Published in Image Processing On Line on 2021–07–26.
Submitted on 2021–04–04, accepted on 2021–06–23.
ISSN 2105–1232 © 2021 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.2021.342

Ground Visibility in Satellite Optical Time Series Based on A Contrario Local Image Matching

Rafael Grompone von Gioi[1], Charles Hessel[1,2], Tristan Dagobert[1], Jean-Michel Morel[1], Carlo de Franchis[1,2]

[1] Université Paris-Saclay, ENS Paris-Saclay, CNRS, Centre Borelli, F-91190 Gif-sur-Yvette, France
[2] Kayrros SAS

Communicated by Pablo Musé *Demo edited by* Charles Hessel and Rafael Grompone von Gioi

Abstract

Assessing ground visibility is a crucial step in automatic satellite image analysis. Some Earth observation satellites are provided with spectral bands specially designed for cloud detection. An alternative approach is to detect ground visibility by comparing locally the images in a temporal series: matching regions are necessarily cloud free. Indeed, the ground has persistent patterns that can be observed repetitively in the time series, while clouds change shape constantly. We describe here a ground visibility detection algorithm based on an a contrario local image matching method, coupled with an efficient greedy algorithm.

Source Code

The reviewed, reference source code for this algorithm is available from the web page of this article[1]. The algorithm is implemented in ANSI C language. Compilation and usage instruction are included in the `README.txt` file of the archive.

Keywords: ground visibility detection; cloud detection; satellite time series; a contrario framework

1 Introduction

Assessing ground visibility is an important step in optical satellite images analysis. Indeed, the presence of clouds and haze concealing the surface of the Earth often causes detection errors in automatic image analysis. This task is usually addressed as a cloud detection problem, where the image pixels are classified into classes such as ground, clouds, cirrus, snow, haze, cloud shadows, etc. [5, 27].

Satellite cloud detection often exploits spectral bands specifically designed for cloud detection [20, 41, 21, 35, 5, 38, 19]. Alternatively, the inter-band delay in push-broom satellites allows cloud

[1]https://doi.org/10.5201/ipol.2021.342

detection by parallax analysis of the color bands [36, 29, 28, 40, 37, 11]. The past decade has seen the launch of constellations with numerous satellites, considerably reducing the revisit time. Very often the revisit is repeatedly made from the same attitude. This opens the way to reliable change detection and therefore to automatic Earth monitoring, provided that atmospheric changes have been discarded [2]. Unfortunately, for reasons of size or cost, recent Earth observation satellite constellations like PlanetScope lack the specially designed spectral bands and use a frame camera, precluding the classic cloud detection approaches. Fortunately the short revisit time of these satellites can, to some extent, compensate the lack of physical or geometric cues about cloud cover.

Instead of detecting clouds, an alternative approach is to detect ground visibility, that is, the parts of each image where the ground is visible [7, 6, 16]. This can be done by comparing the corresponding parts of a time series and selecting matching regions. The rationale is that the ground has persistent patterns that are observed repetitively in the time series, while clouds are changing phenomena appearing differently each time. This approach is based on the following three hypotheses:

1. the images in the time series are well registered;
2. the region of interest changes slowly in the observed time series;
3. each part of the region of interest is visible at least twice in the time series.

Under these hypotheses, any pattern observed more than once at the same position must correspond to a visible region of the ground. Of course, the method cannot be applied when these conditions are not satisfied. For example, the sea is continuously changing and does not satisfy the second hypothesis. It will therefore always be marked as not visible and should be detected separately, for example by the NDWI index.

The method works as follows. Given a set of N registered images, the ground visibility masks for the N images are all initialized as *not visible*; then the $\frac{N(N-1)}{2}$ pairs of images are compared and when a local match is found, the corresponding parts are marked as *visible* in both ground visibility masks.

There is a large literature on how to perform local image comparison [26, 24, 3, 1]. For our problem, three properties are desirable: robustness to contrast changes (as satellite image dynamics may change), a good control of false detections, and a low computational cost. The last two requirements are imposed by the considerable size of data. There are many fast local image comparison methods, and most achieve contrast invariance by relying on the image gradient orientation, like the celebrated SIFT algorithm [26, 33]. An effective ground visibility detection algorithm based on SIFT image descriptors was presented in [7, 6].

Concerning the control of false detections, the *a contrario* statistical approach [9, 10] was introduced to provide a well founded mechanism for setting detection thresholds while producing few false detections. This framework has been successfully applied to local image comparison [4, 32, 31, 18, 34], again based on the image gradient orientation for contrast change robustness. These methods work by comparing the gradient orientation in corresponding image patches, see Figure 1. While effective, there are three limitations when applied to our current problem. First, the patch shape and size need to be specified, and the optimal choice may be content dependent. Second, an exhaustive local patch comparison may be time-consuming. Third, a patch based decision implies a loss of precision on the boundary of visible regions.

The method described here uses an a contrario local image matching approach that can be applied to connected regions of arbitrary shapes. A greedy algorithm provides a fast way of building candidate matching regions. This procedure, coupled with a final morphological area filter, results in an effective ground visibility detection algorithm for satellite time series. This method was first introduced in [16].

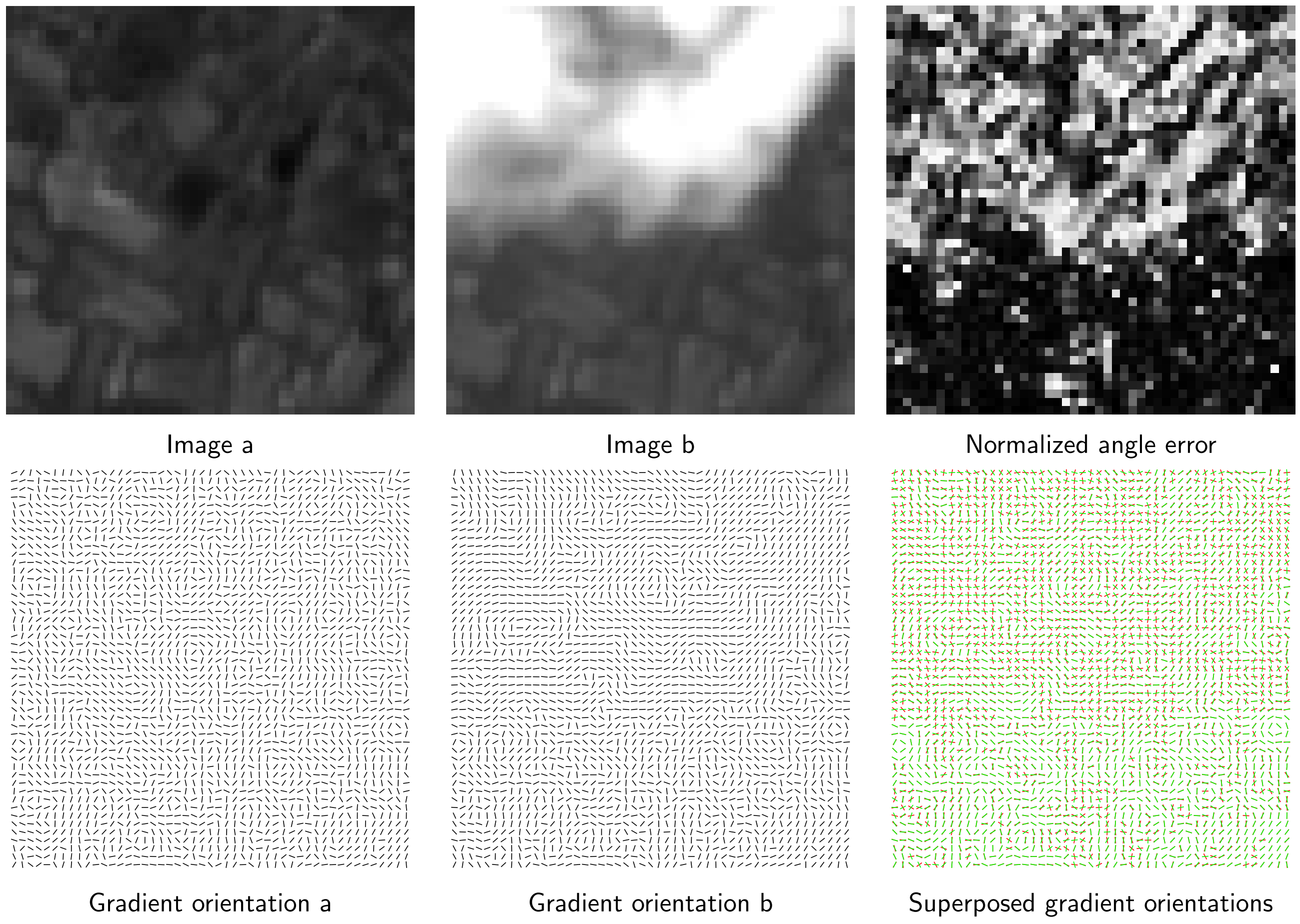


Figure 1: Two Sentinel-2 images, **a** and **b**, to be compared. Below each of the two images is represented the gradient orientation field. One can see how these orientations roughly agree in the visible zone while mainly differing where a cloud is present. In the normalized gradient orientation error between the two images, black corresponds to zero error (same gradient orientation in both images) and white corresponds to the worst error (the gradient orientations have opposite direction).

This paper is organized as follows. Section 2 introduces briefly the a contrario approach which is then applied in Section 3 for local image matching. Then Section 4 details the ground visibility detection algorithm and its computational complexity is analyzed in Section 5. The method is illustrated with some experiments in Section 6. Section 7 presents the conclusions.

2 The A Contrario Approach

The a contrario theory [9, 10] is a statistical framework used to set detection thresholds automatically in order to control the number of false detections. It is based on the non-accidentalness principle [39, 25] which informally states that an observed structure is meaningful only when the relation between its parts is too regular to be the result of an accidental arrangements of independent parts. In the words of D. Lowe, "we need to determine the probability that each relation in the image could have arisen by accident, $P(a)$. Naturally, the smaller that this value is, the more likely the relation is to have a causal interpretation" [25, p. 39].

A stochastic background model $\mathcal{H}_0$ needs to be defined, where the structure of interest is not present and can only arise as an accidental arrangement. For example, many image matching methods (including the one described here) are based on the orientation of the image gradient [26, 4, 32, 31, 18, 34]. Some geometrical feature detection methods such as line segments or elliptical arcs [15, 30] are also based on the image gradient. In such cases, the background model $\mathcal{H}_0$ assumes that the gradient orientations at each pixel are independent random variables, uniformly distributed in $[-\pi, \pi)$. Under this background model, a region of the image where the gradient orientation follows a regular structure would be a rare accident and is detected as such.

