VoxelTerrain

Voxel Terrain Research

Here is the overview of research made in 2009, comments are welcome to Vladimir Kajalin

Little tutorial for level designers can be found here: Voxel Terrain Tutorial

Overview

The system is internally based on the sparse voxel octree (SVO), allowing complete 3D editing. The SVO contains information about topology and surface types. From this data, meshes and textures are generated and then used for rendering on usual hardware. Texturing works in mega-texture way and allows pre-baking of decals, roads or terrain detail layers into high-resolution final texture.

Visualization is supported now on PC, Xbox360, PS3 and even iPad1. Real-time modifications or editing works only on PC.

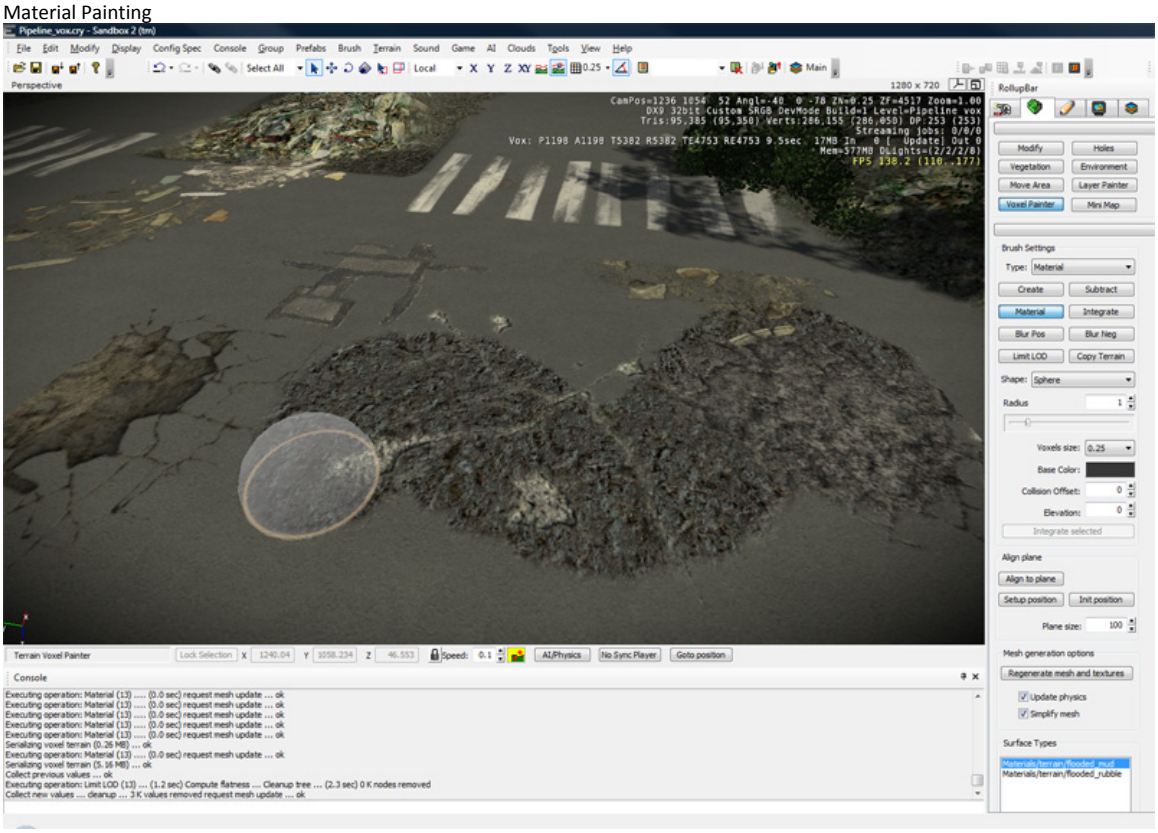
Good example of VoxTerrain usage. Except vegetations rest is VoxTerrain (including roads and man-made structures)

Key Advantages (Compared to the Old System)

- Multi-resolution editing: The designer can create geometry features and texturing with any level of detail, allowing very detailed environments in important gameplay areas, and saving of memory usage, disk space, and performance on faraway, never reachable areas.
- 3D editing of mesh: Any type (and any size) of cave, cliff, or canyon is possible. Editing tools are based on the improved voxel object editor. Optionally, the old heightmap editor can also be used.

