IMAGE PROCESSING ON LINE

Published in Image Processing On Line on 2017–12–30.
Submitted on 2017–06–26, accepted on 2017–10–20.
ISSN 2105–1232 © 2017 IPOL & the authors CC–BY–NC–SA
This article is available online with supplementary materials,
software, datasets and online demo at
https://doi.org/10.5201/ipol.2017.214

A Contrario 3D Point Alignment Detection Algorithm

Alvaro Gómez[1], Gregory Randall[1], Rafael Grompone von Gioi[2]

[1] IIE, Universidad de la República, Uruguay ({agomez, randall}@fing.edu.uy)
[2] CMLA, ENS Cachan, France (grompone@cmla.ens-cachan.fr)

Communicated by Bertrand Kerautret *Demo edited by* Álvaro Gómez

Abstract

In this article we present an algorithm for the detection of perceptually relevant alignments in 3D point clouds. The algorithm is an extension of the algorithm developed by Lezama et al. [J. Lezama, J-M. Morel, G. Randall, R. Grompone von Gioi, A Contrario 2D Point Alignment Detection, IEEE Transactions on Pattern Analysis and Machine Intelligence, 37 (3), pp. 499-512, 2015] for the case of sets of 2D points. The algorithm is based on the a contrario detection theory that mathematically formalizes the non-accidentalness principle proposed for perception: an observed structure is relevant if it rarely occurs by chance. This framework has been widely used in different detection tasks and leads to algorithms with a single critical parameter to control the number of false detections.

Source Code

The reviewed source code and documentation for this algorithm are available from the web page of this article[1]. Compilation and usage instruction are included in the README.txt file of the archive.

Keywords: point alignment detection; 3D point alignments; a contrario method; Gestalt theory

1 Introduction

The method described in this article is based on the *a contrario* methodology proposed by Desolneux, Moisan and Morel [2, 3]. It is a mathematical formalization of the *non-accidentalness principle* proposed for perception [11, 1, 10] (sometimes called *Helmholtz principle*). In this approach, an observed structure is considered relevant if it rarely occurs by chance. This is implemented assuming a null-hypothesis H_0 for the data where no detections should occur (the a contrario model). The rarity or non-accidentalness of a structure is quantified as the probability of observing that structure under the H_0 hypothesis.

[1] https://doi.org/10.5201/ipol.2017.214

The 3D algorithm described in this article is an extension of the 2D algorithm developed by Lezama et al. in [4] and also available in IPOL [5].

In order to detect significant alignments, the algorithm estimates a local density around a candidate alignment, evaluates the regularity of the spacing of the points in the alignment, and defines a criterion and a procedure to select the best interpretation among redundant detections.

Given a data set of 3D points, a candidate alignment consists of a thin cylinder in space defined by two points of the data set. The idea is to evaluate if the point density inside the cylinder is significantly high with respect to the local background. To provide a local estimation of the background density, a coaxial larger cylinder is used to count the points surrounding the candidate alignment (hereinafter we will refer to this density as the "local background density", or abbreviated as "local density").

A meaningful alignment should also have regularly spaced points inside the candidate cylinder. In order to evaluate this regularity the candidate cylinder is divided into equally shaped intervals or pill-boxes and the number of occupied boxes is counted. A candidate alignment will be validated if the number of occupied boxes is significantly larger in a statistical sense relative to the local background density.

The last step of the algorithm consists of a redundancy reduction. Given a meaningful alignment, many smaller or larger cylinders overlapping the main alignment may also be meaningful. This masking phenomenon can involve points that belong to the real alignment as well as background points near but not necessarily part of the alignment. To this aim the non-maximal detections are removed. Candidate alignments are first ordered by decreasing significance. A test is conducted to know if an event masks the neighbouring ones. The candidate alignments are kept if they are not masked by a more significant one.

The rest of this article is organized as follows: Section 2 describes the methodology of the alignment detector, including the redundancy reduction procedure. Section 3 describes the algorithm, with implementation details and an analysis of the computational complexity. Finally, Section 4 shows experiments on synthetic and acquired 3D data.

2 Method Description

Consider a set of N points defined in a 3D domain D with volume V_D. We are interested in detecting groups of points that are well aligned. A reasonable a contrario hypothesis H_0 for this problem is to suppose that the N points are the result of a random process where points are independent and uniformly distributed in the domain. This does not mean that the method will only work when the background points follow exactly this hypothesis. What is important is that this is a good model for isotropic elements where any alignment is accidental.

The validation is done using the a contrario framework. This methodology consists in assuming a null hypothesis H_0 for the data, where no detections should occur, and defining a detection as a violation of this hypothesis (i.e. events that could hardly happen under the hypothesized model). The fundamental quantity of the a contrario approach is the Number of False Alarms (NFA) associated with an event e (up to a certain precision) and a set of points X. Given an a contrario hypothesis H_0 for the set of points in the domain, the NFA of an event is a bound on the expected number of occurrences of the event under H_0. Given an event e, its NFA is

$$NFA(e) = N_{tests} \cdot P_{H_0}(e), \tag{1}$$

where N_{tests} is the number of tested events and $P_{H_0}(e)$ is the probability of the event e under H_0.

In order to implement the a contrario approach, the proposed algorithm is organized in two main parts: The first part comprises the search and validation of candidate alignments. This part implies an exhaustive search across the candidates determined by each pair of points in the domain, the estimation of the local density of points around each candidate and the estimation of the regularity of the point locations along the alignment. A candidate alignment will be validated if the regularity of point locations is significantly large in a statistical sense given the local point density. In the second part, the previously validated candidates undergo a non-maximum suppression strategy to reduce redundancy. In this part a candidate alignment is kept if it is not masked by a more significant one.

2.1 Candidate Alignment Cylinder

Candidate alignments will be formed by taking every pair of points and constructing a cylinder r, whose axis ends at those points, and has variable width. With N points in the domain this gives a total of $\frac{N(N-1)}{2}$ candidates.

Figure 1 shows a schema of the cylindrical regions considered for the tests. Given a pair of points in the domain we can define:

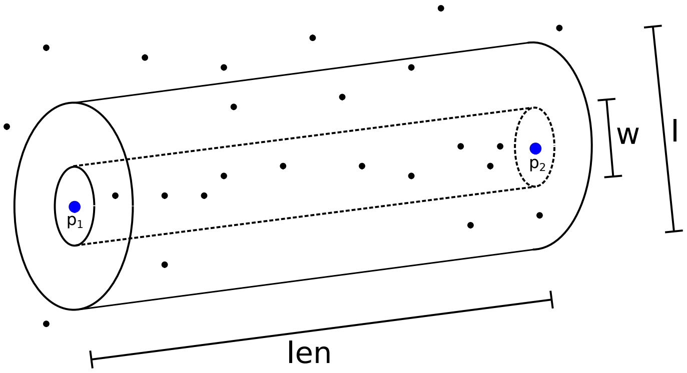

Figure 1: A schematic representation of the evaluated alignment determined by a pair of points in the domain. For each candidate, a set of alignment cylinders (with diameter w) and local cylinders (with diameter l) are considered. See the text for further details.

- $\mathbf{p_1}$: first point in the pair that determines the candidate alignment,
- $\mathbf{p_2}$: second point in the pair that determines the candidate alignment,
- **len**: distance between p_1 and p_2 and also the height of the cylinders considered for the test,
- **w**: diameter of the candidate alignment cylinder. Points within this cylinder *belong* to the alignment,
- **l**: diameter of the local cylinder. The local cylinder is used to estimate the local background density around the candidate alignment.

Since an alignment should be an elongated structure, the ratio $\frac{len}{w}$ must be kept over a reasonably minimum. In our implementation, a minimum $\frac{len}{w}$ ratio of 10 is used.

For each candidate alignment a set of diameters w and l are considered in order to take into account a range of possible density relations. In the implementation a set of $W = 8$ different candidate diameters and $L = 8$ local diameters are considered. The values w and l follow a geometric series with a factor of $2^{-\frac{1}{4}}$ or equivalently $\frac{1}{\sqrt[4]{2}}$.

$$w = \frac{len}{10},\ \frac{len}{10}\frac{1}{\sqrt[4]{2}},\ \frac{len}{10}\frac{1}{\sqrt[4]{2}}\frac{1}{\sqrt[4]{2}},\ \ldots\ \ldots,\ \frac{len}{10}\left\{\frac{1}{\sqrt[4]{2}}\right\}^{W-1}$$

$$l = \frac{len}{\sqrt{10}},\ \frac{len}{\sqrt{10}}\frac{1}{\sqrt[4]{2}},\ \frac{len}{\sqrt{10}}\frac{1}{\sqrt[4]{2}}\frac{1}{\sqrt[4]{2}},\ \ldots\ \ldots,\ \frac{len}{\sqrt{10}}\left\{\frac{1}{\sqrt[4]{2}}\right\}^{W-1}$$

2.2 Density Estimation

In order to evaluate if a configuration of points forms a statistically significant alignment, the method requires an estimation of the density of points around that candidate. The density should be estimated locally around the candidate since the significance is related to the saliency of the alignment with respect to its near background.