We also need to define a family of events of interest T. For feature detection the family of events is the set of all the geometrical events considered, i.e., all the line segments, elliptical arcs, etc., considered in the image domain. Then, we need to assess the accidentalness of a candidate feature. For example, if a line segment is present in an image, the gradient orientation at the corresponding position would be orthogonal to the line segment. Given a candidate line segment, one measures how well the image gradient corresponds to the candidate event, and evaluates the probability of observing such a good agreement by chance in the background model $\mathcal{H}_0$. A rough agreement could arise just by chance and thus does not correspond to an interesting event; conversely, a very good agreement would be rare and suggests the presence of a structure instead of just a lucky accident. In other words, when this probability is small enough, there exists evidence to reject the null hypothesis and declare the event meaningful. However, one needs to consider that multiple candidates are tested. If 100 tests were performed, for example, it would not be surprising to observe among them one event that appears with probability 0.01 under random conditions. The number of tests N_T needs to be included as a correction term, as it is done in the statistical multiple hypothesis testing framework [14].

Following the a contrario methodology [9, 10], we define the *Number of False Alarms* (NFA) of an event e observed up to an error $k(e)$ as

$$\text{NFA}(e) = N_T \cdot \mathbb{P}\Big[K_{\mathcal{H}_0}(e) \leq k(e)\Big], \tag{1}$$

where the right hand term is the probability of obtaining an error $K_{\mathcal{H}_0}(e)$ smaller or equal to the observed one $k(e)$ in the background model $\mathcal{H}_0$. The smaller the NFA, the more unlikely the event e is to be observed by chance in the background model $\mathcal{H}_0$; thus, the more meaningful. The a contrario approach prescribes accepting as valid detections the candidates with NFA $< \varepsilon$ for a predefined value ε. It can be shown [9, 10] that under $\mathcal{H}_0$, the expected number of tests with NFA $< \varepsilon$ is bounded by ε. As a result, ε gives an a priori estimate of the mean number of false detections under $\mathcal{H}_0$. In many practical applications, including the present one, the value $\varepsilon = 1$ is adopted. Indeed, it allows for less than one false detection on an image or on a set of images, which is usually quite tolerable.

3 A Contrario Local Image Matching

Given two images u and v defined on the same domain Ω (of size $X \times Y$), and a set of pixels R, we would like to know whether both images are similar in the region R. To this aim, we will use the distance

$$s_{u,v}(R) = \sum_{\omega \in R} \frac{|\text{Angle}(\nabla u(\omega), \nabla v(\omega))|}{\pi}, \tag{2}$$

namely the sum of all normalized gradient angle errors in R [31, 18]. This distance can take values between zero and $|R|$ (the size of the region). Zero corresponds to a perfect match, while $|R|$ corresponds to the case where the gradient orientation in the two images are opposite at every pixel of R. Using only the image gradient *orientation* renders this distance invariant to illumination changes. As defined, this distance is valid only for single-channel images; the definition could be adapted for handling multi-channel images.

For a given region R, we need to decide whether the distance $s_{u,v}(R)$ is small enough, indicating whether the corresponding parts of the images are similar or not. An a contrario formulation is used for this. As explained before, a natural background model $\mathcal{H}_0$ is that the gradient orientations at

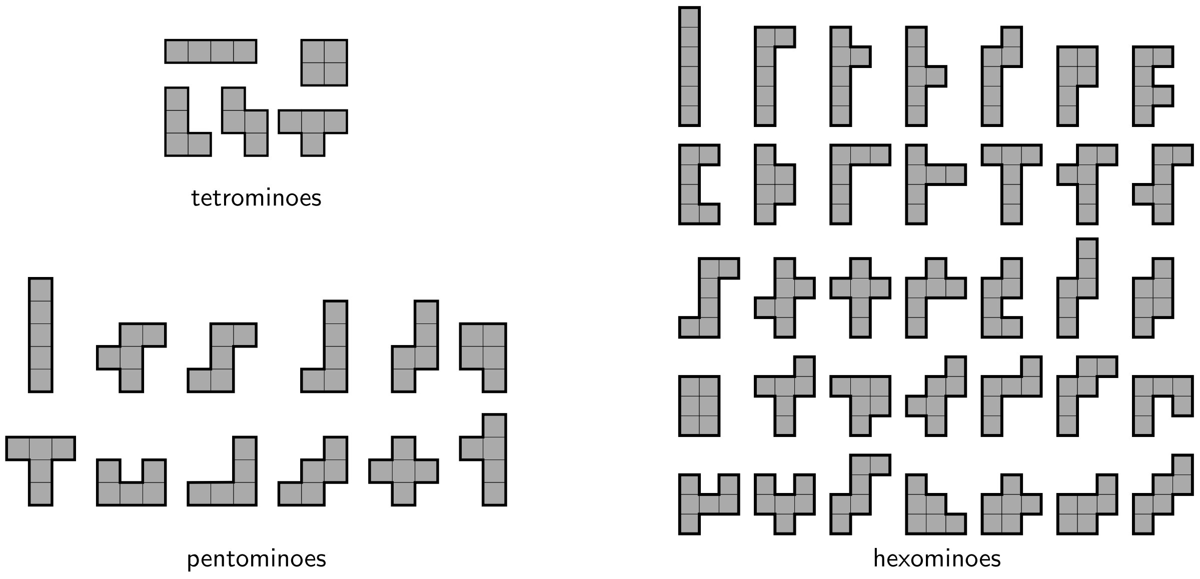


Figure 2: Polyominoes of four (tetrominoes), five (pentominoes) and six (hexominoes) elements. When rotations and reflections are not considered, there are 5 tetrominoes, 12 pentominoes and 35 hexominoes, as shown here. When rotations and reflections are also considered distinct, there are 19 tetrominoes, 63 pentominoes and 216 different hexominoes. The formula in Equation (5) gives the correct order of magnitude of these numbers, as $0.316915\frac{4.062570^4}{4} \approx 21.6$ for tetrominoes, $0.316915\frac{4.062570^5}{5} \approx 70.1$ for pentominoes, and $0.316915\frac{4.062570^6}{6} \approx 237.4$ for hexominoes.

each pixel are independent random variables, uniformly distributed in $[-\pi, \pi)$. (This will happen, for example, if one of the images contains a cloud covering the region.) Following the a contrario framework, we will define the NFA associated to a candidate region match as

$$\mathrm{NFA}(u, v, R) = N_T \cdot \mathbb{P}\Big[S_{\mathcal{H}_0}(R) \leq s_{u,v}(R)\Big], \tag{3}$$

where $S_{\mathcal{H}_0}(R)$ is a random variable corresponding to the distance $s_{U,V}(R)$ for random images U and V whose gradient orientation follows $\mathcal{H}_0$. But under $\mathcal{H}_0$ the gradient orientations are uniformly distributed in all directions, which implies that the normalized angle errors at each pixel are independent random variables following a uniform distribution in $[0, 1]$. As a result, $S_{\mathcal{H}_0}(R)$ corresponds to the sum of $|R|$ independent and uniformly distributed random variables taking values in $[0, 1]$. Thus, $S_{\mathcal{H}_0}(R)$ follows the Irwin-Hall distribution [23] and for a given s, with $0 \leq s \leq |R|$, we obtain

$$\mathbb{P}\Big[S_{\mathcal{H}_0}(R) \leq s\Big] = \frac{1}{|R|!} \sum_{i=0}^{\lfloor s \rfloor} (-1)^i \binom{|R|}{i} (s - i)^{|R|}, \tag{4}$$

where $\lfloor s \rfloor$ is the integer part of s and $\binom{n}{i}$ is the binomial coefficient.

To complete the formulation we still need to specify the family of tests T and the corresponding number of tests N_T. Instead of using rectangular patches as in [31, 18, 34], here we are interested in connected regions of any shape. We consider *4-connectivity*, in which a pixel (x, y) is connected to the pixels at coordinates $(x \pm 1,\ y)$ and $(x,\ y \pm 1)$. Groups of pixels connected under 4-connectivity correspond to the figures called *polyominoes* [13, 12], see Figure 2. The exact number b_n of different polyomino configurations of given size n is not known in general, but there are good approximations of this number [22]. In our case, it is enough to use an estimate of the order of magnitude, so the approximate formula given in [22] is sufficient for our needs. It reads

$$b_n \approx \alpha \frac{\beta^n}{n}, \tag{5}$$

where $\alpha \approx 0.316915$ and $\beta \approx 4.062570$. We need to consider that each particular polyomino may be placed at any position in the image, thus we need to multiply b_n by XY to consider all possible placements (this estimation is not exact as it counts some polyominoes extending outside the image domain; nevertheless, it gives the right order of magnitude). Also, we consider connected regions of different sizes, from 1 to XY, the latter being the case where all the pixels are part of one polyomino. This calculation is not exact due to the restrictions imposed by the total size of the image; e.g., among the b_{XY} polyominoes of size XY, only one is actually possible, the one using all the pixels in the images; again, this rough estimation is useful for our case. The number of tolerated false detections ε is divided into XY categories, and to each one we allow to produce $\frac{\varepsilon}{XY}$ false detections. In this way, the total number of false detections, including the ones obtained for polyomino regions of size from 1 to XY, will add up to $\sum_{n=1}^{XY} \frac{\varepsilon}{XY} = \varepsilon$, as desired. Thus, for a given region size, the test becomes

$$XY\, b_{|R|} \cdot \mathbb{P}\Big[S_{\mathcal{H}_0}(R) \le s\Big] < \frac{\varepsilon}{XY}. \tag{6}$$

This is equivalent to setting the number of test to $X^2Y^2b_{|R|}$ and comparing directly to ε. Finally, in our setting, if there are N images in the time series, we will try to find matches between all the possible pairs. Thus, an extra factor $\frac{N(N-1)}{2}$ needs to be included. The final number of tests is then

$$N_T = \frac{N(N-1)}{2} X^2Y^2 b_{|R|}. \tag{7}$$

All in all, we define the NFA of a candidate match by

$$\mathrm{NFA}(u, v, R) = \frac{N(N-1)}{2} X^2Y^2 \cdot b_{|R|} \cdot \mathbb{P}\Big[S_{\mathcal{H}_0}(R) \le s_{u,v}(R)\Big], \tag{8}$$

where $s_{u,v}(R)$ is the distance defined in Equation (2). We set $\varepsilon = 1$; candidates with $\mathrm{NFA}(u, v, R) < 1$ are considered detected local matches.

4 The Ground Visibility Detection Algorithm

The former section described the a contrario formulation to decide whether a couple of images are similar or not in a given region. But trying every possible 4-connected region in the image domain is impossible as it would take too much time. Instead of testing all possible regions, a simple heuristic proposes a reduced number of regions to be evaluated, performing the comparison of all pairs of images in a reduced time. Algorithm 1 provides a full overview and the steps are discussed in what follows. Figure 3 illustrates the procedure on two images.