- 3D layer painting: High-resolution terrain base textures may be painted even on vertical walls (without texture stretching), allowing the creation of cliffs (for example) that look good without the use of detail material passes (compared to the old system, it saves draw calls and fill rate).

Material Painting



- In the case of PC, multi-threaded generation of meshes and textures allows the stall free reconstruction of terrain segments on the fly, from a relatively small, sparse octree dataset. Alternatively, for low-specification PCs or consoles, mesh and textures can be streamed from the disk, minimizing the usage of the system memory, CPU, and GPU.

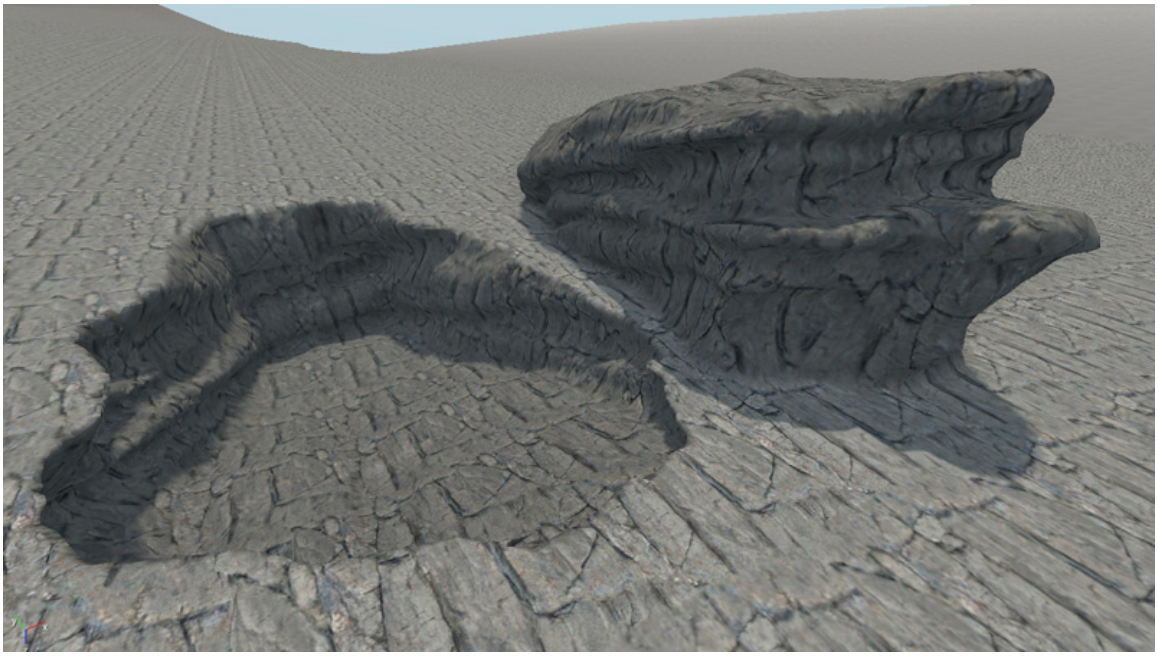
- This technology allows the reduction of the number of objects in the level. Earlier, designers placed thousands of brushes, like rocks on a terrain, mostly because it was not possible to create interesting terrains by using only the heightmap. Now, a terrain can look exactly how the designer wants it, without additional objects being used. This method helps us save draw calls.

- An unlimited number of roads and decals can be easily baked into a high-resolution terrain texture, allowing draw calls to be saved and completely eliminating the alpha blending (fill rate) overhead. Baking is GPU accelerated and absolutely transparent in terms of level design. A designer works with decals and roads in the same way as before. It is baked on the fly so that designer can immediately see the final look. The number and size of decals is unlimited. For example, an 8k x 8k satellite photo can be applied to an entire terrain as a decal or hundreds of small decals can be used on every square meter of the terrain, to get a nice look. All those decals will not affect the performance in the final game.

