Canopy gaps from KH-9 HEXAGON - illustrated user guide

Author: Ladislav Šumichrast

RETROCHANGE / EOSC Federation · Poloniny canopy gaps Notebook: `Poloniny_canopy_gaps_1975.ipynb`

RETROCANOPY starts from a panchromatic image clipped to the forest area of interest, with pixels outside it marked as NoData. The notebook checks the input, corrects local illumination and detects dark canopy depressions using an explicit brightness threshold and minimum area. Users inspect the candidates and can add or remove features with correction polygons. The workflow then measures gap fraction, density, size distribution, distance to openings and spatial arrangement, checks sensitivity to detection settings, and exports maps, tables, figures and method records. Detections remain image-based candidates rather than height-verified canopy gaps.

The figures and results in this guide come from the supplied notebook run on the Havešová dataset. To reproduce the example outputs, rerun the notebook with the same input data, settings and correction layers.

Contents

1 · What you get 2

2 · The five-minute version 3

3 · Getting and preparing the data 5

4 · How the method works, in pictures 7

5 · The decisions that are yours 9

6 · Reading each result 12

7 · How much of it is convention 19

8 · The review loop 21

9 · Reporting the numbers 22

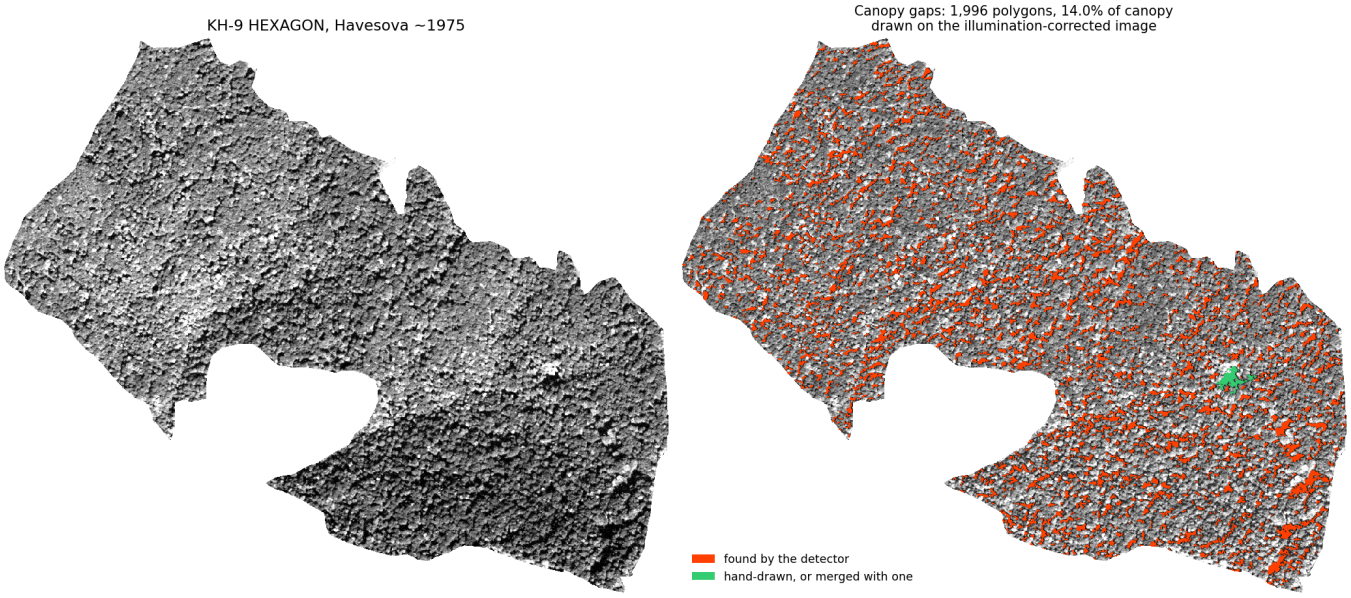
10 · Extending the use case 23

11 · File index 24

Verification and diagnostic edge cases (November 2026) 26

1 · What you get

One panchromatic image in, a measured canopy-gap regime out.



Left: KH-9 HEXAGON over Havešová, ~1975. Right: the 1 996 gaps found in it, drawn on the illumination-corrected image. Green is the one opening added by hand.

quantity	Havešová 1975	what it answers
gap **fraction**	13.96 % of the canopy	how much of the canopy was open
gap **density**	13.79 per hectare	how many openings there were
median gap	50.5 m²	midpoint of the count - it follows the size floor, so compare only at equal floors
half-area size	183 m²	half of all gap area lies in gaps larger than this
largest gap	2 690 m² (0.27 ha)	nothing near stand-replacing
arrangement	random at 15-140 m (p = 0.415)	consistent with openings forming independently
gap influence	99.99 % of canopy within one tree height	no interior remote from disturbance

Beech regenerates almost entirely in canopy gaps, so this *is* the stand-scale disturbance process. A 1975 baseline turns "this reserve is undisturbed" from an assumption into something measurable.

What it is not. Not a gap map in the Brokaw or Runkle sense - those are *height* criteria and this image has no height. It produces **candidates** measured under a stated convention. §7 and §9 say exactly what that buys and what it costs.

2 · The five-minute version

What you need

Input	one **panchromatic raster clipped to your area of interest**, with everything outside it set to the file's nodata value (§3.3). The Havešová example is 0.5 m KH-9, EPSG:5514, 3681 × 3079 px, 144.7 ha of valid pixels.
Software	Python ≥ 3.9 with numpy, scipy, pandas, matplotlib, rasterio, geopandas, **shapely ≥ 2.0**, scikit-image, scikit-learn - and nothing else. On a managed JupyterHub these are often already there: **run §4.1 of the notebook and it tells you.** To build the environment yourself, `env/environment-canopy-gaps.yml` (conda) records the versions this was verified with, `env/requirements-canopy-gaps.txt` is the pip equivalent.
Optional	QGIS, if you want to review and hand-correct the result (§8).
Not needed	an elevation model, sun-angle metadata, field plots, a forest mask, a second image, or network access at run time.

Running it

```
conda env create -f env/environment-canopy-gaps.yml
conda activate retrochange-canopy-gaps
jupyter lab            # open Poloniny_canopy_gaps_1975.ipynb
```

Edit **one cell** - §3 of the notebook - and *Run All*:

```
SRC   = pathlib.Path("data/Havesova.tif")   # your raster, already clipped to the reserve
LABEL = "Havesova"                             # goes into every output filename

GAP_MIN_AREA_M2 = 20        # the smallest gap you will map

RUN_ARRANGEMENT = True      # §12  Monte Carlo null  (~2½ min, then cached)
RUN_APPENDIX    = True      # App. C + App. A  the sweeps and the threshold search
```

Runtime on Havešová - an 11 M pixel array, 5.8 M of them inside the 145 ha reserve, ~2 000 gaps - is **under 4 minutes** with everything on. Most of the workflow scales with analysed pixels (Stužica, 48 M, is roughly four times the work and wants ~4 GB of RAM); §12 scales with the *number of gaps* instead, so that is the switch to turn off first on a large scene.

From the command line, without opening JupyterLab - useful for running several reserves one after another:

```
jupyter nbconvert --to notebook --execute --inplace \
    --ExecutePreprocessor.kernel_name=python3 \
    Poloniny_canopy_gaps_1975.ipynb
```

On the EOSC EU Node

https://eu-1.notebooks.open-science-cloud.ec.europa.eu - you get a pre-built JupyterLab image and, usually, no way to create a conda environment. The notebook is written to work without one, and **there are two things to do.**

Upload the notebook and your clipped raster:

```
Poloniny_canopy_gaps_1975.ipynb
data/<your clipped raster>.tif
```

That is the whole list. `outputs/` is created if missing, **nothing from `env/` is needed on the node**, and `data/` may sit beside the notebook or one level up - §3 of the notebook looks for `data/` in the working directory first and

`../data/` second.

Then *Run All*. Most likely that is the entire experience. Everything below is only what to do if it is not.

The cell to watch: §4.1

Run All reaches the environment self-test after about two seconds. It prints the version of every library, then *exercises* the API details the workflow actually relies on, and either gets out of the way:

```
Python      3.11.14
numpy          1.26.4           ok
scipy          1.17.1           ok
...
scikit-image   0.26.0           ok
Environment OK: every library this notebook needs is present and behaves as expected.
```

or stops and names what is missing or too old.

If it stops

Paste the line it hands you. It installs into your own user environment and needs no admin rights:

```
%pip install --upgrade 'scikit-image>=0.19' 'shapely>=2.0' 'geopandas>=0.13' \
                       'rasterio>=1.3' 'scikit-learn>=1.0'
```

Then *Kernel → Restart Kernel* and *Run All* again. `env/requirements-canopy-gaps.txt` is the same list in a file, if you would rather upload it and `%pip install -r` it.

Two things you never have to think about

Nothing reaches outside the working directory and nothing needs network access at run time, so the whatever the node allows out to the internet cannot affect the run. And if the instance you are given is small, switch `RUN_ARRANGEMENT` off in §3 before running - it is the part that scales with the number of gaps.

