# Shape tracing: An extension of sphere tracing for 3D non-convex collision in protein docking

Adam Leach[‡], Lucas S.P. Rudden[†], Sam Bond-Taylor*,
John C. Brigham[‡], Matteo T. Degiacomi[†], Chris G. Willcocks*
Department of {Computer Science*, Physics[†], Engineering[‡]}, Durham University
{adam.leach, l.s.rudden, samuel.e.bond-taylor, matteo.t.degiacomi,
christopher.g.willcocks}@durham.ac.uk

***Abstract*—This paper presents an algorithm, similar to implicit sphere tracing, that ray marches 3D non-convex shapes for efficient collision detection. Instead of finding points on the surface where individual rays strike, an entire shape is marched in unison by a lower bound of the boundary distance, calculated at the closest point between the two surfaces. Advancing one shape towards the other by this new bound allows us to identify a contact in few steps. This method supports arbitrary non-convex shapes, and can be run in parallel. We apply this to protein-protein docking and show that we can identify around 80 docking poses per second featuring contact but no overlap, irrespective of proteins' specific geometry. This paves the way to future fast docking algorithms, building upon implicit surface representations to quickly find a well-distributed subset of close candidate solutions for further investigation.**

## I. Introduction

Proteins are biopolymers directly responsible for the vast majority of essential cellular functions. Any organism, from a single bacterium to the human body, contains millions of proteins with roles as diverse as sensing, transportation, catalysis, defence or structural support. These biological tasks are often carried out via the formation of specific complexes of minimal energy. One of the hardest challenges in computational structural biology is predicting how individual proteins with a known atomic structure arrange into such assemblies.

Despite the advent of increasingly sophisticated methods over the last 20 years [1], the problem is far from solved. A yearly community-led assessment of current docking algorithms, CAPRI, reveals that at present, no algorithm exists that is capable of consistently yielding accurate results [2]. Existing approaches to this problem can be generally divided into two categories. In the first, proteins are represented explicitly as a collection of atoms. The objective here is to identify a protein arrangement that minimizes a scoring function composed of a sum of physical terms such as electrostatics, van der Waals, desolvation energy and other empirical quantities. In the second, proteins are represented as a geometrical shape derived from the known atomic structure, and the associated scoring function typically maximizes shape complementarity. Independently from the chosen representation and scoring function, protein docking is a hard optimization problem. Indeed, protein-protein interactions feature a multitude of potential binding sites associated with a local energy minimum. The exploration of this complex search space, with high Lipschitz constants, is traditionally tackled either by brute force [3], or via derivative-free optimization algorithms such as Particle Swarm Optimization [4] or Monte Carlo [5]. Many candidate solutions generated during the docking process will feature either intersecting or contactless protein pairs, and optimization will often converge to local minima.

We present a method for the rapid exploration of the search space associated with the matching of two three-dimensional surfaces of arbitrary roughness and demonstrate its usage for protein docking. Instead of taking the traditional route of explicitly representing a protein surface [6; 7], we represent the receptor surface (i.e. the largest protein) implicitly, allowing for easy intersection and distance query. The ligand (i.e. the smaller protein) can then be marched by the lower bound of the boundary distance, as with traditional sphere tracing, but with a modification to the bound that allows tracing of arbitrary non-convex shapes. Our method features two key contributions. First, it leverages on a novel extension of sphere tracing [8], derived without heuristic, for detecting collisions between approaching non-convex shapes. Second, it adopts an implicit approach for finding where surface contact area is maximized, shown to be effective in a foundational outer-loop Monte Carlo method.

