



2008 ESRI Developer Summit

March 17–20, 2008 • Palm Springs, CA

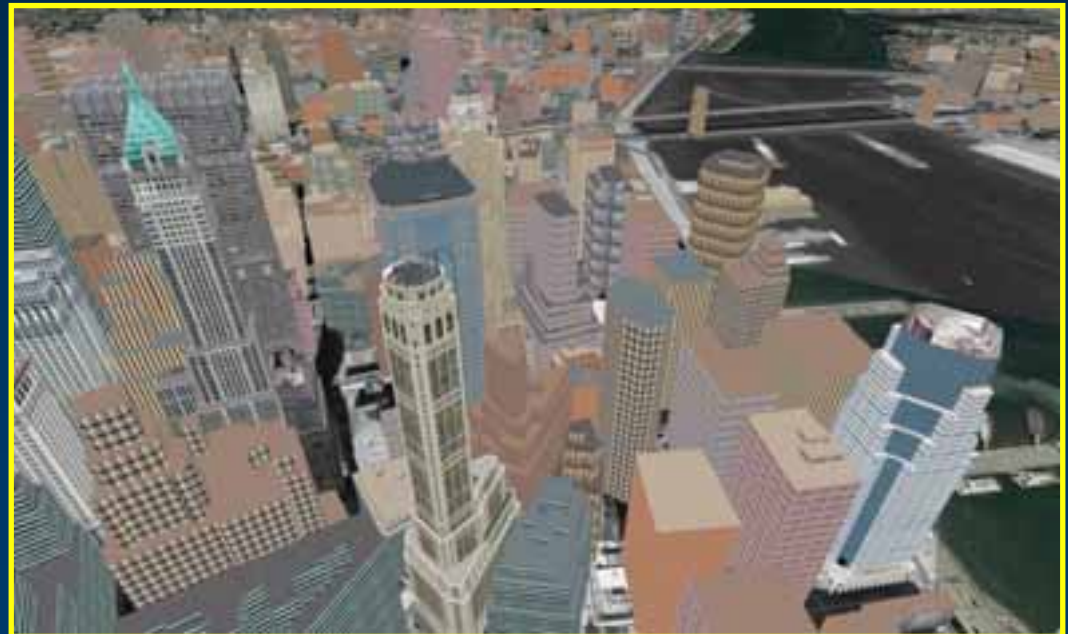
Creating a 3D Virtual City Using ArcGIS

Milad Ershaghi
Product Engineer, 3D Analyst

Deepinder Deol
Product Engineer, 3D Analyst

Overview

- **Creating the 3D Dataset**
- **MultiPatch Geometries**
- **Authoring the 3D View**
- **Serving the 3D View**



Outline: Creating the Dataset using MultiPatches

- Plan

- Data Collection
- Data Display
 - Out of the Box
 - Custom

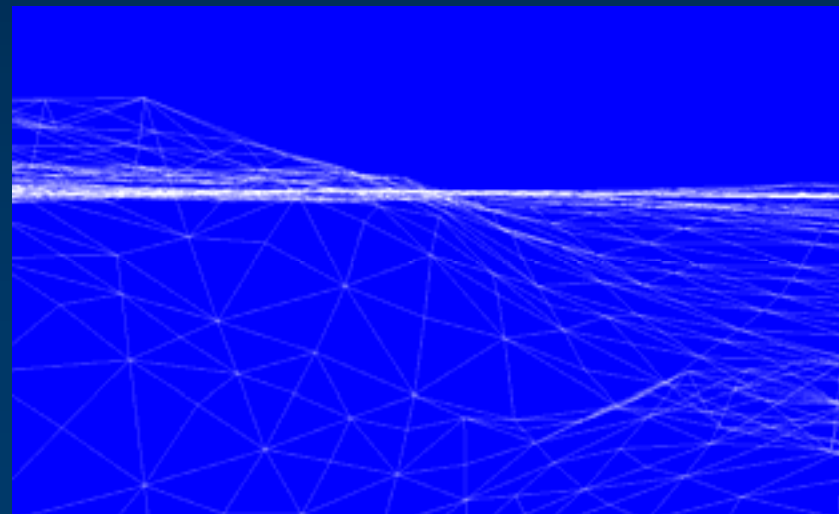
- Develop

- Single Building
 - MultiPatch Geometry
 - IGeneralMultiPatchCreator
- Entire City
 - Workflow
 - Custom Solution



