Algorithm: Isometric Triangle Flattening (UV Unwrapping)

This algorithm takes a 3D triangle (T_2 defined by points B, C, D) and "unfolds" it into a 2D plane (where $y = 0$) relative to an already flattened neighbor triangle (T_1 defined by A, B, C).

---

1. Analyze the 3D Geometry (The "Source")

To preserve the true shape of the mesh, we extract the geometric relationships from the 3D coordinates.

- **Define 3D Vectors from vertex B:**
  - $\vec{V}_{BC} = C - B$
  - $\vec{V}_{BD} = D - B$
- **Calculate $\cos(\beta)$ using the Dot Product:**

$$\cos(\beta) = \frac{\vec{V}_{BC} \cdot \vec{V}_{BD}}{\|\vec{V}_{BC}\| \cdot \|\vec{V}_{BD}\|}$$

- **Derive $\sin(\beta)$:** Using the Pythagorean identity:

$$\sin(\beta) = \sqrt{1 - \cos^2(\beta)}$$

---

2. Establish the 2D Basis (The "Target")

In the UV plane ($y = 0$), we use the shared edge bc to create a local coordinate system. This ensures the triangles stay connected.

- **Edge Length:** $L = \|c - b\|$
- **Local X-axis ($\vec{u}$):**

$$\vec{u} = \frac{c - b}{L}$$

- **Local Z-axis ($\vec{v}$):** Rotate $\vec{u}$ by 90° in the 2D plane to create an orthogonal (perpendicular) axis:

$$\vec{v} = (-u_z, u_x)$$

---

3. Calculate Local 2D Coordinates

We determine the position of d relative to b in our new 2D "grid" using the true 3D distance $d_1 = \|\vec{V}_{BD}\|$.

- $x_{local} = d_1 \cdot \cos(\beta)$
- $z_{local} = d_1 \cdot \sin(\beta)$

---

4. The Flip Logic (Overlap Prevention)

To ensure the triangles "unfold" like a piece of paper rather than folding back on top of the previous triangle, we use point a as a reference.

1. **Reference Vector:** $\vec{w} = a - b$
2. **Alignment Test:** Calculate the dot product of $\vec{w}$ and our perpendicular basis vector $\vec{v}$:
   - $Side\_A = \vec{w} \cdot \vec{v}$
3. **Final Placement:**
   - If $Side\_A$ is **positive**: Point a lies in the direction of $+\vec{v}$. Therefore, we must place d in the **opposite** direction ($-\vec{v}$):

     $$d = b + (x_{local} \cdot \vec{u}) - (z_{local} \cdot \vec{v})$$

   - If $Side\_A$ is **negative**: Point a lies in the direction of $-\vec{v}$. Therefore, we place d in the **positive** direction ($+\vec{v}$):

     $$d = b + (x_{local} \cdot \vec{u}) + (z_{local} \cdot \vec{v})$$