Retro Texel Splatting Toolkit
Asset Store page: https://u3d.as/42RD?aid=1100l8xYY&pubref=Unity
Support: support@ke3dassets.com
Discord: https://discord.gg/NXjZvrk3VG

CONTENTS

1. Introduction
2. Requirements
3. Quick Start
4. Putting Texel Materials on Your Objects
5. Settings Guide
6. Scene Helper Components
7. Included Shaders in Detail
8. Building Your Own Shaders with Amplify Shader Editor
9. Performance Tips
10. Troubleshooting
11. Credits and Support

1. INTRODUCTION

The Texel Splatting Toolkit turns any URP scene into stable, chunky 3D pixel art. Not a screen filter. The pixels (we call them texels) live in the world, stuck to your surfaces, so they hold better while the camera moves. That stability is what makes it read as voxel art instead of a pixelated screenshot; it's not perfect but the imperfections are part of the charm.

The toolkit quietly captures your scene from a point near the camera, works out lighting, shadows and outlines for every texel in that snapshot, crushes the colors into clean pixel art bands, and then rebuilds the picture out of small world locked squares. You do not need to model anything differently or unwrap anything. It works on your existing geometry, in fact you can get very interesting results by simply "converting" existing assets.

What you get out of the box:
- Perspective stabler 3D pixel art on any opaque geometry
- Color posterization with optional dithering (Bayer or blue noise style)
- Ray traced sun and point/spot light shadows within the texel world
- Silhouette and crease outlines, baked or screen space
- World space snow with volume controls and thickness
- A procedural day/night sky
- Optional sharpen and a fixed virtual resolution for that 320x240 look
- Helper components: blob shadows, light blockers, shadow proxies, and a day night cycle

2. REQUIREMENTS

- Unity 2022.3 or newer with the Universal Render Pipeline (URP 14 to 17).
- A GPU with compute shader support. This rules out WebGL2. Web builds are possible with the newer WebGPU target on Unity 6, though performance there is modest and can be challenging.
- On Unity 6 the toolkit runs through Render Graph, which is the default, so there is nothing to switch on. If your project already has Compatibility Mode enabled it quietly uses that instead. Both paths are supported and look identical. Unity 6.4 removed Compatibility Mode, so Render Graph is all there is there.
- MSAA must be off. The Config window (Window > Texel Splatting > Config) checks this for you and can turn it off in one click.

3. QUICK START

The easy way (auto configurator)

1. Open Window > Texel Splatting > Config.
2. Click Add Texel Render Feature to Active Renderer. This adds the render feature to your active URP renderer and wires it to the included settings.
3. If the window warns that MSAA is on, click Turn MSAA Off. That is all it needs.

The manual way

1. Select your URP Renderer asset (the one your pipeline asset points at).
2. Click Add Renderer Feature and pick Texel Splatting.
3. Drop a Settings asset into the feature's Settings slot. Use the included one from the Settings folder to start. You can create your own later with Assets > Create > Texel Splatting > Settings.

Example

Open the demo scene (Scenes/DemoScene) and press Play. Fly around, watch how the pixels stay glued to the world. From there, try a Texel material on one of your own objects (next section) and play with the resolution and settings of the Render Feature.

A note on what gets rendered: objects only appear in the texel world when they use a Texel material (any shader with a TexelCapture pass, which all the included ones have). There is also a fallback toggle in the settings that captures everything else as flat grey, handy for a quick look at a scene you have not converted yet.

4. PUTTING TEXEL MATERIALS ON YOUR OBJECTS

The fastest route is the Material Converter:
1. Open Window > Texel Splatting > Material Converter.
2. Select one or more materials in the Project window.
3. Pick a target shader (Toon Unlit is the recommended default).
4. Click Convert. It copies each material's albedo texture, tint and tiling over, and optionally the normal map, then swaps the shader - the tool looks for common shader property names but it's not 100% accurate.

Or by hand: create a material, set its shader to one of the TexelSplatting shaders, assign your albedo texture. Best option!