The two things that actually break on an environment you did not build

scikit-image	`remove_small_objects(..., max_size=…)` is keyword-only and recent (≥ 0.25). Older releases spell the same idea `min_size`, with an off-by-one difference in meaning. §4 of the notebook defines wrappers that call whichever exists - verified to give **identical** results on both APIs - so this is handled, and §4.1 proves it on your installation.
shapely	must be **≥ 2.0**. §12's null model uses the 2.x vectorised array API (`STRtree.query` on an array, `shapely.distance`, `shapely.union_all`). There is no shim for this one; on shapely 1.8 you must upgrade, or set `RUN_ARRANGEMENT = False` and lose §12 only.

`env/environment-canopy-gaps.yml` records the exact versions the delivered run was executed with, read back from the live interpreter. Note that the project-wide `env/environment.yml` pins numpy 2.4.6 and rasterio 1.4.4, which are **not** what actually ran (1.26.4 and 1.4.3) - use the canopy gaps environment file for this workflow.

3 · Getting and preparing the data

3.1 · The image

KH-9 HEXAGON frames are public domain, from **USGS EarthExplorer** (https://earthexplorer.usgs.gov/ → *Data Sets → Declassified Data → Declass 2 / Declass 3*). Choose a frame with **summer acquisition** (leaf-on, and a high sun, so shadows are short enough that a gap is still distinguishable from the seam between two crowns), little cloud over the stand, and the highest scan resolution offered.

The scenes used here are 0.5 m products. That number is the **scan pitch, not the resolution**: measurement on these frames shows only ~1 % of forest brightness variance below 1.2 m and nothing separable below about 3 m. The workflow is built around that - it measures at crown-group scale and never asks the image for detail it does not hold.

3.2 · Orthorectification

The frames arrive as scanned photographs and must be orthorectified before use; the RETROCHANGE scenes were supplied already orthorectified in EPSG:5514. However it is done, resample with **nearest-neighbour, not bilinear** - bilinear smooths crown edges, and every quantity in this workflow is computed from exactly those edges.

> *The elevation model used for orthorectification is a different thing from an elevation model in the analysis. **The analysis uses none.** See §9.4.*

3.3 · The area of interest

Clip the raster to your reserve before running. The notebook takes the AOI from the raster's own valid pixels - there is no polygon parameter.

```
gdalwarp -cutline reserve.gpkg -crop_to_cutline -dstnodata 255 \
         -r near scene.tif reserve_clipped.tif
```

or, in QGIS, *Raster → Extraction → Clip raster by mask layer* with **"Assign a specified nodata value to output bands"** ticked. Use a value the image does not contain - 255 works for 8-bit KH-9, which has no true-white forest.

Then check three things, in this order:

1. **Nodata is set.** `gdalinfo` must report a *NoData Value*. Without it, the fill outside the reserve is read as very dark, perfectly valid forest, and the map fills with false gaps.
2. **That value does not occur inside the reserve.** The notebook prints the nodata value and the count of DN == 0 pixels at §5, which catches the commonest version of this mistake:

```
Scene        : Havesova.tif   3681 x 3079 px at 0.50 m
CRS          : EPSG:5514
Nodata value : 255.0   (pixels with DN == 0: 0)
Reserve area : 144.7 ha  (51% of the raster; the rest is fill outside the reserve)
```

3. **The clip really is closed-canopy forest.** This is the check people skip, so it is worth being exact about what goes wrong.

The detector has no idea what it is looking at. Its entire rule is *darker than the mark, and at least 20 m²*, and **nothing caps how large a patch may be**. A permanently open feature inside the outline therefore corrupts the result **automatically, with no action from you**, in one of two ways:

- **if it reads dark** - a shaded rock face, a ravine floor, a track under the canopy's shadow - it is detected as **one very large gap**, and both gap fraction and mean gap size inherit it;

- **if it reads bright** - a sunlit meadow, open scree - it is *not* detected, and instead sits in the denominator as canopy that is not canopy, **diluting** the gap fraction.

Neither is what `ADD_GAPS` is for: that layer only *adds* openings the detector missed (§8). `REMOVE_GAPS` can erase a false one after the fact, but the right fix is to clip the feature out of the raster before you start.

The quick diagnostic is the largest-gap number, printed by §8 and listed in `*_gap_areas.csv`. At Havešová the five largest gaps are 2 690, 2 468, 1 527, 1 374 and 1 368 m² - a plausible tail for beech treefall. A "gap" of a hectare or more is a non-forest feature.

If the reserve genuinely contains permanent openings you cannot clip out, the result describes *openness*, not gap dynamics, and should be reported that way.

3.4 · What is not needed

No elevation model, no sun-angle metadata, no field plots, no forest mask, no second image, no network access at run time. That is deliberate: it is what makes the workflow deployable on the EOSC EU Node for any area where a KH-9 frame exists.

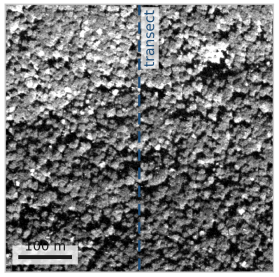
4 · How the method works, in pictures

Finding 1975 canopy gaps in declassified spy-satellite imagery

Havešová primeval beech forest, Poloniny NP · KH-9 HEXAGON ~1975, 0.5 m · every panel shows the same 500 × 500 m of forest

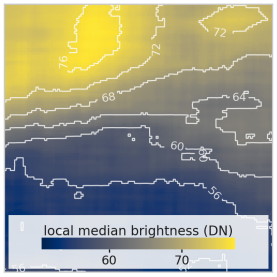
RETROCHANGE
EOSC Federation

UC2_Poloniny_NB2_Havesova_canopy_gaps_methods.ipynb



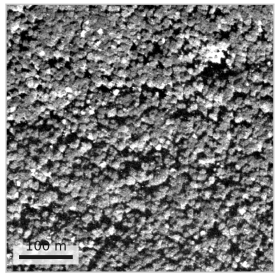
1 The image as it comes

KH-9 HEXAGON, ~1975, 0.5 m. Sunlit crowns are bright and the spaces between them dark — but so is the whole shaded side of the hill. Brightness alone cannot tell the two apart.



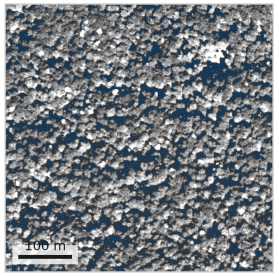
2 How bright the ground is here

The median of every pixel within 150 m. This is the terrain shading, measured from the image itself: 27 DN across this square — larger than the difference between a crown and a gap.



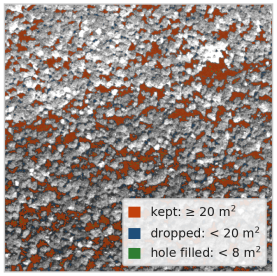
3 Each pixel against its own neighbourhood

anomaly = (DN − median) ÷ median^0.5. The hillside is gone and the canopy is left: openings now look the same whether they sit in sun or in shade.



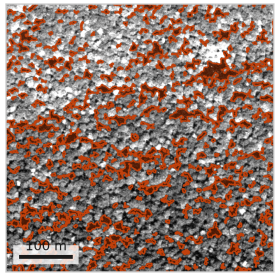
4 Darker than the 15% mark on the brightness range

The one number set by hand. The mark sits 15% of the way from p5 to p95 of the brightness range, here at -2.82 — a place on the brightness axis, not a share of the pixels, so two forests can differ. Everything below it is a candidate — real openings, but also single shaded pixels wedged between crowns.



5 Sieving the candidates

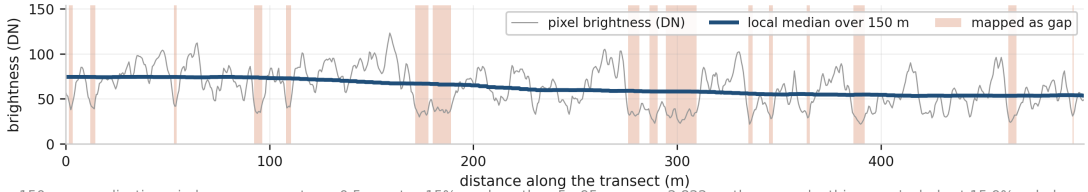
Holes under 8 m² are filled, so one sunlit crown cannot split a gap in two. Patches under 20 m² are dropped: below that size a dark patch is shadow between crowns, not an opening in the canopy.

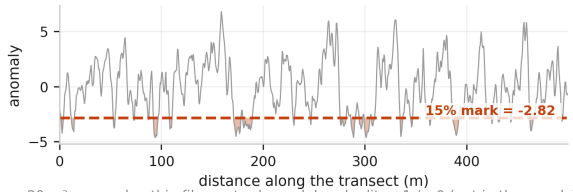


6 Canopy gaps, ≥ 20 m²

Written out as polygons carrying area, perimeter and shape. The exported map is then cleaned of thin shadow filaments and edited by hand (1 added, 0 erased), not drawn here.

Why step 2 is needed — the single line of pixels marked on panel 1. The ground it crosses darkens by 21 DN from end to end, as much as separates a crown from a gap, so the same opening reads darker at one end than at the other. Dividing by the local median (right) puts every opening on the same footing, and a single threshold then works along the whole line.





150 m normalisation window · exponent α = 0.5 · cut = 15% mark on the p5-p95 range = -2.822 on the anomaly, this scene's darkest 15.9% · holes < 8 m² filled · gaps ≥ 20 m² mapped · thin filaments cleaned, hand edits +1 / −0 (not in the panels)

The method shown on one 500 × 500 m square of Havešová. The supplied figure illustrates the processing steps and settings documented in the analysis notebook.

