ACM Transactions on Graphics 2019

Hao-Yu Liu, Xiao-Ming Fu, Chunyang Ye,
Shuangming Chai, Ligang Liu

Atlas Refinement with Bounded Packing Efficiency

Presented by Jerry Yin

Packing efficiency

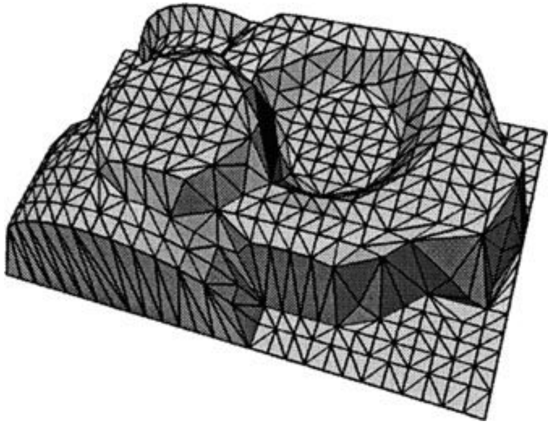
- Store high-frequency information in rectangular 2D texture images.
- Textures are mapped to 3D surfaces using UV coordinates.
- Unused parts of the image are wasted
- A *texture atlas* is composed of *charts*.
- *Packing efficiency* is

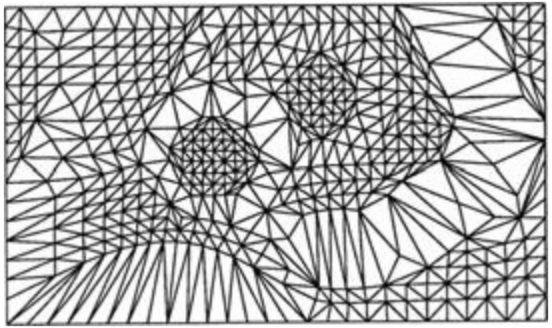
  area(atlas) / area(atlas bounding box).

Model by Geoffrey Marchal.

Packing problem

- Fixed boundary parameterization can give perfect packing efficiency, but has high distortion.
- Making each triangle its own chart gives zero distortion *and* good packing efficiency, but leads to poor performance and potential artifacts.
- We want
  - high packing efficiency,
  - low distortion, *and*
  - short boundaries.