Which shader to pick:
- TexelSplatting/Toon Unlit — albedo plus optional normal map. Start here.
- TexelSplatting/Toon Emissive — same, plus an emission channel for glow.
- TexelSplatting/Metal — a fake highlight that tracks the camera.
- TexelSplatting/Glass — simple transparency, skipped by snow.
- TexelSplatting/Toon Unlit Cutout Shadows — alpha cutout that also casts holed shadows (fences, foliage, grates - non-flat surfaces are bad for performance).
- Skybox/SimpleGradient — a quick gradient sky background.

Do not worry that they are called Unlit. Lighting, shadows and posterization all happen inside the texel pipeline afterward. The material's job is only to say what color the surface is.

5. SETTINGS GUIDE

Everything lives on the Settings asset assigned to the render feature. Changes apply live, so tune while in Play mode. The most important settings for the look are Texel Resolution, Posterization Bands and the outline strengths.

Resolution

Texel Resolution: texels per cubemap face edge (128 to 384). This is THE pixel size dial. Lower is chunkier and much faster; cost scales with the square of this number. 256 is a good middle. 128 for strong retro or for weaker hardware.

Capture Projection

Near and far planes of the internal scene capture. Lower the far plane to roughly match your scene size for better depth precision. You rarely need to touch the near plane. The far plane now doubles as how far out the toolkit looks for geometry to capture, so keeping it close to your scene size is the cheapest win in this section.

Wide Capture Culling: works out what is visible once for the whole cube instead of once per cube face. Cheaper, and it stops lights behind the camera dropping out of the shading. Render Graph requires it, so this toggle only changes anything on the older path. Leave it on unless the wider search upsets something else in your project, like a very heavy shadow pass.

Probe Grid
Grid Cell Size: how far the camera moves before the texel world re anchors. Inside one cell the pixels are frozen; cross into the next and the anchor hops and refreshes. Bigger cells are more stable and cheaper but lean harder on the close range fill. Recommended: leave it, go larger only if the world feels like it shifts as you move.

Eye Sub Grid Cell Size: a finer snap for the close range fill. The default values are tuned; small adjustments here trade stability against how quickly freshly revealed areas fill in. Smaller fills revealed gaps faster with a bit more shimmer, larger is calmer but pops in coarser. Recommended: leave it unless disocclusion gaps look slow to fill, then nudge down a step.

Cell Transition Duration: length of the dissolve when the anchor moves. The old and new captures cross dissolve so the hop does not pop. Longer hides it better, zero pops. Recommended: default, raise slightly if you can see cell crossings.

Splat Size and Grazing Scale: how much texel quads fatten to close gaps at silhouettes and on ground seen edge on. Raise Grazing Scale a touch if you can see the background peeking through distant floors. Splat Size covers seams at silhouette edges, Grazing Scale covers foreshortened surfaces like floors stretching away. Recommended: leave Splat Size unless edges leak, raise Grazing Scale gradually if distant floors show background.

Shading
Posterization Bands: how many lightness steps colors are crushed into. Fewer bands, flatter and more cartoon. This shapes the whole image.
Posterize Dither + Pattern + Strength: dithers between bands so gradients read smoother without adding bands. Bayer is the classic cross hatch, Blue Noise is softer and less patterned.
Ambient Intensity: how much the scene ambient fills shadows. Lower it if everything looks washed out and flat.

Virtual Resolution
Locks the final image to a fixed pixel height (like 180 for a 320x180 feel) no matter the window size, via a point upscale as the last step. Great for a fully committed retro presentation.

Procedural Sky
A stylized gradient sky with sun, moon, stars, clouds and haze, driven by your main directional light's angle. Rotate the light (or use the Day Night Cycle component) and the sky sweeps dawn, day, dusk, night on its own. Posterize Sky matches it to the texel look. All the sliders here are safe to play with; they only affect the sky.

Shadows (directional)
Enable Shadows: ray traced shadows inside the texel world. They are baked into the texels, so they do not swim with the camera.
Shadow Strength and Distance: darkness and reach of shadow rays.
Shadow Start Epsilon: raise slightly if flat surfaces speckle with self shadowing.
Dynamic Shadows: keep this on if objects move and should shadow correctly. Turn it off for fully static scenes to save performance.
GPU Shadow BVH: builds the moving caster data on the GPU. Leave on.
Cull Dynamic Casters + Distance: ignore far away moving casters. Leave on for big scenes.
Cull Backface Shadows: cheaper and cleaner for closed meshes, but leaks on single sided things like planes and fins. Off by default.
Rebuild Shadow BVH: tick once after moving static scenery at runtime.