## II. Related work

Protein docking algorithms can be broadly classified according to their sampling strategies [9]. These are Fast-Fourier Transform (FFT) grid-based searches [3; 10; 11; 12], Monte Carlo (MC) [5; 13; 14], Genetic Algorithms [15; 16] and Particle Swarm Optimisation (PSO) [17; 6; 18]. An additional strategy, Geometric Hashing [19; 20], can only be adopted in conjunction with protein shape representations. Since proteins are inherently flexible molecules, conformational changes, from interfacial side-chain repacking to large-scale domain level rearrangements, should be accounted for by these algorithms. The majority of approaches use atomistic representations, which require computationally expensive additional minimization steps to promote side chains packing upon binding. Such considerations are needed as minor alterations in atomic positions may have profound consequences on the score of a pose. This step can be bypassed by modelling the protein as a shape, accounting for the uncertainty of side chains positions from the outset [21]. A method developed by Rudden

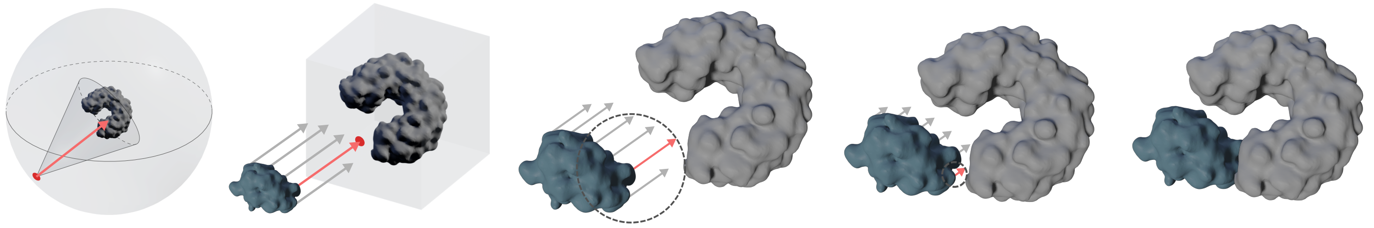

Fig. 1. Docking with shape tracing: the source shape (ligand) is initialised on random points on a sphere around the target shape (receptor) with inward facing cones (1). The source shape is then analytically moved to be just inside the target's bounding box (2). The shape tracing algorithm iteratively (3-5) samples the target's signed distance function $\phi$ at surface positions. The shape is marched by the bound from the closest point (dashed circle, the minimum of this sample).

and Degiacomi proposes a molecular surface representation, 'Spatial and Temporal Influence Density' (STID) [6], which maps points in space to a surface probability $\phi : \mathbb{R}^n \to \mathbb{R}$ based on a Molecular Dynamics simulation. The map and subsequent surface complementarity scoring function [6] led to a success rate of 56% across a benchmark of 224 proteins, which is competitive with the current best atomic based methods [5; 14], and contrastingly performed remarkably well with flexible proteins. However, the time required to complete a full docking run using PSO was significant, often $> 1$ day for larger complexes. Aside from computational time, there are still significant desirable improvements for protein docking algorithms. These include a scoring function which is exact and does not rely on heuristics, and an optimizer which can rapidly and reliably find correct solutions.

## III. Methodology

The task of identifying the best arrangement of two non-convex protein shapes involves: (1) A shape tracing algorithm which efficiently marches the shape through space converging in a few steps and (2) A high-level optimizer to find solutions where contact between two proteins' surface area is maximised.

### *Shape tracing algorithm*

The shape tracing algorithm updates points $\mathbf{x} \in \mathbb{R}^{n\times 3}$ on the boundary of the source shape (the ligand), moving them in a specified direction $\vec{v}$ until they collide:

$$S(\mathbf{x}, \phi, b, \vec{v}) = \mathbf{x}', \tag{1}$$

where $\phi$ is an input signed distance function (SDF) defined on a volumetric grid for the target shape (the receptor), and $b \in \mathbb{N}^3$ is the side length (in voxels) of each axis of this volume. The SDF $\phi$ can be calculated efficiently from the STID map [6] with the fast marching method as an approximate solution to the Eikonal equation. Any values sampled outside of the bounds of the SDF volume $\phi$ are set to $\infty$. The shape tracing method is outlined in Algorithm 1:

1) Compute the analytical intersections from rays cast on the source shape (ligand) at points $\mathbf{x}$ in direction $\vec{v}$ to the target (receptor) bounding box (lines 1-2) as in [22].
2) If any rays hit (line 4), advance all points $\mathbf{x}'$ by the closest distance (line 5). For glancing rays, push $\mathbf{x}'$ just inside box by the sign of $\vec{v}$ (line 6) as in [23].
3) Now we know one of the points in $\mathbf{x}'$ is inside the bounds of $\phi$, find the closest distance to the receptor (line 7).
4) While the shapes are not touching $\delta > \epsilon$ and while the shape is still inside the bounds of $\phi$ (line 8), keep moving the whole shape $\mathbf{x}'$ by the closest distance (lines 9-10).