The input is a time series composed of N registered images $u_1, \ldots, u_N$ defined on the same domain Ω of size $X \times Y$. In the current formulation, the method only works on single-channel images. A priori, it can be applied to any satellite band, detecting visible or stable structures in the corresponding band. The method could be extended to multi-channel images by adapting the distance of Equation (2) or by combining the visibility masks resulting of the different channels. Also, the algorithm could be easily adapted to work on images defined on different domains, provided that the common parts are well-registered; using a common rectangular domain Ω just makes the formulation simpler. Exploring these extensions is left for future work.

The algorithm depends on two parameters, ρ and λ, to be detailed below. The output is a set of N ground visibility masks $V_1, \ldots, V_N$, defined on the same domain Ω; at a given pixel ω, the mask $V_k(\omega)$ can take two values, **visible** or **not visible**, indicating the ground visibility of pixel ω of the image u_k.

Algorithm 1: Ground visibility detection algorithm

```
input     : $N$ images $u_1, \dots, u_N$ defined on $\Omega$ $(X \times Y)$
parameter: region growing angular tolerance $\rho = \frac{1}{5}$
parameter: grain filter region size $\lambda$
output    : $N$ ground visibility masks $V_1, \dots, V_N$
1  $\alpha \leftarrow 0.316915$                                   polyominoes count constants $\alpha$ and $\beta$
2  $\beta \leftarrow 4.062570$
3  $V_k(\Omega) \leftarrow$ not visible                            initialize all masks as not visible
4  $g_k \leftarrow \mathrm{GradientAngle}(u_k)$                    compute gradient orientation, see Algorithm 2
5  for $(a,b) \in \mathrm{Pairs}(N)$ do                            loop on every pair of images
6      for $\omega \in \Omega$ do                                  compute normalized angle error $\xi$ between images a and b,
7          $\xi(\omega) \leftarrow \frac{|\mathrm{Angle}(g_a(\omega),\, g_b(\omega))|}{\pi}$    in pixels where $g_a$ or $g_b$ are not defined, $\xi$ is set to 1
8      for $\omega \in \Omega$ do                                  loop on seed pixels
9          if $\xi(\omega) \le \rho$ then                          start on seeds with $\rho$ condition
10             $R,\ s \leftarrow \mathrm{region}(\omega, \xi, \rho)$   set $R$ of $\rho$-neighbors and its error sum $s$, see Algorithm 3
11             $\mathrm{NFA} \leftarrow \frac{N(N-1)}{2} X^2 Y^2 \alpha \frac{\beta^{|R|}}{|R|} \frac{s^{|R|}}{|R|!}$   NFA upper bound given by Equation (11)
12             if NFA $< 1$ then                                   meaningful match found
13                 $V_a(R) \leftarrow$ visible                     mark the region $R$ as visible in mask a
14                 $V_b(R) \leftarrow$ visible                     mark the region $R$ as visible in mask b
15 $V_k \leftarrow \mathrm{GrainFilter}(V_k, \lambda)$   fill visible and not visible holes of size smaller than $\lambda$, see Algorithm 4
```

The first two steps set the two constants α and β described in Section 3, which are used to count polyominoes. Then, step 3 initializes all the pixels of all the ground visibility masks to **not visible**.

Step 4 of the main algorithm computes the gradient orientation $g_k(\omega)$ at each pixel ω of each image u_k. In our implementations the gradient is computed using central differences, as it is described in Algorithm 2. In the case of a flat region of the image, the two centered differences Δ_x and Δ_y are zero. Then, the image gradient orientation is not defined and $g_k(\omega)$ is set to a particular **not defined** value to prevent its use. Notice that the $\frac{1}{2}$ factors in the centered differences are not needed (and are not included in the reference code) because they cancel out in the computation of the angle as $\arctan \frac{\Delta_y}{\Delta_x}$. Also, in the reference implementation in C language, there is no need for a different handling for the cases in which Δ_x is zero; the function `atan2(y,x)` in the standard C library takes the numerator and denominator as different parameters to give the correct result in all cases.

Returning to the main algorithm, after computing the gradient orientations, all the $\frac{N(N-1)}{2}$ pairs of images u_a and u_b are evaluated (step 5). The normalized gradient angle error map ξ between u_a and u_b is computed (steps 6 and 7) and defined as

$$\xi(\omega) = \frac{|\mathrm{Angle}(\nabla u_a(\omega),\ \nabla u_b(\omega))|}{\pi} = \frac{|\mathrm{Angle}(g_a(\omega),\ g_b(\omega))|}{\pi}. \tag{9}$$

Instead of testing every 4-connected region in Ω, which are a huge quantity, a heuristic approach is used to propose a reduced number of regions R, which are then evaluated using the a contrario formulation described before. A threshold ρ determines pixels with low normalized gradient angle error $\xi \le \rho$. Then, every 4-connected region of pixels satisfying $\xi \le \rho$ is a candidate to be evaluated. The value ρ is a parameter of the method.

A greedy algorithm is used to compute the 4-connected regions with $\xi \le \rho$. Every pixel ω in the domain is a possible seed pixel for a new region (step 8), provided that it satisfies the condition $\xi(\omega) \le \rho$ (step 9). Starting from the pixels ω, a region growing algorithm (step 10) iteratively adds

Algorithm 2: Compute gradient orientation

```
input  : $N$ images $u_1, \dots, u_N$ defined on $\Omega$
output: $N$ gradient orientation maps $g_1, \dots, g_N$ defined on $\Omega$
1 for $k \in \{1, \dots, N\}$ do
2     for $(x, y) \in \Omega$ do
3         $\Delta_x \leftarrow \frac{u_k(x+1,y) - u_k(x-1,y)}{2}$                centered differences
4         $\Delta_y \leftarrow \frac{u_k(x,y+1) - u_k(x,y-1)}{2}$                the $\frac{1}{2}$ factors are not really needed as they cancel in step 6
5         if $\Delta_x^2 + \Delta_y^2 > 0$ then
6             $g_k(x, y) \leftarrow \arctan \frac{\Delta_y}{\Delta_x}$
7         else
8             $g_k(x, y) \leftarrow$ not defined
```

neighbor pixels in which the normalized gradient angle error ξ is smaller than ρ. This region growing step is described in Algorithm 3 and is detailed below. The result is a region R in which all the pixels are connected to each other by a 4-connection chain of pixels of R and in which all its pixels have $\xi < \rho$. The same step also computes and returns the sum s of normalized angle errors ξ along the region R, as defined in Equation (2).

Then, the $\text{NFA}(u_a, u_b, R)$ is computed at step 11. When its value is smaller than 1, a meaningful match is found between images u_a and u_b at region R. Thus, both ground visibility masks V_a and V_b are updated to **visible** at the pixels of R (steps 12 to 14).

To accelerate the procedure, regardless of whether a match is validated or not, the region growing procedure in step 10 puts the values of the normalized angle error ξ to 1 for each pixel that is added to the region R (more details below); as a result, none of these pixels will satisfy the condition of being smaller than ρ more than once, and it will not start, nor be added to a new region, in further iterations of step 10. As a consequence, each pixel is part of at most one region per image pair.

As can be seen in Figure 3, the initial ground visibility masks have many isolated white spots. These "holes" are filled by a simple grain filter in step 15, that removes connected **not visible** regions of less than λ pixels. The same is done for isolated **visible** spots, which appear in accidental very precise angular agreements; this is observed in particular when the pixel values are strongly quantized, resulting in an angular bias as shown in [8], rendering the isotropic hypothesis in the background model $\mathcal{H}_0$ less accurate. The parameter λ is adjusted according to the satellite resolution, the size of the clouds, and the size of the structures of interest. The grain filter is detailed in Algorithm 4 which is described below.

Some remarks are required. First, the NFA is computed according to Equation (8) but using an approximation to the probability term $\mathbb{P}\Big[S_{\mathcal{H}_0}(R) \le s\Big]$. It can be shown that the Irwin-Hall distribution in Equation (4) can be upper-bounded by the first term of the sum,

$$\mathbb{P}\Big[S_{\mathcal{H}_0}(R) \le s\Big] \le \frac{s^{|R|}}{|R|!}. \tag{10}$$

This implies that

$$\text{NFA}(u, v, R) \le \frac{N(N-1)}{2}\, X^2 Y^2\, \alpha \frac{\beta^{|R|}}{|R|}\, \frac{s^{|R|}}{|R|!}. \tag{11}$$

The latter expression is much simpler to compute. Due to the upper-bound, the condition $N_T \frac{s^{|R|}}{|R|!} < 1$ implies that the NFA is also smaller than one. Hence, regions detected with this upper bound are guaranteed to be meaningful. However, some regions could be incorrectly rejected when the exact

Algorithm 3: Compute ρ-neighbors, a 4-connected region with $\xi \leq \rho$ starting at pixel ω

```
input     : normalized angle error $\xi$ defined on $\Omega$
input     : seed pixel $\omega$
parameter: region growing angular tolerance $\rho = \frac{1}{5}$
output    : a region of pixels $R$
output    : a positive value $s$, the sum of normalized angle errors in $R$
1  $R \leftarrow \{\omega\}$
2  $s \leftarrow \xi(\omega)$
3  $\xi(\omega) \leftarrow 1$                      put error $\xi$ to 1 so the pixel $\omega$ is not used again
4  repeat
5      for $(i,j) \in R$ do
6          for $\gamma \in \{(i,j-1),(i,j+1),(i-1,j),(i+1,j)\}$ do      4-connectivity neighbors
7              if $\xi(\gamma) \leq \rho$ then
8                  $R \leftarrow R \cup \{\gamma\}$                        add new pixel
9                  $s \leftarrow s + \xi(\gamma)$                          add error in $\gamma$ to the sum $s$
10                 $\xi(\gamma) \leftarrow 1$                 put error $\xi$ to 1 so the pixel $\gamma$ is not used again
11 until R does not change
12 return $R$, $s$
```

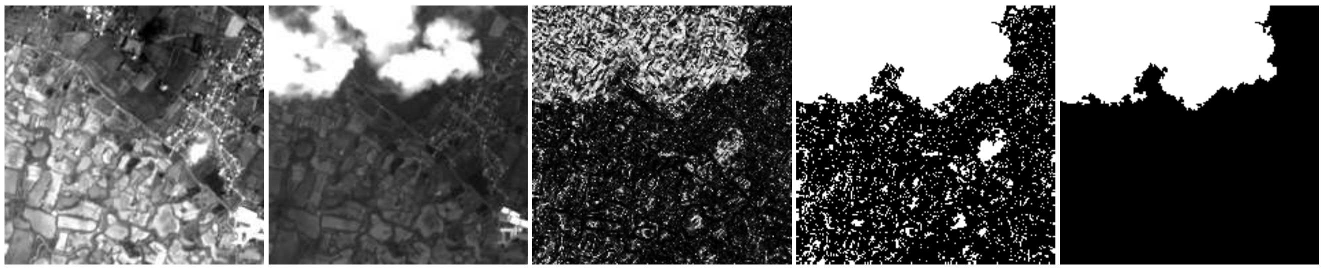


Figure 3: Two images to be compared and the intermediate and final result of the image matching method. In the normalized gradient orientation error map, black corresponds to zero error (same gradient orientation in both images) while white corresponds to the worst error (the gradient orientations have opposite direction). In the two ground visibility maps, black pixels correspond to **visible** ground while white corresponds to **not visible** ground. In the initial ground visibility mask one can see some "ground holes", that is small white spots surrounded by black pixels. Those ground holes are filled for the final ground visibility mask. In this simple example with only two images, a common mask can be produced, and the zones that do not match are declared as not visible. However, when more images are included in the series, zones that are not at first detected as visible in the first image might match with other images in the series and therefore be declared as visible. This shows the relevance of the third hypothesis of the method: each zone should be visible at least twice in the time series.

value of NFA is close to 1. Nevertheless, numerical experiments confirm that this effect is minor. Indeed, when similar structures are present in two images, the NFAs and the upper-bound are usually both very small.

