Iterative Rotations and Assignments (IRA): a shape matching algorithm for atomic structures

Miha Gunde, Nicolas Salles, Anne Hémeryck, Layla Martin-Samos

Iterative Rotations and Assignments (IRA): a shape matching algorithm for atomic structures

Miha Gunde[a,b], Nicolas Salles[b], Anne Hémeryck[a], Layla Martin-Samos[b]

[a] *LAAS-CNRS, Université de Toulouse, CNRS, 7 avenue du Colonel Roche, 31031 Toulouse, France*
[b] *CNR-IOM, Democritos National Simulation Center, Istituto Officina dei Materiali, c/o SISSA, via Bonomea 265, IT-34136 Trieste, Italy*

Abstract

IRA is a Fortran library that solves the shape matching problem for atomic structures, stored as sets of points representing the atomic positions. In the case of exact- and near-congruence, IRA provides the optimal rigid transformation between the structures, given by the atomic assignments, the rotation/reflection matrix, and the translation vector. IRA is also able to operate on structures containing a non-equal number of atoms, i.e. matching a structure to any of its fragments. Any application that requires the solution of a shape matching problem could benefit from IRA.

Keywords: shape matching, structural superposition, linear assignment problem, pattern recognition, off-lattice kinetic Monte Carlo

Code metadata

Nr.	Code metadata description	
C1	Current code version	*v1.0.0*
C2	Permanent link to code/repository used for this code version	*https://github.com/mammasmias/IterativeRotationsAssignments/tree/master/IRA*
C3	Permanent link to Reproducible Capsule	*https://codeocean.com/capsule/1161768/tree/v1*
C4	Legal Code License	*Double license: GNU General Public License v3.0, and Apache 2.0.*
C5	Code versioning system used	*git*
C6	Software code languages, tools, and services used	*Fortran*
C7	Compilation requirements, operating environments & dependencies	*Fortran compiler, lapack library*
C8	Link to developer documentation/manual	*https://github.com/mammasmias/IterativeRotationsAssignments/blob/master/IRA/README.md*
C9	Support email for questions	*miha.gunde@gmail.com*

Introduction

The shape matching problem is very well known in the computer vision community, where it is generally referred to as the point-set registry problem [1, 2]. When applied to atomic structures, it is sometimes referred to as the alignment problem [3, 4], structural superposition [5, 6], or similar. The algorithm Iterative Rotations and Assignments (IRA) has been developed [7, 8] to solve the shape matching problem, specifically for application in the field of materials modeling at atomic scale. The problem can be described as finding a rigid transformation between two structures, such that the two structures match as best as possible when overlaid one on the other.

*Corresponding author: Miha Gunde
E-mail addresses: miha.gunde@gmail.com, nsalles33@gmail.com, anne.hemeryck@laas.fr, marsamos@iom.cnr.it

Our IRA algorithm accurately and efficiently resolves the shape matching problem between two generic atomic structures, including structures with different numbers of atoms. The solution is given in the form of a rotation matrix, a translation vector, and a permutation matrix. Performance of the algorithm in the sense of matching success rate, and benchmarking against two other software, is reported in Refs. [7, 8].

The development repository of the code is accessible at https://github.com/mammasmias/IterativeRotationsAssignments.

Problem statement

In general terms, the shape matching problem can be stated as follows. Two structures A and B are considered congruent, or equivalent, when they are related by a transformation that preserves distances, such as rigid rotation, reflection, translation, and permutation of indistinguishable points (atoms). The congruence relation can be written as:

$$P_B B = \mathbf{R}A + \mathbf{t} \tag{1}$$

where $\mathbf{R}$ is a rotation/reflection matrix, $\mathbf{t}$ is a translation vector, and P_B is permutation. The problem is to find the transformations corresponding to $\mathbf{R}$, $\mathbf{t}$, and P_B that best match A to B. This can be achieved by optimizing a distance D between the two structures, which is variant on $\mathbf{R}$ and $\mathbf{t}$, and invariant on P_B:

$$\arg\min_{\mathbf{R},\mathbf{t}} \Big(D(\mathbf{R}A + \mathbf{t}, B) \Big). \tag{2}$$

