Recognising Geometric Patterns - Example 15

Peter Fletcher, 6th October 2009.

The problem

Figure 1 shows an image containing two crosses overlapping with two hexagonal tessellations.

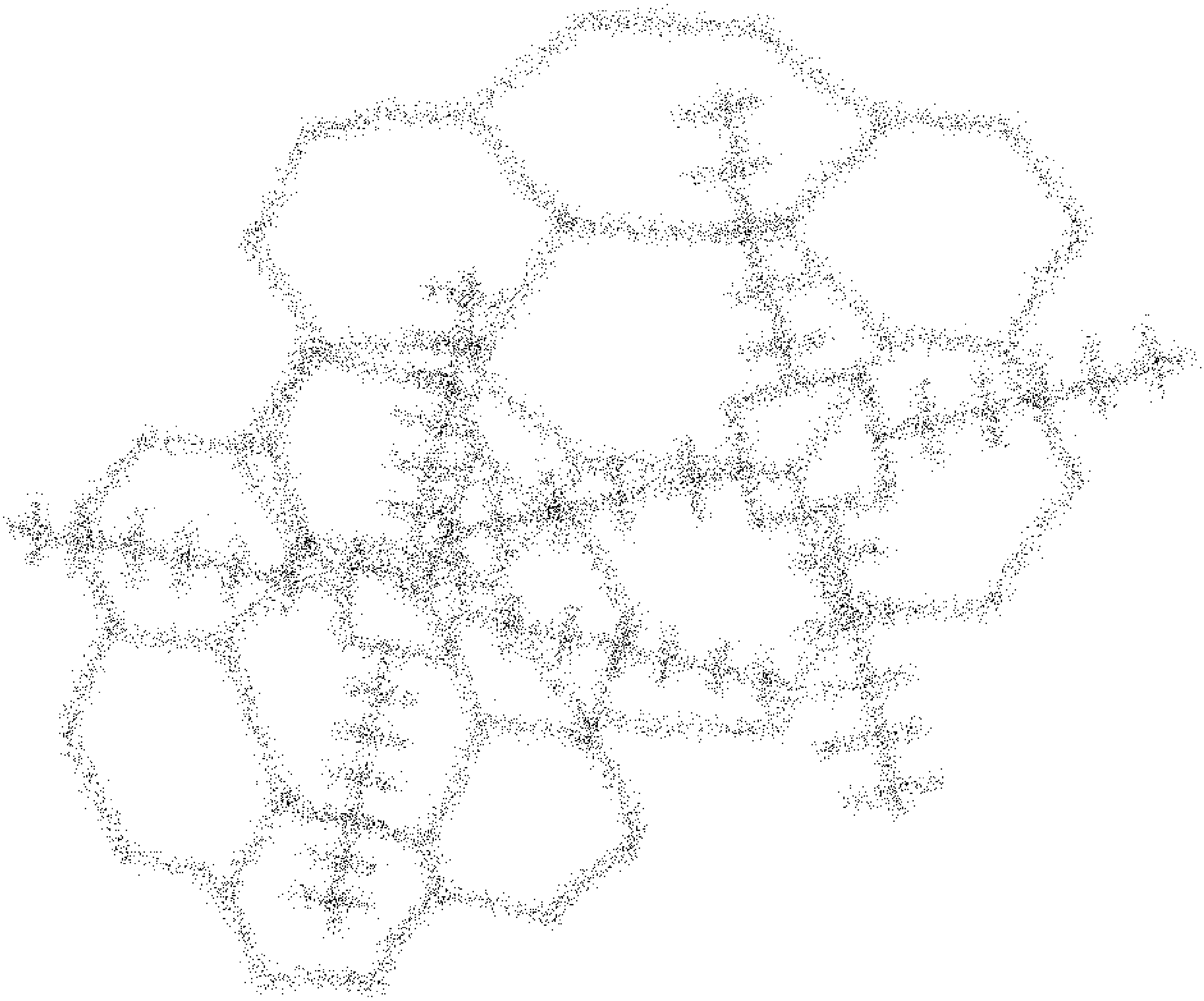

Figure 1: the original image.

The following pages show the results of the program. See example 8 for an explanation of the use of subsymbols in crosses.