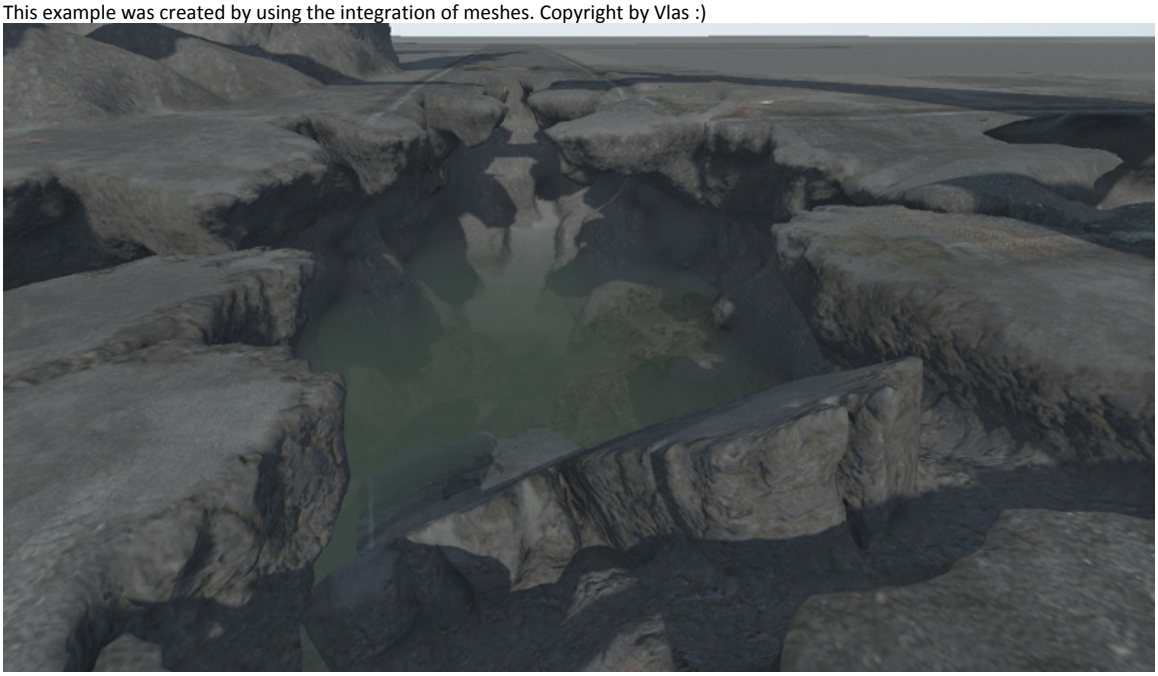
- It is possible to optimize the levels by integrating objects (like rocks) into the terrain mesh. In the current system, we render a lot of invisible triangles because a lot of them are located inside of the other objects or terrains. After the integration, all the invisible triangles get cut, like with Boolean operations. The hundreds of objects that intersect each other get transformed into a nice, single mesh containing triangles only on the visible surface.

- The possibility to integrate meshes into the voxel octree opens a new way of terrain editing. Instead of manually sculpting an entire terrain, the designer can use the integration of the existing brush meshes to speed up the process. Meshes can not only be added into the terrain, but also subtracted in a way that is similar to Boolean operations.

The mesh has been integrated twice here: once in the additive way and once in the subtractive way.