**Algorithm 1:** Shape Tracing

```
   Input: $\mathbf{x}, \phi, b, \vec{v}$
   Output: $\mathbf{x}'$
 1 $t_{\text{nears}} = \max\big(\min\big((1/\vec{v}) \cdot (-\mathbf{x}), (1/\vec{v}) \cdot (b - \mathbf{x})\big)\big)$
 2 $t_{\text{fars}} = \min\big(\max\big((1/\vec{v}) \cdot (-\mathbf{x}), (1/\vec{v}) \cdot (b - \mathbf{x})\big)\big)$
 3 intersects = $\{t_{\text{nears}} > t_{\text{fars}}\}$
 4 if intersects $\neq \{\}$ then
 5     $\delta = \min(t_{\text{nears}}[\text{intersects}])$
 6     $\mathbf{x}' = \mathbf{x} + \delta\vec{v} + \text{sign}(\vec{v})$
 7     $\delta = \min(\phi(\mathbf{x}'))$
 8     while $\delta > \epsilon$ and $\delta \neq \infty$ do
 9         $\mathbf{x}' = \mathbf{x}' + (\delta/2)\vec{v}$
10         $\delta = \min(\phi(\mathbf{x}'))$
11     end
12 end
```

The value for $\epsilon$, 0 by default, can be increased for faster convergence if certain tolerances are acceptable, such as within 1 Å in docking. The $\delta/2$ in line 9 reduces the step (by a value proportional to the maximum derivative of $\phi$), a common strategy in ray marching. While in theory we do not need to reduce this step as $|\nabla\phi| = 1$, in practice $|\nabla\phi| \approx 1$ due to the discretization of $\phi$.

### *Outer-loop Monte Carlo docking*

The shape tracing algorithm can be used to quickly move a shape through space without intersection. This is demonstrated in an outer-loop Monte Carlo docking method, which randomly rotates and moves the ligand to points on a sphere around the receptor, then fires the ligand towards the receptor at a random inward angle in a cone (Figure 1 left). This method is outlined in Algorithm 2:

1) Initialise a product manifold of two random points on a unit sphere (initial translation around the receptor and for the cone), and a random rotation (lines 3-5).
2) Rotate the ligand and translate it to the surface of the receptor's bounding sphere (lines 6-7).

3) Set the ray direction towards the receptor's centre, with some random variation $\gamma$ to form a cone (lines 8-9).
4) Fire the ligand at the receptor (shape tracing), updating the positions $\mathbf{x}'$ (line 10).
5) Sum the contact surface area at the solution $\mathbf{x}'$ (Equation 2), and save the solution parameters if there is more contact than the previous best (lines 11-14).

```
Algorithm 2: Outer-loop Monte Carlo Docking
   Input: $\mathbf{x}_{\text{orig}}, \phi, b, \gamma$
1  $\alpha_{\text{best}} = 0$                          ▷ surface area to maximize
2  while true do
3      $\vec{t}$ = random point on unit sphere       ▷ init translation
4      $\vec{c}$ = random point on unit sphere          ▷ for cone
5      $R$ = random rotation matrix                  ▷ for ligand
6      $s = \max(b)$                      ▷ max receptor side length
7      $\mathbf{x} = R\mathbf{x}_{\text{orig}} + \vec{t}s$               ▷ rotate & translate points
8      $\vec{v} = (1-\gamma)(-\vec{t}) + \gamma\vec{c}$              ▷ construct cone
9      $\vec{v} = \vec{v}/\|\vec{v}\|$
10     $\mathbf{x}' = \text{ShapeTracing}(\mathbf{x}, \phi, b, \vec{v})$
11     $\alpha_{\text{cur}} = \mathcal{L}(\phi, \mathbf{x}')$                        ▷ contact area
12     if $\alpha_{\text{cur}} > \alpha_{\text{best}}$ then
13         $\alpha_{\text{best}} = \alpha_{\text{cur}}$
14         save parameters
15     end
16 end
```