The results of the program

The program successfully finds the crosses and the hexagonal tessellations.

Figure 2 shows the result of the program, with one hexagonal tessellation highlighted. The long narrow red and blue rectangles show the lines identified by the program. The black arrows mark the parts of the tessellation (these parts are lines and also 'dummy' symbols at the edge of the tessellation). The black discs indicate the connections between the parts. The large green rectangles (only partially visible) show the bounding boxes of the hexagonal tessellations. The tessellations also contain subsymbols, including hexagons (shown in mauve) and others in various colours around the edge of the tessellation.

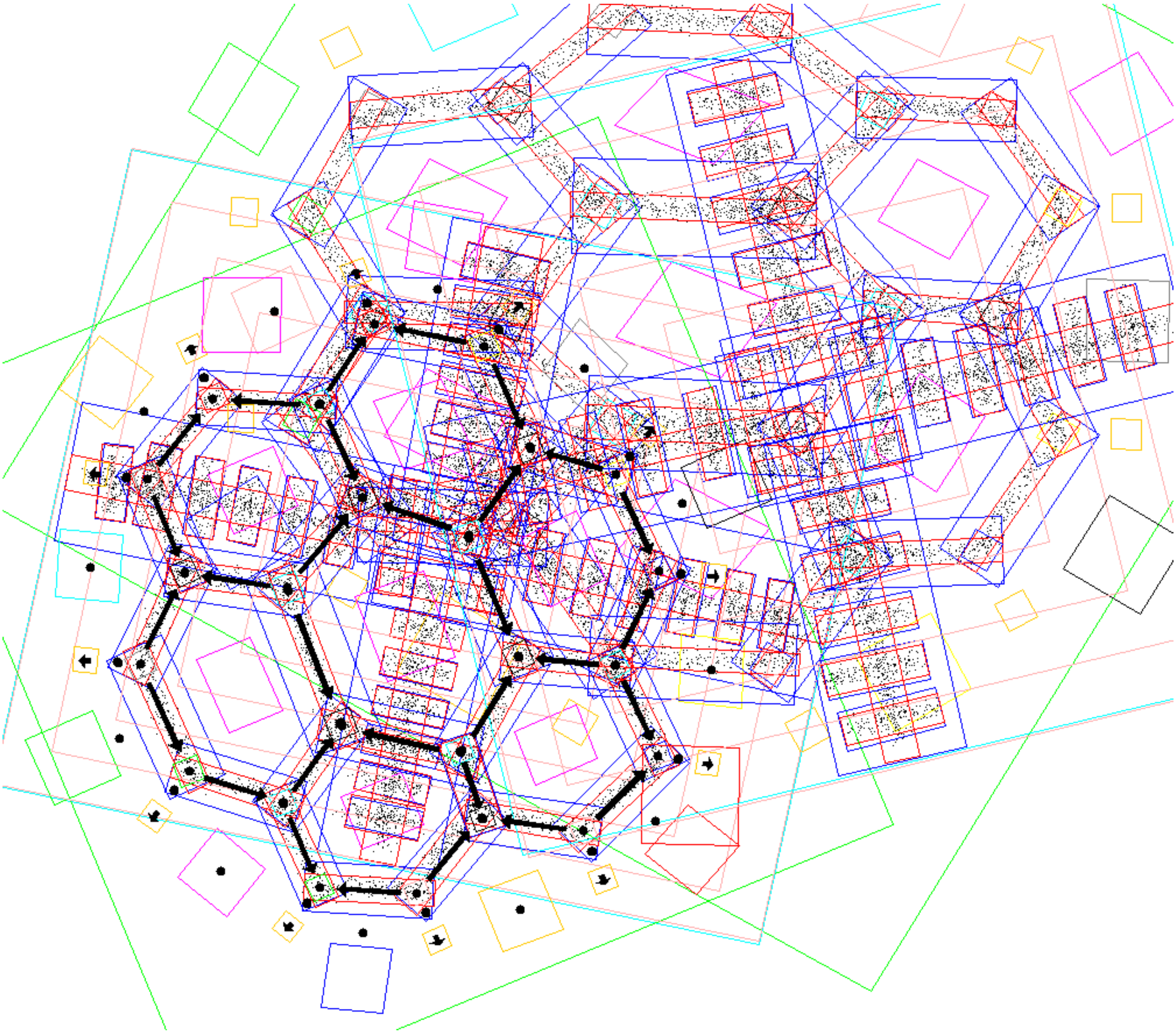

Figure 2: one hexagonal tessellation found by the program.

