Automatic Marker Making*

[Abstract for Third CCCG, 1991]

Victor Milenkovic, Karen Daniels, Zhenyu Li
Harvard University
Center for Research in Computing Technology
Cambridge, MA 02138

April 5, 1991

1 Introduction

An essential step in the manufacture of clothing is the generation of a cutting plan or *marker*. The marker determines how the parts that make up an article of clothing are cut from a bolt of cloth. To improve cloth utilization, the parts for many articles of clothing are included in the same marker: in the case of blue jeans, 100 to 200 parts are packed onto a rectangle of cloth about two yards wide and 8 to 12 yards long. Generating an optimal marker (the shortest marker of a given width containing a given set of parts) is theoretically intractable (specifically, it is NP-complete). Using a CAD system, well trained people can generate near-optimal markers manually, but it is a difficult and time-consuming job. Automatic generation of markers would better enable manufacturers to keep up with customer demands for different styles and sizes.

Automatic marker generation is an exciting area of research because an effective solution could almost immediately be "plugged in"-used in practice. We are currently engaged in a three year research project at Harvard in the area of automatic marker making. The goal of our research is to match human performance in the generation of pants markers 80% of the time. Current algorithms fall far short of human performance for two reasons: they are not domain-specific, trying to solve all clothing types instead of focusing on pants or blouses, and as a result they are limited to variations of first-fit, a relatively weak computational technique. Our strategy is to create an algorithm that first applies domain-specific knowledge to greatly limit the space of plausible markers and that then applies powerful computational techniques such as exhaustive search, simulated annealing, elastic networks, or others to be devised, to find a good marker. We believe that creating an algorithm specific to pants will enable us to understand how to create algorithms for other domains.

The following three sections summarize three areas which we will present in our talk. Section 2 discusses the practical aspects of marker making in industry today. As one might expect, there are a few modifications to the theoretically "pure" problem of polygon placement. Section 3 shows that marker making is NP-hard (trivial) and it is also in NP (somewhat harder). The proof of

*This research was funded by the Textile/Clothing Technology Corporation from funds awarded to them by the Alfred P. Sloan Foundation.

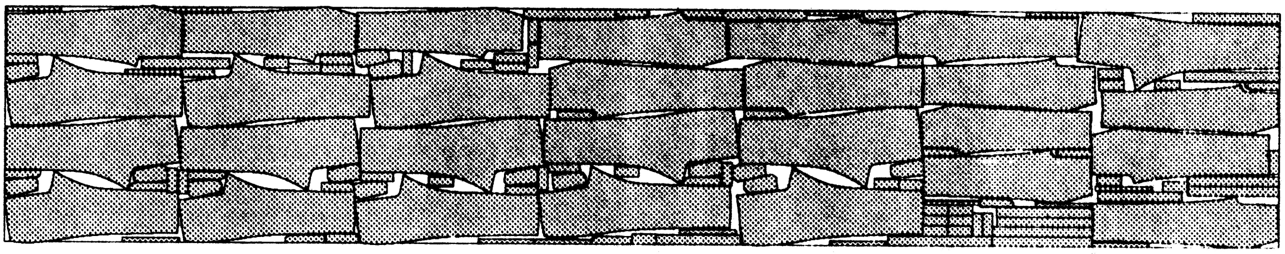

Figure 1: Pants Marker

the latter leads to a *local compaction* algorithm which may prove useful to manual marker making and which may become an important part of the automatic marker making algorithm. Finally, Section 4 covers our current research plan for matching human performance in marker making.

2 Practical Aspects of Marker Making

Figure 1 depicts a small pants marker with 14 pairs of pants on it. It is one yard wide and 8.688 yards long, and it contains 126 pieces. As one can see, a pair of pants consists of two large panels (which are duplicated by using multiple layers of cloth) plus a number of smaller parts called trim. The shorter dimension of the rectangle is called the *width* of the marker, and the longer one, the *length*. For any marker making task, the set of parts is determined by the range of sizes and styles required for a particular cutting. The width is determined by the width of the bolt of cloth in stock. The job of the marker maker is to pack the parts in a rectangle of smallest length. Some parts may be rotated by 180 degrees or flipped along the x-axis. Some parts may also be rotated a small amount, usually no more than 3 degrees. Parts cannot be rotated by arbitrary angles because even solid colored fabric such as denim has a grain.

The *efficiency* of a marker is the ratio of part area to total area. Efficiencies for pants markers are typically in the range 85-91%. The marker in Figure 1 has an efficiency of 89.60%. Sixty layers of cloth are cut simultaneously, so that making the marker an inch shorter (0.25%) saves about $7 of cloth each time the marker is used.

Unfortunately, marker making is not a simple matter of polygon placement. Markers must also satisfy rules for cuttability. Rules for a particular company might run up to 20 pages and they include some of the following:

- align edges to minimize the number of turns for the automatic cutter;
- pieces should have a common cutting line or be separated;
- don't let curves touch flat edges or sharp corners;
- no sharp corners in the selvage (upper one-half inch of the marker).

Current CAD/CAM systems *do not* check for violations of these rules, although they do enforce the constraints on 180 degree rotations, flips, and small rotations.