The final score we maximize is the contact surface area between the receptor and the ligand: shape tracing, which prevents intersections, can also support symmetric profiles demonstrated by a $C^\infty$ smooth delta, regularized by $\beta = 1$ (for 1 Å), with the loss $\mathcal{L}$:

$$\mathcal{L}(\phi, \mathbf{x}') = \int_\Omega \frac{\beta/\pi}{\beta^2 + \phi(\mathbf{x}')^2} \mathrm{d}\mathbf{x}' \qquad (2)$$

This increases as the ligand approaches the receptor boundary, and is not influenced by points away from the surface (such as the back of the ligand).

## IV. Results & Discussion

We compared the performance of Monte Carlo with and without Shape Tracing in docking surfaces generated by STID maps. We took the average values of the best dock over 10 runs and, for each run, sampling was terminated after 5000 iterations as further iteration yielded little improvement. Docking using shape tracing produced better solutions than naive sampling of ligand positions within the unit sphere (see Figure 2).

We investigated the relationship between cone width and solutions quality by varying the cone parameter $\gamma$. Best solutions after 5,000 iterations were averaged over 10 runs. A successful dock was defined as a ligand position within $\epsilon = 0.0001$ Å. We found that in general larger values of $\gamma$ led to worse solutions (see Table I) while a value of $\gamma = 0.05$ produced the best results. Slower tracing for higher values of $\gamma$ is due to near-misses and oblique collisions requiring more steps of the marching algorithm than collisions perpendicular to the receptor's surface. While MC sampling without ST is significantly faster, it fails to produce many successful docks. In general, shape tracing produces more viable solutions to the docking problem.