The second remark refers to the background model $\mathcal{H}_0$. As stated in the previous section, $\mathcal{H}_0$ implies that the gradient orientations at each pixel are independent random variables. However, as described before (Algorithm 2), the actual implementation computes the image gradient using centered differences. Thus, even if the initial pixels are independent, the computed gradient of neighbor pixels is not strictly independent due to the scheme used.

There are three simple approaches to cope with this problem. The first is to sub-sample the image

Algorithm 4: Grain Filter

```
input     : $N$ ground visibility masks $V_1, \dots, V_N$ defined on $\Omega$
parameter: minimal region size $\lambda$
1  for $V \in \{V_1, \dots, V_N\}$ do                                   loop on visibility maps
2      for $v \in \{$visible, not visible$\}$ do               remove small visible and not visible grains
3          available($\Omega$) $\leftarrow$ true               initialize map of used pixels in each image and state
4          for $\omega \in \Omega$ do
5              if available($\omega$) and $V(\omega) = v$ then
6                  $\hat{R} \leftarrow \{\omega\}$                                   initialize new region from $\omega$
7                  available($\omega$) $\leftarrow$ false               prevent this pixel from being used again
8                  repeat
9                      for $(i, j) \in \hat{R}$ do
10                         for $\gamma \in \{(i, j-1), (i, j+1), (i-1, j), (i+1, j)\}$ do 4-connected neighbors
11                             if available($\gamma$) and $V(\gamma) = v$ then
12                                 $\hat{R} \leftarrow \hat{R} \cup \{\gamma\}$                              add new pixel
13                                 available($\gamma$) $\leftarrow$ false       prevent this pixel from being used again
14                 until $\hat{R}$ does not change
15                 if $|\hat{R}| < \lambda$ then                                     remove region if too small
16                     $V(\hat{R}) \leftarrow \bar{v}$                   if $v$ is visible, $\bar{v}$ is not visible, and vice versa
```

gradient map by a step of 3 pixels in both directions. As a result, the computed gradients remain independent if the initial pixels were independent. This is the correct approach as it is coherent with the proposed background model $\mathcal{H}_0$. Nevertheless, this implies some steps to down-sample the gradient and then up-sample the resulting visibility map. Moreover, there is also a loss in resolution on the visibility map. A second approach is to perform an approximation of what would be obtained by this sub-sampling but without actually doing it: after a region R is computed, the number of pixels $|R|$ is divided by 9, and the sum of normalized gradient angle errors s is also divided by 9. Sub-sampling by a factor 3, implies keeping one pixel out of nine. If the errors are roughly regularly distributed, dividing both the sum of errors and the number of pixels by nine, would result in the same result as performing the sub-sampling. The last approach is based on the observation that, even if strictly speaking there are dependencies on the gradient when computed by centered differences, actually the correlation is weak and the gradient angles are fairly decorrelated when the input image is white noise; this was studied empirically in [17, 15]. As a result, one can, as an approximation, model the image gradient without sub-sampling as if they were independent. As will be shown in Section 6, the three approaches produce very similar results (see Figure 6); for practical considerations, the last one will be kept.

A third remark is related to the numerical representation. The NFA can reach extremely small values, which may underflow the usual IEEE 754 number representation. Our implementation in the C programming language, which uses IEEE 754 number representation, computes $\log_{10}(\text{NFA})$ instead of NFA, allowing for a larger numeric range. Any logarithm base is equally useful for this purpose; the 10 base makes it slightly easier to read the order of magnitude of the NFA values. Of course, the test must now compare $\log_{10}(\text{NFA})$ to $\log_{10}(\varepsilon)$, which for $\varepsilon = 1$ is zero.

A final remark concerns the parameter ρ. This value has an impact on the heuristic method used to propose the candidate regions R. However, it has no impact on the final validation step based on Equation (8). The parameter ρ is fixed to a very relaxed value $1/5$, which implies that gradients angles with a difference of up to 36 degrees are grouped together. If all the pixels in the region have an error of about 36 degrees, this would rarely lead to a meaningful match; in good matches, however, many pixels have much smaller errors, leading to meaningful validations. Also, using a value smaller than 45 degrees helps to prevent a potential bias in the gradient orientations due to the discrete nature of digital images [8]; indeed, the horizontal, vertical and diagonal orientations are more probable.

Algorithm 3 provides a detailed description of the region growing step to compute the candidate region R. The region R is initialized as the set composed only of the seed pixel ω, step 1. The sum s of normalized angular errors is initialized to the error at the seed pixel, $\xi(\omega)$, step 2. As explained before, each time a pixel is added to a region R, its normalized angular error ξ is set to 1 so it will never again satisfy the ρ condition, and it will never be used again in a different region; for the seed pixel this is done in step 3.

Then, iteratively, the neighbors of pixel in R are considered to be added to the region; this process continues until no new pixel is added to R (steps 4 to 11). The 4-connected neighbors of each pixel in R (steps 5 and 6) are evaluated, and when the ρ condition is satisfied (step 7) the new pixel is added to the region R (step 8); the error sum s is updated by adding the error of the new pixel (step 9). Again, the normalized angular error ξ of the added pixel is set to 1 to prevent its use in further regions. The final region R and the corresponding sum of errors s are returned.

Note that the partition of the domain Ω into the candidate regions does not depend on the seed pixels or the particular order in which the region growing algorithm works; the result is fully determined by the 4-connected regions of $\xi \leq \rho$, thus by the angle error map ξ, the use of 4-connectivity, and the tolerance parameter ρ.

Algorithm 4 details the grain filter. The aim of this step is to remove *grains*, which are small 4-connected regions of **visible** pixels; the same is done for **not visible** grains. The same process is repeated for each of the N visibility maps (step 1); also, the same process is done for the values **visible** and **not visible** in the visibility maps (step 2).

The grain filter follows the same steps as the region growing algorithm described before: starting from a pixel, it grows a region by adding 4-connected neighbors with the same value in the map (**visible** or **not visible**). For this, an auxiliary map named **available** is used to keep track of the pixels already used. This map is initialized to **true** before processing each new map and with each of the two values, step 3; the value **true** indicates the pixel can be used in a new grain.

Now, each pixel of the domain Ω is tried as a starting point for a new grain (step 4) and the grain is effectively started if the pixel is available and has the current map value v being evaluated (step 5). In that case, a new region $\hat{R}$ is started composed initially of only that pixel (step 6). Note that these regions $\hat{R}$ are not related to the regions R computed in the main algorithm or in the region growing step described in Algorithm 3. (In our implementation, however, the same memory buffer is used to store both kind of regions; this is just to avoid allocating new memory and because there is no risk of conflict as the grain filter is used after all the other processing is already done.) When the pixel is added to the region $\hat{R}$ it is marked as not available so it is not used again (step 7).

Once the region $\hat{R}$ is initialized, the neighbors with the same visibility map value are iteratively added to the region. The process is the same as in the region growing step. The 4-connected neighbors to each pixel in $\hat{R}$ are evaluated (steps 9 and 10), and when they are available and have the same visibility map value (step 11) are added to the region $\hat{R}$ (step 12); also, they are set as not longer available (step 13). This is iterated until no new pixel is added to $\hat{R}$ (step 14).

Finally, the number of pixels in region $|\hat{R}|$ is compared to the parameter λ (step 15); if the region has less than λ pixels, it is removed; this is done by changing the value in the visibility map to the other value: a **visible** grain is removed by setting its pixels to **not visible** and vice versa. The result of the grain filter is the set of updated visibility maps.

5 Computational Complexity

The first two steps of the algorithm, the initialization and the gradient computations, require both a computational cost proportional to the number of images and to the number of pixels per image. The main processing is done on each of the $\frac{N(N-1)}{2}$ pairs of images. For each pair of images, the computation of the normalized angle error has a complexity proportional to the number of pixels. Then, the region growing step uses each pixel only once, so the computational cost is again proportional to the number of pixels. Finally, the grain filter works on the N visibility masks and visits each pixel at most twice, once processing the **visible** map, the other while processing the **not visible** map. All in all, the bottleneck is processing the $\frac{N(N-1)}{2}$ pairs, so the computational complexity is $O(N^2XY)$, where N is the number of images, each one of size $X \times Y$.

In practice, processing 10 images of size 500×500 requires about one second in a current computer running on an 1.2 GHz Intel Core M processor.

There is some room for improving speed by parallel processing. Indeed, the $\frac{N(N-1)}{2}$ image pairs could be processed in parallel. This would require either sharing the visibility map memory (which may reduce slightly the speed) but leads to no potential race condition as the parallel processing, when changing the visibility maps would be always in the same sense, to set it to **visible**. An alternative would be that each pair produces a resulting map and the final visibility maps are obtained by their merging.

When a long time series needs to be processed, the N^2 factor may become important, resulting in a large computational cost. At the same time, when the number of images is large, the hypothesis of the method are probably also satisfied for a subset of the time series. The most important condition in this sense, is that each part of the region of interest is visible at least twice in the time series. The full time series could be divided, for example, into two time series of size $\frac{N}{2}$. Doing this, the processing time would be approximately half the one required to process the entire time series. In general, there may be a number N_0 for which a consecutive set of N_0 images from the time series are enough to produce good results. Then, the algorithm can be applied by batches of N_0 instead of to the full time series. This would result in $\frac{N}{N_0}$ batches, each one gaining a speed factor of $\left(\frac{N_0}{N}\right)^2$; thus, the computational time would be reduced by a factor of about $\frac{N_0}{N}$.