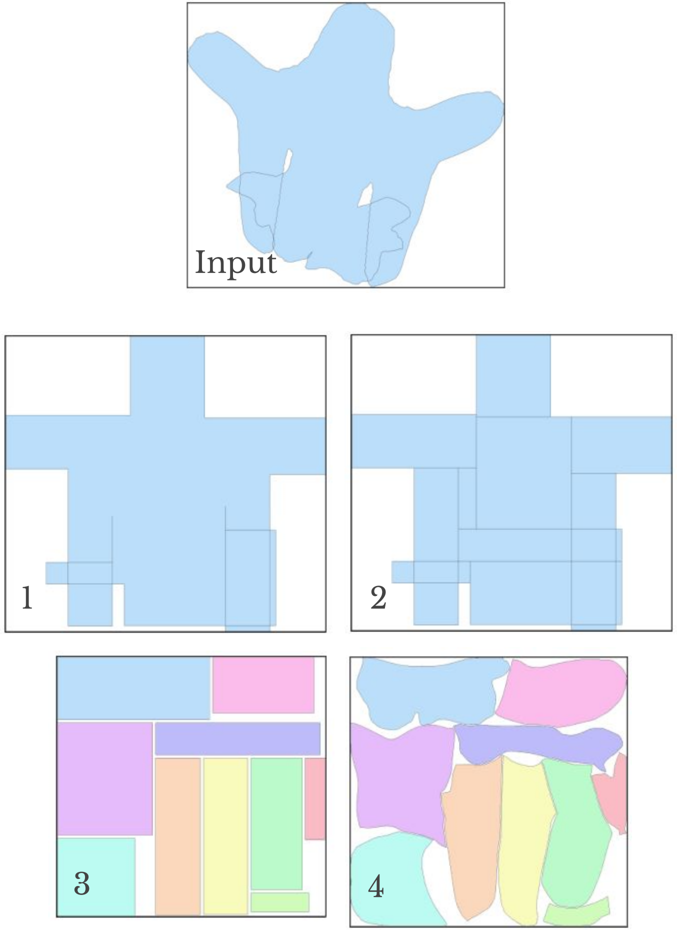




Algorithm overview

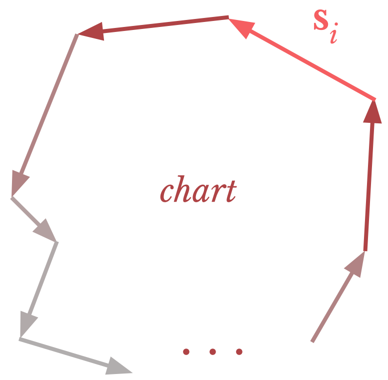
Given an existing texture atlas:

1. Deform input charts so that their boundaries are axis-aligned.
2. Cut axis-aligned charts into rectangles.
3. Pack rectangles.
4. Relax boundaries of rectangles to reduce distortion.



Axis-aligned boundaries

- Want to make boundary edges of all charts axis-aligned.
- To minimize corners, (Gaussian) smooth direction of all boundary edges.
    - Each direction is repeatedly set to weighted average of neighbourhood's directions.
- Find optimally axis-aligned rotation R of each chart by optimizing $\min_R \sum \|\mathbf{s}_i\| \Phi(R\mathbf{s}_i)$ where $\Phi(x, y) = x^2 y^2$

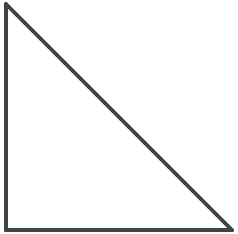


Aside: Gauss-Bonnet theorem

- α_i: interior angle at vertex i

$$\sum_i (\pi - \alpha_i) = 2\pi$$

- Corollary: if we increase the interior angle somewhere, we must decrease it by an equivalent amount somewhere else (possibly in multiple places).



$$\left(\pi - \frac{\pi}{2}\right) + \left(\pi - \frac{\pi}{4}\right) + \left(\pi - \frac{\pi}{4}\right) = 2\pi$$

$$4\left(\pi - \frac{\pi}{2}\right) = 2\pi$$

Axis-aligned boundaries (cont'd)

- $\Gamma(\mathbf{s}_i)$: closest axis direction to $\mathbf{s}_i$
- Update each interior angle α_i to be

$$\alpha_i + \angle(\mathbf{s}_i, \Gamma(\mathbf{s}_i)) - \angle(\mathbf{s}_{i+1}, \Gamma(\mathbf{s}_{i+1}))$$

- Result may have foldovers ($\alpha_i \leq 0$).
  - Find boundary verts ($\alpha_j = 180°$) adjacent to α_i, and set $\alpha_j = 90°$ and add $90°$ to α_i.
- More optimizations: corners can be merged or moved sometimes to reduce distortion.