The local density can be estimated by counting the points in the local cylinder of diameter l which has a volume $V_l = len \cdot \frac{\pi l^2}{4}$. Although this is a reasonable approach when the points are more or less uniformly distributed in space it can lead to unwanted detections in the case of flat borders between regions of different density as shown in Figures 2 and 3. When the points are gathered in these flat structures the estimated local density in the border can be low with respect to the density in the alignment cylinder and points in the border of the structure can be detected as an alignment.

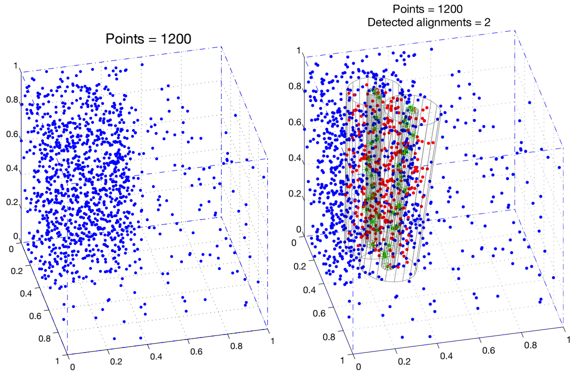

Figure 2: Detection of alignments in the border of two regions with different density. If the local density is estimated by counting in the whole local cylinder, the computed density in the border is low with respect to the density in the alignment cylinder leading to the deceptive detection of unexpected alignments. If seen in color, green points are part of the alignment and red points are part of the local cylinder but not of the alignment cylinder.

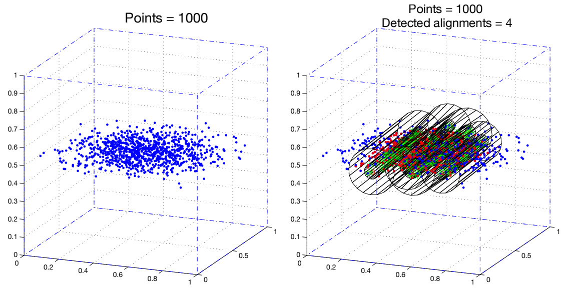

Figure 3: Detection of alignments in a flat region. If the local density is estimated by counting in the whole local cylinder, the computed local density is low with respect to the density in the alignment cylinder leading to the deceptive detection of unexpected alignments. If seen in color, green points are part of the alignment and red points are part of the local cylinder but not of the alignment cylinder.

To overcome the problem in this kind of structures, the local density can be estimated conservatively by dividing the local cylinder in sectors and considering the maximum count among the sectors. Figure 4 depicts the idea in the case of four sectors. The volume of the local cylinder, excluding the alignment cylinder, is divided in S sectors (4 in this example) named S_1 to S_S. Let M_1 to M_S be the respective point counts in those sectors. Let M be the point count in the alignment cylinder. A conservative estimate of the local number of points can be computed as

$$n^*(R, \mathbf{x}) = S \cdot \max(M_1, \ldots, M_S) + M. \tag{2}$$

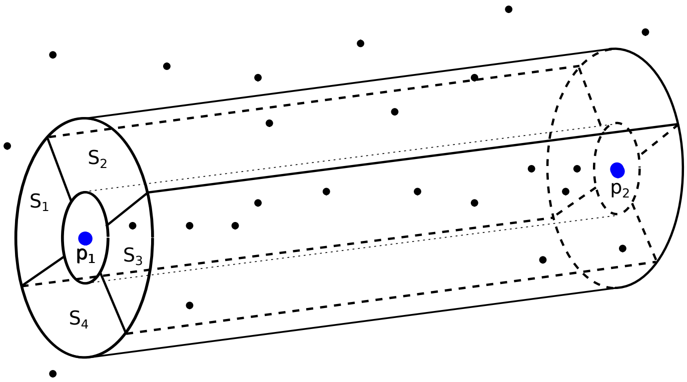

Figure 4: In order to conservatively estimate the local point density, the local cylinder is divided into sectors and the maximum count for the sectors is considered for the estimation. This approach is appropriate to avoid the unwanted detection of simple borders as alignments in structures such as corners or flat regions.

2.3 Point Regularity

As suggested in other studies [7, 8, 9], apart from the density with respect to the background, a significant alignment should also have regularly spaced points inside the candidate alignment cylinder.

This reasonable condition is called the law of *constant spacing* by Gestaltists and is an important factor that makes an alignment perceptually meaningful. The condition is also mandatory to avoid the detection of cluster-like structures as alignments. Figure 5 shows an example of this cluster-issue.

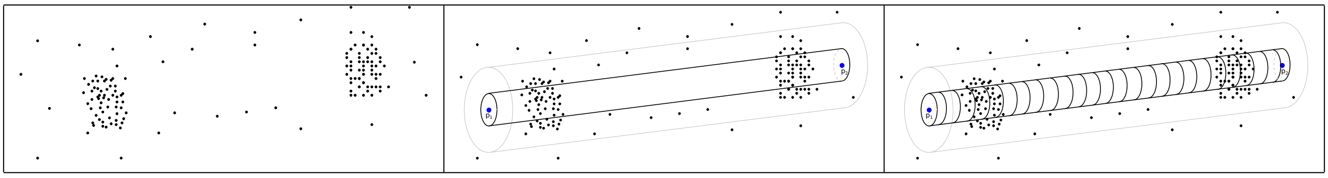

Figure 5: Left: Dot pattern with two point clusters but no alignment. Center: Although there is no “true” alignment, a thin cylinder with a high point density with respect to the local background may be found across the two clusters giving a false detection. Right: If the cylinder is divided into pill-boxes and only the occupation of the boxes is considered in the estimation, this misleading cluster effect can be avoided.

In order to evaluate this regularity the candidate cylinder is divided into equally shaped intervals or pill-boxes as shown in Figure 6. Instead of counting the total number of points in the alignment cylinder, the algorithm counts the number of boxes that are occupied by at least one point. We call them *occupied* boxes. In this way, the minimal NFA is attained when the points are perfectly distributed along the alignment. In addition, a concentrated cluster in the alignment has no more influence on the alignment detection than a single point in the same position.

A candidate alignment will be validated if the number of occupied boxes is statistically significant given the local point density. A set C of different values are tried for the number of boxes c into which the cylinder is divided. C can be roughly estimated as $C = \sqrt[3]{N}$ noticing that in a cube with N evenly distributed points, the longest alignments will have $\sqrt[3]{N}$ points.

2.4 Candidate Validation

The validation is done using the a contrario framework. The a contrario methodology consists in assuming a null hypothesis H_0 for the data, where no detections should occur, and finding violations of this hypothesis (i.e. events that could hardly happen under the hypothesized model). In the case of

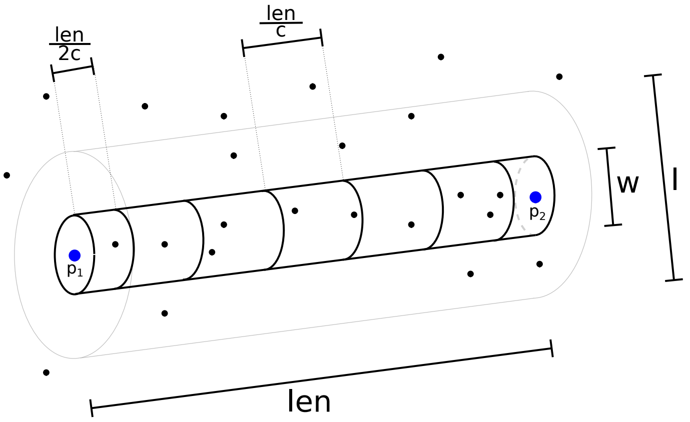

Figure 6: A schematic representation of the division of a candidate alignment cylinder into c pill-boxes. Two half boxes are placed at the extremes of the cylinder, and only the full boxes in the middle are count. This is because the two points forming the alignment must not be counted in the statistical test since they are the ones that define the test.

this algorithm, we consider for the null hypothesis H_0 that the points are independent and uniformly distributed in the domain.

2.4.1 Number of Tests

For each pair of points in the domain a set of $W \times L \times C$ candidate alignments are considered where:

- **W** is the number of different diameters considered for the alignment cylinder,
- **L** is the number of different diameters considered for the local cylinder,
- **C** is the number of different values considered for the number of boxes into which the alignment cylinder is divided.

A candidate alignment is determined by a pair of points. Then, with N points in the domain, we will need to evaluate N_{tests} candidate alignments defined as

$$N_{tests} = W \cdot L \cdot C \cdot \frac{N(N-1)}{2} \tag{3}$$

2.4.2 Non-Accidentalness / Number of False Alarms

A candidate alignment will be accepted based on two principles:

a) a good candidate should be non-accidental,

b) any equivalent or better candidate should be kept as well.

