Spider-Man IG-Impostors

Cityscapes and Beyond

Xray Halperin
Senior Technical Artist
Insomniac Games
Burbank, CA, USA
siggraph@agentxray.com

David Santiago
Principal Technical Artist
Insomniac Games
Burbank, CA, USA
dsantiago@insomniacgames.com

Abdul Bezrati
Senior Engine Programmer
Insomniac Games
Burbank, CA, USA
abezrati@insomniacgames.com

ABSTRACT

Spider-Man's traversal though Manhattan in the video game Marvel's Spider-Man (2018) for Sony's PS4 platform allows the player to climb any building and jump off any structure while rotating the view 360 degrees at 30 frames per second and displayed at 4k resolution. Flat card "billboard" style impostor systems could not represent the city environment at the desired quality so Insomniac Games developed the 3D IG-Impostor system to represent the mid to distant cityscape for Marvel's Manhattan in an efficient and persistent cache. This environment data cache was then available for multi-view rendering used by other systems within the Insomniac Engine. There are no 2D impostors in Marvel's Spider-Man.

CCS CONCEPTS

• Interactive Games • Computer Graphics

KEYWORDS

video games, level of detail, open world, LOD, PlayStation, Spider-Man

Spider Man IG-Impostors for cityscapes:

Xray Halperin, David Santiago and Abdul Bezrati. 2018. Spider-Man IG-Impostors: Level of Detail for Cityscapes and Beyond. In *Proceedings of ACM SIGGRAPH Asia 2018 Technical Briefs, Tokyo, Japan, December 4-7, 2018 (SA '18 Technical Briefs), 4 pages.*

https://doi.org/10.1145/3283254.3283259

ACM ISBN 978-1-4503-6062-3/18/12…$15.00

Figure 1: All distant background buildings are IG-Impostors while the ones directly beneath Spider-Man are full quality.

1 Introduction

A small team of technical artists and core engine programmers at Insomniac Games evolved the processes first developed for the game Sunset Overdrive (2014) into the Spider-Man IG-Impostor system to represent the entirety of Manhattan in full 3D with high quality reflective and emissive surface properties while residing in the smallest possible memory footprint.

Billboard-based 2D impostor techniques [Maciel and Shirley, 1995] lack the parallax and surface properties required to render the mid to distant vistas and cityscapes for Manhattan in the game Marvel's Spider-Man (2018) with the fidelity required. Camera based billboard textures solve parts of the parallax issue, but the memory footprint of the textures required for such a system is prohibitive due to the number of assets and large size of the open world. Furthermore, 2D billboard-based impostor systems are unable to deal with real time reflections representing weather and time of day on glass and mirrored surfaces.

2 IG-Impostor Requirements

The requirements of the IG-Impostor system developed for Marvel's Spider-Man are:

1. Always maintain fidelity under all circumstances
2. Eliminate far clipping plane draw-in by keeping the cache of geometry and textures in the smallest possible amount of persistent global memory
3. Maintain geometric relief details of the source assets
4. Support fully emissive and reflective surfaces for glass, mirrored towers and night time illumination
5. Transition as seamlessly as possible between high-res instanced geometry and the corresponding IG-Impostor.
6. Inform other game systems using the persistent IG-Impostor cache

3 Workflow and Pipeline

When artists wish to generate an IG-Impostor from an asset in the Orbit layout editor they choose the IG-Impostor tool from a pulldown menu to launch a background process that generates optimized geometry and textures that are ultimately published to the Insomniac Games IG-Impostor format.

Buildings in the Insomniac Engine are composed of model instances of architectural elements such as walls, doors, and windows. Each model may have multiple materials applied to its various LOD's. There may be many thousands of such model instances in an individual building asset. An IG-Impostor consists of three elements: A proxy geometry model, a texture atlas, and a set of polygons representing high-quality reflective and emissive surfaces. IG-Impostors swap the collection of individual high-resolution model instances and their assigned materials with the three IG-Impostor components.

Figure 2: Marvel's Manhattan at sunset. All buildings on the island of Manhattan in this screen capture are IG-Impostors.