6 Experiments

Figure 4 shows the result of the ground visibility detection on a time series of ten Sentinel-2 images. Here the hole filling parameter λ was set to 500 pixels. The first and third rows show the ten input images and the second and fourth rows show the resulting ground visibility masks. Here white corresponds to **not visible** and black to **visible**. The algorithm managed to provide a fairly good evaluation of the general ground visibility. Images 3, 7 and 10 are fully covered by clouds and so indicate the complete white masks. In most remaining images the masks are mainly black, correctly indicating the general ground visibility of those images. The white spots correspond to the presence of clouds or of zones were the topography does not satisfy the slow change hypothesis, in this case essentially zones covered by water (lower left and lower right corners).

A second experiment in Figure 5 shows similar results. But here there are also **not visible** parts which are due to the lack of repetitive structure; this is the case in some small fields where no relevant texture can be matched from one image to another. This illustrates the main limitation

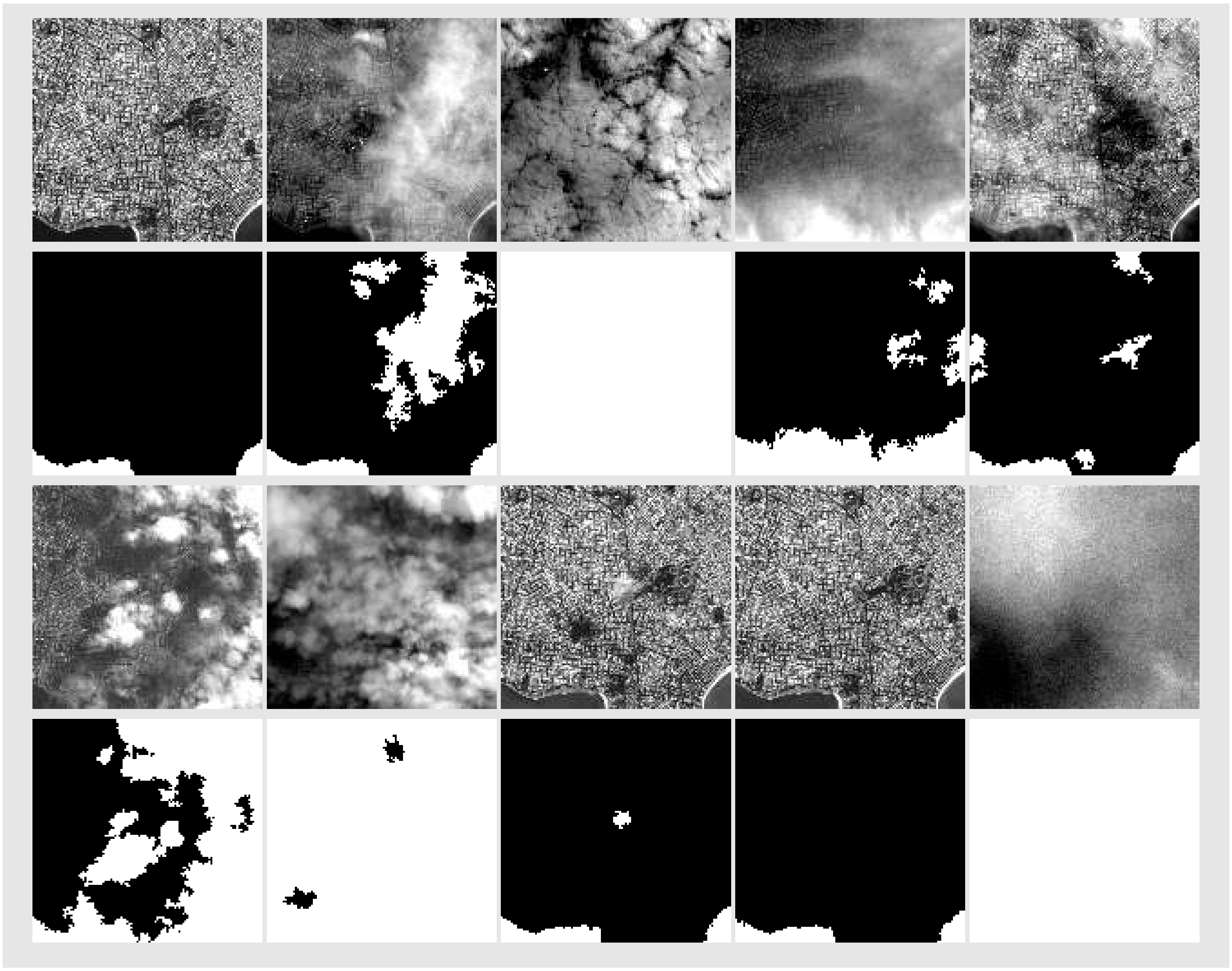


Figure 4: Ground visibility computed with the proposed algorithm in a time series of ten images from the Sentinel-2 constellation. The gray background helps to visualize the white parts of visibility masks. The grain filter parameter was set to $\lambda = 500$. First and third rows: input images. Second and fourth rows: resulting ground visibility maps, where black pixels correspond to **visible** while white corresponds to **not visible**. The images are 462×449 and the ten ground visibility masks were jointly computed in about one second on an 1.2 GHz Intel Core M processor.

of the algorithm: the "non visible" parts do not always correspond to non visible ground, but just to unstable ground structures. Continuously changing objects, as the sea, will be systematically marked as **not visible**. This may not be a problem, as the location of the sea or other continuously changing objects can be obtained from other sources, or even be deduced as locations always marked as **not visible** in long time series. Nevertheless, when the object of interest is located precisely in or near such a zone (for example for observing boats), this algorithm cannot provide a useful evaluation of ground visibility. Moreover, when the object of interest is changing rapidly and significantly, the change hypothesis is again not satisfied and this method would not be useful.

Figure 6 illustrates the results obtained using three different approaches to cope with the gradient angle dependence problem as described in Section 4. As the figure illustrates in a particular case, the results are in general very similar. For simplicity of the implementation and because it gives a few more details in the masks, the full-scale approach is kept.

The only parameter that needs a certain tuning is λ, which controls the size of the regions to be removed in the grain filter. The following experiments show the result of the algorithm on the same

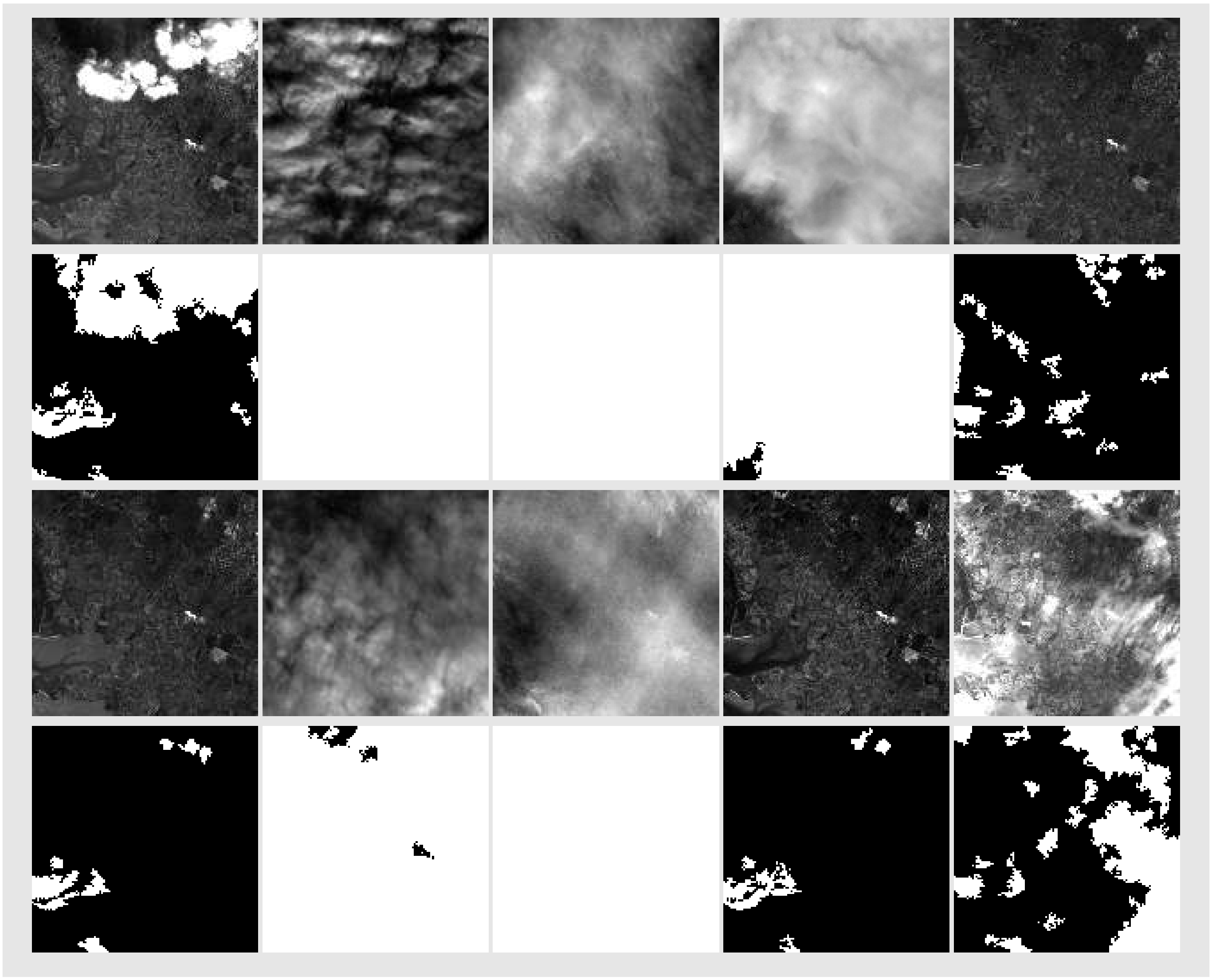


Figure 5: Ground visibility computed with the proposed algorithm in a time series of ten images from the Sentinel-2 constellation. The gray background helps to visualize the white parts of visibility masks. The grain filter parameter was set to $\lambda = 500$. First and third rows: input images. Second and fourth rows: resulting ground visibility maps, where black pixels correspond to **visible** while white corresponds to **not visible**. The images are 496×496 and the ten ground visibility masks were jointly computed in about one second on an 1.2 GHz Intel Core M processor.

time series of Figure 4 but setting $\lambda = 0$ in Figure 7 and $\lambda = 5000$ in Figure 8. Setting $\lambda = 0$ is equivalent to not performing the grain filter and Figure 7 illustrates its need. Indeed, the visibility maps contain small grains to be removed. Many small **not visible** (white) grains, mostly of one or two pixels, are due to small gaps between the detected matches found by the a contrario test. These are the main target of the grain filter. Occasionally, small objects not satisfying the change hypothesis also produce detections that can be removed by the grain filter.