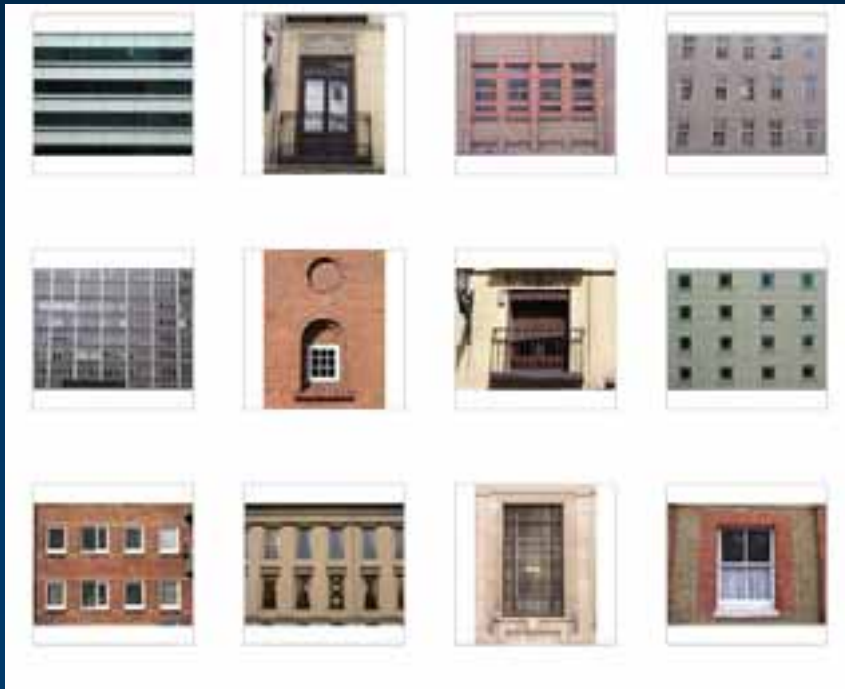
Data Collection: Backdrop

- **Transportation Routes**
- **Aerial Imagery**
- **Elevation Data**



Data Collection: Buildings

- Building Footprints
