Stability analysis of kinetic oriented bounding boxes

Wouter Meulemans[1], **Kevin Verbeek**[1], **and Jules Wulms**[1]

1 **Department of Mathematics and Computer Science, TU Eindhoven, the Netherlands**
`[w.meulemans|k.a.b.verbeek|j.j.h.m.wulms]@tue.nl`

1 Introduction

There exist many applications that desire an algorithm to be *stable*: small changes in the input lead to small changes in the output. Stability is especially important when analyzing or visualizing time-varying data. A sudden change to the input renders an unstable visualization completely ineffective, as it will be hard or impossible to follow the temporal patterns. It is therefore of interest to develop stable algorithms that deal with such changes in the input in an elegant way, so that the output does not lose its effectiveness over time.

Shape descriptors are simplified representations of (complex) shapes and can be used to effectively summarize data, even as it changes over time. They play an important role in fields that rely on shape analysis, like computer vision (shape recognition) [4, 6, 29], computer graphics (bounding boxes for broad-phase collision detection) [1, 12, 16, 24], medical imaging (diagnosis or surgical planning) [7, 13, 15, 30], and machine learning (shape classification) [22, 26, 27, 28]. In this abstract we focus on a shape descriptor that captures the overall orientation of the underlying shape (represented by a point cloud). Specifically, we study the *oriented bounding box* (OBB): an oriented bounding box that contains all points (see Figure 1 left). We say that oriented bounding boxes of smaller area are of better *quality*. Unfortunately, oriented bounding boxes of optimal quality are unstable and may have discrete "flips" in their orientation, even for continuously moving point sets (Figure 1 right). Hence, we want to sacrifice some quality for stability.

Problem description. The main goal of this abstract is to formally analyze the trade-off between quality and stability for an OBB. Our input consists of a set of n moving points $P = P(t) = \{p_1(t), \ldots, p_n(t)\}$ in 2 dimensions, where each $p_i(t)$ is a function $p_i\colon [0,T] \to \mathbb{R}^2$. We assume that, at each time t, not all points are at the same position. We further assume that each point moves with at most unit speed, that is, $\|p_i'(t)\| \leq 1$ for all times t. As output we want to compute an OBB containing all the input points. The oriented bounding box essentially describes an optimization function, which captures the quality of the shape descriptor, that is, how well the descriptor represents (the orientation of) the underlying shape. The optimization function is the area of the bounding box and it can be minimized over all orientations. That is, we consider the optimization of function $f_{\text{OBB}}(\alpha, P)$ which represents the area of the smallest bounding box with orientation α on point set P.

We can now also consider bounding boxes of suboptimal quality, with slightly larger area than the smallest one. This allows us to make a trade-off between quality and other desirable properties of OBB, such as its stability. Since the optimization function optimizes over orientations, the output consists of an orientation $\alpha(t)$ for every time step $t \in [0,T]$. This orientation is an element of the real projective line $\mathbb{RP}^1$, but we represent $\alpha(t)$ by a unit vector in $\mathbb{R}^2$ and implicitly identify opposite vectors, which is equivalent. Furthermore, we assume that the output $\alpha(t)$ is computed for all real values $t \in [0,T]$.

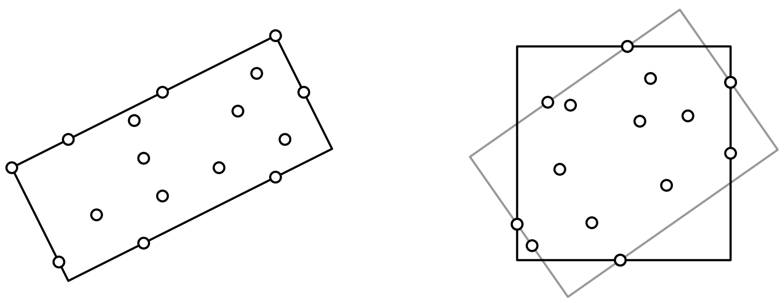

Figure 1 Examples of OBB on different point sets. On the right two optimal boxes exist.

For OBB we typically compute more than just an orientation. However, it is easy to see that the stability of OBB is mostly affected by the optimal orientation. We therefore ignore other aspects of the shape descriptor to analyze the stability, and assume that these aspects are chosen optimally for the given orientation without any cost with regard to the stability.

