

Polygon Packing: Paper Model Publishing

Paper Model Publishing

- 3D Model from folded 2D Polygons.
- 2D Polygons must be placed as few pages as possible
 - 2ary: free space on last page.
- Min.separation between polygons = 5mm
- Only external polygons matter. Internal fold lines, textures, glueing aids, etc. do not concern us.
 - semi-sanitized datasets provided.



- Pieces labeled with “group”, preferably packed together.
- E.g.: head, legs



Paper Model Publishing

- Last rule: pieces may rotate, but not flip over
 - the paper can only be printed on one side
- Real, commercial problem: see <https://tamasoft.co.jp/pepakura-en/>
- Consultant: Claudio Dias : <https://www.paperinside.com/home>
- Current Pepakura solution is far from enough
 - Extensive hand-made adjustments needed. Try yourself the auto-packing of their demo models (windows only)
-

Paper Model Publishing: The Good

- Current Pepakura solution is far from enough
 - Extensive hand-made adjustments needed. Try yourself the auto-packing of their demo models (windows only)
- Possibly, a useful solution may be reached even with clearly sub-optimal, relatively simple heuristics.
- Current pepakura solution forcefully (not preferably) packs the groups together, often one group on each page. Then it may try to join groups, often unsuccessfully.
 - Packing per group solves a few smaller problems instead of one larger problem.
- For the purposes of the course, you can choose different criteria about how to deal with the groups (always, preferably (how?), or don't care).

Paper Model Publishing: The Good

- Claudio provides SVG files (an image file in XML format)
 - Group tag (e.g. head, torso)
 - Piece tag
 - Various coordinates lists tags (besides the borders, there are internal fold lines)
 - Semi-sanitized: no textures.
 - Tags available to translate and rotate each piece
 - Including rotation around a given point as center of rotation
 - Maybe groups can also be translated/rotated (not sure)
- Programmer may deal only with XML
 - Input and output are SVG files → easy to plot/draw them.
- Claudio says it is relatively easy to generate others as needed
 - Just remember he has a job...
- [Show example file]

Paper Model Publishing: The Bad

- Hard problem: Bin packing on fixed rectangular container with arbitrary polygons of different size.
- Claudio provides SVG files (an image file in XML format)
 - Semi-sanitized: no textures.
 - But there are fold lines and internal paths inside the piece
- Pre-processing needed
 - For each polygon, find the external vertices, ignoring internal paths
 - Then, or work with polygons, or convert into polyminos.
 - Known solution with polyminos in integer programming

Paper Model Publishing: The Ugly

- Hard problem: Bin packing on fixed rectangular container with arbitrary polygons of different size.
- Pre-processing needed
 - XML library required (or SVG, even better)
 - For each polygon, find the external vertices
 - Then, or work with polygons, or convert into polyminos.
- Various auxiliary functions needed before thinking on your heuristic
 - Group of polygons → external vertices
 - Polygon → polymino
 - Distance(piece1, piece2) or Distance (polymino1, polymino2)
 - Dealing with translations and rotations
 - around which center?
 - Polygon/polymino → center of mass ?

Paper Model Publishing: The Good Again

- For the polymino approach, a coarser grid means faster results at the expense of optimality.
- All preprocessing / auxiliary code may be shared between students.
 - Examples given in the previous slide
 - Only packing heuristics needs to be your own work
- Datasets may be very different. For some, all pieces have similar sizes. For others, each of a few larger pieces may require a page for itself, and then the problem is where to fit the remaining pieces in the remaining spaces. So, no heuristic is expected to always win.

Paper Model Publishing: Other references

- SVG manual
 - <https://developer.mozilla.org/en-US/docs/Web/SVG>
 - Transform attribute
 - <https://developer.mozilla.org/en-US/docs/Web/SVG/Attribute/transform>
- Puzzles resolvidos com Z3 - https://yurichev.com/writings/SAT_SMT_by_example.pdf
 - Código: https://yurichev.com/SAT_SMT_tree/other/tiling/kangaroo/
 - Especialmente https://yurichev.com/SAT_SMT_tree/other/tiling/kangaroo/tiling.py
 - A lib dele eh essa: https://yurichev.com/SAT_SMT_tree/libs/
 - Other dataset (dinosaur), and he allows reflexions (we dont)
- Game: pack polyminos in board <https://en.wikipedia.org/wiki/Blokus>
 - Solutions for small sized problems: <https://cemulate.github.io/polyomino-solver/>
 - With 3 implementations: <https://github.com/cemulate/polyomino-solver>
 - Including Knuth Alg X, <https://github.com/taylorjg/DlxLib> Another uses Z3