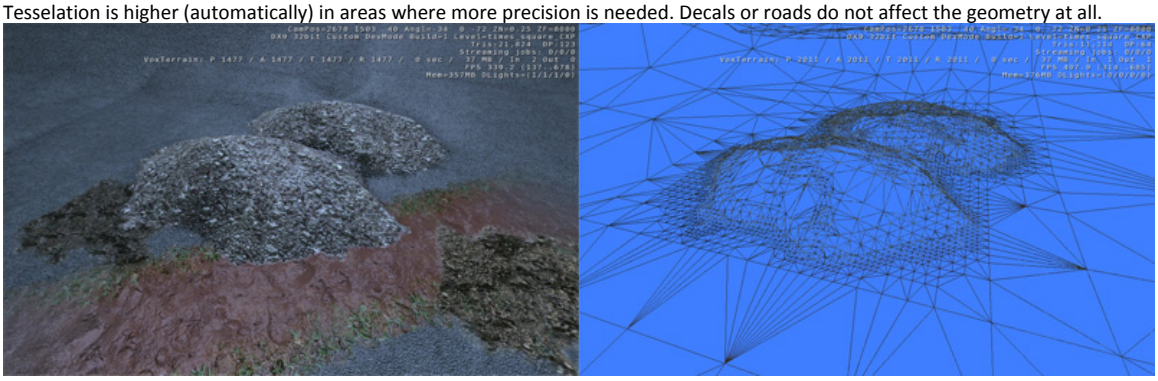


This example was created by using the integration of meshes. Copyright by Vlas :)



- Adaptive mesh resolution allows to increase the render speed because less triangles and vertices are used, compared to the old uniform grid system, which was based on the heightmap.

Tesselation is higher (automatically) in areas where more precision is needed. Decals or roads do not affect the geometry at all.



- There is no old compatibility problem between voxel objects and the heightmap; everything is handled by one general system. In the old system, when a lot of voxel objects were used (for example, for big alien caves), there was a problem of too many draw calls. In the new system, the far geometry has been efficiently merged and simplified into single meshes, saving draw calls, memory space, and fill rate.
- A terrain base texture contains not only diffuse color, but also the per-pixel normal and specular factor for hi-quality lighting.

Lighting; all decals and texture layers have been pre-baked.

- Real-time modifications of terrains will be possible from mid-specification PCs, allowing the creation of detailed craters (for example) from explosions, or even huge step marks from T-Rex.
- Old levels can be converted into the new system.

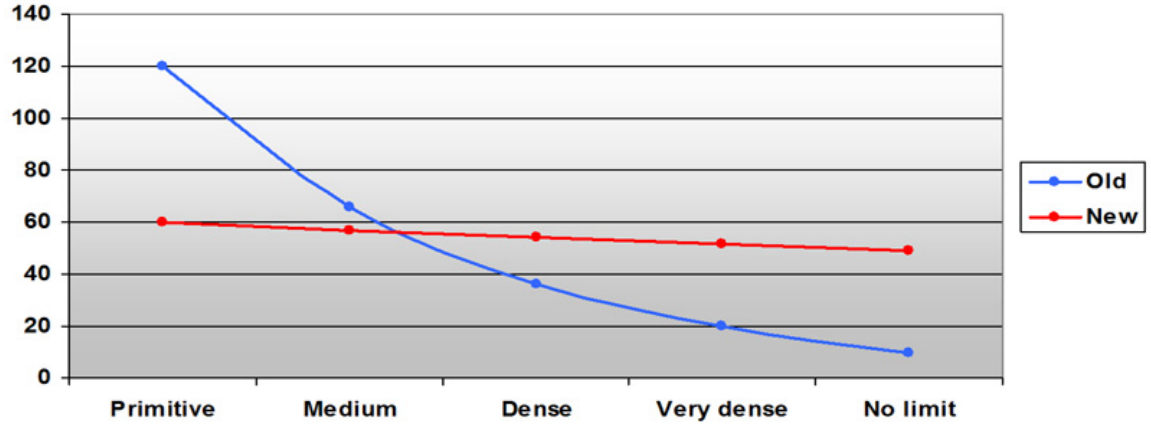
Key Disadvantages and Missing Features

- Editing is not polished for now. Some operations may take a long time if high detail levels are used.
- If texture prebaking is used, levels may take some of the disk space. For example, the Game04 benchmark level when converted into the new system takes about 160 MB and the GDC 2009 demo level takes about 200 MB on the disk.
- In the case of primitive levels, switching to the new system does not bring any performance or memory benefits.

More Performance

Performance increase depends a lot on the properties and complexity of the scene. In the case of simple scenes (flat low-resolution terrain, simple layers setup, little amount of decals and roads), switching to the new system will not bring about any performance increase and may even work slower. However, if the scene has a lot of decals, roads, terrain layers, and complex terrain topology, the new technology will render faster than the old solutions. This technology attempts to decouple the rendering speed from the scene complexity so that the designer can create more complex scenes, with less worry about performance.