3.1 Geometry Processing

Once an architectural asset is created in the Orbit level editor the first part of the IG-Impostor pipeline passes the geometry to Side Effects Software's Houdini® digital content creation (DCC) application where the geometry is decimated and optimized using several techniques and error checks designed to maintain geometric integrity while reducing the total vertex and polygon count. Small details such as rooftop furniture, air conditions and other assorted details are tagged with a "skip-impostor" attribute and ignored by the system.

The geometry decimation step consists of three distinct loops. Firstly, the geometry is grouped by connectivity and polygon normal facing direction. For each group of connected polygons that share facing normals in common, shared edges are detected and removed. Self-intersections are resolved, and the summed area of the result is compared with the sum of the area of the source geometry. If the summed areas vary more than a user specified threshold the original geometry is retained.

Secondly, for each group of like facing connected polygons in the result of the first step, the polygon edges are resampled, and a new triangle mesh is generated. The vertices are consolidated within a user specified threshold and once again shared edges are removed to create an n-gon representing the outline of the shape. In-line points generated by the resampling step along the edges are removed and self-intersections are resolved. Once again, the sum of the area of the result is compared against the sum of the area of the source geometry, keeping the original geometry if the variance is too high.

Lastly, an ambient occlusion test tags occluded geometry for deletion by converting the n-gon geometry into triangles and scattering points across the geometry. If all the sampled points on a given triangle are marked occluded the triangle in question is removed.

The resulting proxy model for the IG-Impostor still has a significant polygon count however this is needed to meet the IG-Impostor requirement of maintaining geometric relief detail.

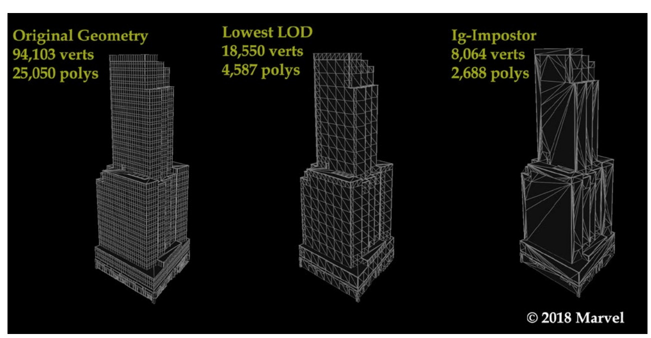


Figure 3: Example of building asset at original LOD, lowest LOD and final IG-Impostor.

3.2 Texture capture and Compression

The decimated geometry is sent to the IG-Impostor generation command line program to generate a compressed geometry file and a pair of four channel texture atlas files. The command line instantiates the Insomniac Engine and renders the full quality high resolution asset. Capturing a set of six orthogonal views for each asset stores the render pass information from the assigned materials into the RBGA channels of the atlas files. The resolution of each IG-Impostor atlas is dependent on the pixels per meter desired for the asset. The area of all polygons in a material group is measured and an appropriate value between two user defined defaults representing the number of pixels per meter to capture the texture is chosen based on the sum area of the polygons of each captured material group. The pixels per meter may be overridden by an artist on an asset by asset basis if the artist desires more texture detail. Polygons representing high-quality reflective and emissive surfaces are also captured at this stage.

3.3 Special Emissive and Reflective Elements

Manhattan is filled with glass towers and signage which require accurate real time reflection and emission. As an integral part of the art direction for Marvel's Spider-Man an adjunct system to deal with these "high-quality" portions of the IG-Impostor assets was developed to minimize the expense of rendering so many emissive and reflective surfaces.

By adding a "High-Quality" attribute to materials applied to emissive and reflective architectural surfaces, such as windows and lit signage, the polygons with such surfaces are isolated into a set of model instances from the lowest artist supplied LOD. Most of the time these are low poly count quads representing panes of glass however this process works equally as well on more complex geometry, such as signage, with the expected additional expense.

The high-quality geometry capture occurs concurrently with the process that captures the snapshot views for the IG-Impostor textures. The high-quality geometry is saved along with the decimated geometry in the proprietary binary IG-Impostor geometry file.