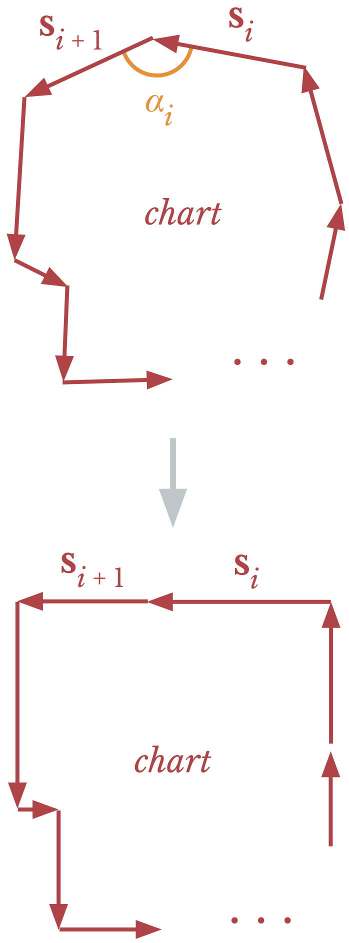


Chart deformation

$$E_{\mathrm{edge}}(\mathbf{b}_i) = \frac{1}{2}(1-\gamma)\underbrace{(\theta_i - \frac{\pi}{2}\Theta_i)^2}_{\text{preserve already obtained angles}} + \frac{1}{2}\gamma\underbrace{(\frac{l_i}{l_i^0} - 1)^2}_{\text{preserve lengths}}$$

$$E_{\mathrm{align}}(\mathbf{c}) = \sum_{i=1}^{N_b} \frac{l_i^0}{l^0} E_{\mathrm{edge}}(\mathbf{b}_i),$$

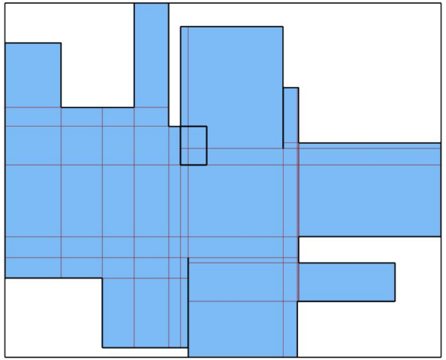
weight according to length

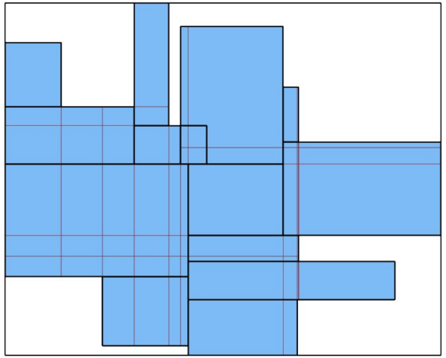
- Interior vertices need to be adjusted.
- Minimize sum of $E_{align}(\mathbf{c})$ (to get close-to-axis-aligned boundaries) and symmetric Dirichlet energy (to minimize isometric distortion) for chart **c**.
  - Increasingly prioritize $E_{align}(\mathbf{c})$ as we iterate.

Afterwards, snap the result to be exactly axis-aligned.

Cutting charts with motorcycles

- *Motorcycle* starts at a corner and travels in a straight axis-aligned line, and stops when it hits a boundary or the trail of a different motorcycle.
- For each two consecutive interior edges of a corner, at least one edge must be assigned a motorcycle in order to get a rectangular partition [Eppstein et al. 2008].
- Partition not unique (choice of motorcycle assignment and ordering).



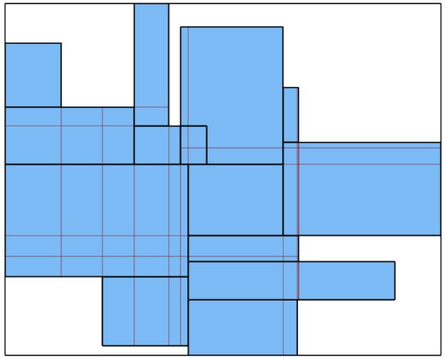


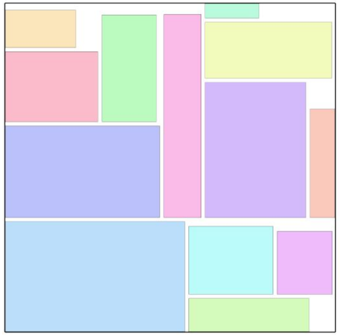
Cutting charts with motorcycles (cont'd)