4.1 · Gaps are dark, not bright

A canopy of height *H* with the sun θ above the horizon casts a shadow *H*/tan θ long. A summer mid-morning KH-9 pass at 49° N puts the sun 45-55° up, so a 45 m beech canopy casts 32-45 m of shadow. Most beech gaps are 20-500 m² - 5 to 25 m across - far below that, so **essentially every gap has no sunlit floor left in a near-nadir view and reads as a dark patch.**

Measured on this scene: the brightest 5 % of pixels is *rougher* than canopy, i.e. sunlit crowns. There is no bright, smooth "gap floor" population to detect.

4.2 · The terrain has to go first (panels 2-3)

Across this reserve the *topographic* brightness swing is larger than the crown-versus-shadow signal being looked for. So brightness is expressed relative to its own surroundings:

$$anomaly = (DN - m_{150m}) / m_{150m}^{\alpha} \text{ with } \alpha = 0.5$$

`m` is the 150 m nodata-aware moving **median** - not a mean, because a mean is dragged down by the gaps inside its own window, so gappy ground gets recorded as shaded ground and the correction partly erases what it is meant to reveal.

The transect at the bottom of the figure is the argument in one line: along 500 m of forest the ground darkens by 21 DN, as much as separates a crown from a gap, so the same opening reads darker at one end than at the other. After dividing by the local median, one threshold works along the whole line.

> *Brightness is deliberately **not** normalised by the local standard deviation. That statistic is inflated by the very gaps being looked for.*

4.3 · The detector is two tests (panels 4-5)

```
gap  =  anomaly < cut            # dark
     &  connected area >= GAP_MIN_AREA_M2   # large enough, after filling holes < 8 m²
```

That is all of it. **No width criterion, no shape criterion, no texture criterion** - each was tested during development and none excluded anything the size floor had not already excluded, while a texture criterion actively made the map worse (§9). The size floor is also what disposes of the inter-crown shadow seams: a seam is far too narrow to reach 20 m² once small holes have been filled.

Afterwards, **thin material is cleaned off**. A dark component often reaches out along the shadow seam between two crowns, so two real openings arrive joined by a filament a metre wide that no field protocol would call gap. Three kinds of thin part are distinguished and only two removed - a **stripe** (≥ 2 m long) is deleted, a **waist** between two lobes is cut so the gap becomes two, and a **sliver** shorter than that is *kept*, because those are the rounded corners of ordinary gaps. At Havešová this takes 0.57 % of gap area and splits 130 gaps.

5 · The decisions that are yours

The method cannot make these for you. All three are recorded in every output file.

5.1 · The smallest gap you will map - `GAP_MIN_AREA_M2`

Default **20 m²**, Brokaw's (1982) convention. This is the **strongest single control on gap density**, and almost none on gap fraction - small openings add count, not area:

min gap (m²)	n gaps	gaps/ha	gap fraction %	mean gap m²	median gap m²
4	4 093	28.29	15.68	55	18
10	2 765	19.11	15.11	79	32
20	**1 895**	**13.10**	**14.26**	**109**	**53**
50	996	6.88	12.28	178	113
100	567	3.92	10.23	261	187

Density moves seven-fold, gap fraction by a third. **Never quote a gap density without the size floor beside it** - this is one reason published beech gap densities are not comparable with each other.

5.2 · The mark - how dark counts as a gap - `GAP_RULER_FRAC`

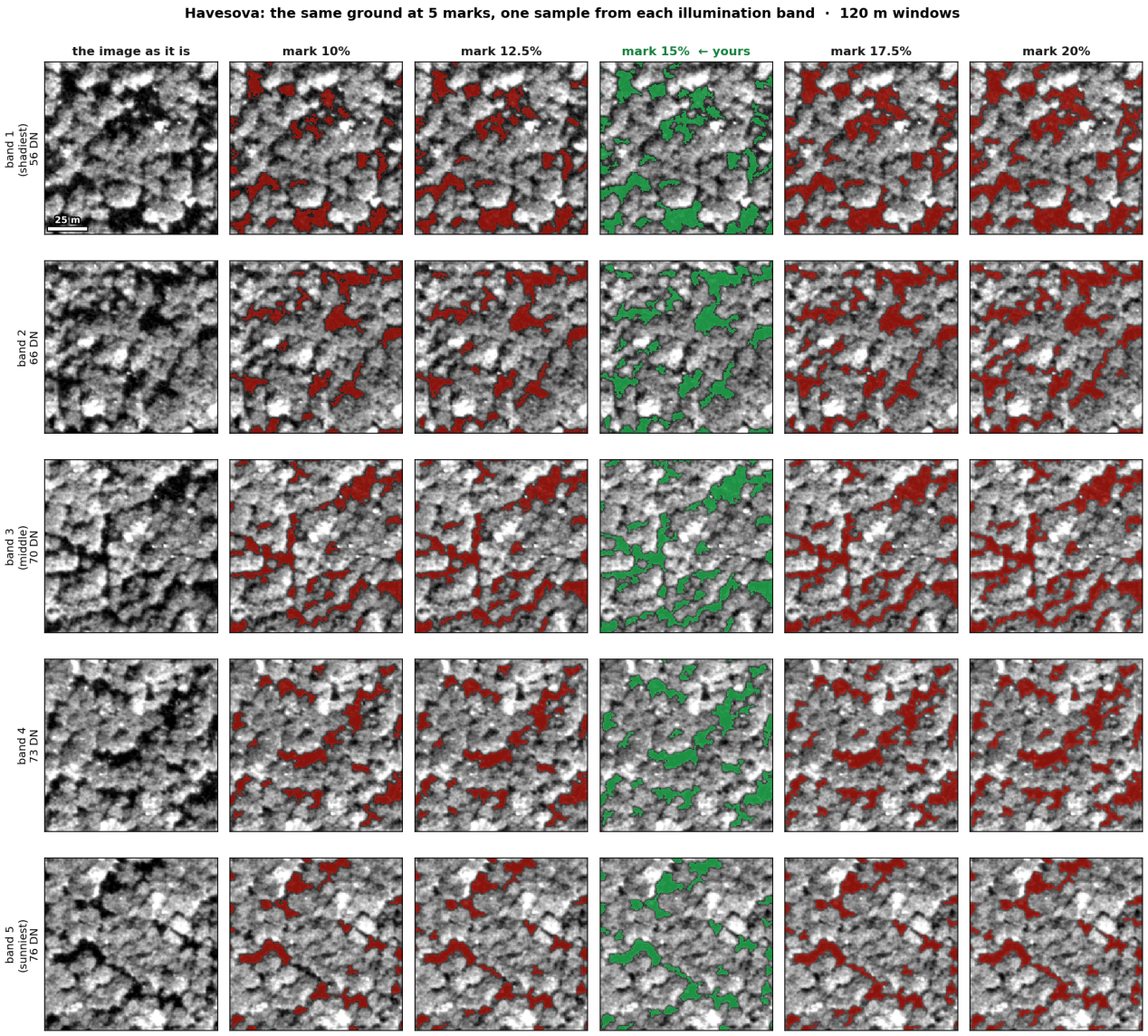
Default **0.15**. This sets the **gap fraction**. The cut is placed on the brightness axis:

> *cut = p_5 + 0.15 × (p_{95} - p_5) of the illumination-corrected brightness range → -2.822 at Havešová, which happens to be this scene's darkest 15.95 %*

Why a position on the brightness axis and not a percentile. "The darkest 15 % of pixels" returns a gap fraction near 13 % in *every* scene by construction, so it can never say that one forest is gappier than another - the statistic is incapable of expressing the difference, whether or not anyone sets out to measure it. A value-anchored mark can: the same physical cut is applied to both scenes, so the **difference** between them is interpretable. Across three Poloniny cores in a first pass, the anchored rule separated them (Rožok 19.8 %, Stužica 14.4 %, Havešová 14.3 %) where the percentile rule returned 12.6-13.3 % for all three - the rule talking, not the forest.

Why anchors and not the extremes. Each extreme is a single pixel and the bright end is sunlit crown highlight and film artefact; a min-max ruler swung 26-fold between reserves. Keep the anchors tight - p1/p99, p2/p98 or p5/p95. Loosening past p10/p90 makes neighbouring gaps **merge**, which shows up as gap density falling while the largest "gap" runs away.

How to choose it: look, do not argue. §7 of the notebook draws one 120 m window from each illumination fifth of your scene - the shadiest ground through to the sunniest - at your mark and its four neighbours, with your column highlighted in green:



Read it as a photo-interpreter would:

- **too low** (left) → only the black cores of the largest openings are caught; small gaps are missed and outlines sit inside the visible opening;
- **too high** (right) → outlines spill past the opening into the shadow around it, and the seams between crowns start joining up into false gaps;
- **about right** → the outlines follow the edge you would have drawn by hand, in the shade as well as in the sun.

The same decision in numbers, written to `*_mark_preview.csv`:

mark %	anomaly cut	this scene's darkest %	gaps	gaps per ha	gap fraction %	mean m²	median m²	
10.0	-3.223	12.04	1 714	11.85	10.29	86.9	48.2	
12.5	-3.022	14.00	1 801	12.45	12.29	98.7	51.8	
15.0	**-2.822**	**15.95**	**1 895**	**13.10**	**14.26**	**108.9**	**52.8**	**← yours**
17.5	-2.621	17.92	1 969	13.61	16.27	119.5	56.0	
20.0	-2.421	19.83	2 025	13.99	18.17	129.9	58.0	

Notice what moves and what does not: gap fraction nearly doubles across the sheet, **density moves by a fifth**. What the mark mostly changes is how far each gap spreads into the shadow around it, not whether the gap is found at all.

