TRIANGLE BISECTION TESSELLATION WITH FRACTIONAL VERTEX LODS

Peter Smith-Lacey & Simon Fenney

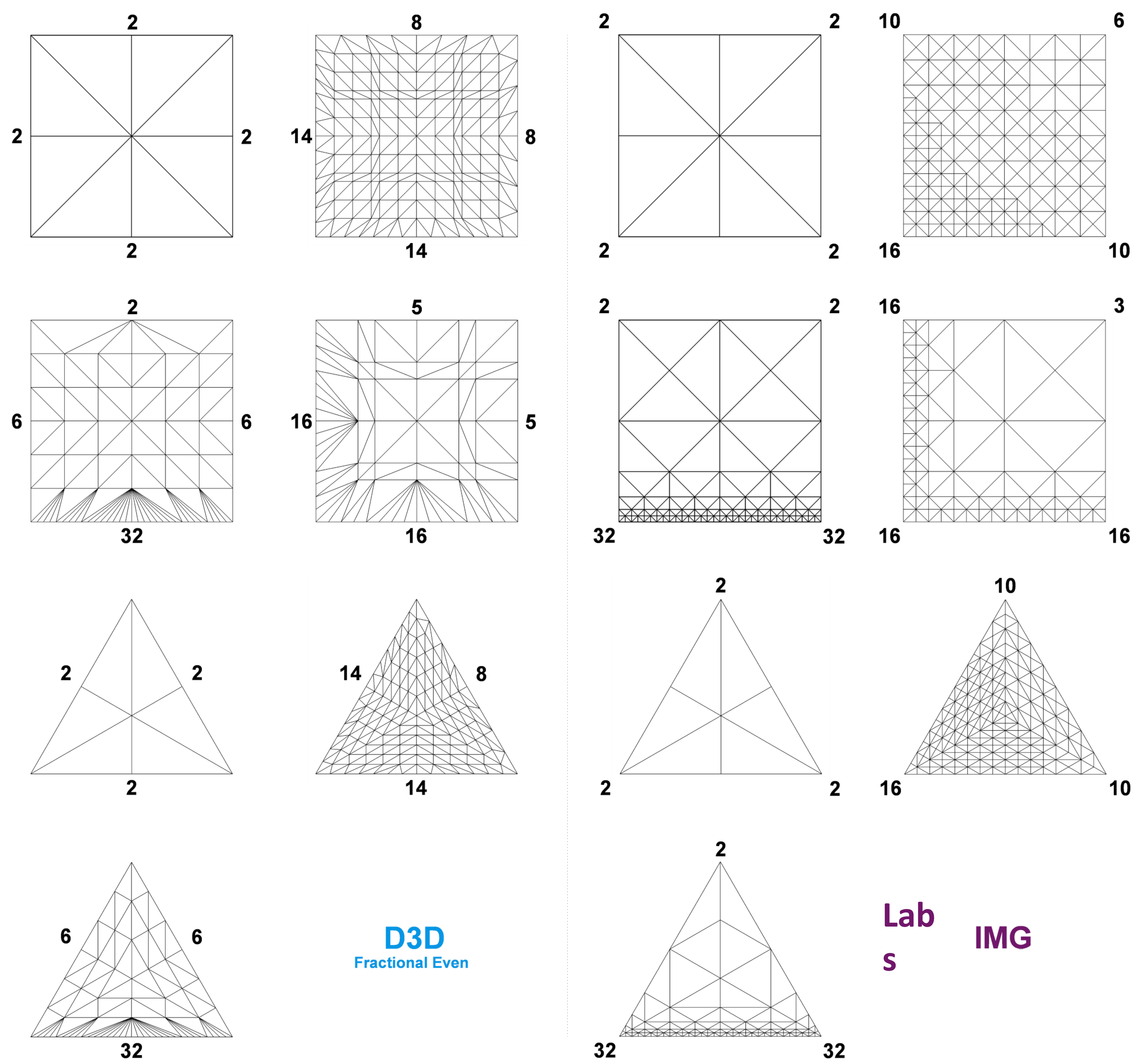
Imagination

INTRODUCTION

Triangle bisection tessellation with fractional LOD for fast subdivision of triangle, quad and isoline patches producing spatially and temporally smooth geometry