Figure 3 shows the same run of the program but with the other hexagonal tessellation highlighted.

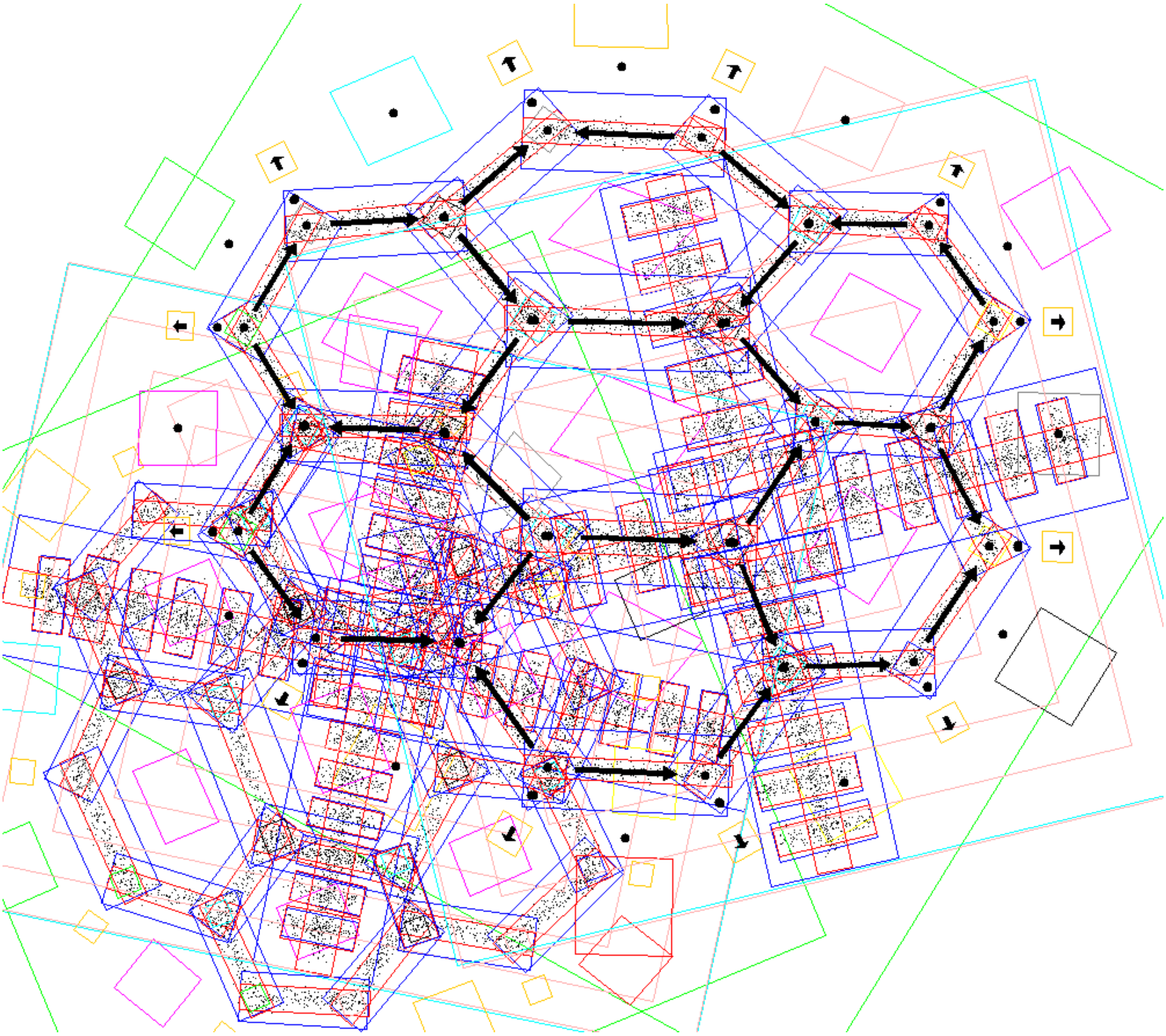

Figure 3: the other hexagonal tessellation found by the program.

