interface patterns

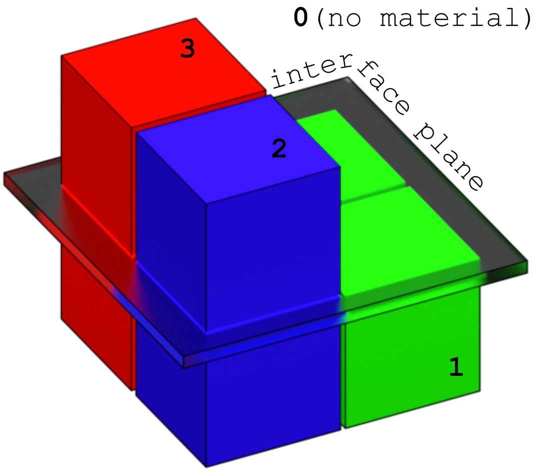

Fig.1 Interesting case

For interfaces nearly the same algorithm as one used for simple bitmaps can be used. In case of interface top m1 and bottom m2 materials can be pacjed into one integer, say the following way: m = m1 + (m2 << 8), further we will mark this int as m2:m1.

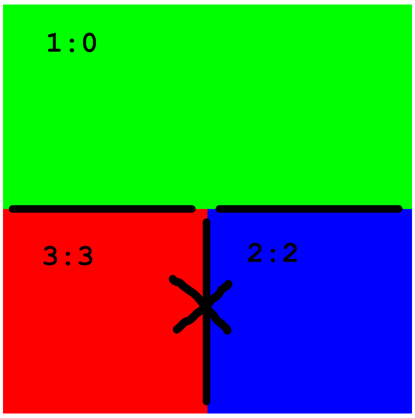

Fig.2 Extra edge

If we will simply mark interface pixels as m2:m1 then in case shown in Fig. 1 for interface we will get Fig. 2 and thus one extra edge will appear between red and blue areas, where we actually do not expect to have any face since red and blue block goes through interfaces under investigation.

Now we can set red and blue areas as void (-1) but in this case we will miss nodes - see Fig. 3. Missing node is marked with red circle in Fig. 3, since nothing change around that node edge is success fully merged, but obviiuosly a vertical edge should go through that missing node.

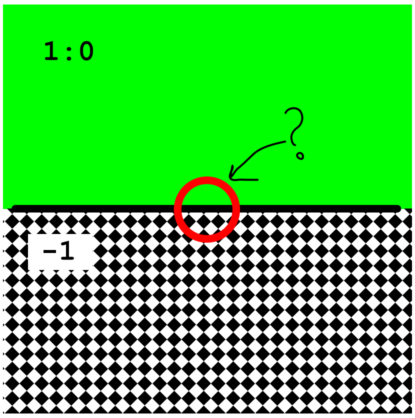

Fig.3 Missing node

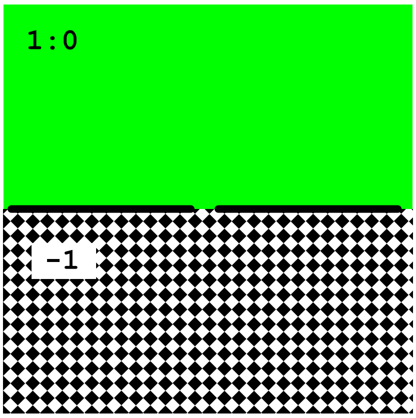

Fig.4 Desired result

To achieve desired result we need to use both representations from Fig. 2 and Fig. 3. Let us call materials from Fig. 3 'altered', so we have packed material *m* and altered packed material *ma*. if both parts of *m* are equal then *ma* is -1. Now we start new edge only if both adjacent pixels *ma*-s are not void (-1) and adjacent pixels *m*-s are not equal.
We finalize edge if one of adjacent *m*-s changed. This keeps missing node and does not create any edge in void area (where faces do not exist).