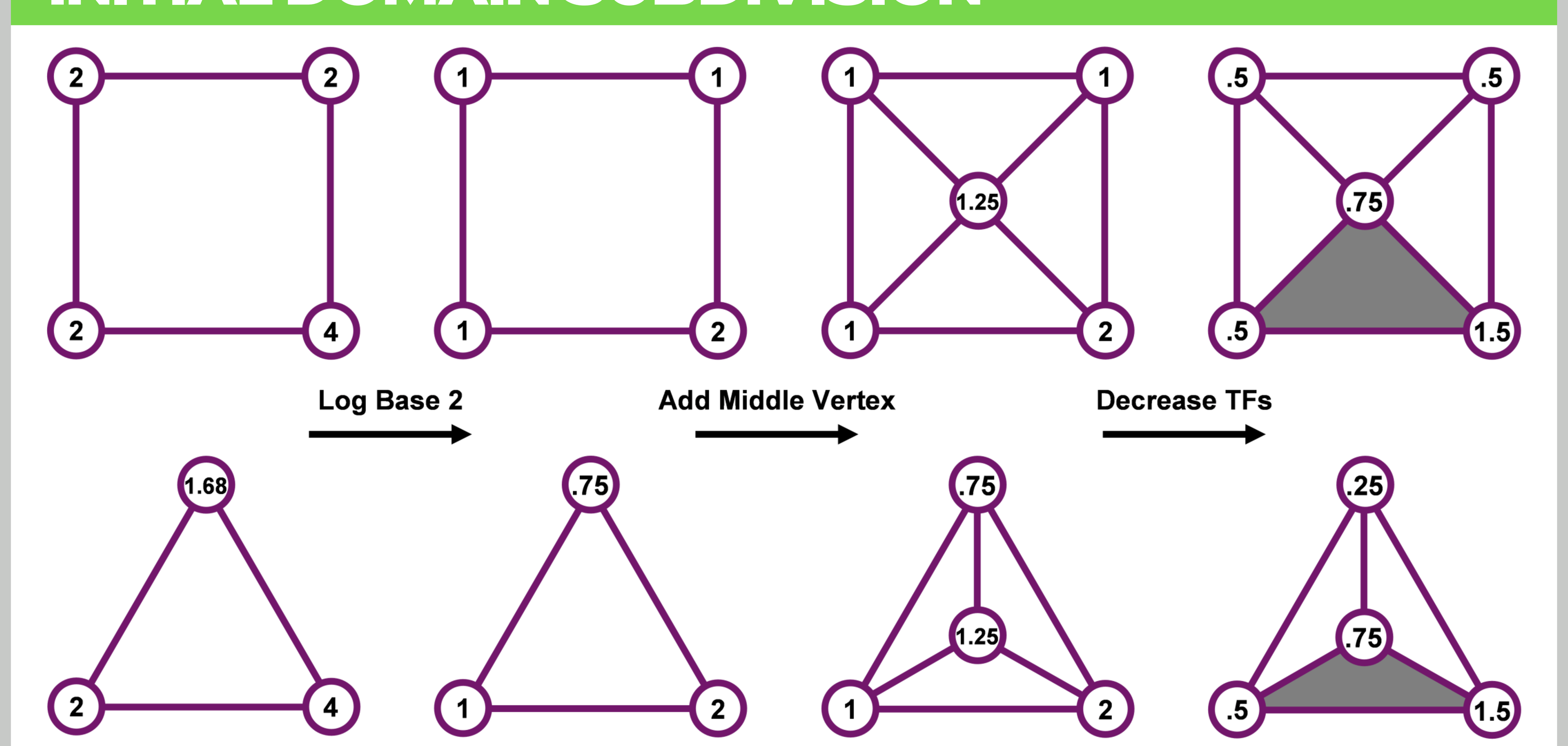
- Tessellation Factors (TFs) defined per vertex rather than per edge
- Fixed vertex UV coordinates for stable geometry
- Fractional tessellation supported through “Blending”
- No thin or redundant primitives produced