Figure 4 shows the same run of the program but with one cross highlighted. The bounding box of the cross is shown in pale blue. Subsymbols (each consisting of four twigs, as in example 8) are shown as concentric pink bounding boxes.

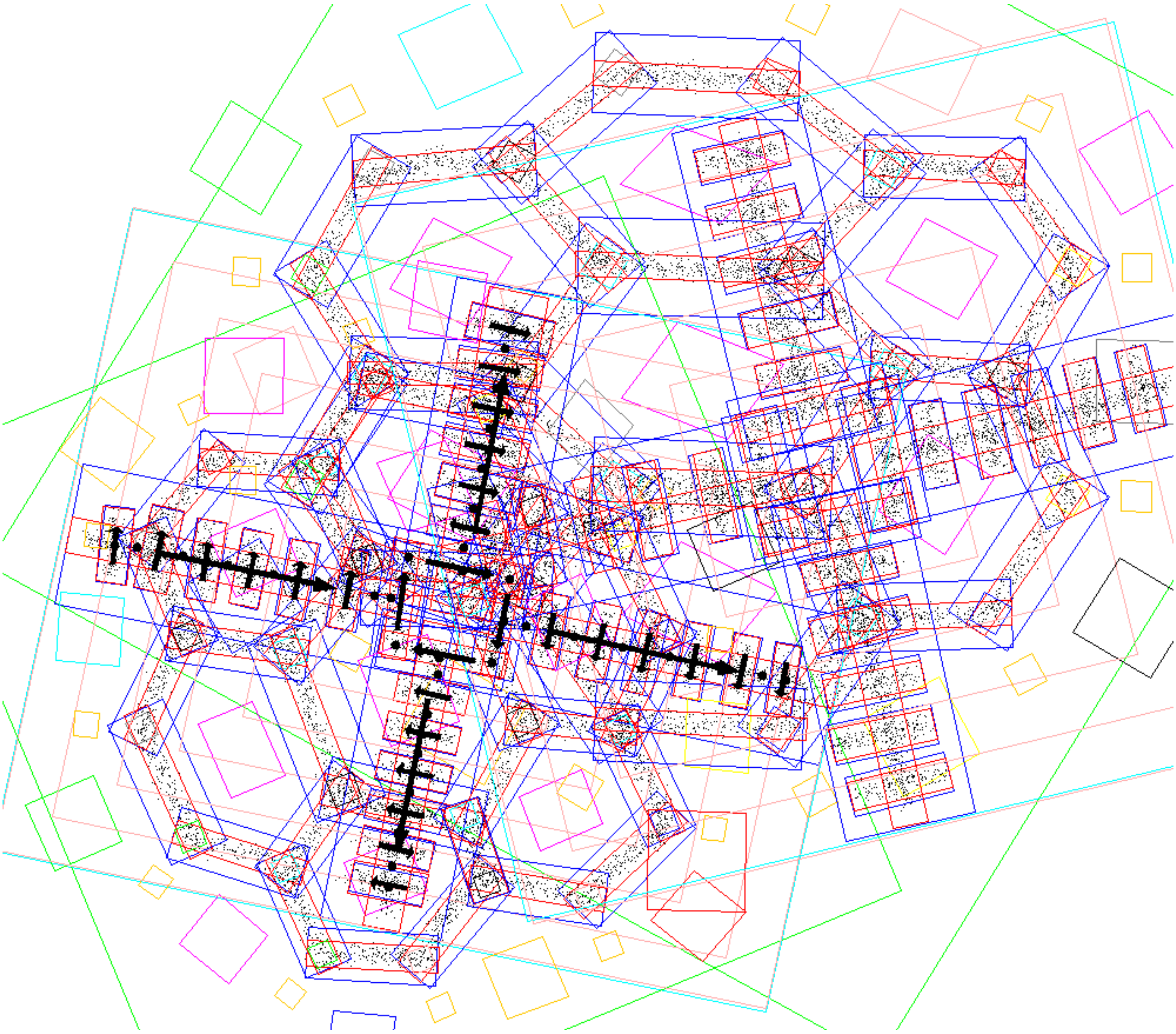

Figure 4: one cross found by the program.