The optimal value for λ depends on the contents of the image, in particular on the image resolution. Figure 8 shows the results using $\lambda = 5000$, that is ten times larger, and produces a similar result to the one in Figure 4. This stronger filtering yields a more clear result but at the same time it removes some of the good detections due to small clouds.

As a rule, the larger the number of images in the time series, the better the result. The final set of experiments illustrates the impact on the result of the number of images in the time series. Figure 9 shows the visibility maps computed with the same algorithm on just the first two images

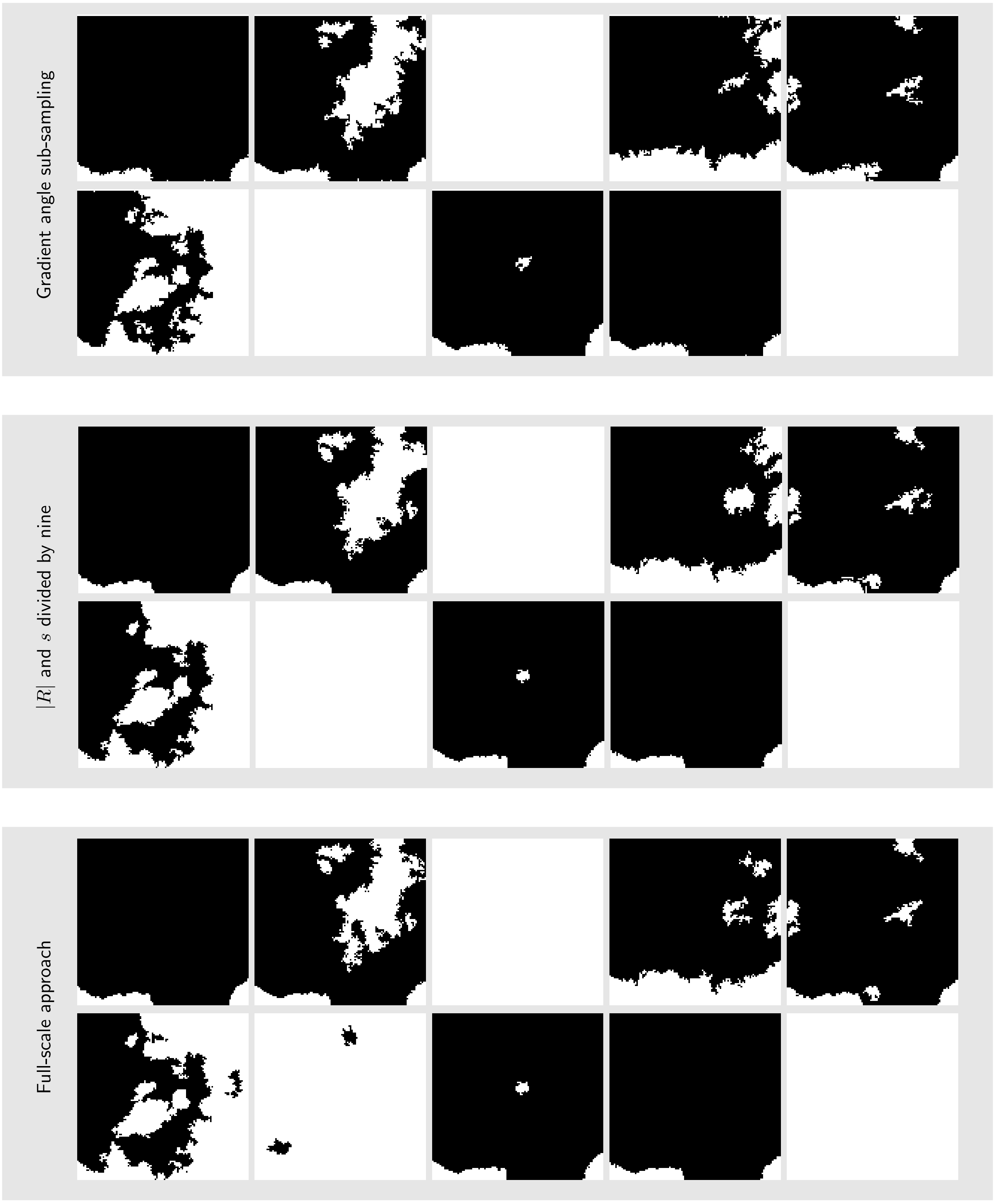


Figure 6: Comparison of ground visibility maps obtained using three approaches to cope with the gradient angle correlation for the ten images used in Figure 4. The gray background helps to visualize the white parts of visibility masks. Sub-sampling the gradient angle maps by a factor 3 before performing the rest of the algorithm; this approach is in agreement with the background model $\mathcal{H}_0$. Dividing R and s by nine is an approximation to the previous case. Using the full-scale gradient angle maps result in fairly correlated gradient angle maps, which strictly speaking does not follow the background model $\mathcal{H}_0$; nevertheless, the model is approximately valid. The results are very similar for the three approaches.

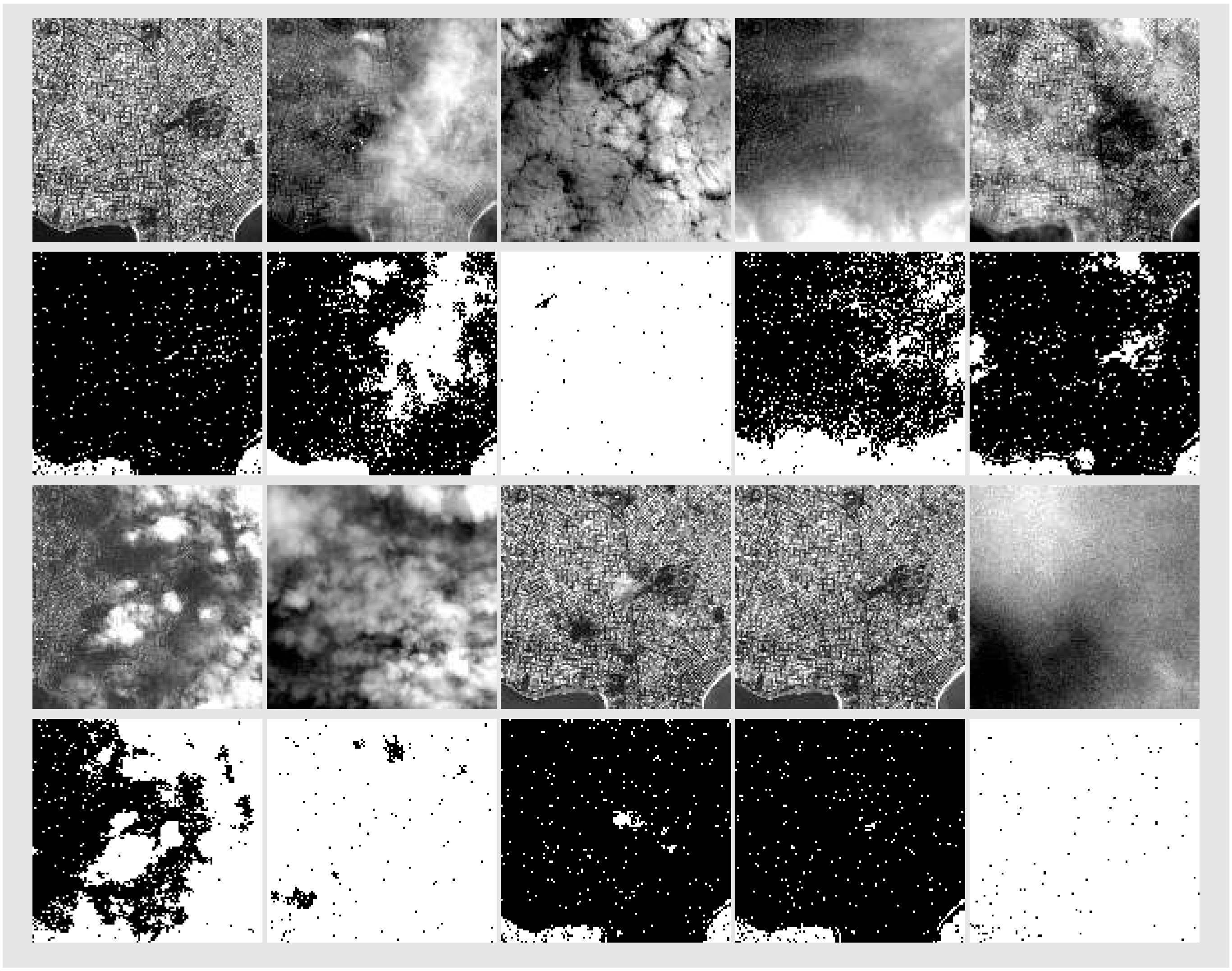


Figure 7: Ground visibility computed with the proposed algorithm in the same time series as in Figure 4 but setting the grain filter parameter to $\lambda = 0$. The gray background helps to visualize the white parts of visibility masks. First and third rows: input images. Second and fourth rows: resulting ground visibility maps, where black pixels correspond to **visible** while white corresponds to **not visible**. The images are 462×449 and the ten ground visibility masks were jointly computed in about one second on an 1.2 GHz Intel Core M processor.

on Figure 4. Given that there is only one image pair, both visibility maps are exactly the same. In this case, the hypothesis that each part of the region of interest is visible at least twice in the time series is not satisfied. Thus, the algorithm is not able to find matches for many visible parts of the first image. As a result, the visibility map for the first image is essentially wrong (even if the visibility map for the second image is roughly correct). A similar experiment is shown in Figure 10 but now using the first six images of Figure 4. The functioning hypothesis are better satisfied and the result is much better. Nevertheless, the result in Figure 4 is a little better, which illustrates that, to a certain degree, the longer the time series, the better the result. This, of course, implies a computational cost. The optimal time series length that gives the best balance depends on the particular application.