Snow
World space snow that tints and thickens up facing surfaces, broken up by noise. Amount, Slope and the Noise controls shape the coverage. Snow Depth physically raises snowy texels for real thickness. Sky Occlusion keeps snow out from under roofs (needs shadows on). Use Texel Snow Volume components (section 6) to control where snow is allowed. Dynamic objects can opt out with the Exclude From Snow toggle on their material.

Outlines
Silhouette Strength/Color: dark edge where an object meets the background.
Crease Strength/Threshold/Color: edge where a surface bends sharply.
Outline Thickness: how many texels wide edges are.
Screen Space Outlines: computes outlines on the final frame instead of baking them into texels. Slightly more stable while moving, one extra pass.

Optimizations
Texel Culling: on. Biggest single performance win, leave it on.
Shade Cache: on. Skips relighting work when nothing changed.
Coplanar Merge: merges flat 2x2 texel blocks into single quads. Helps on scenes with lots of flat surfaces.

Sharpen
An edge aware sharpen over the final image. It crisps the chunky edges without needing any extra detail in your art. Try it with Strength around 0.5 and adjust Edge Boost to taste. The Edges debug view shows what it considers an edge.

Compositing
Capture Fallback Geometry: draws unconverted objects as flat grey so they at least show up. Turn it off once your scene is converted, and keep it off if you use cutout materials (the grey fill plugs their holes).

Clear Color Before Splat + Background Color: show only the texel world over a solid color, instead of overlaying it on the normal render.

Integration
Apply Fog: applies the scene's URP fog to the texel world.

Debug
Debug View: inspect the raw position, normal, depth or albedo data if something looks wrong. None for normal rendering.

6. SCENE HELPER COMPONENTS

All under Add Component > Texel Splatting.

Texel Snow Volume
A box that controls where snow may appear. Exclude mode carves snow out of interiors (put one inside your building). Include mode flips the logic: if any Include volume exists, snow only appears inside them. The box is simply the object's transform, so scale it like a cube.

Texel Light Blocker
A cheap invisible box that blocks light rays. Use it to seal walls or ceilings so the sun or a point light does not bleed into an interior, without adding geometry to the shadow system. Add an empty, add the component, scale the box over the wall.

Texel Shadow Proxy
Put this on a high poly object that casts shadows. It swaps the mesh for an auto generated low poly hull inside the shadow system only, which keeps shadows cheap. The visible mesh is untouched. Bake the hull from the component's context menu, or let it auto generate at runtime.

Texel Blob Shadow
A soft fake shadow blob that sticks to the ground under an object and fades as it rises. Perfect for billboards, particles and anything the real shadow system should skip. Everything it needs is generated at runtime.

Texel Day Night Cycle
Drop it on your main Directional Light. It rotates the sun over time, which drives the procedural sky through a full day and night. Has speed, arc and a scrubbable time of day.

7. INCLUDED SHADERS IN DETAIL

All Texel shaders do the same core job: they describe the surface (color, normal, emission) to the capture step. The pipeline handles everything else. That means no per material lighting settings to manage, and materials stay very light.

TexelSplatting/Toon Unlit
Albedo map, tint, optional normal map. The base shader, use it unless you have a reason not to.

TexelSplatting/Toon Emissive
Adds an emission map and color. Emission glows through the lighting and still posterizes into the palette, so neon and screens sit nicely in the pixel art look.

TexelSplatting/Metal
A stylized metal with a highlight that roughly follows the camera. Because true reflections do not exist in the texel world, this fakes the read of metal convincingly at pixel art scale.

TexelSplatting/Glass
Simple transparent material rendered outside the texel capture. It does not posterize and snow ignores it. Good for windows and bottles where you just need believable see through.

TexelSplatting/Toon Unlit Cutout Shadows
Alpha cutout with a matching cutout shadow caster. Fences, grates, leaves and other holed surfaces cast correctly holed shadows. If you use cutout materials, remember to turn off Capture Fallback Geometry (section 5, Compositing), otherwise the fallback fills the holes with grey.
PERFORMANCE NOTE: Non-flat surfaces (spheres, capsules, etc) can be very slow to compute; if you see a drop in performance, you might have enabled it somewhere on a non-flat surface.