Figures 5 and 6 show the same run of the program with the other cross highlighted. Figure 5 shows an intermediate state of parsing; the program has falsely identified six twigs on two of the trunks (the outermost twig of each trunk should not be there). Because of the use of subsymbols, the program is able to recognise this as an inconsistent state and subsequently removes the spurious twigs and realigns the subsymbols, producing the final version shown in figure 6.

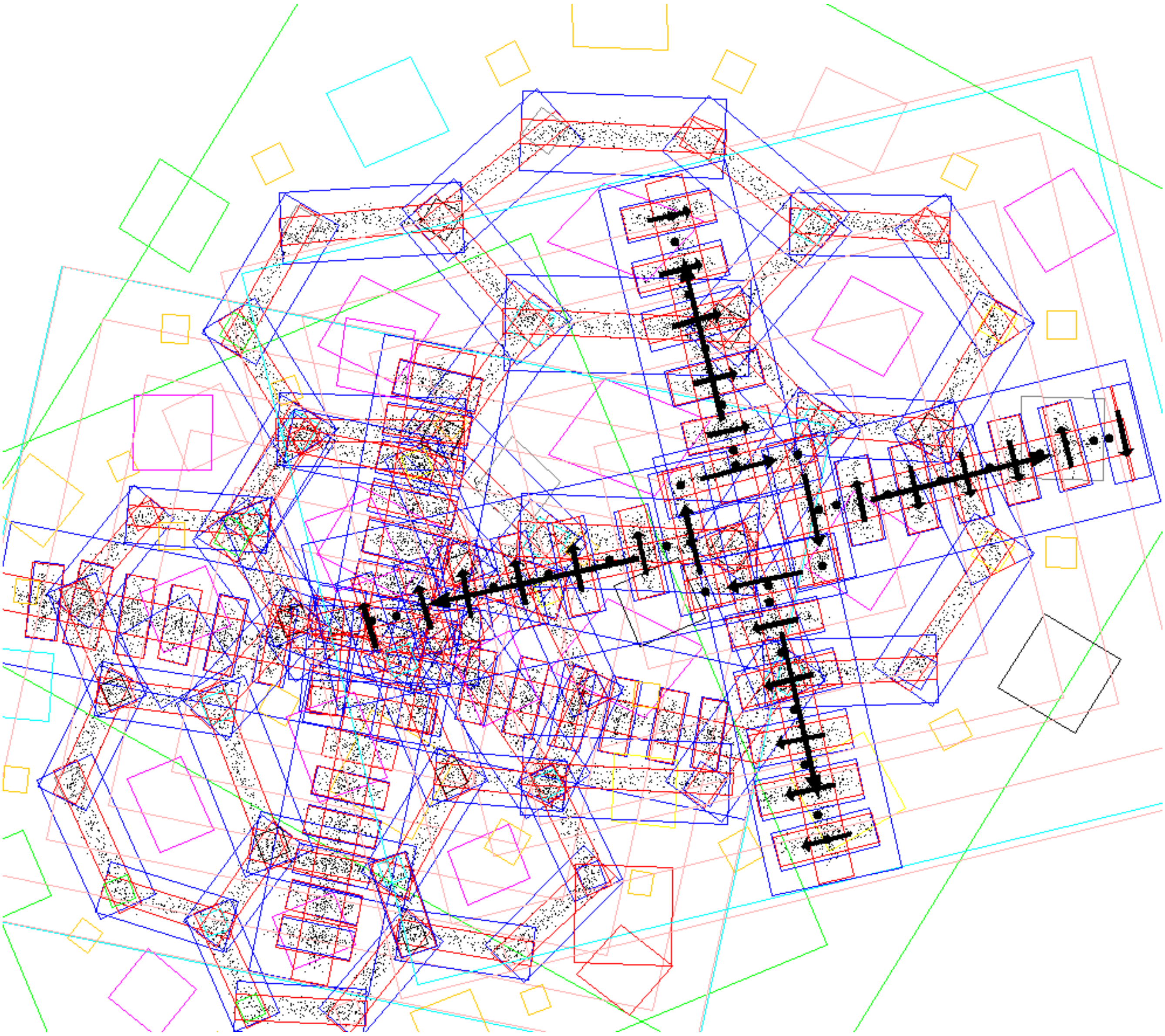

Figure 5: the other cross found by the program (an intermediate state).