This is a loop, and running it once is not the end of it. The sheet is also written to `outputs/<prefix>_00_mark_preview.png`, so you can open it beside the raster in QGIS. If a column other than the green one is the map you would have drawn by hand, put **that column's percentage** into `GAP_RULER_FRAC` in §4 of the notebook - as a fraction, so the 17.5 % column is `0.175` - and re-run from §7. Nothing below that point depends on anything but the value it produces.

Judge it on the shaded rows as well as the sunlit ones: a mark that looks right in the sun and too generous in shade is too high.

5.3 · Whether to hand-correct - `ADD_GAPS` / `REMOVE_GAPS`

Optional; the workflow is complete without it. See §8.

5.4 · One setting to check on a new site - `CANOPY_HEIGHT_M`

Default 45 m, measured in Havešová (Drössler & von Lüpke 2005). It is the yardstick for the distance-to-gap figures and for the shadow argument in §4.1. Set it from whatever stand-height figure you can defend, and say where it came from.

6 · Reading each result

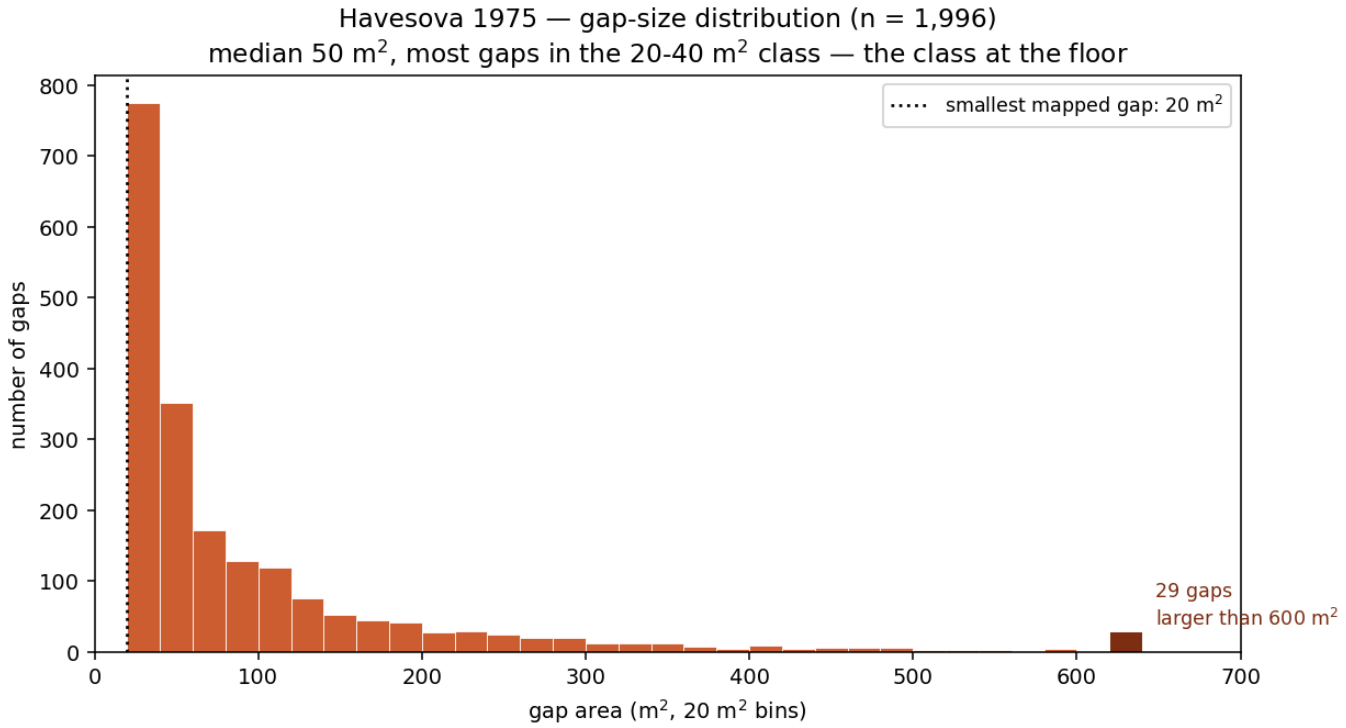
6.1 · How much, and how many

The notebook's §8 prints the headline block, and `*_stats.csv` records it together with every parameter that produced it:

```
CANOPY GAPS at Havesova, 144.7 ha analysed
  number       : 1,996
  density      : 13.79 gaps per hectare
  gap fraction : 13.96% of the canopy
  mean / median: 101 / 50 m2
  p90 / largest: 227 / 2690 m2
  shaded/sunlit: 1.33
```

The last line is a check, not a result: the ratio of gap fraction on the shadiest fifth of the reserve to the sunniest. The illumination bands played no part in choosing the threshold, so it is an independent measure of how far the mark leans towards shaded ground (§7.3).

6.2 · How big - firm by count, convention-bound by area



size class (m²)	n gaps	% of gaps	gap area (ha)	% of gap area	% of the reserve
0-100	1 426	71.4	6.24	30.9	4.31
100-200	333	16.7	4.61	22.8	3.19
200-500	196	9.8	5.82	28.8	4.02
500-1 000	32	1.6	2.13	10.6	1.47
1 000 and larger	9	0.5	1.40	6.9	0.97

Three columns, three different questions, and they are easy to confuse. **% of gaps** is dominated by the small ones. **% of gap area** is the disturbance-severity signal. **% of the reserve** sums to the gap fraction, so it says

directly how much of the stand was in gaps of each size.

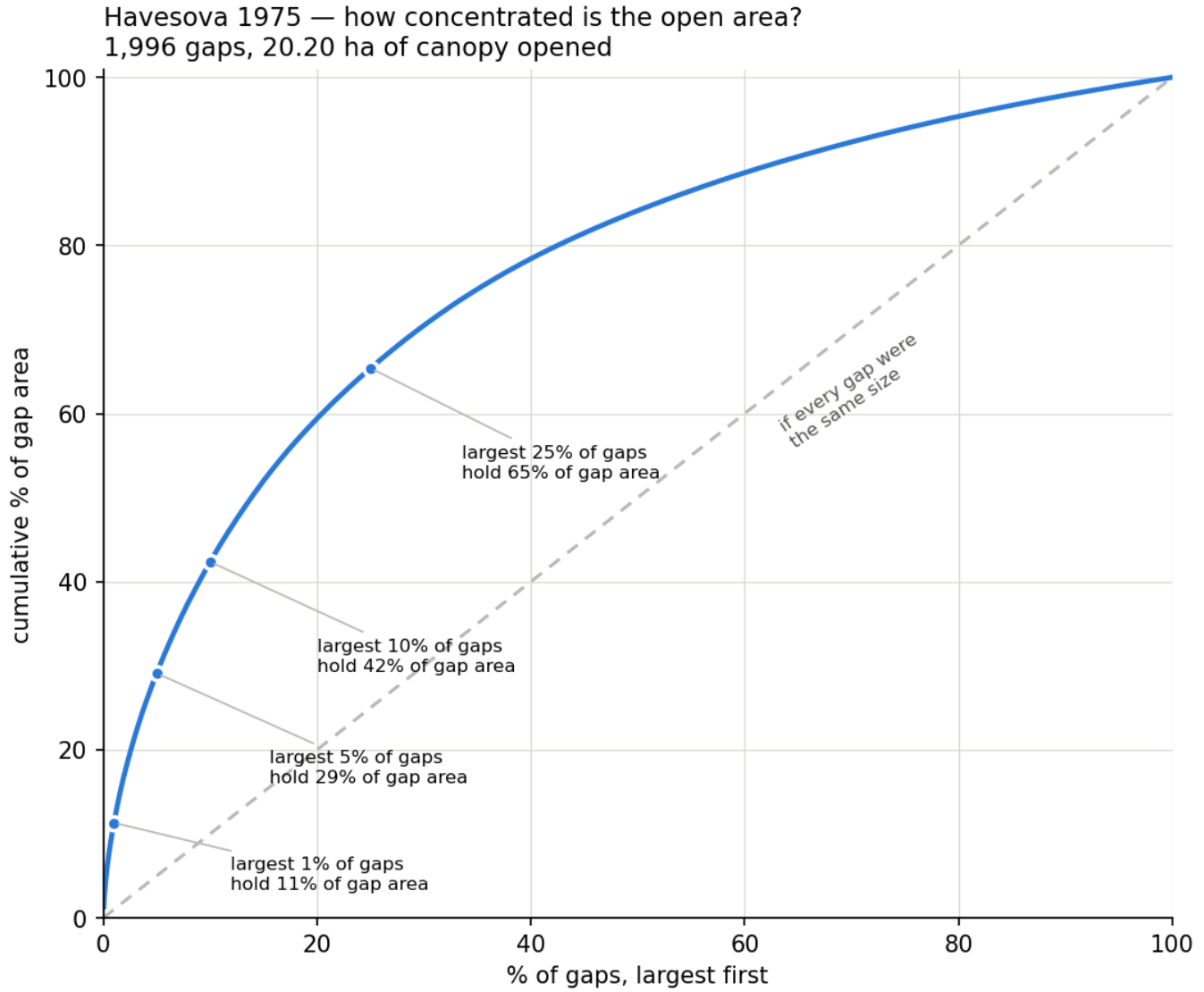
The reading at Havešová: **71 % of the openings are under 100 m² but deliver only 31 % of the opened canopy**, while the nine gaps over 1 000 m² - half a percent of the count - deliver 7 %. A canopy opened a few trees at a time, with nothing near stand-replacing.