Every Texel material also has an Exclude From Snow toggle. Turn it on for characters and props that move, so they do not accumulate snow.

8. BUILDING YOUR OWN SHADERS WITH AMPLIFY SHADER EDITOR

The toolkit includes a template that lets you build custom Texel shaders visually, no code. This is optional; the hand written shaders cover the common cases.

Get ASE here:
https://assetstore.unity.com/packages/tools/visual-scripting/amplify-shader-editor-68570?aid=1100l8xYY&pubref=Unity

Starting a new shader

The included ASE shaders (ASE Texel/TexelSplatting Main, Emissive, and the two Billboards) are all built on the template. The quickest start is to duplicate one of them and open it in ASE. TexelSplatting Main is the plain base, Emissive adds a glow input, and the Billboard pair make camera facing quads (the DOOMER-Style one mimics classic sprite enemies).

What your graph controls

Your nodes drive the surface: albedo color, normal, emission, alpha clip, and vertex position. Plug textures, blends, masks, vertex animation, whatever you like into those. World bending, scrolling textures, wind sway, dissolves through alpha clip, all of that works.

What the pipeline controls

Lighting, shadows, posterization and outlines are applied later by the toolkit itself. So do not build lighting into the graph; it would fight the pixel art shading. Think of your shader as answering one question: what color and shape is this surface right here.

Things to know

- Keep materials opaque or alpha cutout. The texel world is built from opaque surfaces; true transparency belongs to separate shaders like the included Glass.
- The template exposes the Exclude From Snow toggle automatically, so your custom shaders get snow control for free.
- Alpha clipped materials automatically cast holed shadows. Nothing extra to set up.
- Screen space tricks (grab pass style effects, depth fades against the camera) will not carry into the texel world, since capture happens from the probe's viewpoint rather than the camera's. Build these effects in world or object space instead. Swap Screen Position and scene color / grab nodes for World Position, World Normal or UV based inputs, and where you would fade against camera depth, fade against world space distance or a world height gradient instead. Anything that must react to the camera (a fake highlight, rim style shading) should read world space direction rather than screen space, and remember it evaluates from the probe near the camera, not the exact camera, so keep it broad rather than pixel precise.
- Billboards face the capture as well as the camera, which is what keeps them

looking right inside the texel render. Use the included billboard shaders as your base rather than a generic billboard setup.
- The shaders ship compiled for URP 14 through 17. If you edit one in ASE it regenerates for your installed URP version, which is normally exactly what you want.

For programmers (hand written shaders)
A shader joins the texel system by adding a pass tagged LightMode = "TexelCapture" that renders the surface into the toolkit's G buffer. The capture pass draws your geometry from each probe face and sets the view/projection and a global _TexelProbeOrigin for you, so your job is only to output surface data in the layout the pipeline expects.

The contract is a four target MRT, defined in Shaders/Core/SurfaceContract.hlsl:

```
SV_Target0 = float4(worldPosition.xyz, chebyshevDepth)   // RGBAHalf
SV_Target1 = float4(worldNormal.xyz,   snowExclude)      // RGBAHalf
SV_Target2 = float4(albedo.rgb,        materialId)       // ARGB32
SV_Target3 = float4(emission.rgb,      0)                // RGBAHalf
```

You do not fill that struct by hand. Include the file and call one of the helpers from your fragment function, returning its result:

* WriteTexelSurface(worldPos, worldNormal, albedo, materialId) for a plain opaque surface.
* WriteTexelSurfaceEmissive(worldPos, worldNormal, albedo, materialId, emission) to add glow.
* WriteTexelSurfaceFull(..., emission, snowExclude) when you also need the snow opt out (pass 1 to keep snow off this surface, for moving objects).

They compute the Chebyshev depth from the probe origin and normalize the normal for you, so the texel reads as valid.

The vertex stage just needs to pass world position and world normal through, and set positionCS with TransformWorldToHClip (that uses the per face matrix the capture pass installed). Suggested pass state matches the built in shaders: #pragma target 4.5, Cull Back, ZTest LEqual, ZWrite On, under a "RenderPipeline" = "UniversalPipeline" SubShader.