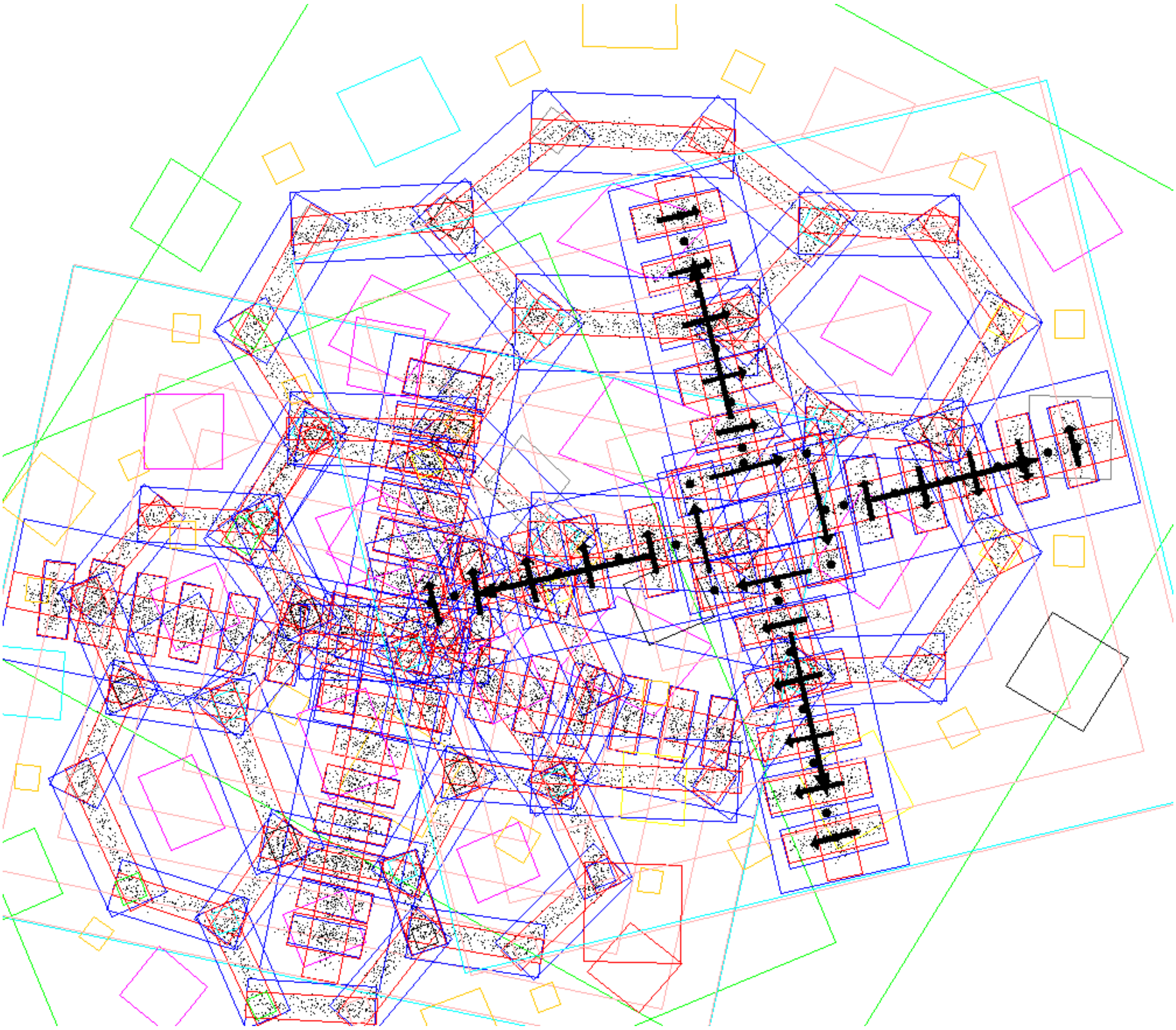

Figure 6: the other cross found by the program (the final parse).