When the IG-Impostor is loaded by the Insomniac Engine at runtime the transforms and hierarchy for the set of all high-quality model instances are concatenated and flattened out as a single pass of model instances rendered with a distance-based Z-bias so the polygons float just over the textured IG-Impostor. The full quality material properties from the high-res assets are assigned to the high-quality polygon set and they are rendered together as a single optimized pass at runtime with the applied Z-bias.

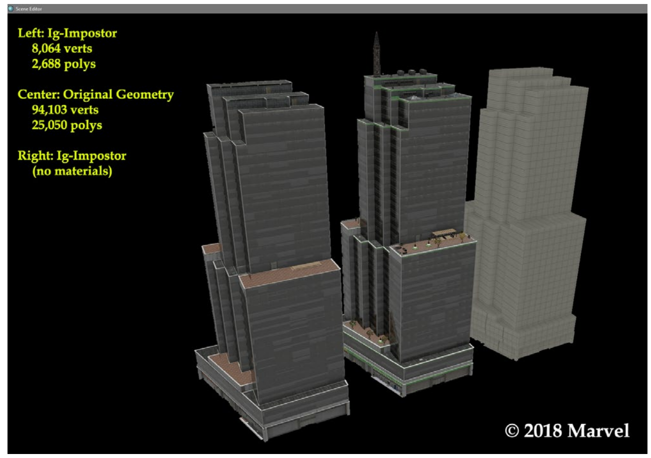


Figure 4: Same building as in *figure 3* rendered in the Insomniac Engine with shading and high-quality polygons for glass reflections

3.4 Swapping IG-Impostors

Marvel's Spider-Man is a tile based streaming game. As the player traverses the environment high-resolution geometry and textures with LOD and MIP-mapping for the current tile stream in and out of memory while the IG-Impostors are loaded once and remain in persistent global memory. Only the lowest MIP from the IG-Impostor texture atlas is persistent. Higher MIPs from the IG-Impostor texture atlas stream in and out based on player distance from the impostor as needed.

The open world consists of 725 tiles, each 128 x 128 meters in size. The contents of each tile are contained in tile .zone files. The IG-Impostors are organized into their own set of .zone files which represent a group of 3 x 3 tiles, an area of about 1.2 square kilometers. The texture atlases for each IG-Impostor asset are packed into a single mipmapped atlas per IG-Impostor zone representing all buildings within the IG-Impostor zone. The per zone texture atlas is variable in size with a maximum resolution of 8192x8192 pixels. The actual size of the texture atlas for a given IG-Impostor zone is dependent upon the physical size of the assets in the zone as represented by some number of pixels per meter per asset. The limiting factor for the number of tiles included in an IG-Impostor zone is the resolution of the IG-Impostor zone texture atlas. All the textures for the assets within an IG-Impostor zone are stored in the pair of texture atlas files representing the diffuse and glossy components of all materials for every asset for 9 tiles. Each IG-Impostor asset within a zone is individually addressable such that any given asset may swap with its corresponding IG-Impostor based on the camera distance from the asset.

The decimated geometry, the high-quality geometry, and the textures are stored in a propriety lossy compressed binary format. While this does create some quantization and decreased precision it's necessary to stay within performance specifications and memory footprint. The total memory footprint for all the IG-Impostor geometry in Marvel's Spider-Man open world is about

101 megabytes. The reflective and emissive high-quality polygons comprise about 30 megabytes of memory. The texture footprint depends on the MIP level and ranges from 90 to 900 megabytes.

When the player is within a tile, the contents of the tile's .zone file streams into the Insomniac Engine at high resolution for full quality rendering. Since the IG-Impostor geometry and lowest MIP of the texture atlas already reside in persistent global memory there is no need to stream them, however the Insomniac Engine needs to fade off the display of any IG-Impostor asset that corresponds with any asset in the currently loaded high-res tile as it fades in. The high-res assets and IG-Impostor representations are alpha blended to fade in and out to minimize popping.