- Generate motorcycle decompositions at random.
- Pack rectangles with small padding around boundary (useful later).
- Evaluate each packing with score

$$\text{packing efficiency} - \omega \frac{\text{boundary len after cut}}{\text{boundary len before cut}}$$

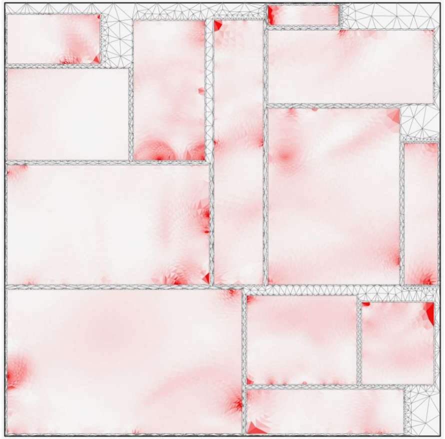
- Pick one with highest score.

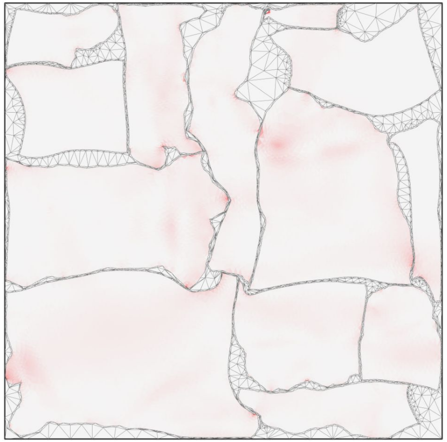




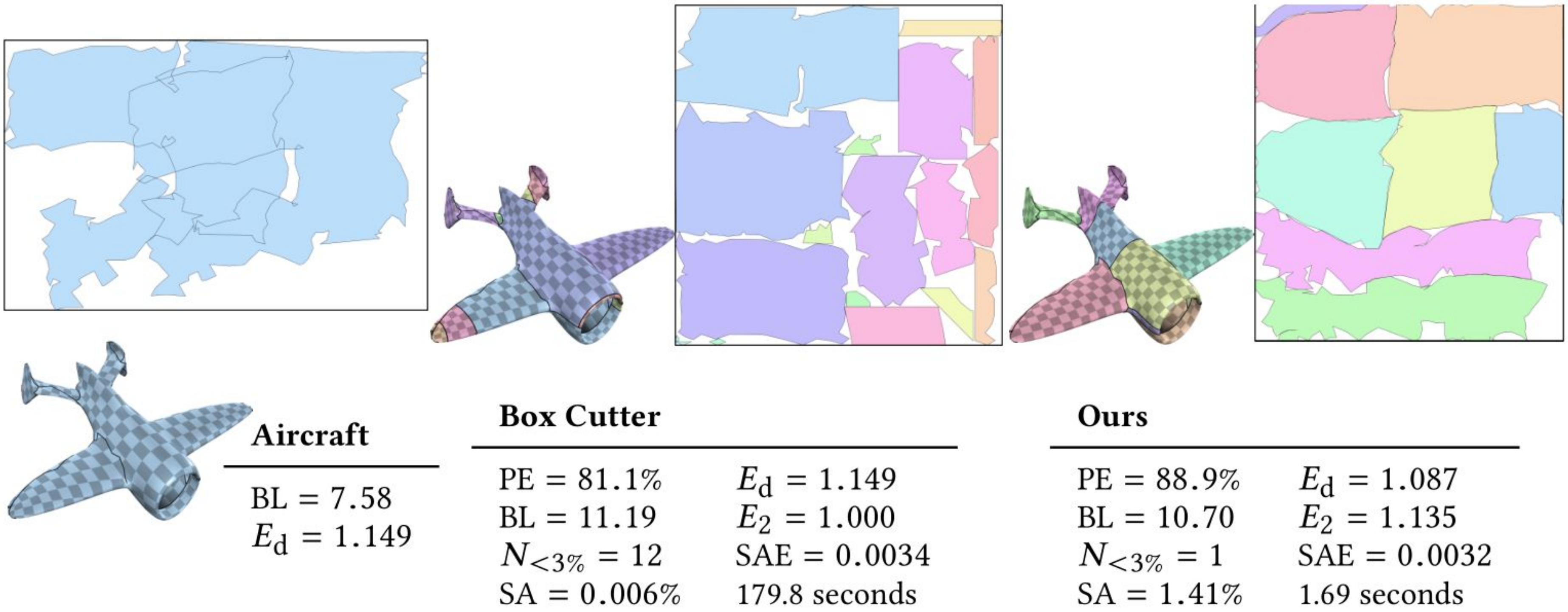
Relaxing the boundaries

- Based on scaffolding method [Jiang et al. 2017].
  - Triangulate empty space, then minimize symmetric Dirichlet energy.
- Allows maintaining padding around charts.
  - Useful for subsequent applications.