The histogram above has no peak. Its tallest bar is the leftmost one, hard against the 20 m² floor, and that bar marks where the counting stops rather than the commonest gap size: lower the floor and the count keeps climbing - 1 895 gaps at 20 m², 2 765 at 10 m², 4 093 at 4 m² - with the median gap following it down, 53 → 32 → 18 m². Nothing in the image says whether the real distribution turns over at some smaller size, so no size read off this figure is "the typical gap", and every count must be quoted with the floor that produced it.

How firm is this? Coarse classes are used because they are what survives the choices that move this map: a bin wide enough to sit clear of the floor does not move when the floor does. But coarse bins do not escape **outline error**, and it is worth knowing which columns of that table are firm. Moving the mark ±5 percentage points changes:

	shift, as % of its own value
number of gaps	16 %
median gap size	18 %
count share of the 0-100 m² class	**10 %**
count share of the 100-200 m² class	13 %
area share of the 200-500 m² class	16 %
area share of the 500-1 000 m² class	81 %
area share of the 1 000 m²-and-larger class	**120 %**
mean gap size	40 %
half-area size	64 %

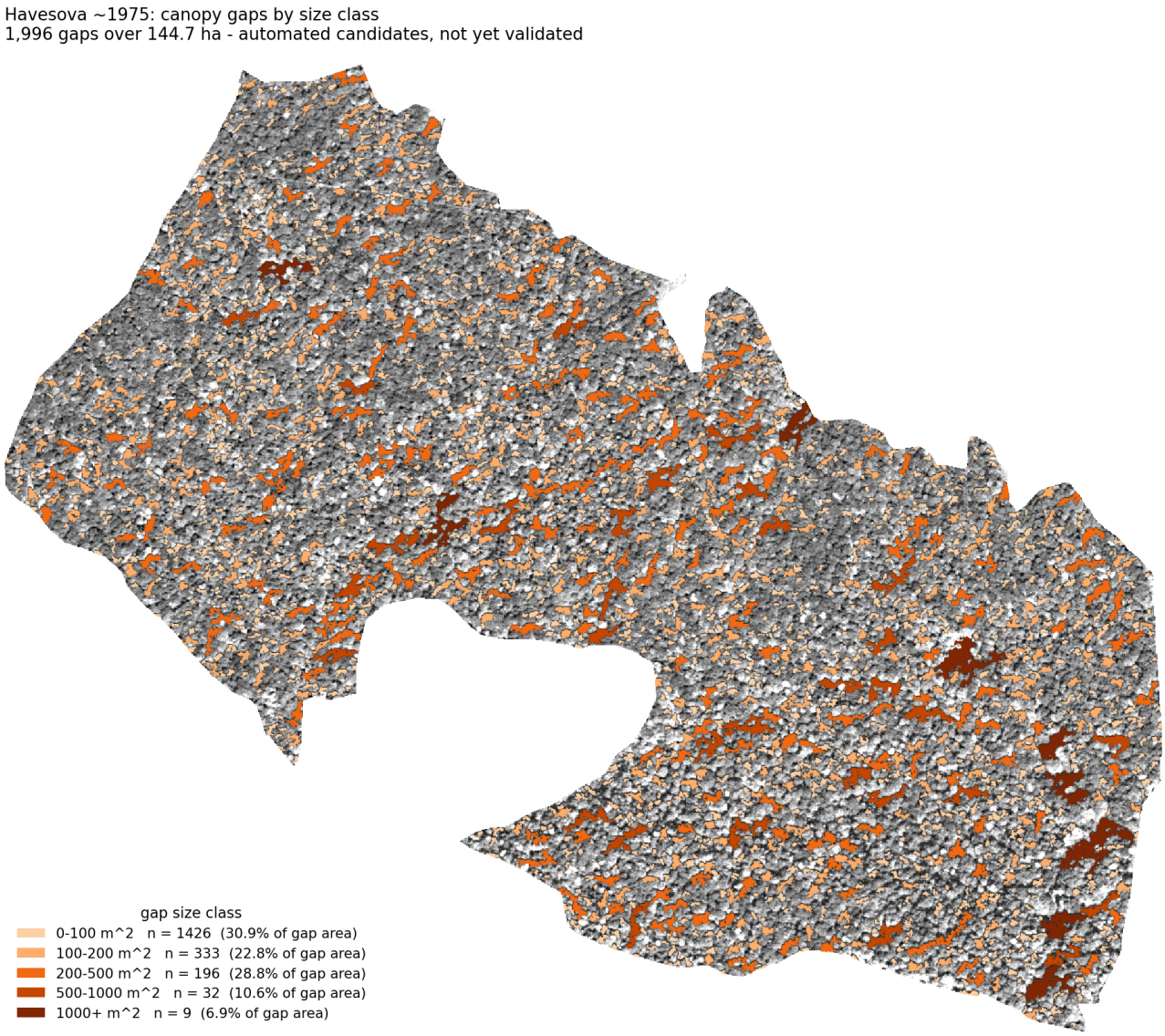
The mark decides how far each opening spreads into the shadow around it, and in absolute terms that lands almost entirely on the large-gap tail. **Read the count columns as the firm result and the area columns as carrying the mark's convention**, exactly as the gap fraction does - comparable between scenes run at identical settings, not quotable as a level on their own.



The concentration curve orders gaps largest first against the share of gap area they have delivered. A straight diagonal would mean every gap is the same size; the further the curve bows above it, the more the open area is the work of a few large events. **This is close to floor-free** - the small gaps the 20 m² floor discards carry almost none of the area - which makes it the sturdiest size result available.

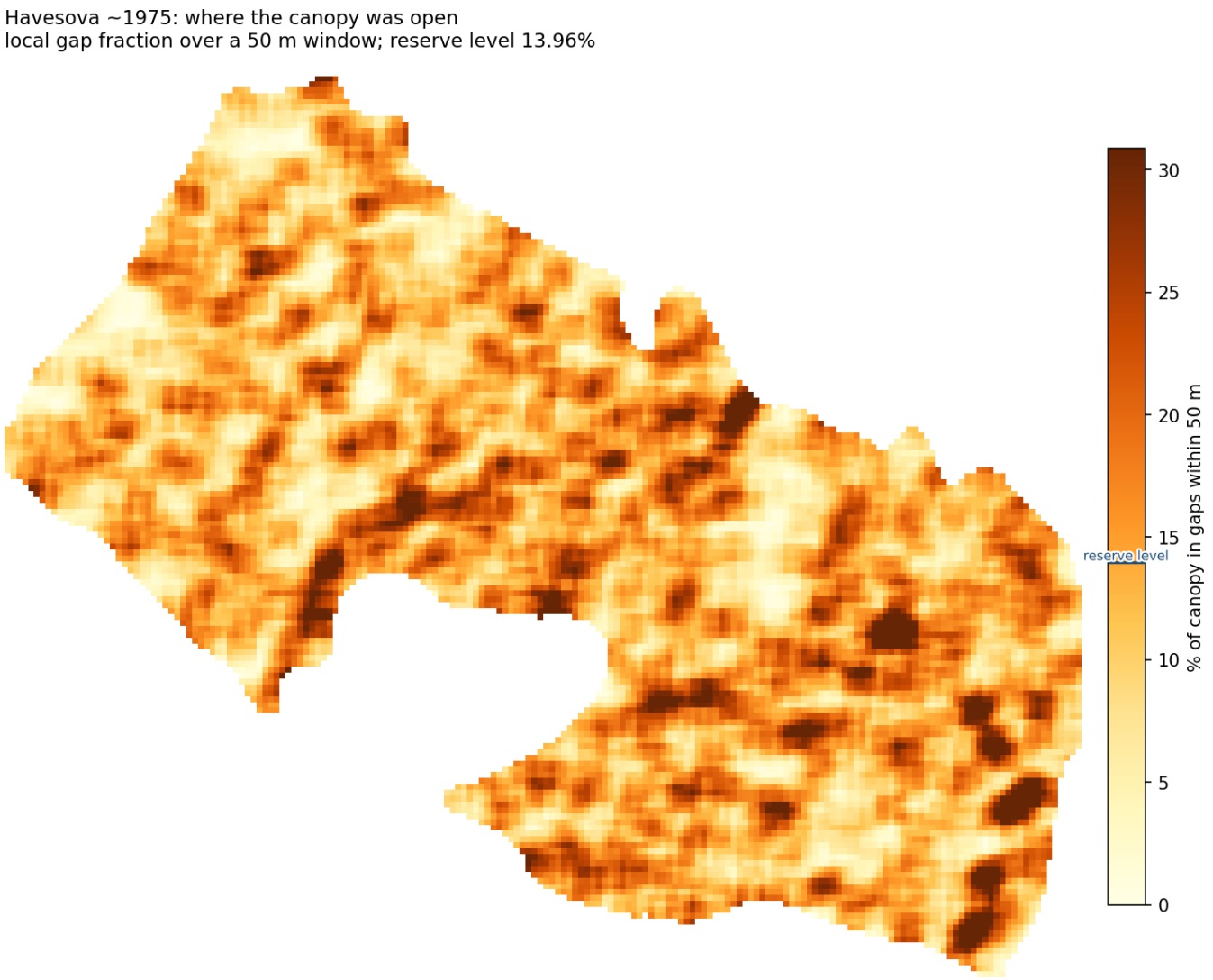
statistic	value	what it means
median gap	50.5 m²	midpoint of the count; follows the size floor, so compare only at equal floors - held to that, it barely differs between reserves
half-area size	183.1 m²	half of all gap area is in gaps smaller than this. It separates sites well, but its *level* moves 64 % across a ±5-point swing of the mark - compare it only at identical settings
mean gap	101.2 m²	pulled by the tail; do not quote alone
p90 / p99 / largest	227 / 689 / 2 690 m²	the tail itself
largest 5 % hold	29.0 % of gap area	concentration, read off the curve

6.3 · Where the large gaps are



The plain gap map answers "where are the gaps"; this one answers "where are the **large** gaps", which is the more useful question given that 71 % of the openings contribute under a third of the area.