TESSELLATION PATTERNS COMPARISON



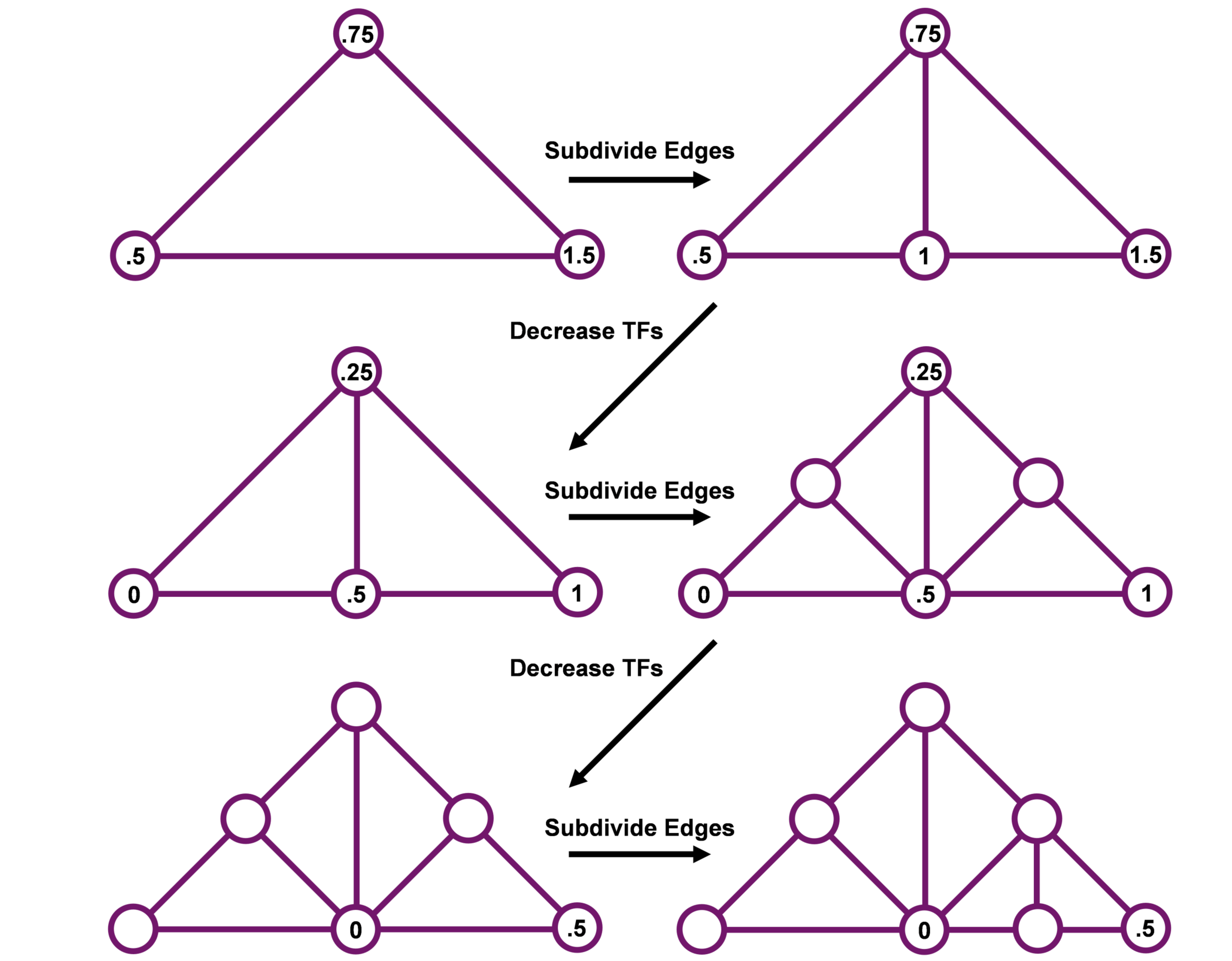
Our new tessellation scheme (right) produces a range of patterns determined by per vertex TFs rather than the per edge TFs of current schemes. Our proposal permits large steps in TFs without resorting to triangle fanning unlike current methods, such as DirectX (left).

INITIAL DOMAIN SUBDIVISION



Our tessellator takes as input a topology plus per-vertex TFs. Each TF is reduced (pseudo) log base 2. The triangle/quad domain is subdivided into 3/4 initial triangle patches by adding a middle vertex as the average of the corner vertices and reducing each TF by 0.5. In the following stage the triangle patches are processed independently and identically. If none of the corner TFs of the triangle domain exceeds 0 then no subdivision occurs to prevent over-tessellation.