- Building Textures
- Building Models



Data Display: Elevation, Imagery



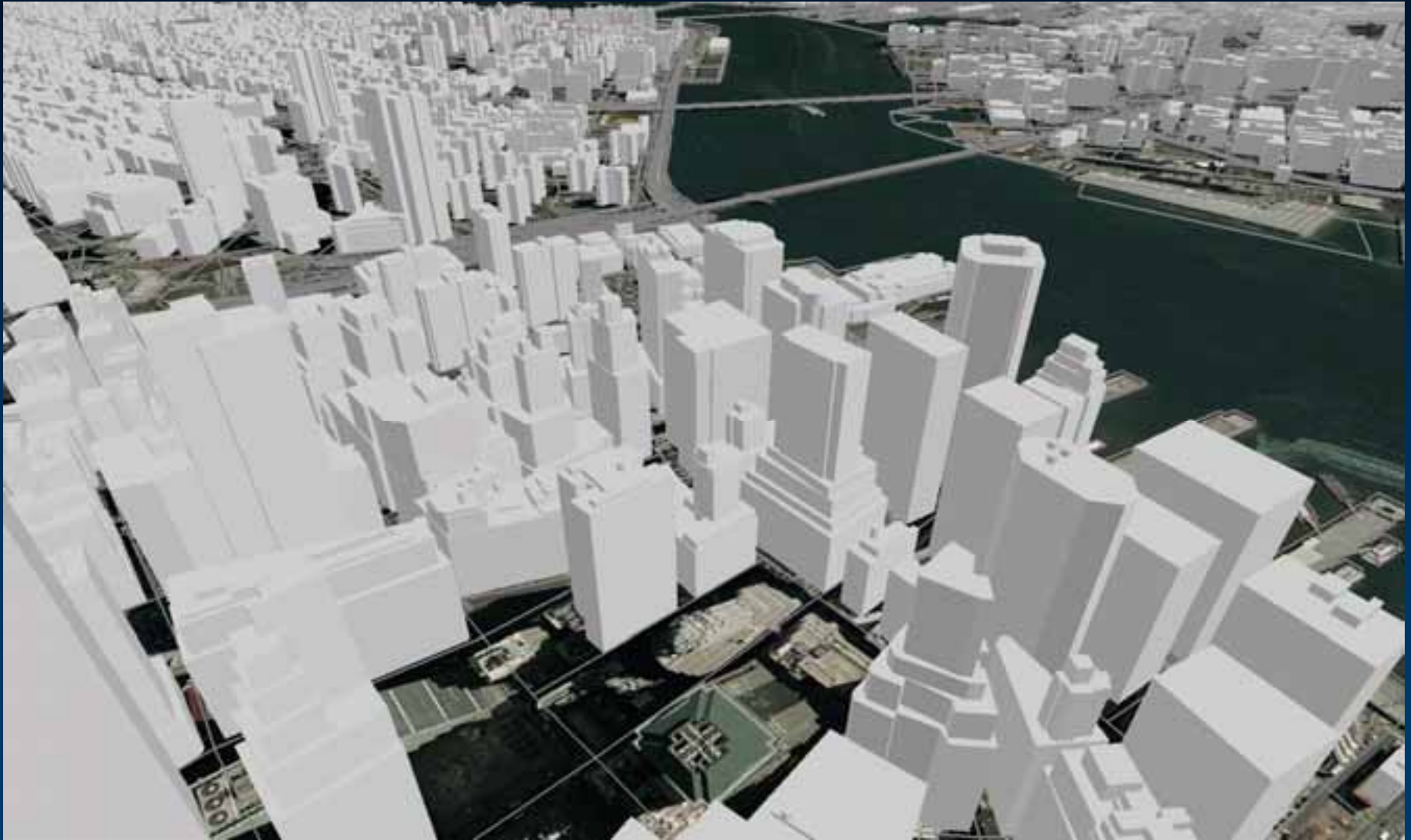
Data Display: Transportation Routes



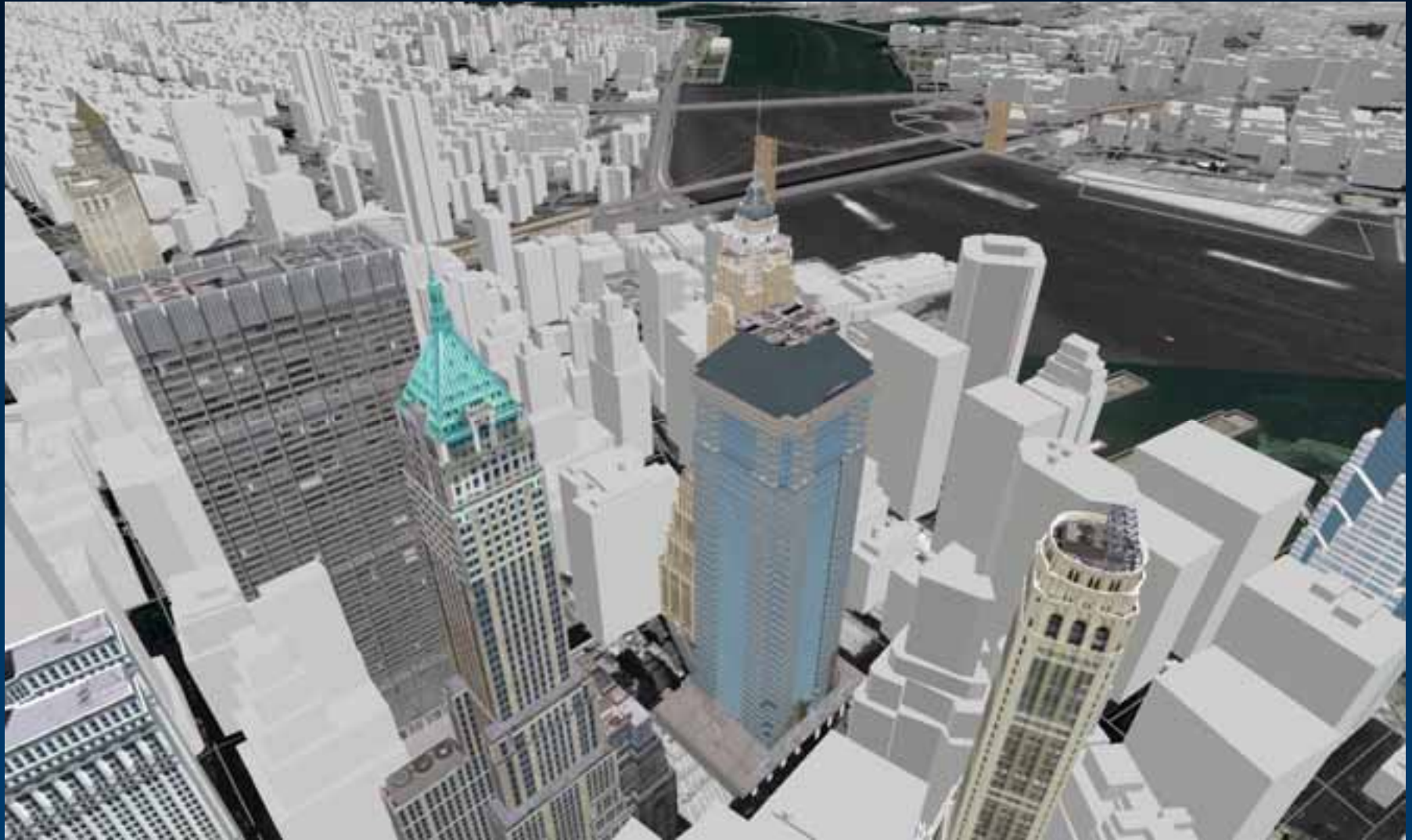
