



# Filtering blips using an R expression

To allow for more flexible blip filtering, you can specify an R logical expression to determine which patches are accepted as blips. Standard patch filtering can be represented as the following logical expression:

```
ns >= min.blip.samples & ( max.blip.samples >= 0 & ns <= max.blip.samples
& area >= min.area & ( max.area >= 0 & area <= max.area
& aspan >= min.aspan & ( max.aspan >= 0 & aspan <= max.aspan
& rspan >= min.rspan & ( max.rspan >= 0 & rspan <= max.rspan
```

which returns a logical vector with one entry for each patch, with TRUE representing patches that pass all filters and so are **blips**. (the highlighted comparisons are present so that you can specify "no max area" by using a negative value for max.area)

For example, to filter out patches that are too irregular in shape, you might use a filtering expression like:

```
perim^2 / (4 * pi * area) < 5
```

which specifies that only a patch whose perimeter is not too large compared to its area is to be accepted. The effect of specifying this filtering expression and checking the *also filter by logical expression* box in the blip processing dialogue is to change the overall filtering expression to:

```
ns >= min.blip.samples & ( max.blip.samples >= 0 & ns <= max.blip.samples
& area >= min.area & ( max.area >= 0 & area <= max.area
& aspan >= min.aspan & ( max.aspan >= 0 & aspan <= max.aspan
& rspan >= min.rspan & ( max.rspan >= 0 & rspan <= max.rspan
& perim^2 / (4 * pi * area) < 5
```

i.e. your filtering expression is ANDed with the other filtering criteria, and a patch is accepted as a blip only if all pieces of the expression are TRUE.

In this example, we square the perimeter so that the index is **scale invariant** (i.e. **dimensionless**): we want two blips of the same shape to have the same index, even if one is twice the size of the other. (If you use the simple ratio of  $\text{perim} / \text{area}$ , the limit "5" depends on the

units in which perim and area are measured. In radR, perim is in metres and area is in  $m^2$ , so the units on "5" would be  $m/m^2$ ). This particular index is always at least 1.0 (achieved only for a circle), and can be interpreted loosely as "how many times more irregular is this blip than a circle". Here, we set a limit of "5 times as irregular as a circle".

**Notes:**

**blip\_filtering\_expression.png**

- **Syntax:** filtering expressions are checked for syntax; if you hit **Enter** after editing an expression, and the field stays yellow, it means there is a syntax error; TODO: add an informative error message.
- **Expression type:** a filtering expression must return a logical vector of the same length as the number of patches, with TRUE indicating the patch is to be accepted as a blip (if it has not already been filtered), and FALSE indicating it is to be rejected. The length requirement is more or less automatically met if the patch variables (described below) are used in the expression.

- **Enabling the expression:** filtering expressions are only effective when *filter blips* is checked **and** the option *also filter by logical expression* is checked
- **Filtering is cumulative:** a filtering expression operates in addition to the usual range criteria: a blip rejected because it has too many samples will not be subsequently accepted by a filtering expression, even if the expression returns **TRUE** for that blip. The filtering expression can be given "supremacy" by making the range criteria very wide; e.g. by setting the minimum sample count to 1 and the maximum to -1.
- **Zone expressions:** each special zone can have its own filtering expression. Even if there is a "global" filtering expression (i.e. one specified in the "blip parameters" dialogue), it does not apply inside a special zone, even if there is no filtering expression for the zone. You can, of course, use the same filtering expression in the special zone as you specify globally. The filtering expression for a zone will be evaluated on the entire set of patches, but the results will only affect the patches in the zone.
- **Patch variables:** certain variable names are special in expressions, because they refer to the properties of patches. These are:
  - **ns:** the number of samples in the patch
  - **area:** the apparent area of the patch (in  $m^2$ )
  - **perim:** the apparent perimeter of the patch (in m)
  - **aspan:** the angular span of the patch (in samples)
  - **rspan:** the radial span of the patch (in samples)
  - **x, y, z, t:** the coordinates of the blip centroid in metres and seconds
  - **max:** the maximum intensity of any sample in the patch (scaled to the range [0, 1])
  - **int:** the mean intensity of samples in the patch (scaled to the range [0, 1])
  - *Technical note:* these patch variables actually refer to "extmat" objects, which normally require you to use "[]" after their names to obtain their values, but radR rewrites the expressions to include the "[]" so you don't have to.

## zone\_parameters\_filter\_expression.png

- **Name collisions:** (TEMPORARY WORKAROUND): to use the R functions `t()` or `max()` in a filtering expression, you must give them new names outside of the expression, and use those new names inside the expression:
  - in the file **radR/startup.windows.R** or **radR/startup.unix.R**, add these lines:

```
TFUN <- t          ## give "t" (transpose) a new name
MAXFUN <- max      ## give "max" a new name
```
  - in your filtering expression, use the new name:

```
MAXFUN(abs(x), abs(y), abs(z)) < 1000
```
- **Filtering history:** filtering expressions are saved in blipmovies. If you wish to see the expression used in creating a blipmovie, pause the movie and enter this into the console window:

```
RSS$scan.info$blip.filter.expr
```
- **Refiltering:** if you re-record a blipmovie with a new filtering expression, any original filtering expressions are saved in the new blipmovie in the "prev" attribute of

```
RSS$scan.info$blip.filter.expr
```

so that a complete history of filtering is present in the blipmovie.