Stability analysis. To analyze the stability of OBB we use the framework introduced by Meulemans et al. [18]. This framework defines various stability measures, one of which is the Lipschitz stability. An algorithm is *K-Lipschitz stable* if its output changes at most K units, when the input points move one unit. We study the Lipschitz stability of OBB and slightly reformulate the definition by restricting it to our setting. Let $\mathcal{A}$ be an algorithm that takes a point set as input and computes an orientation. The definition of the Lipschitz stability ratio depends on the metrics (distance functions) on the input and output space. This is straightforward: the input space is $\mathbb{R}^{2n}$ and the output space is topologically equivalent to S^1, hence we can use the Euclidean and circle metric respectively. We define the K-Lipschitz stability ratio of OBB as follows:

$$\rho_{LS}(\textsc{obb}, K) = \inf_{\mathcal{A}} \sup_{P(t)} \max_{t\in[0,T]} \frac{f_{\textsc{obb}}(\mathcal{A}(P(t)), P(t))}{\min_\gamma f_{\textsc{obb}}(\gamma, P(t))} \tag{1}$$

where the infimum is taken over all algorithms $\mathcal{A}$ for which $\mathcal{A}(P(t))$ is K-Lipschitz. Thus, the difference in orientation or angle is at most K radians per unit of change in the input. Informally, ρ_{LS} is the best approximation ratio any algorithm can achieve for a given K.

Kinetic algorithms. Algorithms for kinetic (moving) input can adhere to different models, which may influence the results of the stability analysis. We focus on *state-aware* algorithms, which map input $I(t)$ on time t to output, but also maintain a state S (typically the output at the previous time step) over time. This contrasts *stateless* and *clairvoyant* algorithms, which depend only on the input $I(t)$ at a particular time or the complete function $I(t)$ respectively.

Interestingly, for stateless algorithms the K-Lipschitz stability ratio is unbounded for any constant K. This can be argued topologically by realizing that a stable stateless algorithm defines a continuous map between the input and output space, and we can show that an algorithm with bounded approximation ratio must define a map between two subspaces that are not homotopic. We therefore move our attention to state-aware algorithms.

Our results. A state-aware algorithm can enforce continuity by keeping the output at the previous time step as the state. We define a *chasing algorithm*, which moves the output of the previous time step with maximum allowed speed towards the optimal solution at the current time. In the remainder of this abstract we analyze the quality and stability of such an algorithm: Section 2.1 explains how stability is ensured, while Section 2.2 shows how the quality is affected.

Related work. Computing OBB for static point sets is a classic problem in computational geometry. In two dimensions, one side of the optimal box aligns with a side of the convex hull and it can be computed in linear time after finding the convex hull [11, 23]; a similar property holds in three dimensions, allowing a cubic-time algorithm [19]. The relevance of bounding boxes in 3D as a component of other algorithms led to efficient approximation algorithms, such as a $(1+\epsilon)$-approximation algorithm in $O(n + 1/\epsilon^{4.5})$ time or an easier algorithm with running time $O(n \log n + n/\epsilon^3)$ [2]. Bounding boxes find applications in tree structures for spatial indexing [3, 14, 20, 21] and in collision detection and ray tracing [1, 12, 24]. Results on the stability of facility location problems [5, 9, 10, 8] and the medial axis [17] all predate a framework for analyzing stability introduced by Meulemans et al. [18]. In [18] the authors apply the framework to maintaining the Euclidean minimum spanning tree on a set of moving points. They show a bounded topological stability for various topologies on the space of spanning trees, and that the Lipschitz stability is at most linear, but also at least linear if the allowed speed for the changes in the tree is too low. Bounds on the topological stability of a problem essentially are lower bounds on its Lipschitz stability. Van der Hoog et al. study the topological stability of the k-center problem [25], showing upper and lower bounds on the stability ratio for various measures. They also provide an algorithm to determine the best ratio attainable for a given set of moving points.

2 Lipschitz stability of a state-aware algorithm

To derive meaningful bounds on the Lipschitz stability ratio, the relation between distances/speeds in input and output space should be *scale-invariant* [18]. This is currently not the case: if we scale the coordinates of the points, then the distances in the input space change accordingly, but the distances in the output space (between orientations) do not. To remedy this problem, we require that diameter D of $P(t)$ is at least 1 for every time t.