Data Display: Building Footprints



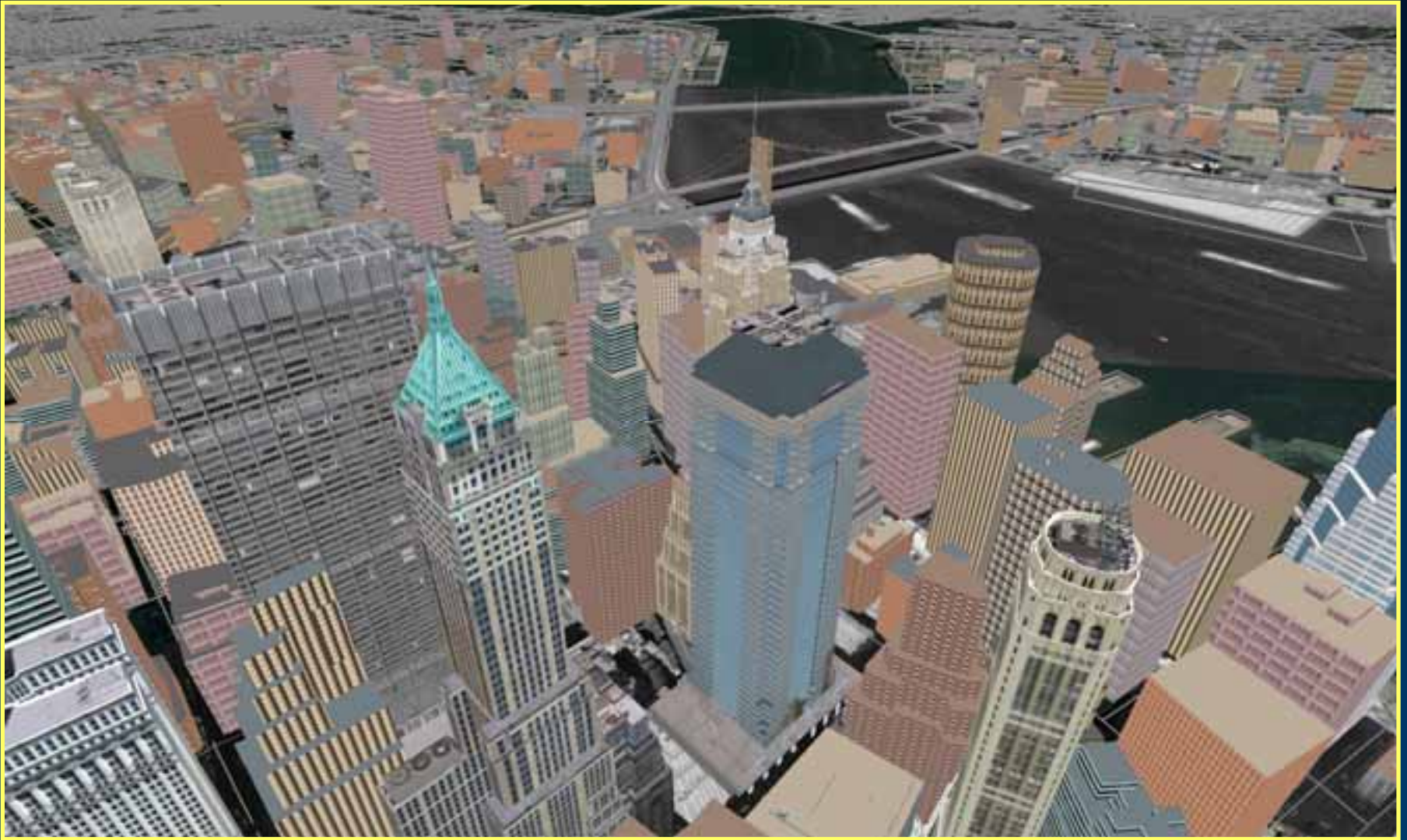
Data Display: Extruded Footprints



Data Display: Building Models

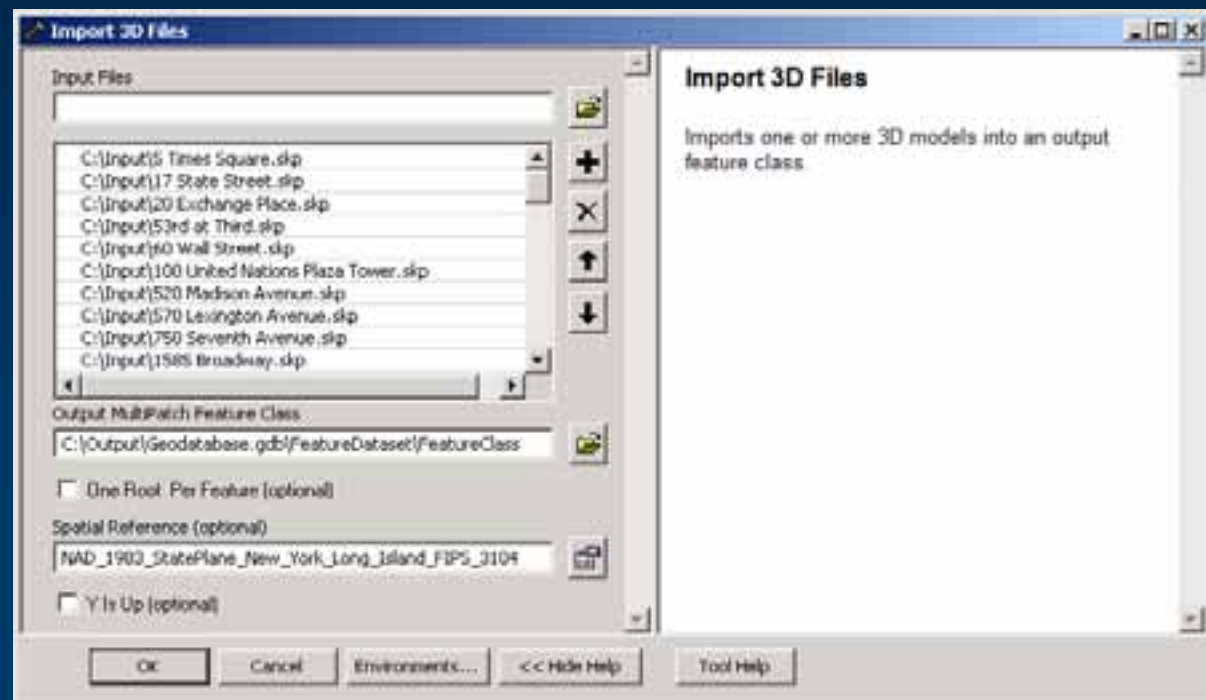