We will denote by $b(r, c, \mathbf{x})$ the observed number of occupied boxes in the candidate alignment cylinder r when divided into c boxes ($\mathbf{x}$ is the actual set of N points in the domain). Let $b(r, c, \mathbf{X})$ be the expected number of occupied boxes under the null hypothesis ($\mathbf{X}$ denotes a random set of N points following H_0).

The degree of non-accidentalness of a candidate alignment cylinder r can therefore be measured by how small the probability $\mathbb{P}\Big[b(r, c, \mathbf{X}) \geq b(r, c, \mathbf{x})\Big]$ is. In the same vein, a candidate r' will be considered at least as good as r given the observation $\mathbf{x}$, if $\mathbb{P}\Big[b(r',c,\mathbf{X}) \geq b(r',c,\mathbf{x})\Big] \leq \mathbb{P}\Big[b(r,c,\mathbf{X}) \geq b(r,c,\mathbf{x})\Big]$.

Given that N_{tests} candidates will be tested, the expected number of cylinders which are as good as r under H_0 is less than

$$\mathrm{NFA}(r,c,\mathbf{x}) = N_{tests} \cdot \mathbb{P}\Big[b(r,c,\mathbf{X}) \geq b(r,c,\mathbf{x})\Big]. \tag{4}$$

This fundamental quantity of the a contrario methodology is denoted the Number of False Alarms (NFA). It will be interpreted as a bound of the expected number of candidate cylinders containing enough *points* to be as rare as r under H_0. When the NFA associated with a candidate cylinder is large, this means that such an event is to be expected under the a contrario model and therefore is not relevant. On the other hand, when the NFA is small, the event is rare and probably meaningful. A rarity threshold ε must nevertheless be fixed for each application. Candidate cylinders with $\mathrm{NFA}(r,c,\mathbf{x}) \leq \varepsilon$ will be called *ε-meaningful cylinders* [3], constituting the detection result of the algorithm.

2.4.3 Calculating the NFA of a Candidate

We want to estimate the expected number of occupied boxes in the background model H_0. The probability of one point falling in one of the boxes is $p_0 = \frac{V_B}{V_L}$, where V_B and V_L are the volumes of the box and the local cylinder respectively. Then, the probability of having one box occupied by at least one of the $n^*(R,\mathbf{x})$ points (where $n^*(R,\mathbf{x})$ is the local number of points estimated in Equation (2)) can be computed as the complement to "none of the points fall in the box"

$$p_1(R,c) = 1 - (1-p_0)^{n^*(R,\mathbf{x})}. \tag{5}$$

As mentioned before, we will denote by $b(r,c,\mathbf{x})$ the observed number of occupied boxes in the candidate cylinder r when divided into c boxes. Finally, the probability of having at least $b(r,c,\mathbf{x})$ of the c boxes occupied is

$$\mathcal{B}\big(c, b(r,c,\mathbf{x}), p_1(R,c)\big), \tag{6}$$

where $\mathcal{B}(n,k,p)$ is the tail of the binomial distribution

$$\mathcal{B}(n,k,p) = \sum_{j=k}^{n} \binom{n}{j} p^j (1-p)^{n-j}. \tag{7}$$

The NFA of the alignment event is then

$$\begin{aligned}\mathrm{NFA}(r,R,c,\mathbf{x}) = {} & \\ & N_{tests} \cdot \mathbb{P}\Big[b(r,c,\mathbf{X}) \geq b(r,c,\mathbf{x}) \,\Big|\, n(R,\mathbf{X}) = n^*(R,\mathbf{x})\Big] \\ & = \frac{N(N-1)}{2} WLC \cdot \mathcal{B}\big(c, b(r,c,\mathbf{x}), p_1(R,c)\big).\end{aligned} \tag{8}$$

The algorithm computes the NFA for each candidate alignment and validates only those with $NFA \leq \varepsilon$. Usually the value of ε is set to 1, which means that, in average, only one false detection will occur in a random data set.

2.4.4 Influence of the Number of Sectors

As mentioned in Section 2.2, the idea of using sectors is basically to avoid deceptive detections in the border between regions of different density. Ideally, n^* (Equation (2)) should give an estimate as close as possible to the number of points of the most dense region. Then, the ideal number of sectors depends on the type of structure where one wants to avoid unwanted detections. If only corner like structures matter, four to eight sectors can be a good choice. But wedge-like and flat structures will require more sectors. As shown in Figures 14 and 15, four sectors are good to avoid detections on the corner-like structure but are not enough to avoid detections on a quasi planar region.

The conservative estimate of the local number of points n^* as computed in Equation (2) always increases with the number of sectors. The estimation of n^* influences the probability p_1 (Equation (5)) and hence the probability of having at least $b(r, c, \mathbf{x})$ of the c boxes occupied given by the binomial tail (Equations (6) and (7)). As n^* increases with the number of sectors, p_1 increases and the tail of the binomial also increases. Then, the computed NFA (Equation (8)) will increase with the number of sectors. Eventually, as sectors increase, the NFA of a significant alignment will surpass the ε limit loosing the alignment detection. The number of sectors must be chosen taking into account the trade-off between removing unwanted detections and the chance of loosing some true alignments.

Figure 7 presents an example with two alignments in uniform noise and the evolution of the NFA with respect to the number of sectors used in the algorithm.

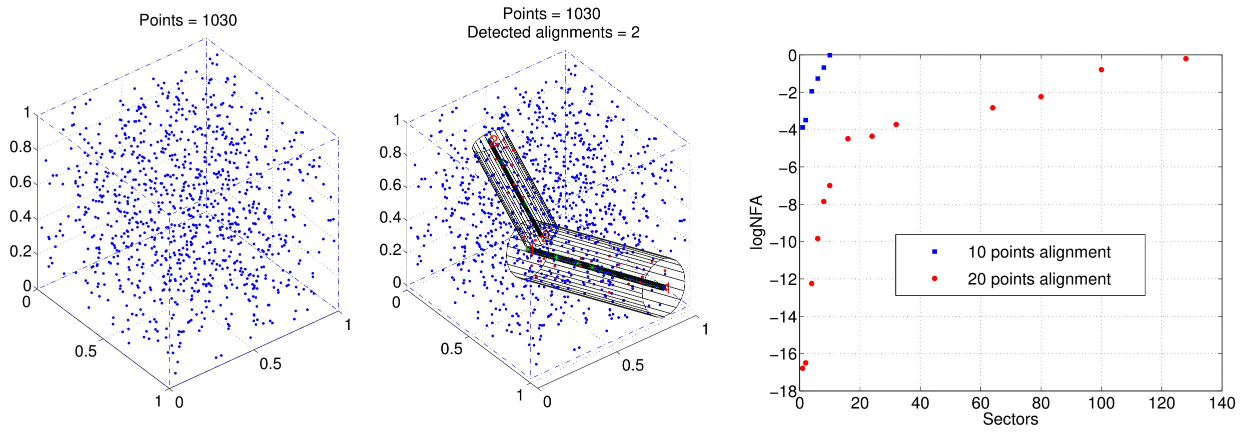

Figure 7: Influence of the number of sectors in the NFA of two alignments. Left: points , Center: the two detected alignments when one sector is used in the algorithm, Right: evolution of the $logNFA$ with the number of sectors. In this example, the detection of the less significant alignment is lost with more than ten sectors.

2.5 Redundancy Reduction

Given a very meaningful alignment, many smaller or larger cylinders overlapping the main alignment may also be meaningful as exemplified in Figure 8. This result is not desirable and redundancy should be eliminated in order to keep the most significant detections.

In this paper, redundancy reduction is done in the same way as in [5] following the masking principle: *A meaningful structure B will be said masked by a structure A if B is no longer meaningful when evaluated without counting its building elements belonging to A*. For the implementation, the alignments detected in the first stage of the algorithm (those with $NFA < \varepsilon$) are ordered by decreasing meaningfulness (increasing NFA). The first alignment in the ordered list is kept. The subsequent alignments are analyzed one by one and kept if it is checked that they are not masked by any one of the previously validated alignments. To check if an alignment B is masked by another

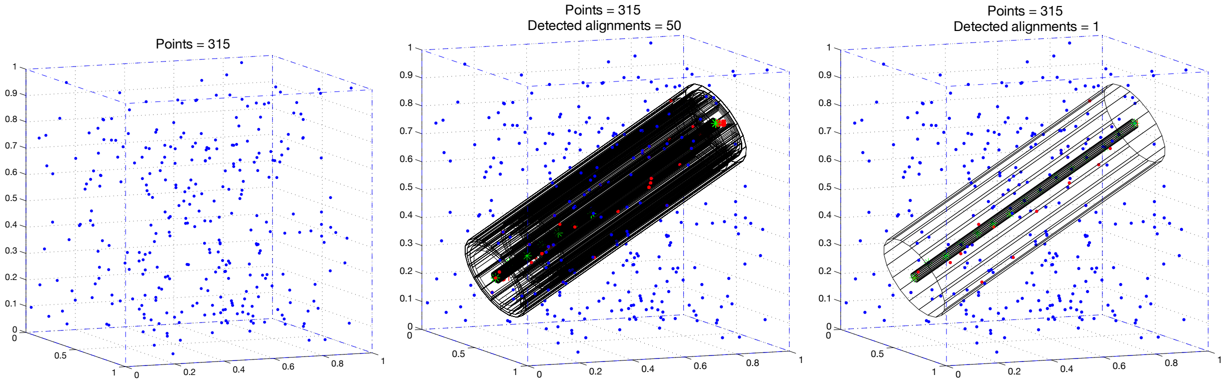

Figure 8: Left: An alignment in noise. Center: 50 most significant alignments. Right: The detected alignment after eliminating the redundant detections. If seen in color, green points are part of the alignment and red points are part of the local cylinder but not of the alignment cylinder.

alignment A, the algorithm recomputes the NFA of B using its same structure but removing the points belonging to A. If the new NFA is no longer significant ($NFA \nleq \varepsilon$), then B is said to be masked by A.