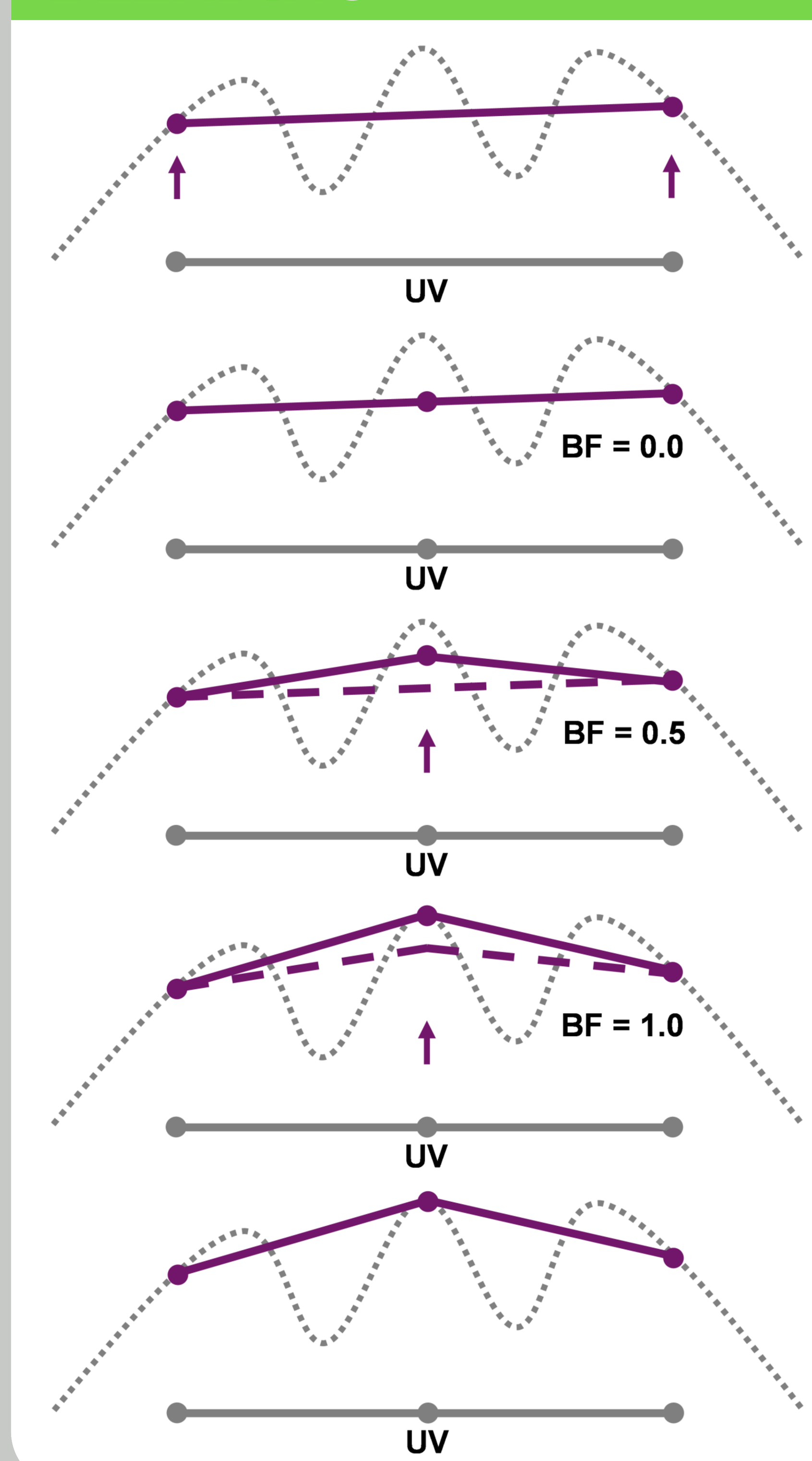
TRIANGLE BISECTION



Each initial and subsequent triangle patch is processed independently and identically by triangle bisection. The example bisections above match the triangle patches from initial subdivision marked in grey. A triangle patch comprises three UVs and three TFs, one for each corner vertex.

If either of the Tessellation Factors on the end of the longest edge (in domain space) exceeds 0 then subdivision occurs, otherwise the triangle patch forms a single primitive. When subdivision occurs, a new vertex is added as the average of the vertices on the end of the bisected edge and two new triangle patches are formed. Lastly, all Tessellation Factors are decreased by 0.5. The same process is then repeated on the two new triangle patches until subdivision terminates.

BLENDING



Our proposed tessellation scheme supports fractional levels of detail by “Blending” newly added vertices. This means that each newly added vertex lies in the tessellated surface of the current integer LOD and continuously changes into its final state in the next integer LOD.

