices with orientation in $[0, \frac{\pi}{2}[$), using the relation: $\theta' = \theta \mod (\frac{\pi}{2})$. Furthermore, when computing the angle between two rotation matrices R_1 and R_2, represented by angles θ_1 and θ_2, the shortest distance in the light of this periodicity should be exploited, using the following computation, so that the measured angle always lies in $]\frac{-\pi}{4}, \frac{\pi}{4}]$:

$$\Theta(R1, R2) = \begin{cases} \theta_2 - \theta_1, & if\ |\theta_2 - \theta_1| \leq \frac{\pi}{4} \\ \theta_2 - \theta_1 - \frac{\pi}{2}\mathrm{sgn}(\theta_2 - \theta_1), & if\ |\theta_2 - \theta_1| > \frac{\pi}{4} \end{cases}$$

These properties can be regarded another way. The rows r_1 and r_2 of a matrix R in $SO(2)$ form a basis in $\mathbb{R}^2$. Considering all the 2D signed permutation matrices P in $SO(2)$ (composed of columns $p_i \in \{\pm e_1, \pm e_2\}$), $R' = PR$ yields 4 different bases, each time with r'_1 and r'_2 lying in two consecutive quadrants. Since the log mapping $\log(R)$ measures the "displacement" between the identity and R, choosing the permutation P that brings r'_1 as close as possible to e_1 and r'_2 to e_2

ESANN 2010 proceedings, European Symposium on Artificial Neural Networks - Computational Intelligence and Machine Learning. Bruges (Belgium), 28-30 April 2010, d-side publi., ISBN 2-930307-10-2.

ensures that the smallest displacement D linking two matrices R_1 and R_2 can be computed using $D = \log(PR_1^T R_2)$.

This can now be extended to the 3-dimensional case. The rows r_1, r_2 and r_3 of R now form a basis of $\mathbb{R}^3$, and with the set of all 3D signed permutations P in $SO(3)$, $R' = PR$ yields 24 bases. The permutation can now be chosen so that r_1' is as close as possible to e_1, r_2' to e_2 and r_3' to e_3, and the relation $D = \log(PR_1^T R_2)$ still gives the smallest displacement linking R_1 and R_2.

Taking the periodicity and symmetry into account, the update equations can finally be written as:

$$\begin{cases} \tilde{v}_i(k+1) &= w(k)\tilde{v}_i(k) + c\alpha_i(k)\log\left(P_1 x_i(k)^T y_i(k)\right) \\ & \qquad + s\beta_i(k)\log\left(P_2 x_i(k)^T \tilde{y}(k)\right) \\ x_i(k+1) &= P_3 x_i(k)\exp(\tilde{v}_i(k+1)), \end{cases}$$

where P_1, P_2 and P_3 are permutation matrices chosen as described above.

3 Results

As a preliminary experiment, the proposed adaptation of PSO was implemented using Matlab. Some data sets (available at `www.inma.ucl.ac.be/~borckmans`) were chosen presenting various sizes (number of nodes on the convex hull) and distributions. Table 3 presents the results of this simulation for three of them and compares the performance of the proposed solution to two state-of-the-art methods: All-PCA and O'Rourke. The optimal volume is the one obtained by O'Rourke since it is an exact method. The relative error is measured for PCA and PSO. Since PSO presents a stochastic component, 100 runs were performed, stopping after 100 iterations or when no improvement was made for 10 consecutive iterations; the best and worst results are presented, as well as the mean and variance of the relative error. The parameters were chosen as follows:

$$size = 20,\ c = 0.5,\ s = 0.5,\ w(k) \text{ decreasing from } 0.9 \text{ (k = 1) to } 0.4 \text{ (k = 100)}$$

Table 3 shows that the proposed algorithm runs faster than the exhaustive O'Rourke method, without compromising the quality of the solutions. PCA, on the other hand, runs extremely fast but only produces rough approximations. The experiment showed that PSO sometimes leads to situations where the particles prematurely converge to a local minimum. This explains the difference between the best and worst results and indicates that the proposed method needs improvements regarding its robustness.

4 Conclusion

This preliminary experiment indicates that, while being a stochastic method, PSO shows to be both efficient and relatively reliable to solve the OBB problem. Furthermore, this application is encouraging the development of PSO over different search spaces and manifolds.

data set	method	time (s)	relative error (%)			
set1	O'Rourke	$29 \cdot 10^0$	0			
132 nodes	PCA	$20 \cdot 10^{-5}$	28.04			
	PSO	$55 \cdot 10^{-1}$	min:	$21 \cdot 10^{-14}$	max:	$10 \cdot 10^{-2}$
			mean:	$17 \cdot 10^{-2}$	var:	$33 \cdot 10^{-3}$
set2	O'Rourke	$14 \cdot 10^3$	0			
6479 nodes	PCA	$38 \cdot 10^{-2}$	114.9			
	PSO	$87 \cdot 10^{-1}$	min:	$15 \cdot 10^{-12}$	max:	$56 \cdot 10^{-2}$
			mean:	$17 \cdot 10^{-2}$	var:	$46 \cdot 10^{-3}$
set3	O'Rourke	$22 \cdot 10^2$	0			
1560 nodes	PCA	$49 \cdot 10^{-4}$	83.8			
	PSO	$67 \cdot 10^{-1}$	min:	$15 \cdot 10^{-12}$	max:	$56 \cdot 10^{-2}$
			mean:	$17 \cdot 10^{-2}$	var:	$46 \cdot 10^{-3}$

Fig. 3: Comparison of the performance of the proposed PSO algorithm with PCA and O'Rourke.

Some extensions of the proposed algorithm can be considered for further work. Among the many variations of the PSO algorithm introduced since 1995, strategies dealing more efficiently with multimodal objectives could be valuable. Approaches involving self-adaptive coefficients may also add robustness to the proposed method. Finally, a hybridization of PSO with a directional search method (compass search, MADS, ...) could be envisaged as proposed in [7].

Acknowledgement

The results presented in this paper benefited from many useful discussions with Samuel Melchior, Bastien Gorissen and Chia-Tche Chang.

References

[1] Joseph O'Rourke. Finding minimal enclosing boxes. *Internat. J. Comput. Inform. Sci.*, 14:183–199, June 1985.

[2] Darko Dimitrov, Christian Knauer, Klaus Kriegel, and Gunter Rote. Bounds on the quality of the pca bounding boxes. *Computational Geometry*, 42(8):772 – 789, 2009. Special Issue on the 23rd European Workshop on Computational Geometry.

[3] C.-T. Chang, B. Gorissen, and S. Melchior. Fast computation of the minimal oriented bounding box on the rotation group $SO(3)$. Submitted to ACM Transaction on Graphics, 2009.

[4] Andrew R. Conn, Katya Scheinberg, and Luis N. Vicente. *Introduction to Derivative-Free Optimization*. Society for Industrial and Applied Mathematics, Philadelphia, PA, USA, 2009.

[5] Andries P. Engelbrecht. *Fundamentals of Computational Swarm Intelligence*. John Wiley & Sons, 2006.

[6] William M. Boothby. *An Introduction to Differentiable Manifolds and Riemannian Geometry*. Academic Press, 2003. Revised Second Edition.

[7] A. I. F. Vaz and L. N. Vicente. Pswarm: a hybrid solver for linearly constrained global derivative-free optimization. *Optimization Methods Software*, 24(4-5):669–685, 2009.