Two rules to keep in mind. First, do not add lighting: no light loops, no shadow sampling, no posterization. Those all run later in the compute shade pass, and doubling them up fights the pixel art shading. You are only describing the surface. Second, stay out of screen space for the same reason ASE shaders do: capture happens from the probe, not the camera, so build anything view dependent from world space data.

The included hand-written shaders (TexelToon_Unlit, TexelToon_Emission, and the cutout one) are compact, copyable examples of exactly this.

9. PERFORMANCE TIPS

In order of impact:

1. Texel Resolution. Quadratic cost. 192 or 128 transforms performance and often looks even more on style.
2. Shadows. Turn off Dynamic Shadows in static scenes. Keep caster culling on. Use Shadow Proxies on high poly casters. Lower Shadow Distance.
3. Cutout materials on non-flat surfaces. Alpha cutout casters can't use a low poly shadow proxy, so they feed their full mesh into the shadow system and sample the alpha map on every ray hit. Cheap on a flat card, expensive on a dense mesh. Keep cutout for flat-ish geometry (foliage, fences); use an opaque material plus a Shadow Proxy for solid complex shapes.
4. Snow. Free to disable if you do not use it. Sky Occlusion is the pricey part of snow; leave it off unless you need it.
5. Keep Texel Culling and Shade Cache on (they are on by default).
6. Capture Far Plane. Match it to your scene size instead of the default. It also sets how far out the toolkit searches for geometry, so this matters more than it used to.
7. Coplanar Merge helps architectural scenes with lots of flat surfaces.
8. Mesh Read/Write: shadow casting needs it enabled on the mesh import settings for static meshes in builds (see Troubleshooting).

The scene captures and relighting only run when the camera crosses a grid cell or something changes, so a calmer camera is naturally cheaper.

10. TROUBLESHOOTING

The Editor is very slow when first opening in the scene view
Give Unity a few seconds to warmup the shaders, it will turn back to normal after.

Everything renders pink
The shaders did not compile for your URP version. Reimport the Shaders folder.

Nothing happens / no texel look
Check the render feature is on your ACTIVE renderer (the Config window tells you) and that a Settings asset is assigned. Both URP render paths are supported now, so you do not need to change Render Graph or Compatibility Mode either way.

Edges look soft or the pixels look smeared
MSAA is on somewhere. Open Window > Texel Splatting > Config and use Turn MSAA Off. If you switch quality levels at runtime, use Turn MSAA Off Everywhere instead, since each quality level can point at its own URP asset and only the active one gets fixed otherwise.

Objects show up flat grey
They are being captured by the fallback. Convert their materials to a Texel shader (section 4), or turn off Capture Fallback Geometry if you want unconverted objects hidden from the texel world.

An object casts no shadow in a build but does in the editor
Enable Read/Write on that mesh's import settings. The shadow system needs CPU access to static meshes, and builds strip it by default. Skinned characters are handled on the GPU and do not need this.

Shadows leak through a wall
Walls thinner than the shadow ray offset can leak. Raise the wall's thickness, lower Shadow Start Epsilon carefully, or place a Texel Light Blocker over the wall (the reliable fix for interiors).

Snow indoors
Put a Texel Snow Volume set to Exclude around the interior. For moving props, enable Exclude From Snow on their material.

Cutout holes filled with grey
Turn off Capture Fallback Geometry (section 5, Compositing).

The sky draws over everything / weird sky layering
Make sure the feature's Injection Point is After Rendering Skybox (the default on fresh installs).

Flicker or crawling pixels while moving
Expected at cell transitions in tiny amounts (the dissolve hides it). If it is strong, raise Cell Transition Duration slightly, or lower Eye Sub Grid Cell Size a step.

Performance is low
See "PERFORMANCE TIPS", Section 9

11. CREDITS AND SUPPORT

Based on "Texel Splatting" by Dylan Ebert (MIT License).
https://github.com/dylanebert/texel-splatting

This toolkit is an independent URP implementation; not a direct 1-1 port.

OKLab color space by Björn Ottosson.

Questions, bugs, ideas:
Email: support@ke3dassets.com

Thanks for supporting the toolkit. Now go make something chunky.