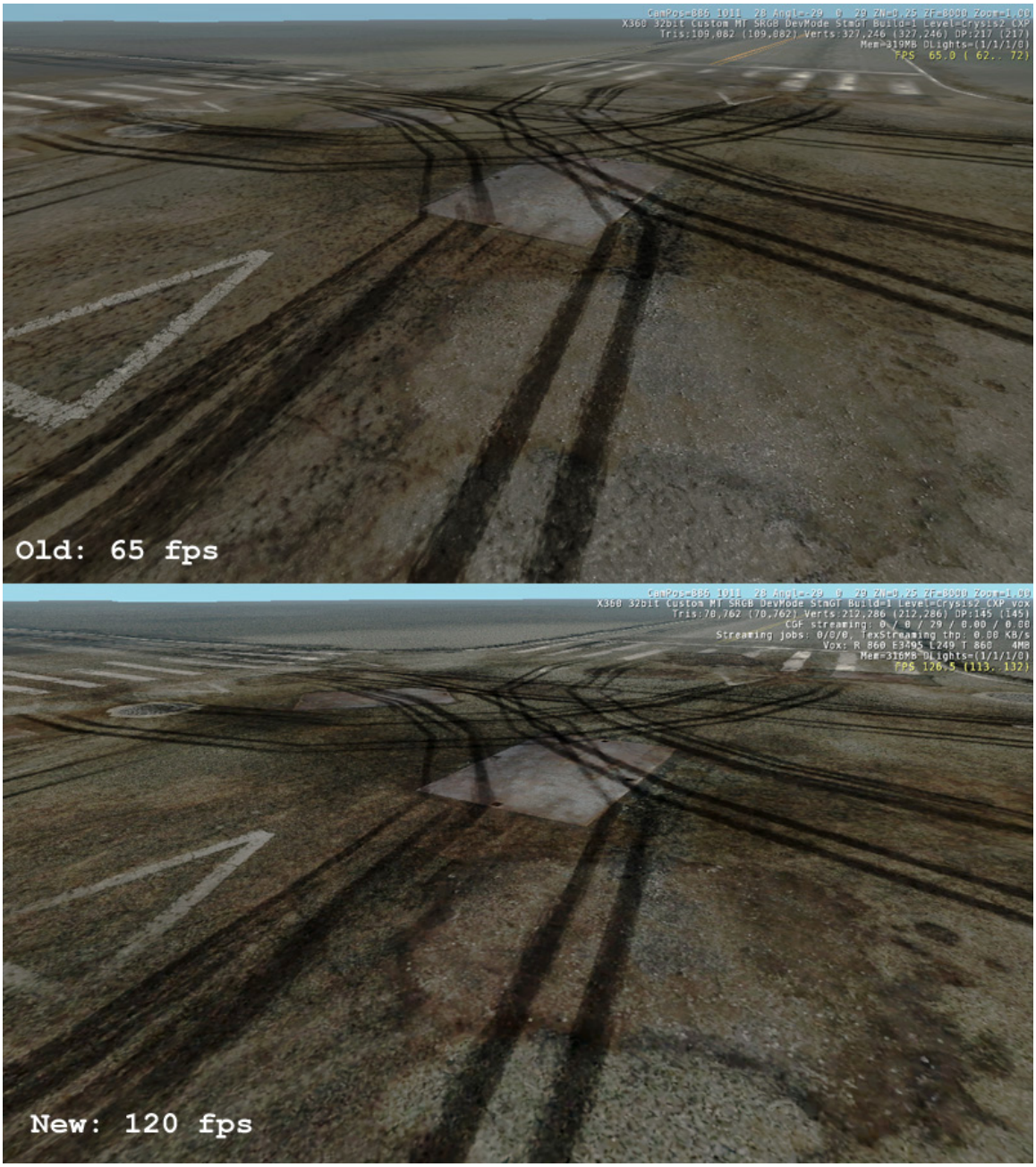
This simple chart will help you to understand how the frame rate depends on the scene complexity in both the old and new systems. As you can see, the new technology is more efficient for complex scenes.