6.4 · Where the canopy was open - the manager's layer



The share of canopy in gaps within a moving 50 m window (0.25 ha, about three openings at this density), written out as `*_opening_surface.tif` at 10 m. This is the layer to put in front of a protected-area manager, and the layer to intersect with anything else you hold - a management compartment, a windthrow record, an exposure map.

where	% of canopy in gaps within 50 m
reserve as a whole	13.96
least open 5 % of the reserve	3.99
lower quartile	9.11
median place	13.32
upper quartile	17.96
most open 5 % of the reserve	26.15

3.5 % of the reserve lies in ground more than twice as open as the reserve average, and 15.0 % in ground less than half as open.

Read the pattern, not the level. The level carries the mark's convention exactly as the gap fraction does; where the open ground *is* does not move with it. The dark line running across the middle of the reserve is a ravine - a real topographic feature, visible here without any elevation model being used.

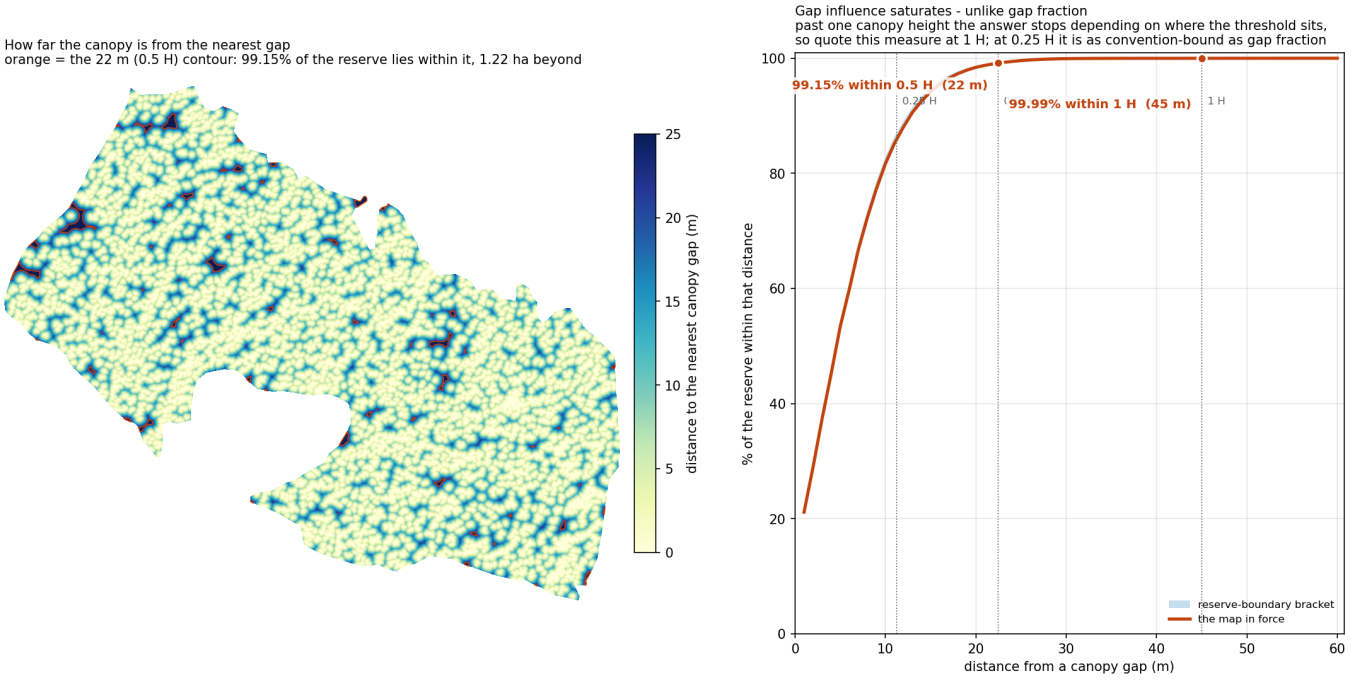
6.5 · Does it change towards the reserve boundary?

distance from boundary (m)	area (ha)	gap fraction %	gaps	gaps per ha	median gap m²
0-10	6.7	10.76	109	16.16	36.5
10-25	10.0	13.91	153	15.36	57.8
25-50	15.5	14.01	204	13.15	49.4
50-100	27.8	14.57	372	13.39	49.0
more than 100	84.7	14.01	1 158	13.67	53.0

Two reasons to ask. **Ecologically**, an old-growth core embedded in managed forest may be more disturbed near its edge. **Methodologically**, if the numbers were driven by the analysis edge rather than by the forest, this is where it would show.

At Havešová the profile is flat - 13.9-14.6 % from 10 m inwards - so neither is happening. The outermost 10 m reads lower (10.8 %) because gaps there are clipped by the outline, which is also why its median gap is the smallest in the table.

6.6 · How far the canopy stood from an opening



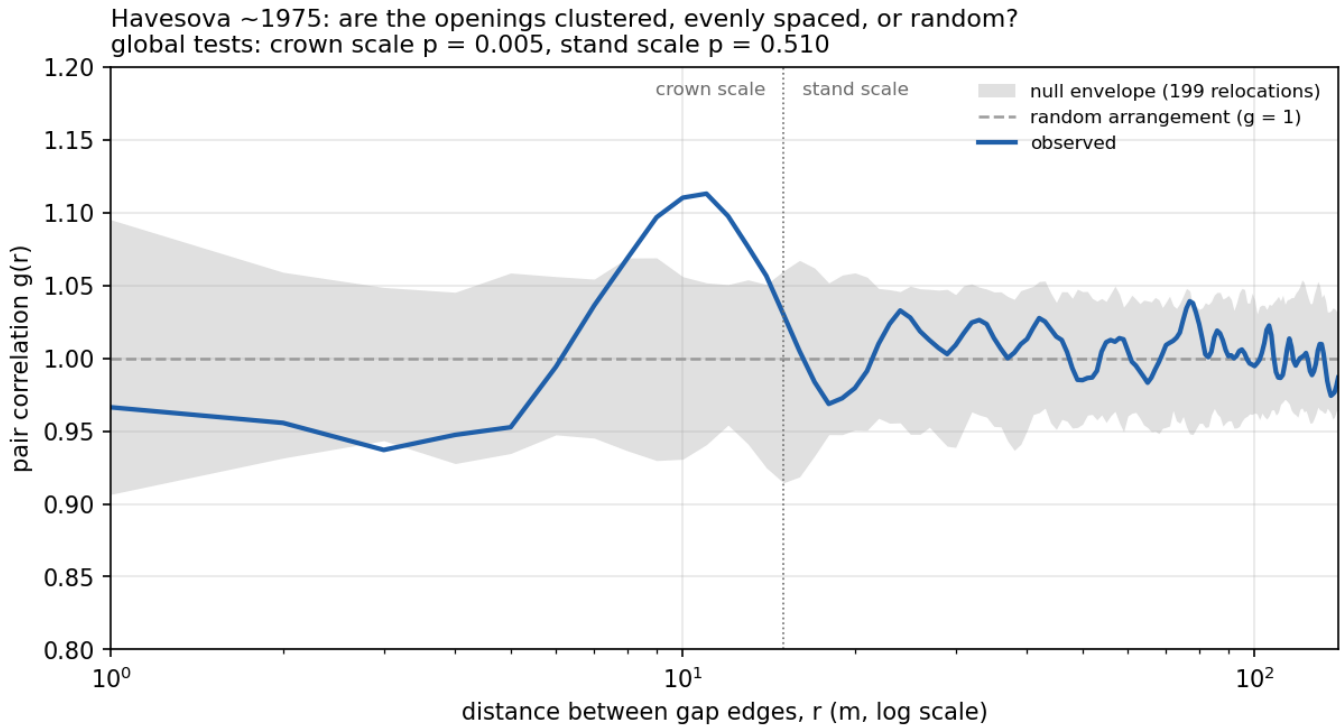
Gap fraction says what share of the canopy **is** a gap. It is not what governs the forest floor: a tree standing 5 m from a 60 m² opening lives in a different light, temperature and moisture regime from one standing 40 m away, even though neither is *in* a gap. The quantity that describes that is the **distance from every canopy point to the nearest opening**, reported in units of canopy height *H*.