Continuity is achieved by interpolating between the average of the edge-end vertices and the final state of the newly added vertex by a weight known as the Blend Factor (BF).

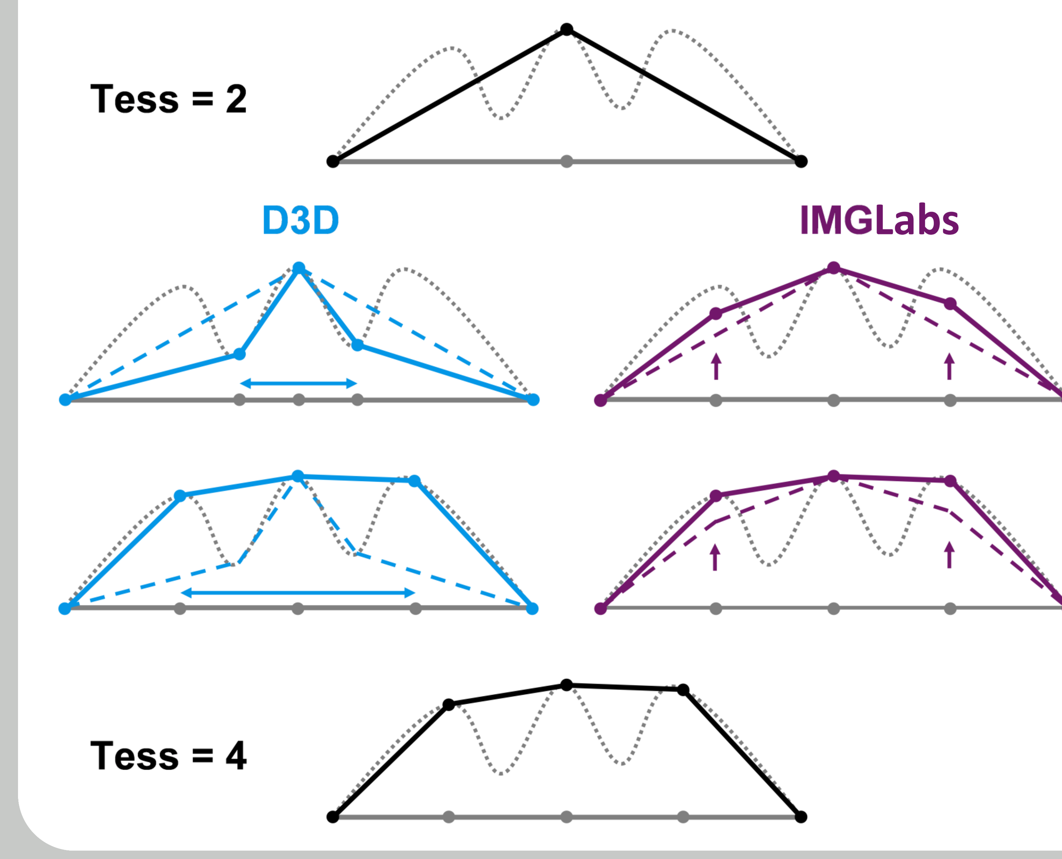
The Blend Factor is a value between 0 and 1, is specified for each vertex by the tessellator and is derived from the fractional parts of the edge-end vertices.

Blending is performed on the attributes of the post Domain Shader vertices.

By keeping UV coordinates of vertices fixed, our technique ensures stable geometry for fractional tessellation.

The left-hand diagram demonstrates the process of Blending from the point that a new vertex is added until it reaches its final state.