We use a chasing algorithm as introduced in Section 1. However, instead of chasing the orientation of OBB, we chase the orientation of a diametrical pair. Although chasing the optimal shape descriptor would be better in general, chasing a diametrical pair is easier to analyze and sufficient to obtain a bounded Lipschitz stability ratio for OBB.

2.1 Chasing the diametrical pair

We denote the orientation of the diametrical pair as $\alpha = \alpha(t)$ and the diameter as $D = D(t) \geq 1$. Furthermore, let $W = W(t)$ be the width of the thinnest strip with orientation $\alpha(t)$ covering all points in $P(t)$, and let $z = z(t) = W(t)/D(t)$ be the *aspect ratio* of the *diametric box* with orientation $\alpha(t)$. We will generally omit the dependence on t if t is clear from the context. Finally, we have a chasing algorithm that has orientation $\beta = \beta(t)$ and the difference in orientation is at most a constant K, when the input has moved one unit.

Approach. The main goal is to keep β as close to α as possible, specifically within a sufficiently small interval around α. The challenge lies with the discrete flips of α. We must argue that, although flips can happen instantaneously, they cannot happen often within a short time-span – otherwise we can never keep β close to α with a bounded speed. Furthermore, the size of the interval must depend on the aspect ratio z, since if $z = 0$, the interval around α must have zero size as well to guarantee a bounded approximation ratio.

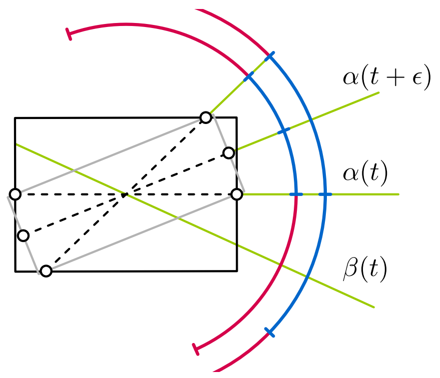

Figure 2 Interval I at time t (outer) and time $t+\epsilon$ (inner). The safe/danger zones and orientations are indicated in blue, red and green respectively. Diametrical pairs are connected by dashed lines.

For the analysis we introduce three functions depending on z: $T(z)$, $H(z)$, and $J(z)$. Function $H(z)$ defines an interval $[\alpha - H(z), \alpha + H(z)]$ called the *safe zone*. We aim to show that, if β leaves the safe zone at some time t, it must return to the safe zone within the time interval $(t, t + T(z)]$. We also define a larger interval $I = [\alpha - H(z) - J(z), \alpha + H(z) + J(z)]$. We refer to the parts of I outside of the safe zone as the *danger zone* (see Figure 2). Although β may momentarily end up in the danger zone due to discontinuous changes, it must quickly find its way back to the safe zone. We aim to guarantee that β stays within I at all times. Let $E = E(t)$ refer to an endpoint of I. We call $J(z)$ the *jumping distance* and say it is valid if $J(z)$ upper bounds how far E can "jump" in a single time step. Note that $J(z)$ is defined recursively through E, so we need to be careful to choose the right function for $J(z)$. For the other functions we choose $T(z) = z/4$ and $H(z) = c \arcsin(z)$ for a constant c (chosen later).

Changes in orientation and aspect ratio. To verify that the chosen functions $T(z)$ and $H(z)$ satisfy the intended requirements, and to define the function $J(z)$, we need to bound how much α and z can change over a time period of length Δt. We refer to these bounds as $\Delta\alpha(z, \Delta t)$ and $\Delta z(z, \Delta t)$, respectively. Note that, since the diameter can change discontinuously, we generally have that $\Delta\alpha(z, 0) > 0$ and $\Delta z(z, 0) > 0$.

▶ **Lemma 2.1.** $\Delta\alpha(z, \Delta t) \leq \arcsin(z + \Delta t(1 + z))$ *for* $\Delta t \leq (1 - z)/(1 + z)$.

▶ **Lemma 2.2.** $\Delta z(z, \Delta t) \leq z - \frac{\sin(\frac{1}{2}\arcsin(z)) - 2\Delta t}{1 + 2\Delta t}$ *for* $\Delta t \leq \sin(\frac{1}{2}\arcsin(z))/2$.