Data Display: Textured Buildings



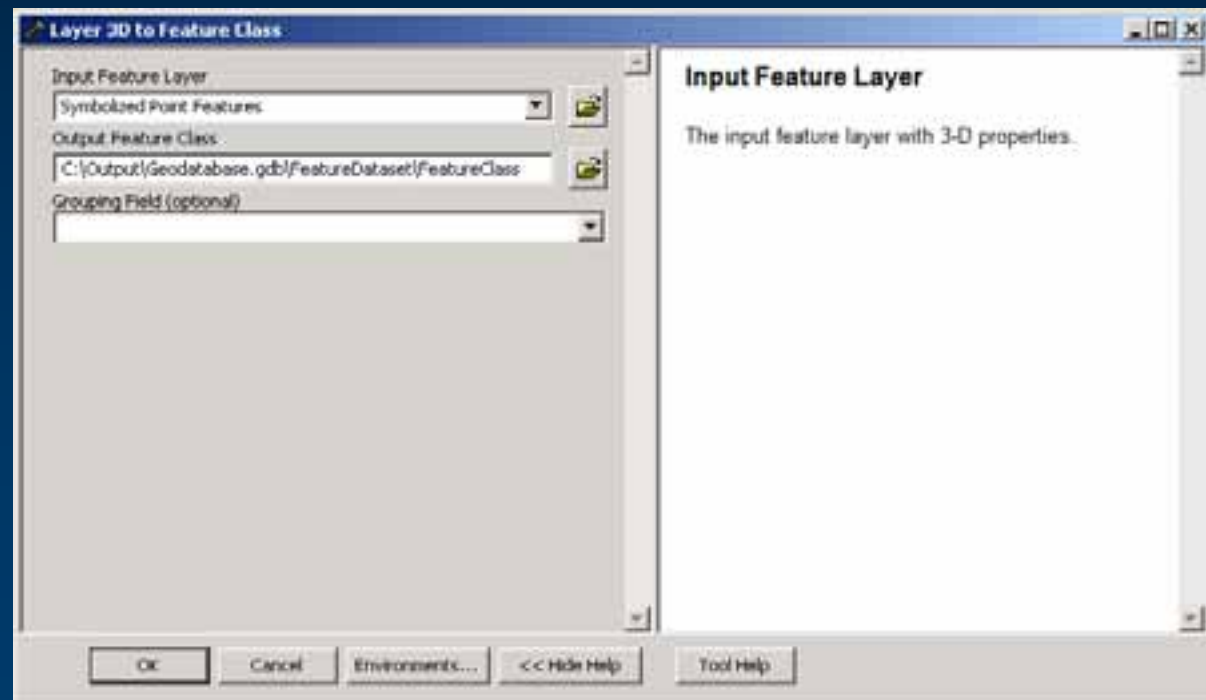
Geoprocessing Tool: Import 3D Files

- Input Formats
 - Collada
 - SketchUp
 - OpenFlight
 - 3ds Max
 - GeoVRML