The approach we have developed (see Refs. [7, 8]) returns the approximate rotation/reflection and translation, $\mathbf{R}_{apx}$ and $\mathbf{t}_{apx}$, and the optimal permutation P_B (which is strictly from Iterative Rotations and Assignments, IRA). The optimal transformations $\mathbf{R}$ and $\mathbf{t}$ which satisfy Eq. (1) are then found as small corrections to $\mathbf{R}_{apx}$ and $\mathbf{t}_{apx}$, by applying the standard Singular Value Decomposition (SVD) [9] to structures A' and B' (where $A' = \mathbf{R}_{apx}A + \mathbf{t}_{apx}$, and $B' = P_B B$.). This correction is only required when the structures A and B are not exactly congruent, but exhibit some small distortions.

To compute the optimal permutations, i.e. to solve the Linear Assignment Problem (LAP), we have developed an algorithm called Constrained Shortest Distance Assignments (CShDA). The atomic assignment from CShDA is used to compute the Hausdorff distance as distance function D in Eq. (2). By imposing the one-to-one assignment constraint of CShDA, we exploit the non-commutativity of the Hausdorff distance to generalize the IRA algorithm to structures containing different number of atoms. Further details can be found in Refs. [7, 8].

Software design and functionality

The software implementing our IRA algorithm is written as a collection of Fortran routines, gathered and compiled into a library. As such, an application calling IRA routines can be compiled by simply statically linking the library. The input atomic structures are saved quantities with `intent (in)` attribute, and are therefore never modified globally by IRA routines.

The low-level routine is the `cshda` routine, which computes the atomic assignments P_B, and constrains the computation of Hausdorff distance. It is called from a higher-level routine `get_gamma_m`, which implements the search on the discretized space of rotations, and returns the approximate rotation $\mathbf{R}_{apx}$ and translation $\mathbf{t}_{apx}$. The setup and the main call to this routine are encapsulated in the driver routines `ira_equal`, and `ira_nonequal`, for the cases of equal, and non-equal numbers of atoms, respectively. The input arguments to the driver routines are the two atomic structures A and B, and their output are the following quantities: the 3×3 rotation matrix $\mathbf{R}_{apx}$, the 3-dimensional translation vector $\mathbf{t}_{apx}$, and the permutation P_B in the form of an N-dimensional array of integers.

The SVD rotation is in `svdrot_m` routine, and returns the optimal $\mathbf{R}$ and $\mathbf{t}$, corresponding to Eq. (1).

The difference in the cases when the structures contain equal, or different numbers of atoms, lies in the choice of the origin point, or central atom. In the case of equal numbers of atoms, the origin point is chosen by default as the geometrical center of each structure. This choice can be modified by selecting

any point common to both structures, such as any known common atom. For a custom choice, the driver routine `ira_equal` must be modified. In the case of a different numbers of atoms, the origin point is an arbitrarily chosen central atom from the structure A. The search over structure B is then performed over all candidate origin points, or central atoms. Similarly to the case of an equal number of atoms, the origin point of structure A can be modified in the driver routine `ira_nonequal`. As discussed in Refs. [7, 8], the speed and accuracy of the algorithm can in this case be greatly improved by reducing the set of candidate central atoms in structure B, which can be done by choosing the central atom of A based on some additional criterion. This additional criterion is the subject of a pre-processing done on the structures, which can be simply based on the chemical types (choosing an atom of the type which is least present in the structure), or can be more involved (finding atoms whose immediate local environment is equivalent).

Impact overview

The IRA algorithm provides a reliable and efficient way of solving the shape matching problem, for congruent, and near-congruent structures, including structures with different numbers of atoms. Any application that requires the solution of a shape matching problem could benefit from IRA as a whole, or any of its parts.

Limitations

The implementation of the library is for now serial. As such, the matching of structures with several hundreds of atoms could become a significant computational bottleneck.

When the structures are far from congruence, the transformation and distance that are returned from our approach are not unambiguous. It is therefore not recommended to use IRA for defining "absolute" metrics between structures.

Future applications

There are a number of applications of IRA within our ideas of the future work. The most immediate application is already being developed and implemented (see Ref. [8]), which is the application of IRA in a general off-lattice kinetic Monte-Carlo (kMC) and in the generation of a catalogue of unique structures.