Jumping distance. Using Lemma 2.1 and Lemma 2.2 we can derive a valid function for $J(z)$. Recall that we require that $J(z)$ is at least the amount E can move in $\Delta t = 0$ time.

▶ **Lemma 2.3.** $J(z) = (c + 2)\arcsin(z)$ *is a valid jumping distance function.*

Proof. By Lemma 2.1 and Lemma 2.2 we get that $\Delta E(z, 0) \leq \Delta\alpha(z, 0) + H(z) - H(z - \Delta z(z, 0)) + J(z) - J(z - \Delta z(z, 0))$. Since $\Delta\alpha(z, 0) \leq \arcsin(z)$ and $\Delta z(z, 0) \leq z - \sin(\frac{1}{2}\arcsin(z))$, we get after simplification that $\Delta E(z, 0) \leq (1 + c/2)\arcsin(z) + J(z) - J(\sin(\frac{1}{2}\arcsin(z)))$. Since we require that $J(z) \geq \Delta E(z, 0)$, it suffices to show that the following holds: $J(\sin(\frac{1}{2}\arcsin(z))) \geq (1 + c/2)\arcsin(z)$. Using the provided function, we get that $J(\sin(\frac{1}{2}\arcsin(z))) = (c + 2)\arcsin(z)/2$ as required. ◀

▶ **Corollary 2.4.** *If* β *is in* I, *then* $|\alpha - \beta| \leq (2c + 2)\arcsin(z)$.

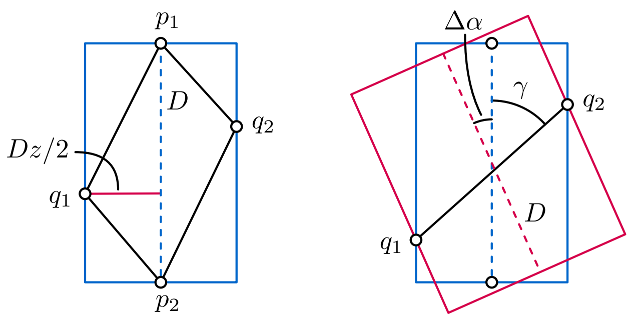

Figure 3 Illustrations supporting proof of Lemma 2.7.

Bounding the speed. To show that the orientation β stays within the interval I, we argue that over a time period of $T(z)$ we can rotate β at least as far as E. As the endpoint of the safe zone moves at most as fast as E, this implies that if β leaves the safe zone at time t, it returns to it in the time period $(t, t + T(z)]$. Thus we require that $KT(z) \geq \Delta E(z, T(z))$. We need to keep up only when the safe zone does not span all orientations, that is, the above inequality must hold only when $H(z) \leq \pi/2$ or $z \leq \sin(\frac{\pi}{2c})$. For the following speed bound we choose a specific value $c = 3$. Hence we only need to chase α when $z \leq \sin(\frac{\pi}{6}) = \frac{1}{2}$.

▶ **Lemma 2.5.** *If $K \geq 40$, then $|\beta(t) - \alpha(t)| \leq 8 \arcsin(z)$ (using $c = 3$) for all times t.*

2.2 Lipschitz stability ratio

What remains is to analyze the approximation ratio of the chasing algorithm for OBB. Corollary 2.4 implies that the orientation β of the chasing algorithm is at most an angle $(2c + 2) \arcsin(z)$ away from the orientation of a diametrical pair of points.

▶ **Lemma 2.6.** $\sin(\lambda \arcsin(x)) \leq \lambda x$ *for* $\lambda \geq 1$ *and* $0 \leq x \leq 1$*:*

▶ **Lemma 2.7.** *If $|\beta - \alpha| \leq (2c + 2) \arcsin(z)$, then $f_{\text{OBB}}(\beta, P) \leq (4c + 6) \min_x f_{\text{OBB}}(x, P)$.*

Proof. Assume that at some time t we have a diametric box with diameter D and aspect ratio z, and let (p_1, p_2) be a diametrical pair. The smallest OBB must contain p_1 and p_2 and must hit the sides of the diametric box at, say, q_1 and q_2 (see Figure 3). Since the smallest OBB must contain the triangles formed by $\{p_1, p_2, q_1\}$ and $\{p_1, p_2, q_2\}$, the area of this box must be at least the sum of the areas of these two triangles, which is $D^2z/2$.

