

## **1. Introduction**

We divide the problem into two cases: small (number of pieces < 10) and big (number of pieces >= 10). Small cases can be solved by hand. Then big cases will be solved by our program.

## **2. Methodology used in the program**

### **2.1 Small case**

It will be easy to solve a case when there are few pieces. Also our program can convert big cases into small ones.

### **2.2 Big case**

The program will find best matching pairs of pieces. Then it will combine them to create a new, bigger piece. This step we will call Combination step. After combination step, number of pieces will be decreased nearly by half. Repeating the combination step will bring us to a small case.

### **2.2 Best matching pairs of pieces**

We will use following assessment function to measure matching of two pieces:

$$f = \alpha * \text{totalEdge} + \beta * \text{totalCorner};$$

Here, totalEdge is the total length of the common edges, totalCorner is the total count of the common corners for the two pieces. Values of the coefficients  $\alpha$  and  $\beta$  are evaluated by experiment.

## **3 Software library**

We used an open source C library – OpenCV to detect edges and corners of pieces. Edge length and corner angle values are important information for our solution.

## **4. Conclusion**

The method we chose will give us concentrate only on finding best matching pairs of pieces rather than solving the whole puzzle. That helped us to save our time too.