3 Algorithm

3.1 Main Body

Algorithm 1: Main body of the algorithm

```
input : A set $\mathbf{x}$ of $N$ points [$W = 8$, $L = 8$, $C$ (set of values for the number of boxes), $\varepsilon = 1$,
        $S$(number of sectors)]
output: A list $\mathbf{m}$ of non-redundant point alignments
Create a symmetric extension of $\mathbf{x}$;
$\mathbf{l} = detect\_alignments(\mathbf{x})$;
$\mathbf{m} = redundancy\_reduction(\mathbf{l})$;
```

Algorithm 1 presents the main body of the algorithm. In the first step, the points are symmetrically extended outside the domain. This extension avoids having border problems when estimating point densities near the frontier of the domain. Figure 9 shows an example of a symmetrization.

In the second step, the N_{tests} tests are performed and the significant candidate cylinders (those with $NFA < \varepsilon$) are kept. The output from the second step may have redundant detections. The third step of the algorithm reduces the redundancy by applying the masking principle (Section 2.5).

3.2 Point Alignment Detection

Algorithm 2 presents the pseudocode for the detection of the candidate alignments. Figure 10 aids with the notation used in the algorithm description. Following are some additional comments:

- Line 4: The searched alignments are expected to be elongated structures. In this implementation, a minimum $\frac{len}{w}$ ratio of 10 is used.

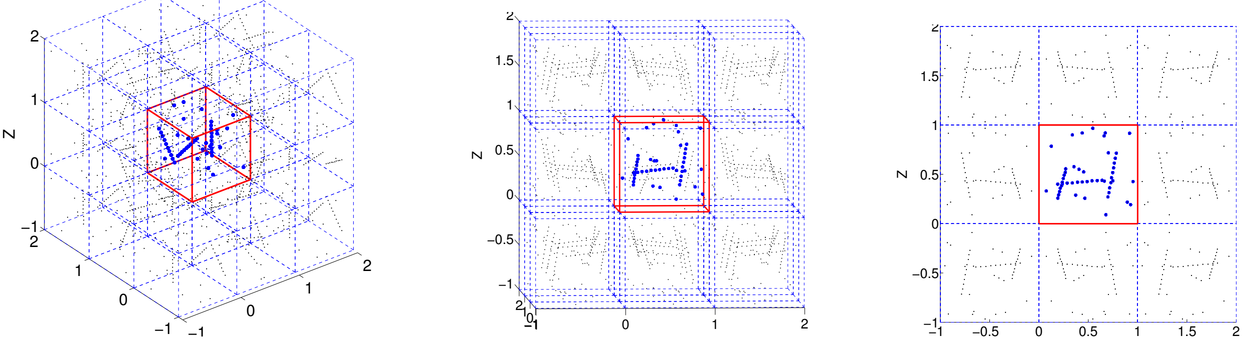

Figure 9: Three views of a symmetrization example. The N points in the domain are extended by symmetrization. This gives a set of $27 \times N$ points that allows to correctly estimate the densities near the frontier of the domain. If seen in color, blue points are the points in the domain and black points are the points extended by symmetrization outside the domain. The boundaries of the domain are outlined in red.

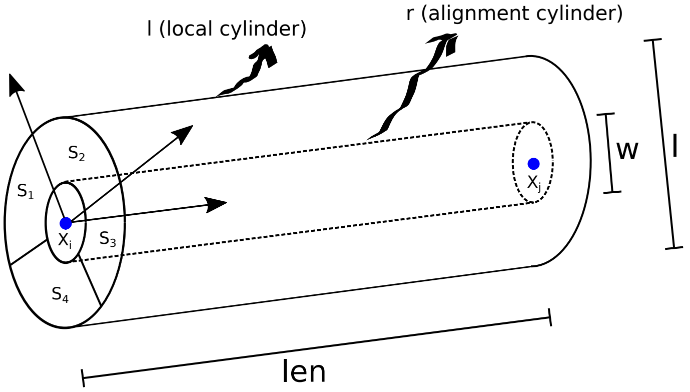

Figure 10: Schema of the variables used in Algorithm 2.

- Line 6: r is the alignment cylinder whose main axis is the segment $\overline{X_i X_j}$ and whose diameter is w.

- Line 7: The local cylinder R surrounds the alignment cylinder and is used to estimate the local background point density related to the tested alignment.

- Line 12: In practice, for each alignment cylinder, the set of values for the number of boxes that are considered is $C = \left[\frac{M}{2}, 2M\right]$ where M is the number of points in the alignment cylinder.

- Line 13: The cylinder is divided into c pill-boxes leaving two half-size boxes at each extreme of the alignment as show in Figure 10. There are two reasons for this. One is that the points defining the statistical test must not be counted in the test. The second reason is that if there are c points perfectly spaced inside the alignment, then these would fall exactly in the center of each of the c boxes.

- Line 14: The probability of one point of the local cylinder falling in one box is given by the ratio of volumes $p_0 = \frac{V_{box}}{V_r} = \frac{box \cdot w^2}{len \cdot l^2}$.

3.3 Redundancy Reduction

Algorithm 3 shows the pseudocode for the redundancy reduction procedure. The procedure deals with the list of alignment candidates and applies the masking principle. The pseudocode is the same

Algorithm 2: Point alignment detection

```
   input  : A set $\mathbf{X}$ of $N$ points $\{X_0, \dots, X_{N-1}\}$
            [$W = 8$, $L = 8$, $C$ (set of values for the number of boxes), $\varepsilon = 1$, $S$(number of
   sectors)]
   output: A list align of point alignments
 1 for $i = 0$ to $N - 1$ do
 2     for $j = i + 1$ to $N - 1$ do
 3         $len \leftarrow$ distance($X_i, X_j$) // length of the alignment cylinder
 4         $w \leftarrow len/10$ // initialize the diameter of the alignment cylinder
 5         for $iw = 1$ to $W$ do
 6             $r \leftarrow$ cyl($X_i, X_j, w$) // $r$ is the alignment cylinder with diameter $w$
 7             $l \leftarrow len/\sqrt{10}$ //initialize the diameter of the local cylinder $R$
 8             for $il = 1$ to $L$ do
 9                 Count $M$, the number of points in the alignment cylinder $r$
10                 Count $M_1 \cdots M_S$, the number of points in the sectors of the local cylinder
11                 Compute $n^*(R, X)$ // conservative number of local points, see Equation (2)
12                 for $c \in C$ do
13                     Divide $r$ into $c$ pill-boxes ($box = \frac{len}{c}$)
14                     $p_0 \leftarrow \frac{box \cdot w^2}{len \cdot l^2}$ //prob. of one point of R falling in a pill-box
15                     Compute $p_1(r, R, c)$ //see Equation (5)
16                     Count $b(r, R, c, X)$, the number of occupied boxes
17                     Compute $NFA(r, R, c, X)$ //see Equation (8)
18                     if $NFA(r, R, c, \mathbf{X}) \leq \varepsilon$ then
19                         align.append($r$)
20                     end
21                 end
22                 $l \leftarrow l/\sqrt[4]{2}$
23             end
24             $w \leftarrow w/\sqrt[4]{2}$
25         end
```

as in [5] for the 2D case.

3.4 Computational Complexity

In the first step (Algorithm 2), an exhaustive search is performed to detect the candidate alignment cylinders. A total of $N_{tests} = W \cdot L \cdot C \cdot \frac{N(N-1)}{2}$ is performed. The number of tests is then $O(\sqrt[3]{N}N^2)$ since the number of tested boxes can be estimated as $C = \sqrt[3]{N}$.

Each test on an alignment implies counting the points in the alignment cylinder and in the local cylinder (steps 9 and 10 in Algorithm 2).