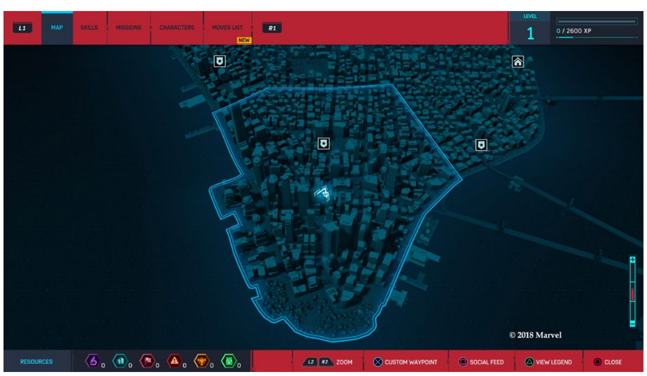
4 Leveraging IG-Impostor data for other game systems

Once the initial requirements of the IG-Impostor system were met other teams at Insomniac Games leveraged the efficient and persistent IG-Impostor cache to improve the performance of other game systems.

4.1 Open World Map

As seen in *Figure 2* the IG-Impostors are quite detailed for mid to distant representations of Manhattan. To construct the open world map for Marvel's Spider-Man, rather than generate a specific map data set the Insomniac Games IG-Impostor team in conjunction with the FX department and UI department leveraged the same data for IG-Impostors to represent the city map, thus saving time, memory and disk space.

Since the IG-Impostor geometry is already loaded into persistent global memory, when the player activates the open world map the in-game camera instantly transforms up into the sky and the IG-Impostor materials are overridden with a material designed to match the other UI elements in the game. With the addition of some art directed content the resulting open world map consistent with the game UI and accurate to the open world is generated.



8,

Figure 5: IG-Impostors used to represent in-game map of Marvel's Spider-Man open world.

4.1 Environment Probes

Marvel's Spider-Man utilizes a system of environment probes to drive global illumination specular response and reflections in the Insomniac Engine. Every tile in Marvel's Spider-Man has several environment probes placed at strategic locations. Each environment probe consists of a 512 x 512 cube map. Initially the environment probe textures were generated offline with high-resolution assets. Generating this data across 725 game tiles, each with multiple environment probes, results in many thousands of environment probe texture files. When it became clear that the sheer number of environment probe textures was prohibitive with regards to disk space the core engineering team developed a system to generate the environment probe textures at runtime by using the persistent IG-Impostor cache.

4.2 Cascaded Shadow Map (CSM) Shadow Occlusion for Distant Light Sources

Cascaded Shadow Maps [Engel, 2006] ensure that the distant light shadows of objects closer to the camera render at higher quality than shadows for distant objects. Implementing a system to test for CSM Shadow Occlusion for distant lighting from the sun shining over Manhattan allows the Insomniac Engine to optimize by including or excluding the objects casting distant light shadows thus saving precious milliseconds of time which would otherwise go to waste rendering shadows that do not contribute to the final image.

The IG-Impostor CSM Shadow Occlusion Cache is regenerated about once per second by rendering an orthogonal shadow buffer from the sun position with over-scan to accommodate any motion within the refresh time. Shadow casting assets are tested against the IG-Impostor CSM Shadow Occlusion Cache to see if they are completely covered by an existing shadow or have a chance of contributing to the distant light CSM. Any asset whose silhouette is subsumed by other IG-Impostor shadows in the CSM Shadow Occlusion Cache is removed from consideration when generating the final distant light shadow map. The CSM Occlusion Cache serves as the furthest slice in the Cascaded Shadow Map in addition to occluding casters into each slice of the final CSM.

ACKNOWLEDGMENTS

Tony Arcuilo, Tom Breeden, Tiffany Chu, Kelly Fitzpatrick, Al Hastings, Jason Hickey, Josh Noble, Jenna Ruth

REFERENCES

PAULO W. C. MACIEL AND PETER SHIRLEY. 1995. Visual navigation of large environments using textured clusters. In Proceedings of the 1995 symposium on Interactive 3D graphics (I3D '95). ACM, New York, NY, USA, pp. 95-102 DOI=http://dx.doi.org/10.1145/199404.199420

WOLFGANG F. ENGEL. 2006. Cascaded Shadow Maps. ShaderX5, Advanced Rendering Techniques, Wolfgang F. Engel, Ed. Charles River Media, Boston, Massachusetts, USA, pp. 197–206.