FRACTIONAL LOD COMPARISON



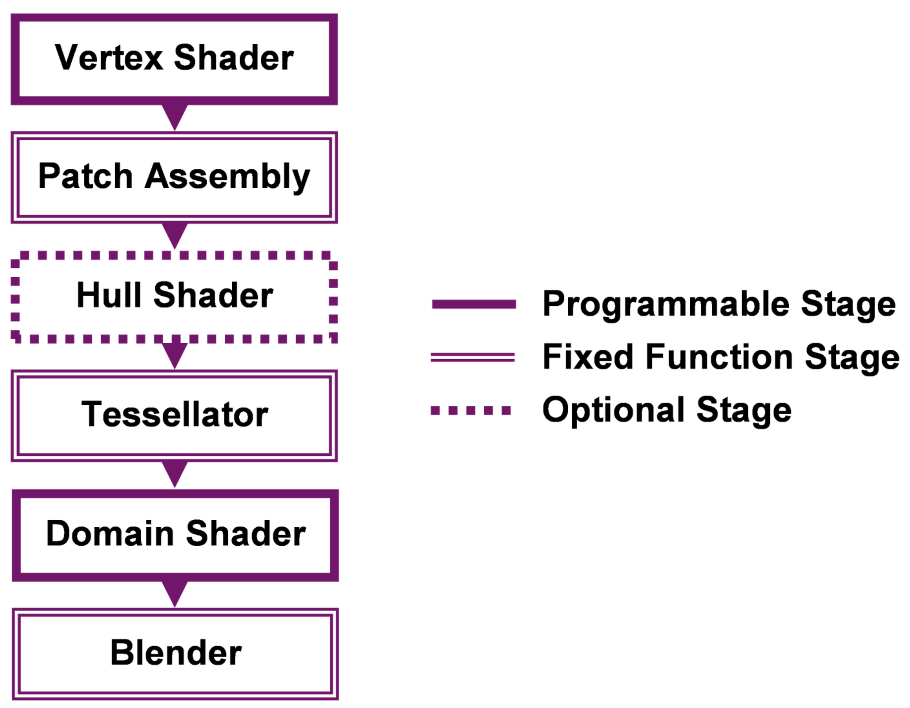
Our tessellation scheme significantly improves on DirectX’s fractional methods by producing stable geometry which is not affected by higher frequencies in attributes of the surface.
In contrast DirectX is known to produce instability artefacts when geometry changes abruptly as vertices move in UV space.
When tessellating higher frequency surfaces DirectX must either sacrifice detail from the surface or over-tessellate. Our proposal remedies these issues.

MODIFIED PIPELINE

The proposal adds one extra fixed function stage to the tessellator stages of the pipeline: the Blender.

This unit performs the Blending on post Domain Shader vertices.

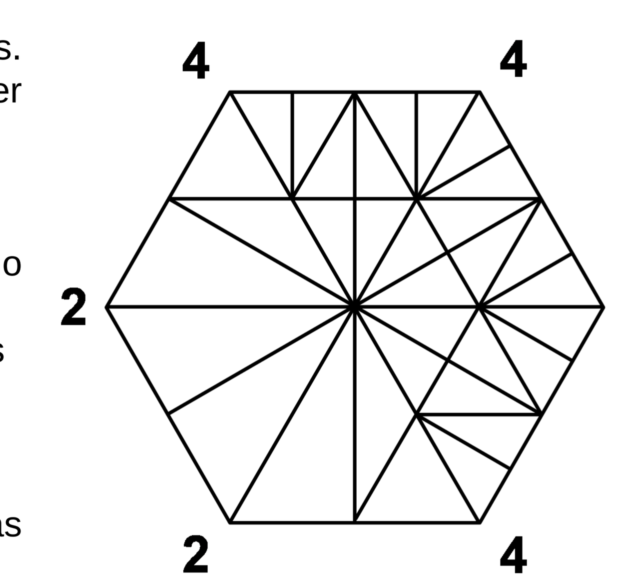
By moving Tessellation Factors from patches to vertices we make the Hull Shader an optional part of the pipeline between the patch assembly and the tessellator, reserved for optional per-patch work.



EVALUATION

Our proposal solves all major issues of current schemes. In particular it has the following advantages over DirectX’s edge based tessellation:

- More user-friendly vertex Tessellation Factors
- Very simple algorithm requiring minimal logic, with no special cases
- Supports fractional LOD via Blending with no artefacts
- Stable geometry is ensured by the fixed UVs
- No thin or redundant primitives
- Spatially and temporally smooth geometry
- Generalises to any polygonal domain such as hexagons or octagons



CONCLUSION

We propose a fast and high quality tessellation scheme of low complexity requiring minimal changes to the existing tessellation stages. By moving Tessellation Factors from domain edges to their corners one eliminates the need for fanning of thin triangles. By fixing vertices in UV space unstable geometry artefacts are removed. By continuously interpolating vertices between discrete levels of detail smooth fractional tessellation is achieved. By subdividing triangle and quad domains in the same fashion our method generalises to any polygonal domain making it more versatile to the modelling process.

HPG 2022

Scan me to watch our example videos!