In the simplest approach, where no special data structure nor nearest neighbor search is used, for an alignment defined by two points, all the other $N - 2$ points in the domain must be tested to see if they are part of the alignment or the immediate local vicinity. Moreover, since we also have symmetrization of the points outside the domain, for the alignments that lie near the boundary of the domain an extra scale factor must be considered (up to $27 \times N$). Finally, each test contributes an N factor and the total complexity of the first step is $O(\sqrt[3]{N}N^3)$

Algorithm 3: Redundancy reduction

```
input  : A list $\mathbf{l}$ of all significant alignments
output: A list $\mathbf{m}$ of maximally significant alignments
$m \leftarrow [\,]$; // initialize the output list
if $isempty(l)$ then
    return;
$l \leftarrow sort(l)$; //sort $l$ by ascending NFA (decreasing significance)
$m.append(l[0])$; // keep the most significant alignment from the sorted list
//check each alignment in $l$ against the already validated alignments in $m$
for $i = 1$ to $length(l) - 1$ do
    $B = l[i]$; //the alignment to check
    $masked \leftarrow False$
    for $j = 0$ to $length(m) - 1$ do
        $A = m[j]$; //an already validated alignment
        $X' = \{x | x \notin r_A\}$; //consider only the points that are not in the alignment cylinder of $A$
        //check the significance of $B$ excluding the points from the alignment $A$
        if $NFA(r_B, R_B, c_B, X') > \varepsilon$ then
            $masked \leftarrow True$;
            break;
    // if B is not masked by the already validated alignments, then add it to the output list
    if $masked == False$ then
        $m.append(B)$
```

A reduction of the N factor can be achieved if, for each alignment, only the neighboring points (in a certain radius depending on the maximum tested background cylinder) are considered. Although the speed up is not easy to determine analytically, this approach significantly reduces the factor in the complexity and enables, in practice, to run the algorithm on larger sets of points.

The source code provided with this article implements both approaches. For the nearest neighbors approach, the Fast Library for Approximate Nearest Neighbors (FLANN) [6] is used.

In the redundancy reduction step (Algorithm 3), a list of final detections is constructed and the alignments of the first step are tested to see if they are masked or not by the already selected final alignments. The complexity of each test is once again $O(N)$. The number of tests is much smaller than in the first step since each validated alignment from the first step is only checked for masking against the already selected as final detections.

4 Experiments

4.1 Experiments on Synthetic Data

Figure 11 shows the results of the algorithm when the points are uniformly distributed random points in the domain. This is the case where the points are in the H_0 hypothesis. Since the NFA of the alignments is limited to $\varepsilon = 1$, at most one false detection is expected to occur in this random datasets. In the examples of Figure 11 no detections were found.

Figure 12 shows the results of the algorithm when the points have a Gaussian distribution and the points are spread in the whole domain ($\mu = [0.5, 0.5, 0.5]^t$, $\sigma = [0.2, 0.2, 0.2]^t$). Although this

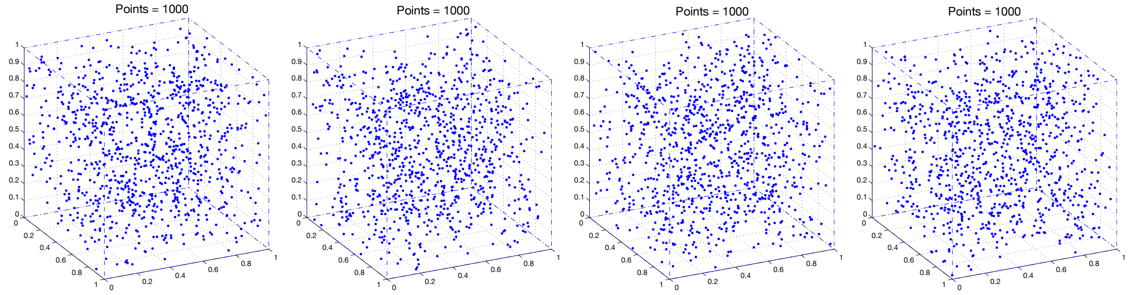

Figure 11: Uniformly distributed points.

distribution does not follow the a contrario model no alignment is detected.

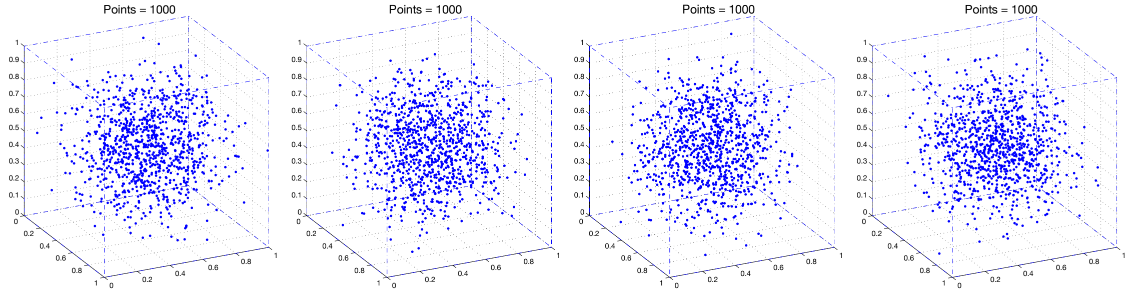

Figure 12: Gaussian distributed points.

Figures 13 and 14 present the case where the points are distributed on regions of different density. In this case alignments can be detected in the interface between regions if only one sector is considered in the local cylinder. As explained in Section 2.2, with only one sector, the background estimation can be underestimated since a big portion of the local cylinder intersects a low density region. The use of sectors allows the conservative estimation of the local density and helps overcome this problem.

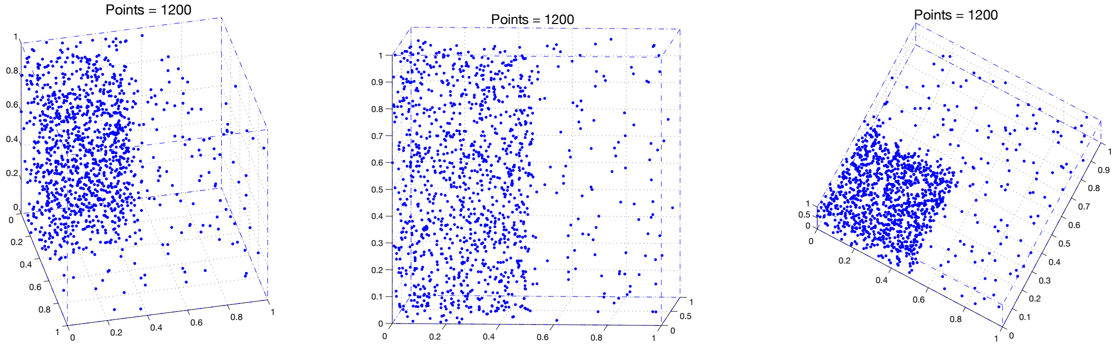

Figure 13: Points distributed in regions of different density. Three different views of the same set of points.

An extreme case for the local density estimation arises when the points group in nearly planar regions. In this case the distribution of the points is clearly far from the underlying hypothesis of the method. However, spurious detections in this kind of regions can be avoided using a larger number of sectors to estimate the local density. Figure 15 shows the results on several experiments with nearly planar regions of points. Figures 16, 17, 18 present the detection of true alignments for several point distributions.

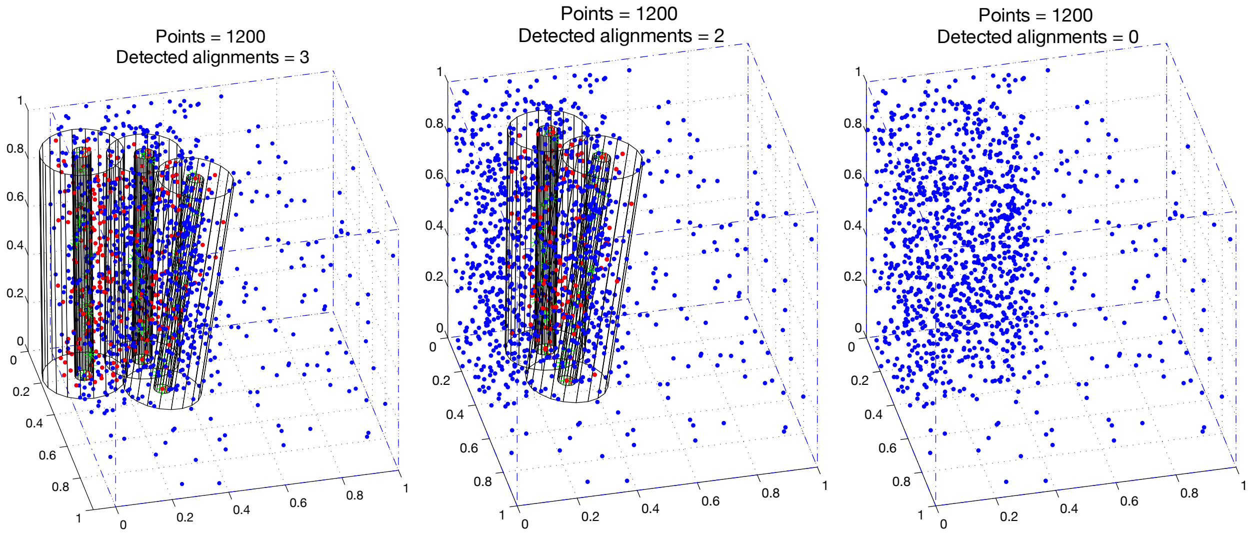

Figure 14: Alignments detected on the points of Figure 13 using different number of sectors. From left to right: detections computed using 1, 2 and 4 sectors. If seen in color, green points are part of the alignment and red points are part of the local cylinder but not of the alignment cylinder.