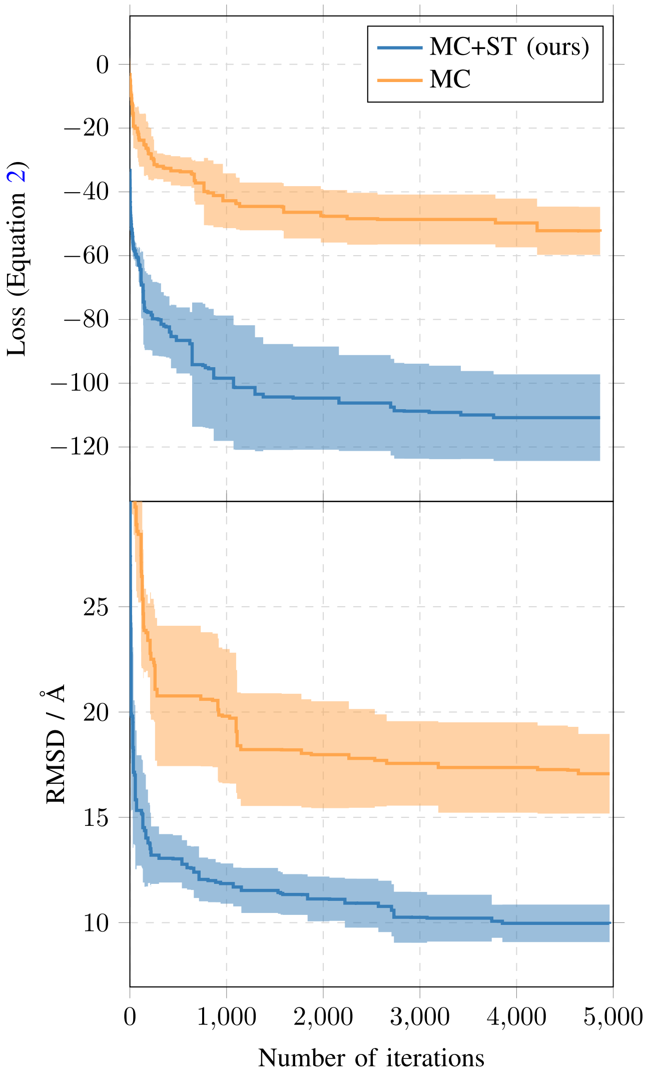

Fig. 2. Combining Shape Tracing (ST) to a Monte Carlo (MC) search improves the performance of shape-based protein docking. MC+ST finds poses with both lower loss (top) and smaller RMSD with respect of the known docked pose (bottom). Example results from CAPRI unbound case 1AKJ are shown (ligand 2CLR, receptor 1CD8).

TABLE I
WIDE CONE ANGLES HAVE MORE MISSES AND FIND WORSE SOLUTIONS

| | Baseline MC | Cone parameter $\gamma$ | | | | |
|---|---|---|---|---|---|---|
| | – | .00 | 0.05 | 0.10 | 0.15 | 0.20 |
| Iterations/s | 1040 | 83.8 | 83.5 | 82.4 | 79.5 | 75.9 |
| Docks/s | 0.001 | 83.8 | 83.5 | 82.4 | 79.5 | 75.6 |
| Misses/s | – | 0.00 | 0.00 | 0.012 | 0.04 | 0.351 |
| Best RMSD at 5k | 17.07 | 9.94 | 9.33 | 9.58 | 11.0 | 10.3 |
| Std dev. at 5k | 1.89 | 0.63 | 1.10 | 0.90 | 0.81 | 1.35 |

## V. Limitations and future work

The current profile must be initialised away from the receptor, rather than also inside it, and therefore it can not find occluded solutions without modification. While in theory the current profile is sensible for finding exact solutions where contact is maximised without intersection, in practice successful docking poses often feature some intersection [6]. Handling such cases can be achieved with an asymmetric energy profile that is negative for $x < 0$. In the future, we will extend our approach by replacing Equation 2 for a 1D function allowing intersections while minimising RMSD and other metrics (e.g. predicted interfacial residues) across a validation dataset.

## VI. Availability & Acknowledgements

The algorithm has been implemented using PyTorch on the GPU and is available at https://github.com/cwkx/ShapeTracing along with data for 14 additional test cases. The work was supported by the Engineering and Physical Sciences Research Council (EP/P016499/1).

## VII. Conclusion

We have found that shape marching with the updated bound significantly improves the convergence of finding non-intersecting solutions. The analytical ray-box intersection allows for quick evaluation of far-away solutions, and the shape tracing algorithm is able to march arbitrary non-convex shapes with a few inexpensive operations that can be calculated in parallel on a GPU. In the future, we will aim at identifying an asymmetric energy profile which allows for some intersection to occur, to facilitate the discovery of biologically relevant docking poses.

## References

[1] N. S. Pagadala, K. Syed, and T. Jack, “Software for molecular docking: A review,” *Biophysical Reviews*, vol. 9, no. 2, pp. 91–102, 2017.

[2] M. F. Lensink, S. Velankar, and S. J. Wodak, “Modeling protein–protein and protein–peptide complexes: Capri 6th edition,” *Proteins: Structure, Function, and Bioinformatics*, vol. 85, no. 3, pp. 359–377, 2017.

[3] R. Chen, L. Li, and Z. Weng, “Zdock: An initial-stage protein-docking algorithm,” *Proteins Struct. Funct. Bioinforma.*, vol. 52, pp. 80–87, jul 2003.

[4] S. Janson, D. Merkle, and M. Middendorf, “Molecular docking with multi-objective particle swarm optimization,” *Applied Soft Computing*, vol. 8, no. 1, pp. 666–675, 2008.

[5] “The rosettadock server for local protein-protein docking.,” *Nucleic Acids Res.*, vol. 36, no. Web Server issue, 2008.