Here, we compare the rendering of 300 decals on Xbox. Old solution on the left - 40 fps; new solution on the right - 130 fps.

Note: Decals on the left have unpredictable, inconsistent sorting; decals on the right have very consistent, "never flicker" sorting

Roads, decals and terrain from real level on XBox. Old solution - 65 fps; new solution - 120 fps.



Less Memory Usage

If we convert some simple level into the new system, the allocations on Xbox will increase by about 5 MB. However, if we take a level with a lot of unique decals, roads, and many terrain detail materials, conversion will reduce the memory allocations because all those objects and textures are not needed on the client PC or console. As the number of normal textures becomes smaller, the texture streaming pool size can be reduced allowing to save even more memory. Tests with Game03 Pipeline level have confirmed that switching to the new technology allows the reduction of the overall memory allocations.

Console Variables

To activate the new terrain system for a level, this line has to be placed into level.cfg (in the level folder) and the level has to be reloaded.

e_VoxTer=1

It is normal for the old heightmap to be not visible after the reload.

Mesh and Texture Resolution Control

Terrain data can use up a lot of disk space, especially if textures prebaking is used (it is the default solution for now). Therefore, it is very important to limit the mesh and texture resolution for not reachable areas and keep full resolution only in areas where the player can come in. Level designer can use shape tool or usual road objects in order to define areas where full mesh and texture resolution is needed.

The terrain near the gameplay area supports up to 25 cm triangle size (it is artificial limit) and up to 1 cm texel size. The resolution is reduced for areas outside the game-play area, depending on the distance. Hence, visually, there will be no quality lost.

Integration of the Meshes into the Terrain

In order to integrate a mesh:

1. Place several brush objects into the level. Convex objects like rocks work better.

2. Select Integrate Pos* (Integrate Positive means add mesh into the terrain) or *Integrate Neg (means subtract mesh from the terrain).

3. Select the voxel size. It affects the tessellation and detail level of the final mesh.

4. Select the terrain layer in the Terrain Texture Layers panel.

5. Set the base color to something dark (for example, 30,30,30).

6. Select the brush objects for integration.

7. Now, click Integrate Selected in the Voxel Painter panel.

Usually, the following text will be displayed in the console:

Preparing brushes for integration ... 3 brushes found, 3 mesh groups produced
, press Ctrl-Break to abort ...

Executing operation: Integrate Pos (13) ... 354 tris ... Integrating (1/3) ... (0.0 sec) request mesh update ... ok

Executing operation: Integrate Pos (13) ... 354 tris ... Integrating (2/3) ... (0.1 sec) request mesh update ... ok

Executing operation: Integrate Pos (13) ... 354 tris ... Integrating (3/3) ... (0.1 sec) request mesh update ... ok

Finished integrating brushes

After this, it will take some additional time to recompute the terrain geometry and textures.

Please note that the texturing of the original mesh is completely ignored in the current version.

Visual Artifacts During Editing

Currently, the terrain mesh may not look correct during modifications. Seams between sections of mesh may be visible or parts of the mesh may be temporarily missing. Mesh computations are very CPU intensive and it may take a few seconds for the designer to get to see the final look. This will improve in future. Also, it is better to disable shadows during editing. Doing this avoids shadows from not-ready (low LOD) geometry.