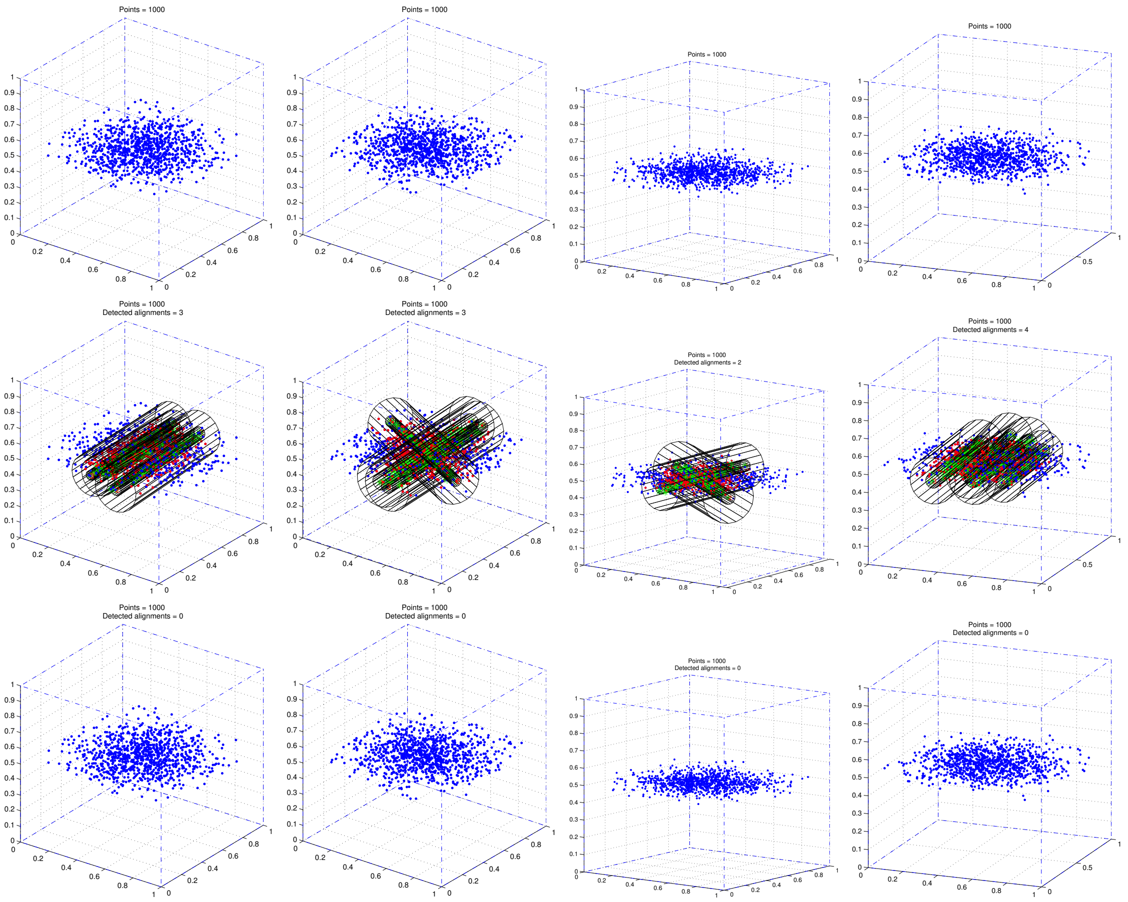

Figure 15: Points grouped in a nearly planar region. The first row shows four experiments where the points follow a Gaussian distribution with $\mu = [0.5, 0.5, 0.5]^t$ and $\sigma = [0.2, 0.2, 0.02]^t$. Second and third rows show the detections with 4 and 180 sectors respectively. If seen in color, green points are part of the alignment and red points are part of the local cylinder but not of the alignment cylinder.

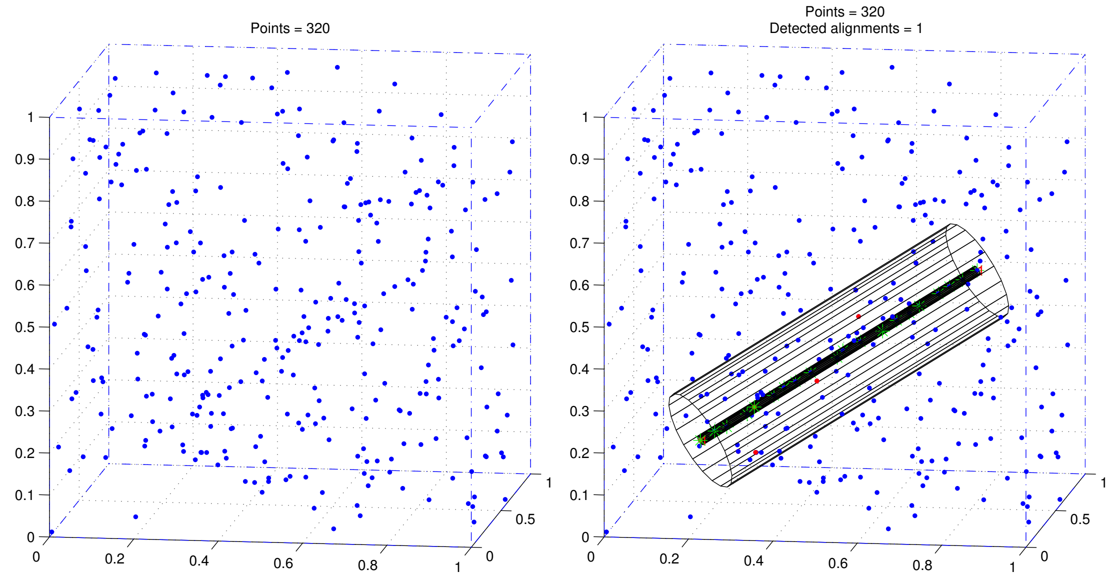

Figure 16: Detection of an alignment in noise. Left: an alignment in uniform noise, Right: detection.

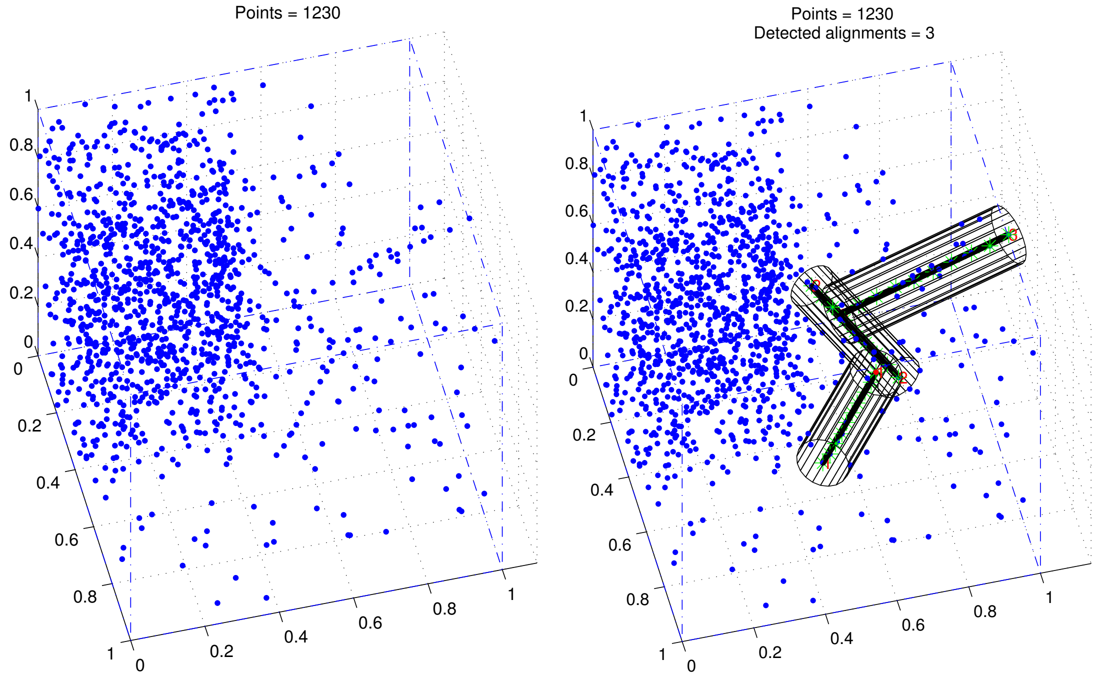

Figure 17: Detection of true alignments. Left: points uniformly distributed in two regions of different density and 3 true alignments, Right: detections with 8 sectors. If seen in color, green points are part of the alignment and red points are part of the local cylinder but not of the alignment cylinder.

