

Channels to channels in any ratio or combination

The `ratio` and `linear-combination` operations of `cv-cat` have been extended to support assignment to multiple channels. Previously, these operations would take up to 4 input channels (symbolically always named `r`, `g`, `b`, and `a`, regardless of the actual contents of the data) and produce a single-channel, grey-scale output. Now you can assign up to four channels:

ratio syntax

```
... | cv-cat "ratio=(r-b)/(r+b),(r-g)/(r+g),r+b,r+g"
```

The right-hand side of the `ratio` / linear combination operations contains comma-separated expressions defining each of the output channels through the input channels. The number of output channels is the number of comma-separated fields, it may differ from the number of input channels. As a shortcut, an empty field, such as in

ratio syntax shortcut

```
... | cv-cat "ratio=,r+g+b,"
```

is interpreted as channel pass-through. In the example above the output has three channels, with channels 0 and 2 assigned verbatim to the input channels 0 and 2 (`r` and `b`, symbolically), and the channel 1 (symbolic `g`) assigned to the sum of all three channels.

As yet another shortcut, `cv-cat` provides a *shuffle* operation that re-arranges the input channels without changing their values:

shuffle syntax

```
... | cv-cat "shuffle=b,g,r,r"
```

In this case, the order of the first 3 channels is reversed, while the former channel `r` is also duplicated into channel 3 (alpha). Internally, shuffling is implemented as a restricted case of linear combination, and therefore, other usual rules apply: the number of output channels is up to 4, it does not depend on the number of input channels, and an empty field in the right-hand side is interpreted as channel pass-through.