Level Exporting

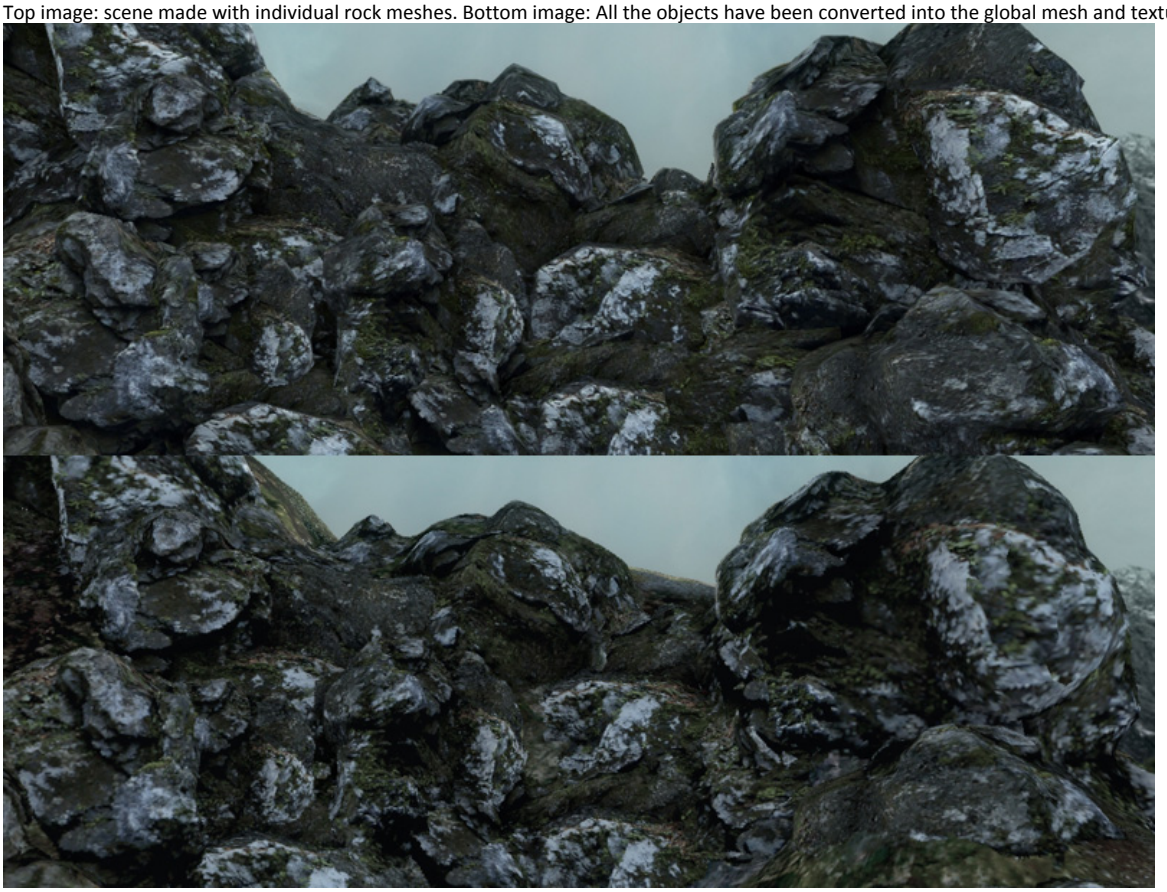
Currently only areas visited by camera will be exported.
During first level exporting process all needed textures and meshes will be first saved into cache directory USER\VoxelTempCache and then exported into level.pak\terrain\voxmap.dat. On next export if all necessary data is already in the cache only fast copy into level.pak will be performed.

Bonus Possibilities

One of the features in research is the possibility to integrate meshes, taking into account not only geometry, but also textures. The goal is to make the scene look the same even after most of the objects have been integrated into the global voxel mesh. The current implementation allows for example the conversion (with texturing) of terrain and all rock objects of the Game04 benchmark level.