Human marker makers require six months to a year to learn to make efficient and cuttable markers. Many markers are required because the bolts of cloth vary in width from about 58 inches to 65 inches and because demand for sizes and styles varies. A large company might have two dozen CAD/CAM stations in operation twenty-four hours a day.

3 Theoretical Aspects of Marker Making

We are given rectangle of positive width (actually height) W and length L, a set of polygons P expressed with integer coordinates, a rotation bound $\theta(p)$ for each polygon $p \in P$ (expressed as a vector $v(p)$ with integer coordinates which forms the angle $\theta(p)$ with the x-axis), and two bit arrays indicating whether each polygon can be rotated by 180 and/or flipped. The marker making problem is to determine if this set of polygons can be placed inside the rectangle without overlapping by means of arbitrary translations and by means of rotations and flips satisfying the constraints.

Marker making is clearly NP-hard because one can reduce bin-packing to marker making. Taking the statement of bin-packing from Gary and Johnson, we have a finite set U of items, a positive integer size $s(u)$ for each $u \in U$, a bin capacity B, and another positive integer K. The question is whether we can partition U into disjoint sets $U_1, U_2, \ldots, U_K$ such that the sum of the sizes of the items in each set is less than or equal to B. We reduce this to marker making by replacing each item $u \in U$ by a rectangle of length 1 and height $s(u)$. We ask the marker maker to place these without rotation into a rectangle of height B and length K. It is clear that if the bin-packing problem is solvable, then the marker maker can place the rectangles in K stacks. If the marker exists, then we can post-process it by sweeping through from left to right and shifting each rectangle as far left as possible, and then sweeping from bottom to top shifting each rectangle as far down as possible. This leads to K or less stacks of height B or less, and thus there is a solution to the bin-packing problem.

It is not clear that the general marker making problem is in NP. Even if the vertices of the input polygons are specified with integer coordinates, the tightest packing may require an irrational translation and rotation for some of the polygons. In general, the question of whether a set of quadratic constraints has a solution is not in NP. It is an open problem whether the special structure of the marker making problem would put it in NP.

If rotations are not permitted, then marker making is in NP. To see this, suppose a powerful being has determined a placement of P into a W by L rectangle. First, he takes note of all vertex-edge and vertex-vertex contacts and determines if this set of contacts uniquely determines the solution. If not, at least one $p \in P$ will be free to move along some line and all the other polygons can move to maintain the set of contacts. The being moves p as far as it can go, but stops when a vertex slides to the end of an edge or when a vertex comes in contact with an edge. When this happens, he adds the new vertex-edge or vertex-vertex contact to the set of constraints. He repeats such motions until the set of contacts determines the solution. This has to happen eventually since he can add only a finite number of contacts to the set. All we have to do is to non-deterministically guess this set of contacts, solve the linear system implied by these contacts to generate a marker of polynomial size, and verify that there are no overlaps in polynomial time.

This "thought" proof leads to an interesting compaction algorithm for finding *locally* optimal markers in the case that rotations are not allowed. Given a particular marker we can improve it by determining an (infinitesimal) motion for each piece that will diminish the length of the marker without causing any immediate overlap. We can find this motion using linear programming. We move the pieces with the implied velocity until a new contact occurs and update the set of contacts, also removing any which have disappeared. We apply this process until no further progress can be made. The resulting marker will not be the global optimum unless we are very lucky. We can also modify this algorithm to allow us to move a particular part in a particular direction without introducing an overlap.

4 Techniques for Automatic Marker Making

We cannot expect to find optimal solutions to an NP-hard problem; however, we can hope to match human performance in marker making. To this end we interviewed experts marker makers and observed them in action. We determined a number of common principles.

- Large pieces (panels) are placed first and the length of the marker is determined at this point. Small pieces (trim) are placed without increasing the length of the marker if that can be avoided.
- In about a third of the cases, the panels can be placed in columns of four as shown in Figure 1. In other cases, they may have to be interleaved.
- A good marker maker will rotate the panels to increase the size of the gaps between them *before* placing any trim. Once trim placement has begun, the panels are not moved or rotated.
- The efficiency of the very best markers vary only by about 2%.

From our observations we believe that human-generated markers are generally within 2% of optimal. It is not likely that we can significantly improve on human performance.

Currently we feel the best plan is to first aim for a good non-interleaved panel marker. We would find this by treating the marker as four rows of seven (instead of seven columns of four). Using dynamic programming, shape matching between panels, and perhaps simulated annealing, we would determine four rows of panels that would be nearly equal in length and fit within the specified width W.

To place the trim, we would first have to generalize the compaction algorithm to allow small rotations, possibly by linearizing the quadratic set of constraints. We would then temporarily increase the width of the panel marker to add space between the panels, place the trim, and then compact the marker back to the original width using the generalized compaction algorithm.

Our talk will present our current plan and accomplishments in full.

5 Conclusion

Marker making is an interesting practical and theoretical problem. Techniques from many areas can be brought to bear, for example, shape understanding and matching, linear programming, quadratic programming, simulated annealing, dynamic programming, and many algorithms from computational geometry for rapid overlap detection and other tasks. We welcome all input and comments.