[6] L. S. Rudden and M. T. Degiacomi, “Protein docking using a single representation for protein surface, electrostatics, and local dynamics,” *Journal of chemical theory and computation*, vol. 15, no. 9, pp. 5135–5143, 2019.

[7] S. Chaudhury, M. Berrondo, B. D. Weitzner, P. Muthu, H. Bergman, and J. J. Gray, “Benchmarking and analysis of protein docking performance in rosetta v3. 2,” *PloS one*, vol. 6, no. 8, 2011.

[8] J. C. Hart, “Sphere tracing: A geometric method for the antialiased ray tracing of implicit surfaces,” *The Visual Computer*, vol. 12, no. 10, pp. 527–545, 1996.

[9] Q. Zhang, T. Feng, L. Xu, H. Sun, P. Pan, Y. Li, D. Li, and T. Hou, “Recent advances in protein-protein docking,” *Current Drug Targets*, vol. 17, pp. 1586–1594, 2016.

[10] D. Kozakov, R. Brenke, S. R. Comeau, and S. Vajda, “Piper: An fft-based protein docking program with pairwise potentials,” *Proteins Struct. Funct. Genet.*, vol. 65, pp. 392–406, aug 2006.

[11] “Gramm-x public web server for protein-protein docking,” *Nucleic Acids Res.*, vol. 34, pp. W310–W314, jul 2006.

[12] T. M. K. Cheng, T. L. Blundell, and J. Fernandez-Recio, “PyDock: Electrostatics and desolvation for effective scoring of rigid-body protein-protein docking,” *Proteins Struct. Funct. Genet.*, vol. 68, pp. 503–515, apr 2007.

[13] M. Zacharias, “Attract: Protein-protein docking in capri using a reduced protein model,” in *Proteins Struct. Funct. Genet.*, vol. 60, pp. 252–256, Proteins, aug 2005.

[14] C. Dominguez, R. Boelens, and A. M. J. J. Bonvin, “Haddock: A proteinprotein docking approach based on biochemical or biophysical information,” *Journal of the American Chemical Society*, vol. 125, pp. 1731–1737, 2003.

[15] E. J. Gardiner, P. Willett, and P. J. Artymiuk, “Gapdock: A genetic algorithm approach to protein docking in capri round 1,” *Proteins Struct. Funct. Genet.*, vol. 52, pp. 10–14, jul 2003.

[16] “Software news and updates autodock4 and autodocktools4: Automated docking with selective receptor flexibility,” *J. Comput. Chem.*, vol. 30, pp. 2785–2791, dec 2009.

[17] I. H. Moal and P. A. Bates, “Swarmdock and the use of normal modes in protein-protein docking,” *International Journal of Molecular Sciences*, vol. 11, no. 10, pp. 3623–3648, 2010.

[18] V. Namasivayam and R. Günther, “Pso@autodock: A fast flexible molecular docking program based on swarm intelligence,” *Chem. Biol. Drug Des.*, vol. 70, no. 6, pp. 475–484, 2007.

[19] E. Mashiach, D. Schneidman-Duhovny, N. Andrusier, R. Nussinov, and H. J. Wolfson, “Firedock: a web server for fast interaction refinement in molecular docking.,” *Nucleic Acids Res.*, vol. 36, jul 2008.

[20] D. Schneidman-Duhovny, Y. Inbar, R. Nussinov, and H. J. Wolfson, “Patchdock and symmdock: Servers for rigid and symmetric docking,” *Nucleic Acids Res.*, vol. 33, no. SUPPL. 2, 2005.

[21] Y. Yan and S.-Y. Huang, “Pushing the accuracy limit of shape complementarity for protein-protein docking,” *BMC Bioinformatics*, vol. 20, no. 25, p. 696, 2019.

[22] T. L. Kay and J. T. Kajiya, “Ray tracing complex scenes,” *ACM SIGGRAPH computer graphics*, vol. 20, no. 4, pp. 269–278, 1986.

[23] C. G. Willcocks, *Sparse volumetric deformation*. PhD thesis, Durham University, 2013.