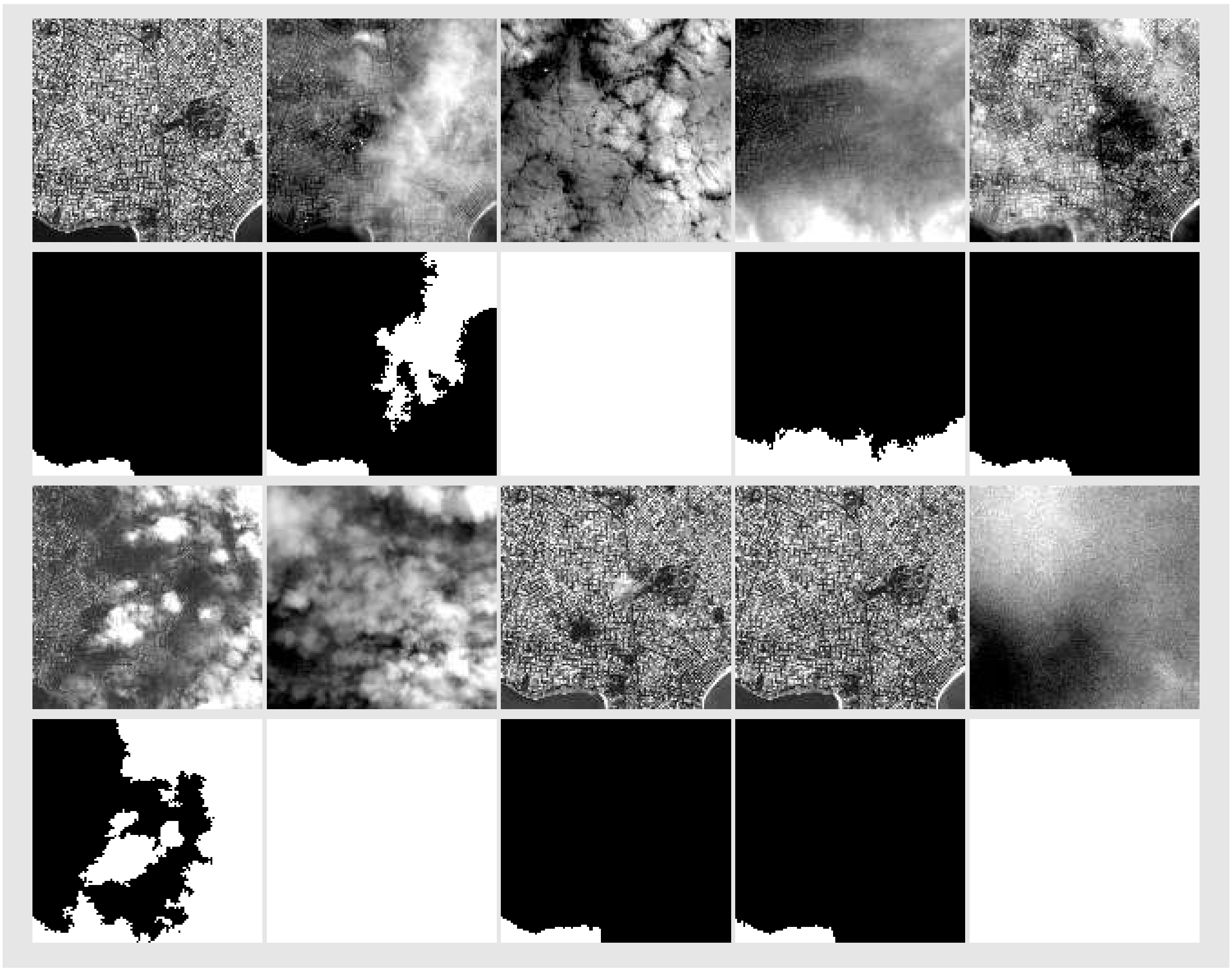


Figure 8: Ground visibility computed with the proposed algorithm in the same time series as in Figure 4 but setting the grain filter parameter to $\lambda = 5000$. The gray background helps to visualize the white parts of visibility masks. First and third rows: input images. Second and fourth rows: resulting ground visibility maps, where black pixels correspond to **visible** while white corresponds to **not visible**. The images are 462×449 and the ten ground visibility masks were jointly computed in about one second on an 1.2 GHz Intel Core M processor.

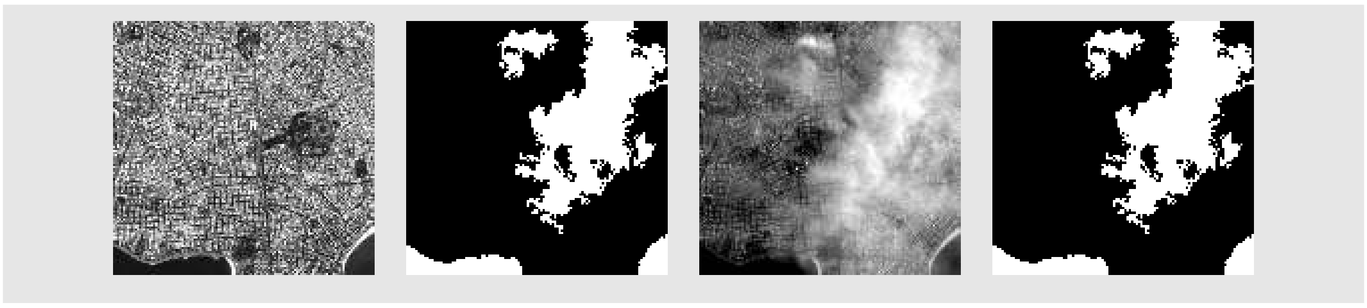


Figure 9: Ground visibility computed with the proposed algorithm in the same time series as in Figure 4 but using only the first two images from the sequence. The gray background helps to visualize the white parts of visibility masks. Given that there is only one image pair, both visibility maps are exactly the same. In this case, one of the hypothesis of the method is clearly not satisfied, namely that each part of the region of interest is visible at least twice in the time series. As a consequence, some clear parts of the first image cannot be matched to second image and are wrongly indicated as not visible.

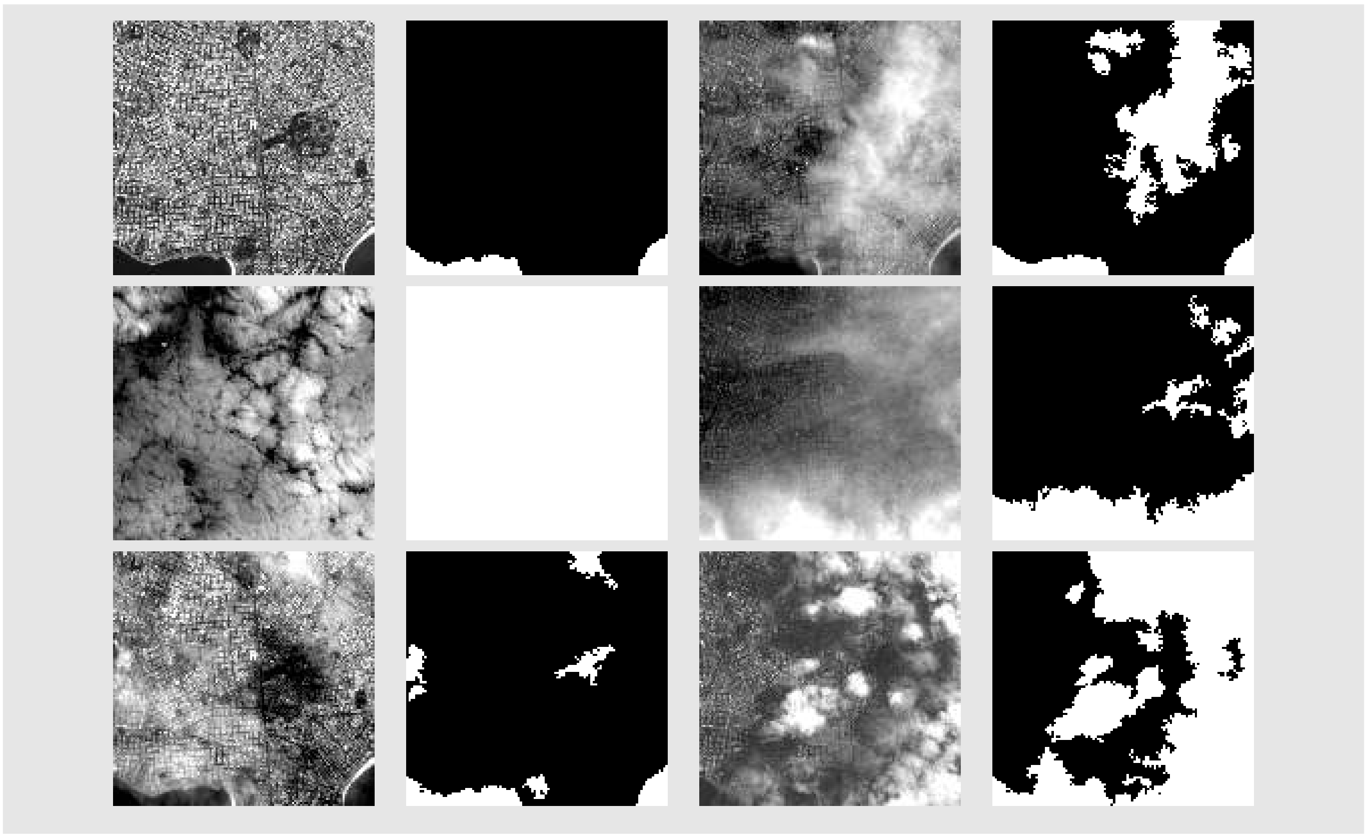


Figure 10: Ground visibility computed with the proposed algorithm in the same time series as in Figure 4 but using only the first six images from the sequence. The gray background helps to visualize the white parts of visibility masks. In this case, the hypothesis that each part of the region of interest is visible at least twice in the time series is roughly satisfied; nevertheless, the results are not as good as when the full series of ten images is used.

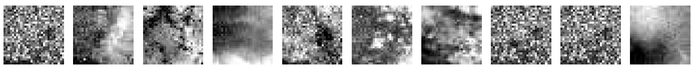
7 Conclusions

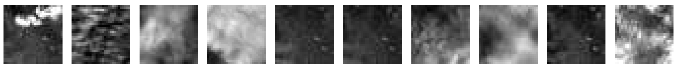
An algorithm for temporal repetition detection in time series was described based on the a contrario statistical framework and using a heuristic procedure to accelerate the result. The method produces good results when its functioning hypotheses are satisfied. Among them, the more demanding one is that the target zone changes slowly relative to the sampling frequency, which is not valid, for example, over the sea.

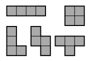
Acknowledgements

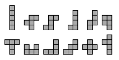
We thank Thibaud Ehret for valuable discussions and suggestions. Work partly financed by Office of Naval research grant N00014-17-1-2552, DGA Astrid project “Filmer la Terre” no. ANR-17-ASTR-0013-01 and Kayrros.