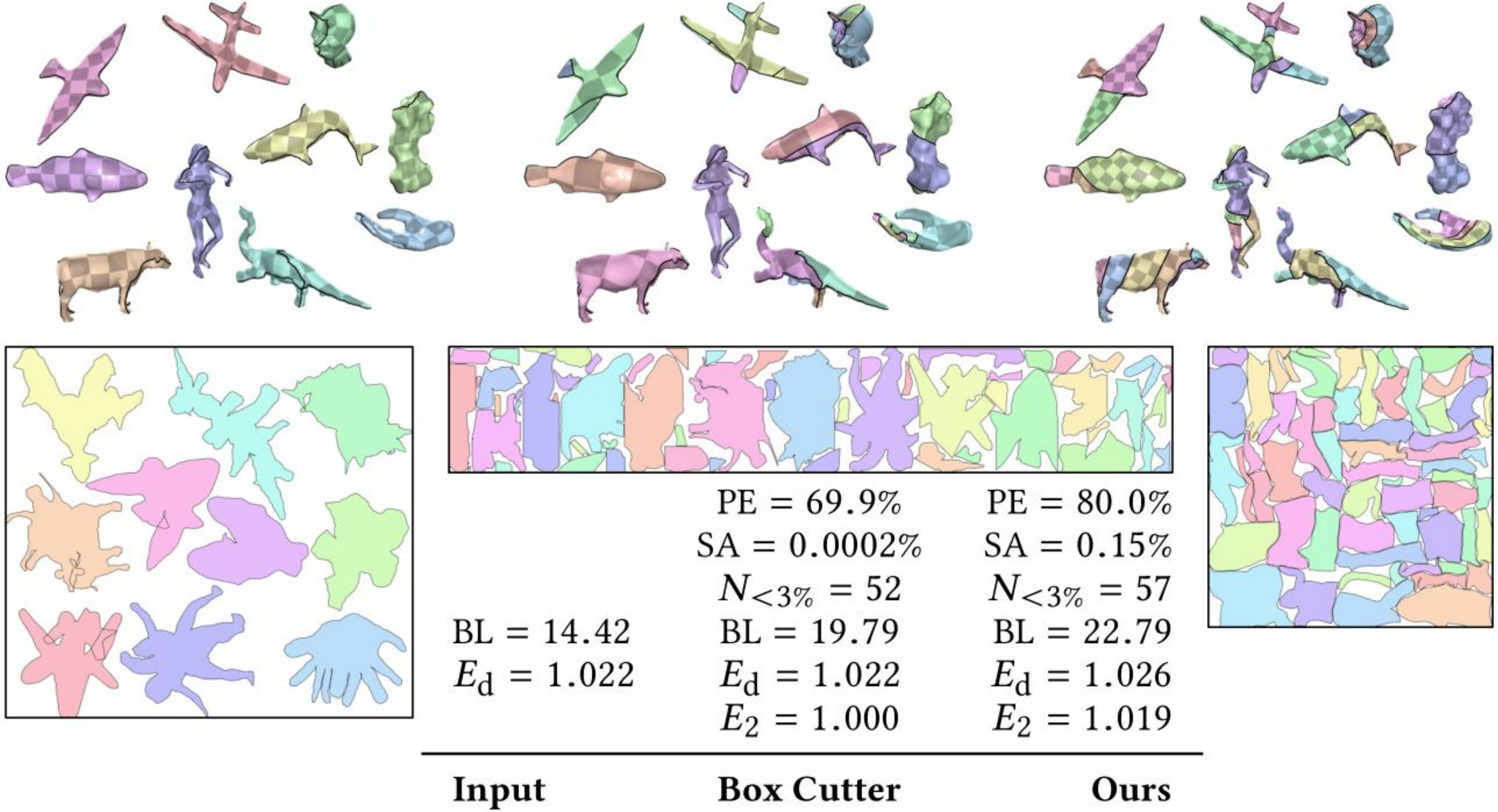


Results

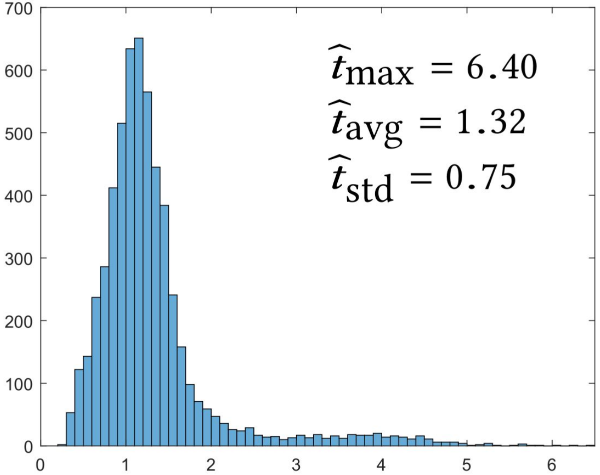


Aircraft
BL = 7.58
E_d = 1.149
Box Cutter
PE = 81.1%
BL = 11.19
$N_{<3\%}$ = 12
SA = 0.006%
E_d = 1.149
E_2 = 1.000
SAE = 0.0034
179.8 seconds
Ours
PE = 88.9%
BL = 10.70
$N_{<3\%}$ = 1
SA = 1.41%
E_d = 1.087
E_2 = 1.135
SAE = 0.0032
1.69 seconds

Results



PE = 69.9%
SA = 0.0002%