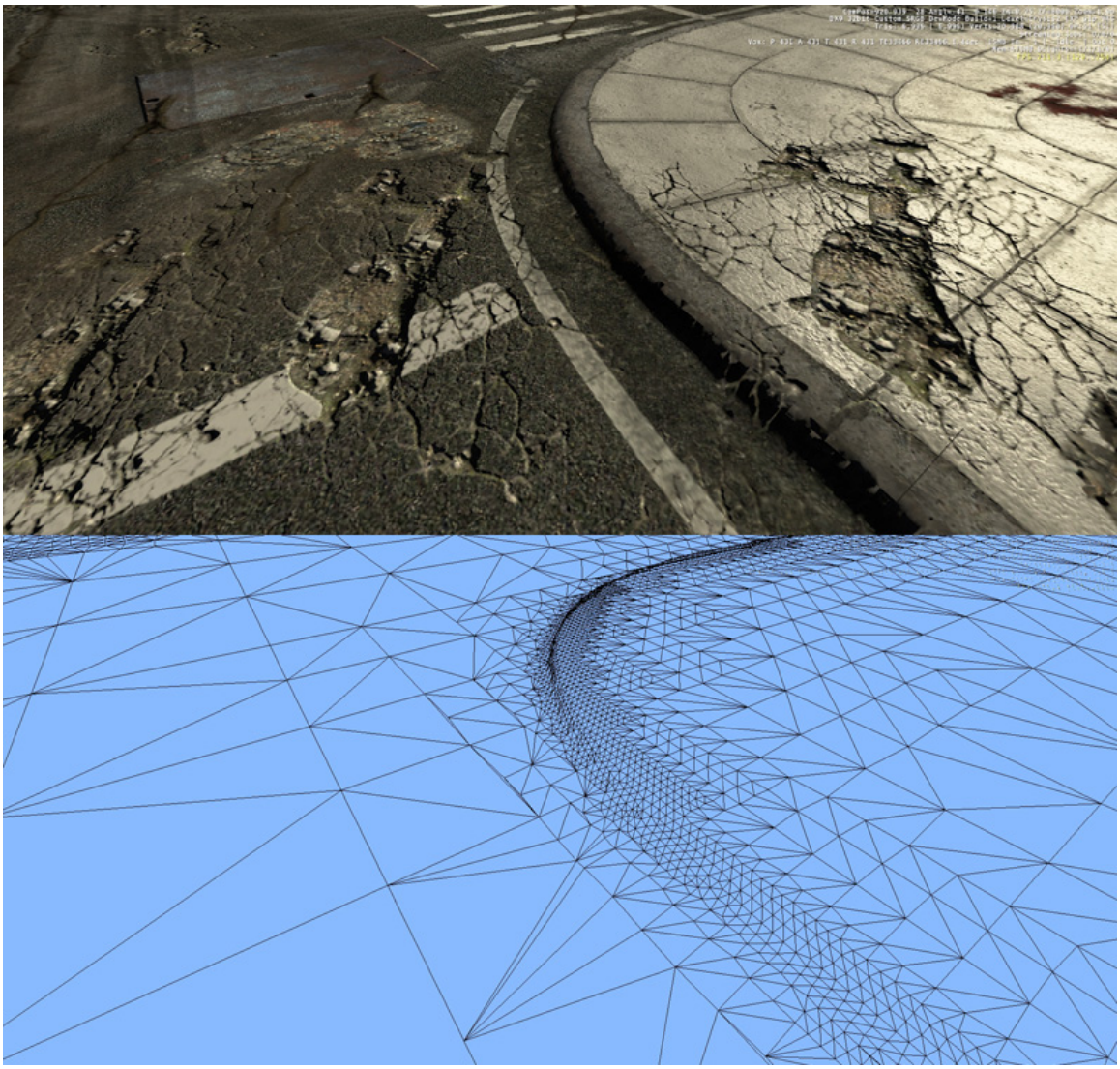
On the left: scene made with the usual brush objects. On the right: meshes and textures have been converted into the new system.

Top image: scene made with individual rock meshes. Bottom image: All the objects have been converted into the global mesh and texture.



This technology has some properties of PolyBump; all the high-resolution content of the level (geometry and textures) gets converted (re-meshed) into new geometry and new textures which can be rendered more efficiently. The small or high-resolution geometry details get automatically converted into texture and may be excluded from the final mesh.

Sidewalk mesh integration example.



Few Videos

RealtimeDeformAndBaking.avi - Very simple in-game volume subtraction and decals baking test

iPadDemo.avi - Voxel_pipeline test level running on iPad1

Binaries

Here is latest working build (from February 2010): (Link is removed)

Let me know if you want to try it, it maybe not obvious how to use it

This build contains Voxel_pipeline level (shown in overview section above) and also 2 test levels for another project. All of them represent different variations of this terrain engine.

8x8 km of mega-textured height-map. Advanced materials blending is used, allowing multiple terrain materials mixed at flexible blend ratios. (In default CryEngine every height-map unit has only one material assigned). Special "Overlay" mode is used for mixing of detail layers with terrain base color. Terrain detail materials are visible at full view distance.

Similar but fully 3D. Surface types blending resolution is 25 cm.