4.2 Sensitivity of the Alignment Significance

For a particular alignment defined by its extreme points, the significance (in terms of the NFA of the alignment) was computed for different amounts of jitter on the interior points. Figure 19 and 20 show examples of different amount of off-axis and on-axis jitter on the interior points of an alignment.

For the off-axis jitter, starting from an ideal alignment, the interior points are allowed to move away from the axis a distance drawn from a normal distribution $N(0, (len \times jitter)^2)$.

For the on-axis jitter, each interior point is allowed to move along the axis following a uniform

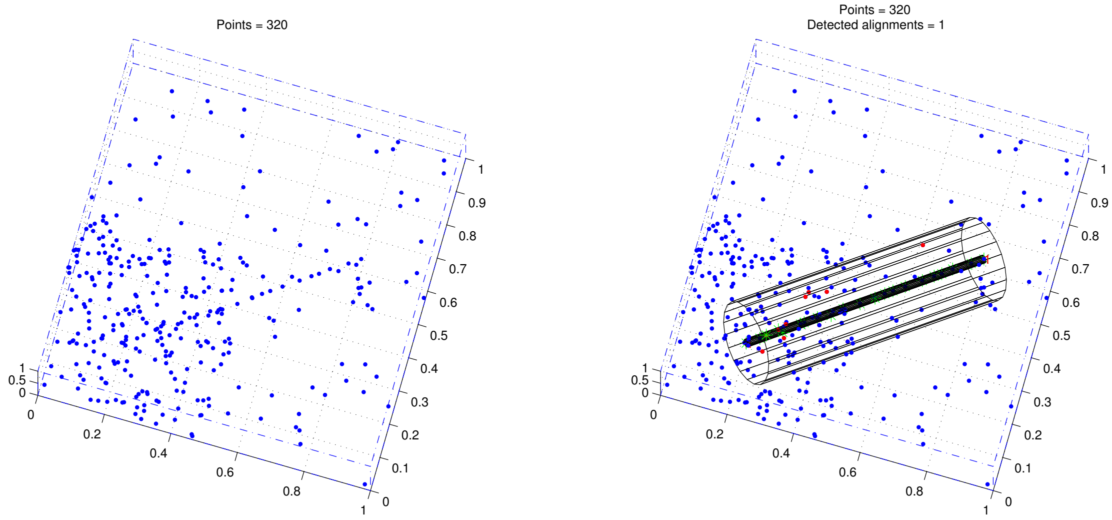

Figure 18: Detection of an alignment across two regions of different density. Left: uniform points in two regions of different density and one true alignment, Right: detection. If seen in color, green points are part of the alignment and red points are part of the local cylinder but not of the alignment cylinder.

distribution $\pm jitter \times ideal\_distance\_between\_points$ centered on the ideal position of the point (the position if the alignment were perfect and the space between points constant). Note that an on-axis jitter of 1 allows each point to move to any position between the ideal positions of its neighbors.

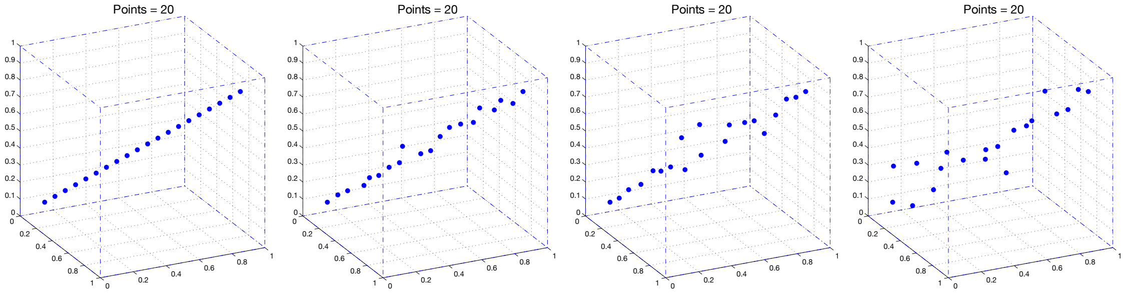

Figure 19: Off-axis jitter. From left to right: jitter=0 (perfect alignment), jitter=0.02, jitter=0.05, jitter=0.08.

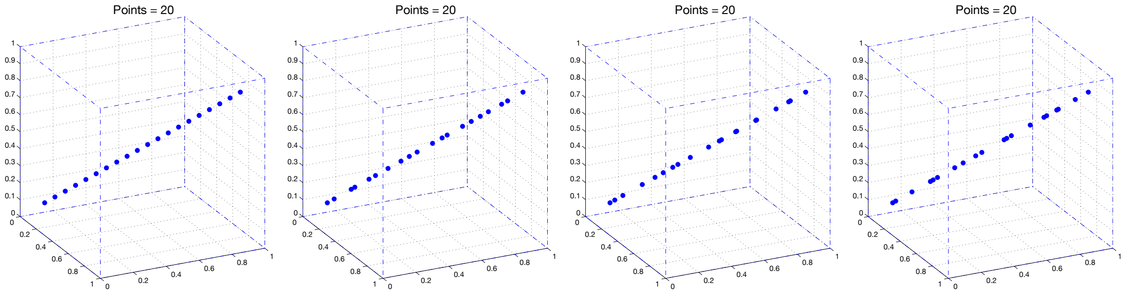

Figure 20: On-axis jitter. From left to right: jitter=0 (perfect alignment), jitter=0.5, jitter=0.9, jitter=1.2.

Figures 21 and 22 present the results (values and standard deviations) for the sensitivity with respect to the off-axis and on-axis jitter. An alignment of 10 points was considered and 20 experiments were performed for each value of jitter.

Off-axis jitter does not change the position of the points with respect to the boxes in the alignment so the number of boxes remains stable. In the on-axis jitter, when jitter is above 0.5, the points may move close to other points so the number of boxes in the alignment may drop.

The NFA value increases (significance decreases) with jitter in both cases but it is, as expected, influenced more by off-axis jitter where the points may move away from the ideal line.

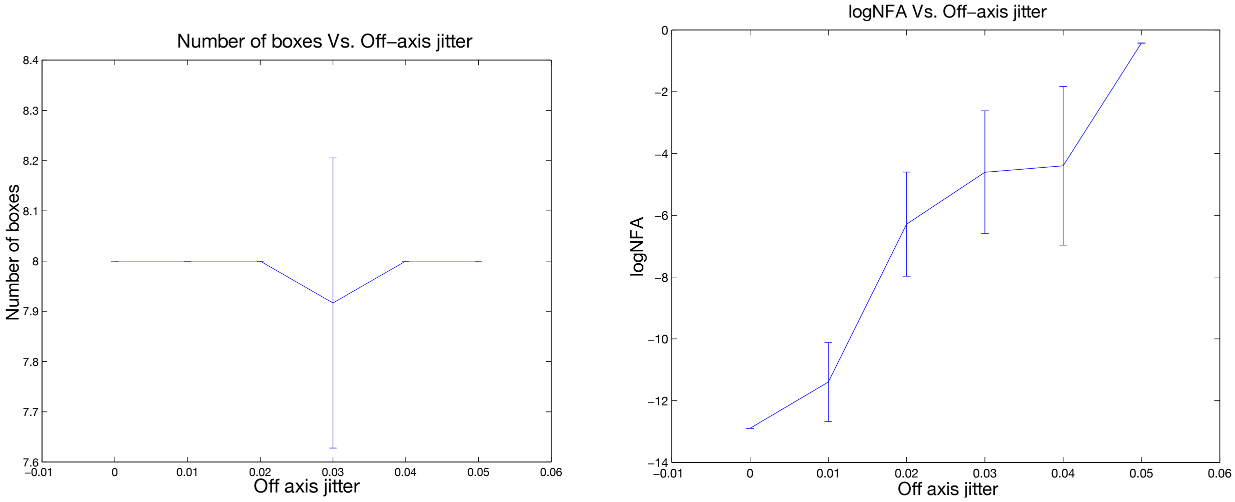

Figure 21: Off-axis jitter results.

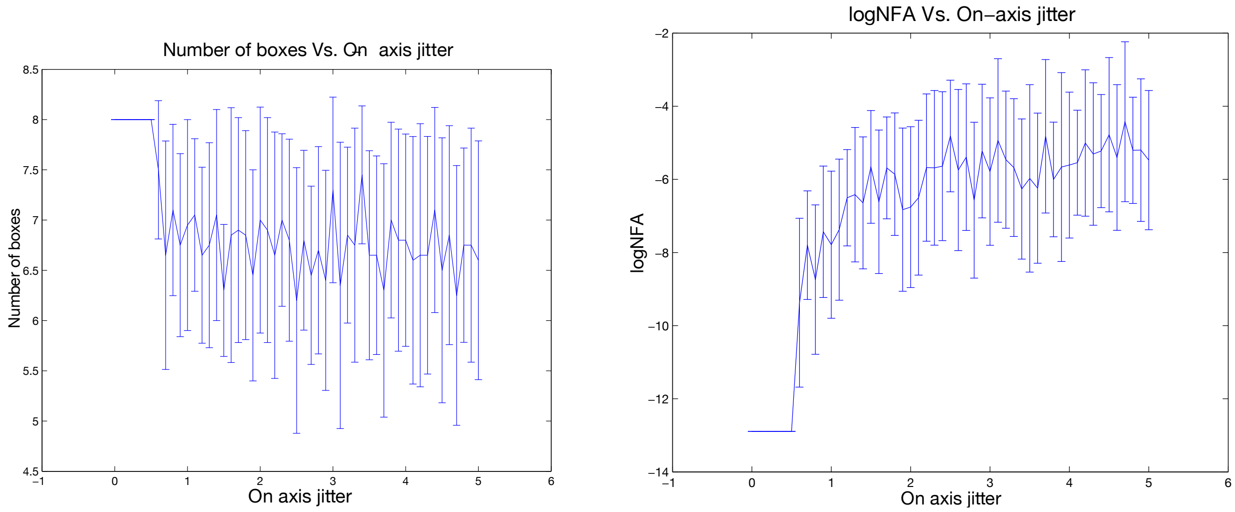

Figure 22: On-axis jitter results.