Why this one is worth reporting when gap fraction is not. It **saturates**: past one canopy height the curves for every threshold from the darkest 8 % to the darkest 25 % converge, because at that distance it no longer

matters whether a borderline dark patch was called a gap - some other gap is nearer anyway. At Havešová the 1 H figure is 99.99 % and spans only 98.1-100 % across those cuts. At a quarter of a tree height it is 87 % and spans 58.8-96.8 %, which is as convention-bound as gap fraction. **Quote it at 1 H, not closer.**

Saturation cuts both ways, and with *H* = 45 m it cuts hard: no point in this reserve is more than about 59 m from a mapped opening, so the 1 H figure is robust largely because it sits close to its own ceiling. Read it as **"this reserve holds no interior remote from disturbance"** - a real and reportable property of the stand - and not as a sensitive index that would separate two old-growth beech forests. These are distances, not measured effects: nothing in the workflow measures light or regeneration.

6.7 · Were the openings clustered, evenly spaced, or random?



In beech old-growth, new openings are often reported to form at the margins of old ones. If that happened here as *separate* openings, gaps would sit closer together than chance allows. A random arrangement looks clumpy to the human eye, which is exactly why the test exists.

The **pair-correlation function g(r)** counts every pair of gaps at each edge-to-edge distance: g = 1 is random, g > 1 clustered, g < 1 regularly spaced. The method follows Nuske et al. (2009), adapted for gaps as polygons, against 199 random relocations of the same polygons inside the same outline.

- **Stand scale, 15-140 m: random** (global DCLF test p = 0.415). No clustering and no regular spacing - Havešová's gaps are not grouped into gappy and closed patches of forest at these scales. A non-rejection is **consistent with openings forming independently, never proof of it**, and it says nothing about *why* a gap formed where it did.
- **Crown scale, below 15 m: structured** (p = 0.005) - but this is where the detector and the forest cannot be told apart. Where only a few metres of canopy separate two openings, that canopy is often in shadow too and the pair is mapped as one gap. Treat the stand-scale range as the reportable result.

7 · How much of it is convention

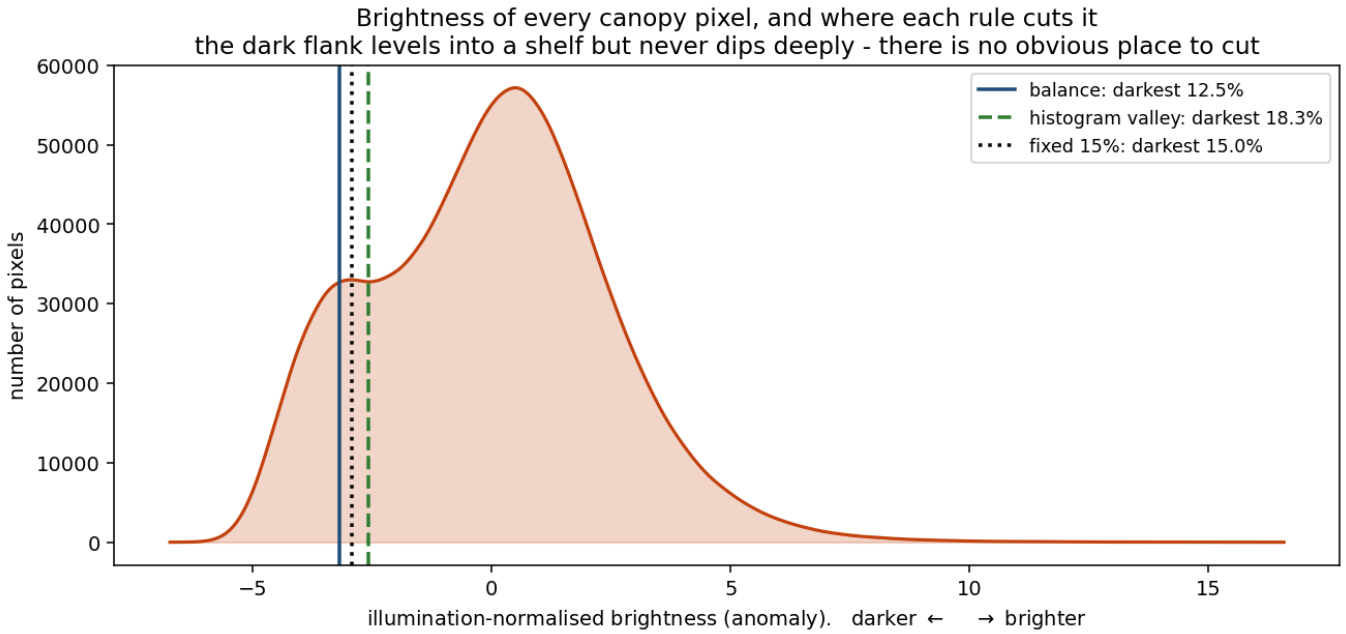
This is Appendix C of the notebook, and the part a reviewer will go to first.

7.1 · The threshold never settles

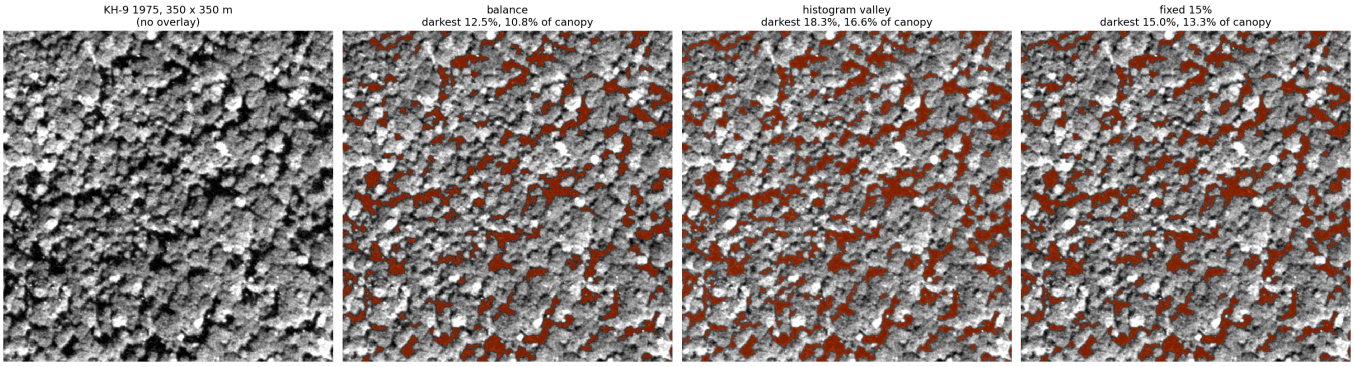
darkest %	gaps/ha	gap fraction %	discovery share	d(gap fraction)/d%
4	4.91	2.55	0.59	0.82
8	8.89	6.25	0.36	0.94
12	11.83	10.25	0.27	1.02
16	**13.10**	**14.31**	**0.20**	**1.01**
20	14.04	18.35	0.19	1.00
24	14.38	22.30	0.15	0.99
28	14.22	26.25	0.14	0.98

Gap fraction moves **0.99 points per percentile of pixels** - essentially one for one, with **no plateau anywhere**. Every percentile of newly darkened canopy ends up inside a gap. Gap density, by contrast, rises, flattens and turns over: the peak is not a natural threshold, it marks where dark patches fuse into one connected shadow network and the map stops resolving individual openings.

Appendix A of the notebook is the full account. Six ways of deriving such a threshold from the image were tried - a histogram trough, a stability plateau, the gap-count maximum, the discovery share, the curvature knee of the dark flank, and an explicit two-component mixture model - and **none locates a boundary**:



The dark flank levels into a *shelf* - a darker population really is present - but it never dips into its own mode, and a mixture fit prefers three or more components over two. The reason is physical: **a gap floor and the shadow between two crowns are both lit by diffuse skylight alone, so they are radiometrically the same surface.** What the threshold sets is how much of the shadow *around* each opening is counted as part of it - a matter of gap definition, exactly the thing a field protocol has to state explicitly rather than derive.



Same forest, same stretch, three candidate rules. What changes between the panels is mostly how far each gap spreads into the shadow around it, not whether the gap is found at all - which is why density barely moves while gap fraction nearly doubles.

7.2 · The lean towards shaded ground

illumination band	mean brightness DN	area ha	gap fraction %
Q1 (most shaded)	55.7	28.9	14.28
Q2	65.0	29.0	**17.06**
Q3	69.5	28.8	15.76
Q4	73.0	29.0	11.98
Q5 (most sunlit)	76.9	29.0	10.74

Read this at both ends **and** in the middle. The ends measure how far the mark leans towards shaded ground: 1.33 here, where the balance rule of Appendix A would force 1.00 by construction - at the cost of assuming shaded ground has no more gaps, which is exactly what cannot be assumed.

The **hump in the middle** is not an illumination artefact, because illumination bias is monotonic in illumination by construction. Topographic position - mid-slopes genuinely more disturbed than crests or valley floors - is the likely reading, and separating it from anything else would need a terrain model.

8 · The review loop

The detector produces candidates. A trained interpreter beats it on the ambiguous cases, and the workflow is built to accept that correction rather than hide it.