Now consider the box of the chasing algorithm, where $\Delta\alpha = |\beta - \alpha| \leq (2c + 2) \arcsin(z)$. We assume that the major axis of the box has length D, which is worst possible. Let the minor axis of the box be bounded by two points q_1 and q_2, where the angle between the line through q_1 and q_2 and the line through the diametrical pair is γ. Note that the distance between q_1 and q_2 is bounded by $\min(D, zD/\sin(\gamma))$. The angle between the minor axis of the box and the line through q_1 and q_2 is $\pi/2 - \gamma - \Delta\alpha$. Thus, the length of the minor axis is $\min(D, zD/\sin(\gamma)) \cos(\pi/2 - \gamma - \Delta\alpha) = \min(D \sin(\gamma + \Delta\alpha), zD \sin(\gamma + \Delta\alpha)/\sin(\gamma))$. Since the function $\sin(\gamma + \Delta\alpha)/\sin(\gamma)$ is decreasing in γ, we attain the maximum when $z/\sin(\gamma) = 1$ or $\gamma = \arcsin(z)$. Hence, the area of this box is at most $D^2 \sin((2c + 3) \arcsin(z))$, which is at most $D^2z(2c + 3)$ by Lemma 2.6. Thus, $f_{\text{OBB}}(\beta, P) \leq (4c + 6) min_x f_{\text{OBB}}(x, P)$. ◀

By combining Lemmata 2.5 and 2.7, we obtain this bound on the Lipschitz stability of OBB.

▶ **Theorem 2.8.** *The Lipschitz stability ratio for* OBB *is bounded by* $\rho_{\text{LS}}(\text{OBB}, 40) \leq 18$.

References

1 Gill Barequet, Bernard Chazelle, Leonidas Guibas, Joseph Mitchell, and Ayellet Tal. BOX-TREE: A hierarchical representation for surfaces in 3d. *Comput. Graph. Forum*, 15(3):387–396, 1996.

2 Gill Barequet and Sariel Har-Peled. Efficiently approximating the minimum-volume bounding box of a point set in three dimensions. *J. Algorithms*, 38(1):91–109, 2001.

3 Norbert Beckmann, Hans-Peter Kriegel, Ralf Schneider, and Bernhard Seeger. The R*-tree: An efficient and robust access method for points and rectangles. In *Proc. 1990 ACM SIGMOD International Conference on Management of Data*, pages 322–331, 1990.

4 Serge Belongie, Jitendra Malik, and Jan Puzicha. Shape matching and object recognition using shape contexts. *IEEE Trans. Pattern Anal. Mach. Intell.*, 24(4):509–522, 2002.

5 Sergei Bespamyatnikh, Binay Bhattacharya, David Kirkpatrick, and Michael Segal. Mobile facility location. In *Proc. 4th Workshop on Discrete Algorithms and Methods for Mobile Computing and Communications*, pages 46–53, 2000.

6 Michael Bronstein and Iasonas Kokkinos. Scale-invariant heat kernel signatures for non-rigid shape recognition. In *23rd IEEE Conference on Computer Vision and Pattern Recognition*, pages 1704–1711, 2010.

7 Dragana Brzakovic, Xiao Mei Luo, and P. Brzakovic. An approach to automated detection of tumors in mammograms. *IEEE Transactions on Medical Imaging*, 9(3):233–241, 1990.

8 Mark de Berg, Marcel Roeloffzen, and Bettina Speckmann. Kinetic 2-centers in the black-box model. In *Proc. 29th Symposium on Computational Geometry*, pages 145–154, 2013.

9 Stephane Durocher and David Kirkpatrick. The steiner centre of a set of points: Stability, eccentricity, and applications to mobile facility location. *International Journal of Computational Geometry & Applications*, 16(04):345–371, 2006.

10 Stephane Durocher and David Kirkpatrick. Bounded-velocity approximation of mobile Euclidean 2-centres. *International Journal of Computational Geometry & Applications*, 18(03):161–183, 2008.