4.3 Experiments on Acquired Images

Figure 23 presents a point cloud acquired with a Kinect sensor. In the cloud there is a person holding linear objects. The original cloud with 59290 points is downsampled to 501 points in order to apply the alignment detector. Downsampling is performed in a voxel grid where all the points in a voxel are approximated with their centroid. The detected alignments are shown in Figure 24.

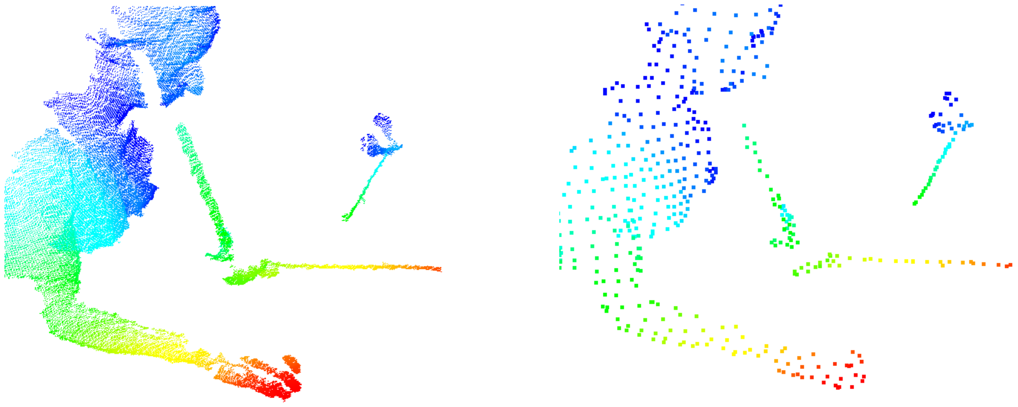

Figure 23: Point cloud acquired with a Kinect. Left: full point cloud of 59290 points. Right: Downsampled point cloud with 501 points.

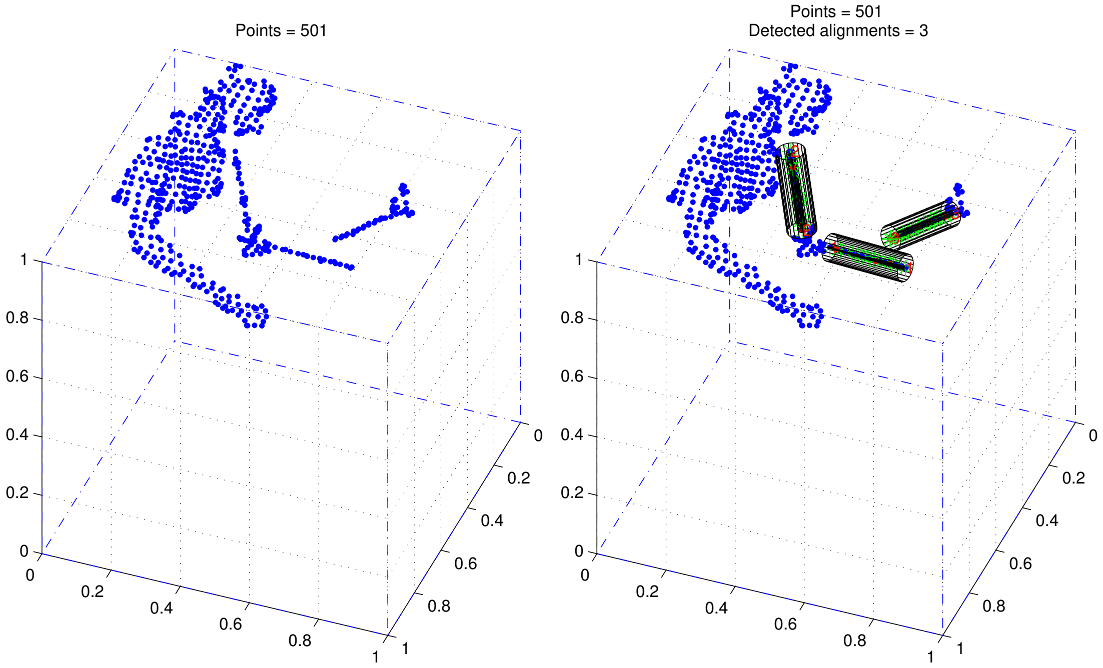

Figure 24: Alignments detected on data acquired with a Kinect sensor. The point cloud is the same as in Figure 23 right. If seen in color, green points are part of the alignment and red points are part of the local cylinder but not of the alignment cylinder.

5 Conclusion

In this work we have presented an alignment detector for 3D point clouds that extends the work done in 2D in [5]. Results on synthetic and acquired data show that the algorithm is effective to detect meaningful alignments giving in general few or none false alarms. The computational complexity of the algorithm constitutes its main drawback and limits the number of points of the cloud that can be processed in a reasonable time. Acceleration strategies are not addressed in this instance and will be the subject of future work.

Acknowledgments

This work was done under an I+D project funded by Comisión Sectorial de Investigación Científica (CSIC), Universidad de la República, Uruguay. CSIC and the Centre de Mathématiques et de Leurs Applications (CMLA - ENS Saclay, France) made possible a meeting of the group at the CMLA. Duke University hosted Gregory Randall during part of this work.

Image Credits

All images by the authors.

References

[1] M. K. Albert and D. D. Hoffman, *Genericity in spatial vision*, in Geometric Representations of Perceptual Phenomena: Arts. in Honor of Tarow Indow's 70th Birthday, D. Luce, K. Romney, D. Hoffman, and D'Zmura M., eds., Erlbaum, 1995, pp. 95–112.

[2] A. Desolneux, L. Moisan, and J.-M. Morel, *Meaningful alignments*, International Journal of Computer Vision, 40 (2000), pp. 7–23. https://doi.org/10.1023/A:1026593302236.

[3] A. Desolneux, L. Moisan, and J-M. Morel, *From Gestalt Theory to Image Analysis: A Probabilistic Approach*, vol. 34, Springer-Verlag New York, 2008. https://doi.org/10.1007/978-0-387-74378-3.

[4] J. Lezama, J-M. Morel, G. Randall, and R. Grompone Von Gioi, *A contrario 2D point alignment detection*, IEEE Transactions on Pattern Analysis and Machine Intelligence, 37 (2015), pp. 499–512. http://dx.doi.org/10.1109/TPAMI.2014.2345389.

[5] J. Lezama, G. Randall, J-M. Morel, and R. Grompone von Gioi, *An Unsupervised Point Alignment Detection Algorithm*, Image Processing On Line, 5 (2015), pp. 296–310. https://doi.org/10.5201/ipol.2015.126.

[6] M. Muja and D.G. Lowe, *Fast approximate nearest neighbors with automatic algorithm configuration*, in International Conference on Computer Vision Theory and Application (VISSAPP), INSTICC Press, 2009, pp. 331–340.

[7] A. K. Preiss, *A Theoretical and Computational Investigation into Aspects of Human Visual Perception: Proximity and Transformations in Pattern Detection and Discrimination*, PhD thesis, University of Adelaide, 2006. http://hdl.handle.net/2440/37820.

[8] S. P. Tripathy, A. J. Mussap, and H. B. Barlow, *Detecting collinear dots in noise*, Vision Research, 39 (1999), pp. 4161–4171. http://dx.doi.org/10.1016/S0042-6989(99)00125-X.

[9] W. R. Uttal, *The effect of deviations from linearity on the detection of dotted line patterns.*, Vision Research, 13 (1973), pp. 2155–2163.

[10] J. Wagemans, *Perceptual use of nonaccidental properties*, Canadian Journal of Psychology, 46 (1992), pp. 236–279. http://dx.doi.org/10.1037/h0084323.

[11] A. P. Witkin and J. M. Tenenbaum, *On the role of structure in vision*, in Human and Machine Vision, J. Beck, B. Hope, and A. Rosenfeld, eds., Academic Press, 1983, pp. 481–543. http://dx.doi.org/10.1016/B978-0-12-084320-6.50022-0.