$N_{<3\%}$ = 52
PE = 80.0%
SA = 0.15%
$N_{<3\%}$ = 57
BL = 14.42
E_d = 1.022
BL = 19.79
E_d = 1.022
E_2 = 1.000
BL = 22.79
E_d = 1.026
E_2 = 1.019
Input
Box Cutter
Ours

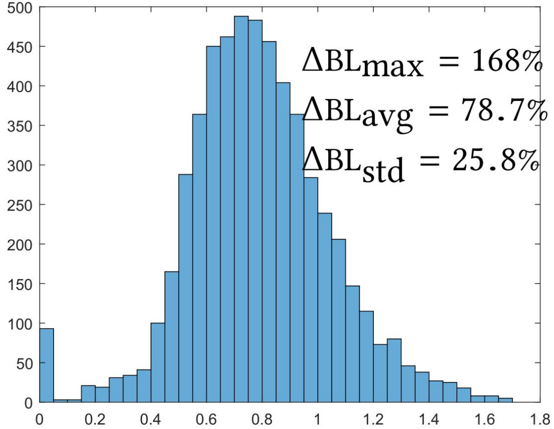


Timings

(runtime (ms?) / # vertices).

Boundary length

elongation.



Isometric distortion

w/r/t original mesh.

Isometric distortion

w/r/t input atlas.