11 Herbert Freeman and Ruth Shapira. Determining the minimum-area encasing rectangle for an arbitrary closed curve. *Commun. ACM*, 18(7):409–413, 1975.

12 Stefan Gottschalk, Ming Lin, and Dinesh Manocha. OBBTree: A hierarchical structure for rapid interference detection. In *Proc. 23rd Annual Conference on Computer Graphics and Interactive Technique*, pages 171–180, 1996.

13 Xianfeng Gu, Yalin Wang, Tony Chan, Paul Thompson, and Shing-Tung Yau. Genus zero surface conformal mapping and its application to brain surface mapping. *IEEE Trans. Med. Imaging*, 23(8):949–958, 2004.

14 Antonin Guttman. R-trees: A dynamic index structure for spatial searching. In *Proc. 1984 ACM SIGMOD International Conference on Management of Data*, pages 47–57, 1984.

15 András Kelemen, Gábor Székely, and Guido Gerig. Elastic model-based segmentation of 3-D neuroradiological data sets. *IEEE Trans. Med. Imaging*, 18(10):828–839, 1999.

16 James Klosowski, Martin Held, Joseph Mitchell, Henry Sowizral, and Karel Zikan. Efficient collision detection using bounding volume hierarchies of k-DOPs. *IEEE Trans. Vis. Comput. Graph.*, 4(1):21–36, 1998.

17 Kyle Sykes David Letscher and Kyle Sykes. On the stability of medial axis of a union of disks in the plane. In *Proc. 28th Canadian Conference on Computational Geometry*, pages 29–33, 2016.

18 Wouter Meulemans, Bettina Speckmann, Kevin Verbeek, and Jules Wulms. A framework for algorithm stability and its application to kinetic euclidean MSTs. In *Proc. 13th LATIN*, LNCS 10807, pages 805–819, 2018.

19 Joseph O'Rourke. Finding minimal enclosing boxes. *International Journal of Parallel Programming*, 14(3):183–199, 1985.

20 Nick Roussopoulos and Daniel Leifker. Direct spatial search on pictorial databases using packed r-trees. In *Proc. 1985 ACM SIGMOD International Conference on Management of Data*, pages 17–31, 1985.

21 Timos Sellis, Nick Roussopoulos, and Christos Faloutsos. The R+-tree: A dynamic index for multi-dimensional objects. In *Proc. 13th International Conference on Very Large Data Bases*, pages 507–518, 1987.

22 Hang Su, Subhransu Maji, Evangelos Kalogerakis, and Erik Learned-Miller. Multi-view convolutional neural networks for 3D shape recognition. In *2015 IEEE International Conference on Computer Vision*, pages 945–953, 2015.

23 Godfried Toussaint. Solving geometric problems with the rotating calipers. In *Proc. IEEE Melecon*, volume 83, page A10, 1983.

24 Gino van den Bergen. Efficient collision detection of complex deformable models using AABB trees. *J. Graphics, GPU, & Game Tools*, 2(4):1–13, 1997.

25 Ivor van der Hoog, Marc van Kreveld, Wouter Meulemans, Kevin Verbeek, and Jules Wulms. Topological stability of kinetic k-centers. *CoRR*, abs/1810.00794, 2018.

26 Manik Varma and Debajyoti Ray. Learning the discriminative power-invariance trade-off. In *Proc. IEEE 11th International Conference on Computer Vision*, pages 1–8, 2007.

27 Jin Xie, Guoxian Dai, Fan Zhu, Edward Wong, and Yi Fang. DeepShape: Deep-learned shape descriptor for 3D shape retrieval. *IEEE Trans. Pattern Anal. Mach. Intell.*, 39(7):1335–1345, 2017.

28 Hao Zhang, Alexander Berg, Michael Maire, and Jitendra Malik. SVM-KNN: discriminative nearest neighbor classification for visual category recognition. In *Proc. 2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 2126–2136, 2006.

29 Yu Zhong. Intrinsic shape signatures: A shape descriptor for 3d object recognition. In *Computer Vision Workshops (ICCV Workshops), 2009 IEEE 12th International Conference*, pages 689–696. IEEE, 2009.

30 Barbara Zitová and Jan Flusser. Image registration methods: a survey. *Image Vision Comput.*, 21(11):977–1000, 2003.