In particular, the kMC algorithm generates a stochastic but statistically correct sequence of events, where each event has a certain probability. As such, a single step of the kMC consists of choosing a single event among a set of possible events. When applying the kMC algorithm to a simulation of phenomena at the atomic scale, an event represents a change in the atomic structure, and the set of possible events is given by a catalogue of events. Each event in the catalogue is defined by two structures, the structure before, and the structure after the event. The system of simulation is expected to contain structures that are locally equivalent (exactly-, or near-congruent) to the structures given in the catalogue of events. However their relative orientation, translation, and permutation of the atoms are generally unknown, but are needed for a correct execution of a chosen event. This is the problem where our IRA shape matching algorithm comes into action. As it has been shown to be a fully reliable algorithm in finding the correct rigid transformations in the cases of exactly- and near-congruent structures [7, 8], it is an ideal algorithm for the described situation within a kMC. The solution of shape matching given by IRA in this situation is used to construct a relative transformation between the structure given by the event catalogue and the structure within the system of simulation. This transformation makes it possible to correctly execute the event chosen by the kMC algorithm in our system of simulation. Moreover, as IRA does not make any assumptions about the structures, i.e. it does not assume an underlying rigid lattice, and it does not even assume an equal number of atoms present in the two structures, the kMC simulation becomes unlimited by these assumptions, and can be generalized to any type of structure: bulk, surface, interface, grain boundary, single defects, complex

defects, extended and non-spherical structures, and amorphous phases.

Moreover, the catalogue of events on which kMC relies, is generated with the help of IRA. More concretely, a separate application would scan the input structures of an event, and extract the most important and relevant atoms participating in an event. Then, IRA is used to determine whether an event with equivalent structures is already present in the catalogue, if not, then add current event into the catalogue. At the same time, a modified version of IRA can be used to determine the possible different directions of execution of the same event, which means finding all the unique symmetry transformations of a local structure (see Ref. [8]). This approach reduces the number of structures needed in an event catalogue, as the different directions of the same event are stored in a single event entry, as opposed to each direction being a separate entry.

Acknowledgements

The authors are active members of the Multiscale And Multi-Model Approach for MaterialS In Applied Science consortium (MAMMASMIAS consortium), and acknowledge the efforts of the consortium in fostering scientific collaboration. This work was partially funded from the European Union's Horizon 2020 research and innovation program under grant agreement No. 871813 MUNDFAB, and by the European Union's Horizon 2020 research and innovation program under grant agreement No 899285 MAGNELIQ.

References

[1] Maiseli B., Gu Y., and Gao H., *Journal of Visual Communication and Image Representation*, 46, 95–106, 2017. https://www.sciencedirect.com/science/article/pii/S1047320317300743

[2] Besl P. and McKay N. D., *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 14, 2, 239–256, 1992. https://doi.org/10.1109/34.121791

[3] Hanson A. J., *Acta Crystallographica Section A*, 76, 4, 432–457, Jul 2020. https://doi.org/10.1107/S2053273320002648

[4] Griffiths M., Niblett S. P., and Wales D. J., *Journal of Chemical Theory and Computation*, 13, 10, 4914–4931, 2017. https://doi.org/10.1021/acs.jctc.7b00543

[5] Kneller G. R., *The Journal of Chemical Physics*, 128, 19, 194101, 2008. https://doi.org/10.1063/1.2902290

[6] Helmich B. and Sierka M., *Journal of Computational Chemistry*, 33, 2, 134–140, 2012. https://onlinelibrary.wiley.com/doi/abs/10.1002/jcc.21925

[7] Gunde M., Salles N., Hémeryck A., and Martin-Samos L., *Journal of Chemical Information and Modeling*, 61, 11, 5446–5457, 2021. https://doi.org/10.1021/acs.jcim.1c00567

[8] Gunde M., "Development of IRA: a shape matching algorithm, its implementation, and utility in a general off-lattice kMC kernel," Ph.D. dissertation, Nov 2021. http://thesesups.ups-tlse.fr/5109

[9] Arun K. S., Huang T. S., and Blostein S. D., *IEEE Transactions on Pattern Analysis and Machine Intelligence*, PAMI-9, 5, 698–700, 1987. https://doi.org/10.1109/TPAMI.1987.4767965