1. **Run once.** You now have `outputs/<prefix>.gpkg` (the gaps) and `outputs/<prefix>_anomaly.tif` (the illumination-corrected image).
2. **In QGIS, open the anomaly raster first**, then the gap layer over it. Judge outlines against the anomaly, **never against the raw photograph** - the raw image is darker on shaded slopes, so a correct outline looks too generous there and too mean in the sun. That is the terrain talking.
3. **Draw the misses** into a new polygon layer, saved as `add_gaps` (GeoPackage layer or shapefile, any CRS - it is reprojected for you).
4. **Draw the over-reach** into a second layer, `remove_gaps`: anything wrongly called gap, or the part of a real gap that has run into the shadow beside it.
5. **Point the notebook at them** in §3 and re-run:

```
ADD_GAPS          = pathlib.Path("data/add_gaps.shp")
ADD_GAPS_LAYER    = "add_gaps"
REMOVE_GAPS       = pathlib.Path("data/remove_gaps.shp")
REMOVE_GAPS_LAYER = "remove_gaps"
```

Both are folded in **before** anything is measured, so every table, figure and export includes them, and the run reports what happened:

```
ADDED BY HAND: 1 polygons from add_gaps.shp (layer 'add_gaps'), 0.05 ha inside the reserve
             1 touch a gap the detector already found and are merged into it; 0 are new gaps
             the size floor and hole filling are not applied to them
```

Rules of the loop

- Draw additions **over the opening**, not across the canopy between two openings - a polygon that bridges two detected gaps fuses all three into one.
- Additions are used **exactly as drawn**: neither the size floor nor the hole filling is applied.
- **Erasing wins.** It is applied last, so where the two layers overlap the erase takes precedence. An erase can trim a gap, split it in two or delete it outright, and each outcome is reported separately.
- Every exported polygon carries `source` (`automatic` / `hand-drawn` / `merged`) and `trimmed`, so the manual contribution can always be separated out again.

> ***Reviewing the detected polygons cannot give you an accuracy figure.*** *It only finds commission error.* ***Omission*** *- the openings the detector never proposed - can only be measured by exhaustively digitising every opening inside a sample of cells, for example ten 100 m squares stratified by illumination, and comparing them against the automatic map. That sampling design is the one addition that would turn these candidates into measurements, and it is* ***not*** *part of the delivered notebook.*

9 · Reporting the numbers

9.1 · The safe order

1. **Gap density**, with the size floor - the sturdiest number. Measured against modelled terrain at Havešová it is flat across aspect (12-14 gaps/ha on every exposure).
2. **Median gap size and the count share of the small classes** - the next sturdiest, 18 % and 10 % across a ±5-point swing of the mark.
3. **Gap influence at one canopy height** - the one headline figure that stops depending on the mark.
4. **Differences in gap fraction** between scenes run at identical settings.
5. **Absolute gap fraction**, always with the rule, the anchors, the mark and the size floor.
6. **Mean gap size, the half-area size and the size classes' *area* shares - last, and with care.** All of them move with the mark as strongly as gap fraction does (40 %, 64 %, up to 120 %), so compare them only between scenes run at identical settings. Mean gap size has a second problem on top - §9.3.

9.2 · The sentence that must travel with the gap fraction

> *Gap fraction of X % (anchored rule, p5/p95, mark 0.15, minimum mapped gap 20 m²).*

Without that parenthesis the number means nothing: published beech gap fractions span 0.75-16 % almost entirely on account of differing gap definitions and detection floors. For the same reason, never quote a published range without both - and **do not read a match with a published figure as validation**, because several different configurations can reach the same number by compensating errors.

§13.1 of the notebook writes a complete methods paragraph with every parameter filled in (`*_methods_statement.md`), so the conventions travel with the numbers automatically:

```
A pixel was called gap where its anomaly fell below a cut placed on the anomaly's own value axis,
cut = p5 + 0.15 x (p95 - p5), giving -2.822 for this scene (15.95% of analysed pixels). Holes
smaller than 8 m2 were filled, thin connecting material was removed (on; stripes at least 2.0 m
long, and waists narrower than 1.0 m, removing 0.57% of gap area), and connected patches of at
least 20 m2 were retained as gaps. 1 hand-drawn polygon(s) covering 0.05 ha were added ...
```

9.3 · Two comparisons that need care

- **Mean gap size between sites.** Within Havešová, mean gap size ran about 48 % larger on the shaded half of the reserve than on the sunlit half, while density stayed flat. Whether that is larger disturbance patches on cool slopes or a longer shadow footprint on slopes facing away from the sun **has not been established**. Either way the consequence holds: compare mean gap size only between sites of comparable aspect composition. Density is unaffected.
- **Arrangement below one crown width** - see §6.7.

9.4 · What "no DEM" means, precisely

No elevation model is read by the notebook, and the workflow runs where none exists. An elevation model was used during development as a **referee** - to decide that the illumination field should be a median rather than a mean, and to check what the map does across aspect. Those checks are cited in the notebook's §4 comments as the *reason for a default*; none of them is an input, and none of their results is reported as a finding of the delivered workflow.

10 · Extending the use case

- **Two localities.** Run the notebook on a second reserve - `SRC` and `LABEL` in §3 are the only edits. **The notebook itself does not compare reserves**: it produces one baseline per scene, and putting two side by side is a separate step. Only identically-configured runs may be compared, and each scene's `*_stats.csv` records the rule, the anchors, the mark and the size floor so that this can be checked.
- **Two dates.** Re-running the same workflow on a current orthophoto of the same stand would give gap persistence, closure and formation rates. **This is not part of the present work**, which delivers a single-date baseline; it is listed because such data would also settle the threshold question, containing as it does gaps known independently of brightness.
- **Cross-border.** The Slovak and Polish sides of the same massif are one forest under two management histories; the workflow treats them identically because it needs nothing but the image.

11 · File index

Every filename begins `canopy_gaps_<label>_`.

For a GIS, or for further analysis

file	use it for
`*.gpkg` (layer `canopy_gaps`)	per-gap area, perimeter, shape index, size class, local illumination, distance from the boundary, provenance. The layer to hand to a colleague.
`*_gap_map.tif`	the binary map; threshold, rule and gap fraction are in its band metadata
`*_opening_surface.tif`	§6.4, the manager's layer
`*_anomaly.tif`	what the detector saw - **review outlines against this**
`*_illumination.tif`	the 150 m brightness field that was divided out

The numbers

file	use it for
`*_stats.csv`	the provenance record: every headline number **and** every parameter. Archive it with the result.
`*_methods_statement.md`	a ready-to-edit methods paragraph
`*_size_classes.csv`, `*_size_summary.csv`	the most robust results
`*_influence.csv`	distance-to-opening at every threshold tested
`*_pair_correlation.csv`	g(r) with its null envelope
`*_threshold_scan.csv`, `*_sensitivity_min_gap_size.csv`, `*_mark_preview.csv`	the sweeps - what each convention costs
`*_threshold_balance.csv`, `*_threshold_comparison.csv`	written by Appendix A

The figures - all publication-scale PNGs, all reproduced in this guide.

Glossary

term	meaning here
anomaly	brightness relative to its own 150 m surroundings: (DN - m) / m^α
illumination field	the 150 m moving median of brightness; at that scale it is terrain shading, not trees
the mark	`GAP_RULER_FRAC` - how far along the anchored brightness range the cut is placed
anchors	the two percentiles of anomaly *value* defining that range (p5, p95 by default)
size floor	`GAP_MIN_AREA_M2`, the smallest patch admitted as a gap
thin-part cleaning	removal of the filaments along which two openings arrive joined through inter-crown shadow
gap influence	distance from a canopy point to the nearest opening, in units of canopy height *H*
half-area size	the gap size below which half of all gap area sits
outline error	uncertainty in *where a gap's edge is*, as distinct from whether the gap exists. The image records shadow, not height, so the boundary the detector finds is where the dark patch ends - and how far that reaches into the shadow beside the opening is what the mark decides. It is the dominant uncertainty in every quantity built from gap *extent*, and barely touches those built from gap *count* (§6.2)
g(r)	pair-correlation function: pairs of gaps at distance *r* relative to a random arrangement
DCLF test	one global test of departure from randomness over a whole distance range

Key references

Brokaw (1982) *Biotropica* 14: 158-160 · Canham et al. (1990) *Can. J. For. Res.* 20: 620-631 · Drössler & von Lüpke (2005) *J. For. Sci.* 51: 446-457 · Loosmore & Ford (2006) *Ecology* 87: 1925-1931 · Nuske et al. (2009) *For. Ecol. Manage.* 259: 107-116 · Runkle (1982) *Ecology* 63: 1533-1546.

Full list, including software citations, in §15 of the notebook.

Verification and diagnostic edge cases (November 2026)

Appendix A.2 compares the threshold candidates available for the current image. If no histogram valley is found, it omits that candidate, reports the omission, and compares the remaining rules by name.

Appendix A.3 samples at most 200,000 finite analyzed pixels without replacement, using the configured random seed. Smaller scenes use all available finite pixels. Fewer than two finite pixels, or a constant sample, produce a skip message. The number of fitted components is limited by sample size and distinct values. These guards apply to this diagnostic; the main workflow still needs a suitable image with detectable candidate openings.

The exported `*_methods_statement.md` follows the active threshold rule: `anchored` reports the anchor formula; `percentile` reports the selected percentile, its selection method, and the anomaly cut. A fixed-percentile run states that the initial share of dark candidate pixels is prescribed; final gap fractions alone do not establish differences in canopy openness.

Run the behavioral checks from the application root in its analysis environment:

```
python -m unittest discover -s tests -v
```

The checks execute the corrected notebook cells directly and complement a complete notebook run. They do not measure classification accuracy.