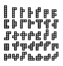
Image Credits


Sentinel-2 images.


Sentinel-2 images.


Adapted from file `All_5_free_tetrominoes.svg` by Anypodetos CC BY-SA 3.0


Adapted from file `All_18_Pentominoes.svg` by R. A. Nonenmacher CC BY-SA 4.0


Adapted from file `All_35_free_hexominoes.svg` by R. A. Nonenmacher CC BY-SA 4.0

References

[1] R. Arandjelovic and A. Zisserman, *Three things everyone should know to improve object retrieval*, in IEEE Computer Society Conference on Computer Vision and Pattern recognition (CVPR), 2012, pp. 2911–2918. https://doi.org/10.1109/CVPR.2012.6248018.

[2] L. Baetens, C. Desjardins, and O. Hagolle, *Validation of Copernicus Sentinel-2 Cloud Masks Obtained from MAJA, Sen2Cor, and FMask Processors Using Reference Cloud Masks Generated with a Supervised Active Learning Procedure*, Remote Sensing, 11 (2019). https://doi.org/10.3390/rs11040433.

[3] H. Bay, T. Tuytelaars, and L. Van Gool, *SURF: Speeded up robust features*, European Conference on Computer Vision (ECCV), 1 (2006), pp. 404–417. https://doi.org/10.1007/11744023_32.

[4] F. Cao, J. L. Lisani, J. M. Morel, P. Musé, and F. Sur, *A Theory of Shape Identification*, Springer, 2008. ISBN 978-3-540-68480-0.

[5] G. Chandran and C. Jojy, *A survey of cloud detection techniques for satellite images*, International Research Journal of Engineering and Technology (IRJET), 2 (2015), pp. 2485–2490.

[6] T. Dagobert, R. Grompone von Gioi, J-M. Morel, and C. de Franchis, *Temporal Repetition Detector for Time Series of Spectrally Limited Satellite Imagers*, Image Processing On Line, 10 (2020), pp. 62–77. https://doi.org/10.5201/ipol.2020.245.

[7] T. Dagobert, J. M. Morel, C. de Franchis, and R. Grompone von Gioi, *Visibility Detection in Time Series of Planetscope Images*, in IEEE International Geoscience and Remote Sensing Symposium, 2019, pp. 1673–1676. https://doi.org/10.1109/IGARSS.2019.8898892.

[8] A. Desolneux, S. Ladjal, L. Moisan, and J.M. Morel, *Dequantizing image orientation*, IEEE Transactions on Image Processing, 11 (2002), pp. 1129–1140. https://doi.org/10.1109/TIP.2002.804566.

[9] 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.

[10] A. Desolneux, L. Moisan, and J-M. Morel, *From Gestalt theory to image analysis: a probabilistic approach*, vol. 34, Springer, 2008. ISBN 978-0-387-74378-3.

[11] D. Frantz, E. Hass, A. Uhl, J. Stoffels, and J. Hill, *Improvement of the Fmask algorithm for Sentinel-2 images: Separating clouds from bright surfaces based on parallax effects*, Remote Sensing of Environment, 215 (2018), pp. 471 – 481. https://doi.org/10.1016/j.rse.2018.04.046.

[12] M. Gardner, *More about the shapes that can be made with complex dominoes (mathematical games)*, Scientific American, 203 (1960), pp. 186–201.

[13] S. W. Golomb, *Polyominoes*, Princeton University Press, 2. ed. ed., 1994. ISBN 9780691024448.

[14] A. Gordon, G. Glazko, X. Qiu, and A. Yakovlev, *Control of the mean number of false discoveries, Bonferroni and stability of multiple testing*, The Annals of Applied Statistics, 1 (2007), pp. 179–190. https://doi.org/10.1214/07-AOAS102.

[15] R. Grompone von Gioi, *A contrario line segment detection*, Springer, 2014. ISBN 978-1-4939-0575-1.

[16] R. Grompone von Gioi, C. Hessel, T. Dagobert, J-M. Morel, and C. de Franchis, *Temporal repetition detection for ground visibility assessment*, in ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences, vol. V-2-2020, 2020, pp. 829–835. https://doi.org/10.5194/isprs-annals-V-2-2020-829-2020.

[17] R. Grompone von Gioi, J. Jakubowicz, J-M. Morel, and G. Randall, *On straight line segment detection*, Journal of Mathematical Imaging and Vision, 32 (2008), pp. 313–347. https://doi.org/10.1007/s10851-008-0102-5.

[18] R. Grompone von Gioi and V. Pătrăucean, *A contrario patch matching, with an application to keypoint matches validation*, in IEEE International Conference on Image Processing (ICIP), 2015, pp. 946–950. https://doi.org/10.1109/ICIP.2015.7350939.

[19] A. Hollstein, K. Segl, L. Guanter, M. Brell, and M. Enesco, *Ready-to-Use Methods for the Detection of Clouds, Cirrus, Snow, Shadow, Water and Clear Sky Pixels in Sentinel-2 MSI Images*, Remote Sensing, 8 (2016). https://doi.org/10.3390/rs8080666.

[20] R. R. Irish, *Landsat 7 automatic cloud cover assessment*, in Proceedings of SPIE, vol. 4049, 2000, pp. 348–355. https://doi.org/10.1117/12.410358.

[21] R. R. Irish, J. L. Barker, S. N. Goward, and T. Arvidson, *Characterization of the Landsat-7 ETM+ Automated Cloud-Cover Assessment (ACCA) Algorithm*, Photogrammetric Engineering & Remote Sensing, 72 (2006), pp. 1179–1188. https://doi.org/10.14358/PERS.72.10.1179.

[22] I. Jensen and A. J. Guttmann, *Statistics of lattice animals (polyominoes) and polygons*, Journal of Physics A: Mathematical and General, 33 (2000), p. L257.

[23] N. L. Johnson, S. Kotz, and N. Balakrishnan, *Continuous univariate distributions*, Distributions in statistics, Wiley, 2. ed. ed., 1995. ISBN 978-0-471-58495-7.

[24] Y. Ke and R. Sukthankar, *PCA-SIFT: A more distinctive representation for local image descriptors*, IEEE Computer Society Conference on Computer Vision and Pattern recognition (CVPR), 2 (2004), pp. 506–513. https://doi.org/10.1109/CVPR.2004.1315206.

[25] D. Lowe, *Perceptual Organization and Visual Recognition*, Kluwer Academic Publishers, 1985. ISBN 978-1-4613-2551-2.

[26] D. G. Lowe, *Distinctive image features from scale-invariant keypoints*, International Journal of Computer Vision, 60 (2004), pp. 91–110. https://doi.org/10.1023/B:VISI.0000029664.99615.94.

[27] S. Mahajan and B. Fataniya, *Cloud detection methodologies: variants and development—a review*, Complex & Intelligent Systems, (2019). https://doi.org/10.1007/s40747-019-00128-0.

[28] K. F. Manizade, J. D. Spinhirne, and R. S. Lancaster, *Stereo Cloud Heights From Multispectral IR Imagery via Region-of-Interest Segmentation*, IEEE Transactions on Geoscience and Remote Sensing, 44 (2006), pp. 2481–2491. https://doi.org/10.1109/TGRS.2006.873339.

[29] C. Panem, S. Baillarin, C. Latry, H. Vadon, and P. Dejean, *Automatic cloud detection on high resolution images*, in IEEE International Geoscience and Remote Sensing Symposium, vol. 1, 2005. https://doi.org/10.1109/IGARSS.2005.1526222.

[30] V. Pătrăucean, P. Gurdjos, and R. Grompone von Gioi, *Joint a contrario ellipse and line detection*, IEEE Transactions on Pattern Analysis and Machine Intelligence, 39 (2017), pp. 788–802. https://doi.org/10.1109/TPAMI.2016.2558150.

[31] V. Pătrăucean, R. Grompone von Gioi, and M. Ovsjanikov, *Detection of mirror-symmetric image patches*, in IEEE Conference on Computer Vision and Pattern Recognition Workshops, 2013, pp. 211–216. https://doi.org/10.1109/CVPRW.2013.38.

[32] J. Rabin, J. Delon, and Y. Gousseau, *A statistical approach to the matching of local features*, SIAM Journal on Imaging Sciences, 2 (2009), pp. 931–958. https://doi.org/10.1137/090751359.

[33] I. Rey Otero and M. Delbracio, *Anatomy of the SIFT Method*, Image Processing On Line, 4 (2014), pp. 370–396. https://doi.org/10.5201/ipol.2014.82.

[34] M. Rodriguez and R. Grompone von Gioi, *Affine invariant image comparison under repetitive structures*, in IEEE International Conference on Image Processing (ICIP), 2018, pp. 1203–1207. https://doi.org/10.1109/ICIP.2018.8451060.

[35] P. L. Scaramuzza, M. A. Bouchard, and J. L. Dwyer, *Development of the Landsat Data Continuity Mission Cloud-Cover Assessment Algorithms*, IEEE Transactions on Geoscience and Remote Sensing, 50 (2012), pp. 1140–1154. https://doi.org/10.1109/TGRS.2011.2164087.

[36] D. Shin and J. K. Pollard, *Cloud height determination from satellite stereo images*, in IEE Colloquium on Image Processing for Remote Sensing, Feb 1996, pp. 4/1–4/7. https://doi.org/10.1049/ic:19960158.

[37] K. Sinclair, B. van Diedenhoven, B. Cairns, J. Yorks, A. Wasilewski, and M. McGill, *Remote sensing of multiple cloud layer heights using multi-angular measurements*, Atmospheric Measurement Techniques Discussions, (2017), pp. 1–23. https://doi.org/10.5194/amt-2017-2.

[38] A. Taravat, S. Proud, S. Peronaci, F. Del Frate, and N. Oppelt, *Multilayer Perceptron Neural Networks Model for Meteosat Second Generation SEVIRI Daytime Cloud Masking*, Remote Sensing, 7 (2015), pp. 1529–1539. https://doi.org/10.3390/rs70201529.

[39] 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. https://doi.org/10.1016/B978-0-12-084320-6.50022-0.

[40] T. Wu, X. Hu, Y. Zhang, L. Zhang, P. Tao, and L. Lu, *Automatic cloud detection for high resolution satellite stereo images and its application in terrain extraction*, ISPRS Journal of Photogrammetry and Remote Sensing, 121 (2016), pp. 143 – 156. https://doi.org/10.1016/j.isprsjprs.2016.09.006.

[41] W. D. Zhang, M. X. He, and M. W. Mak, *Cloud detection using probabilistic neural networks*, in IEEE 2001 International Geoscience and Remote Sensing Symposium, vol. 5, 2001, pp. 2373–2375. https://doi.org/10.1109/IGARSS.2001.978006.