Geoprocessing Tool: Layer 3D to Feature Class

- 3D Symbology
- Extrusion Settings
- Texture Downsampling



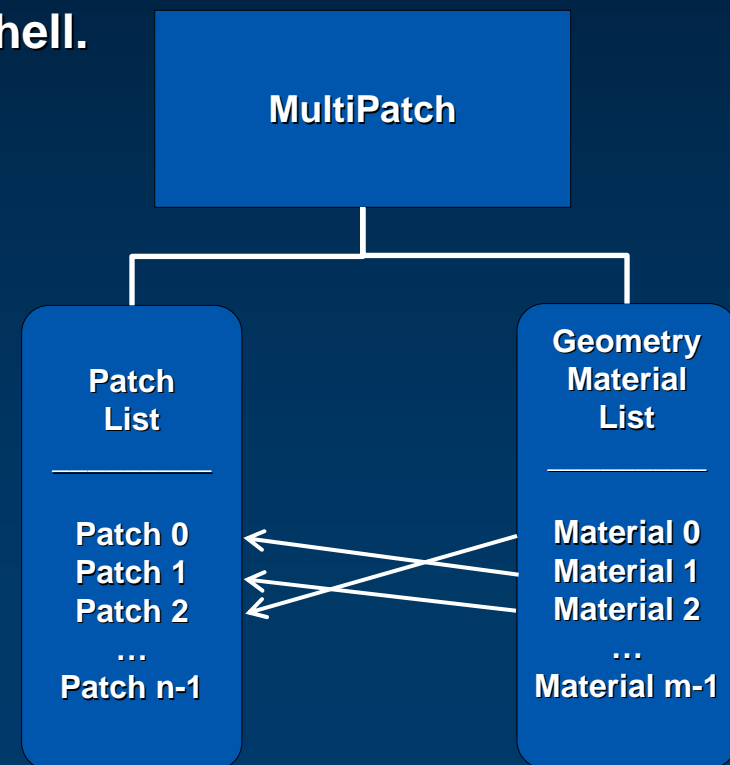
Custom Solution: Textured Buildings Generator



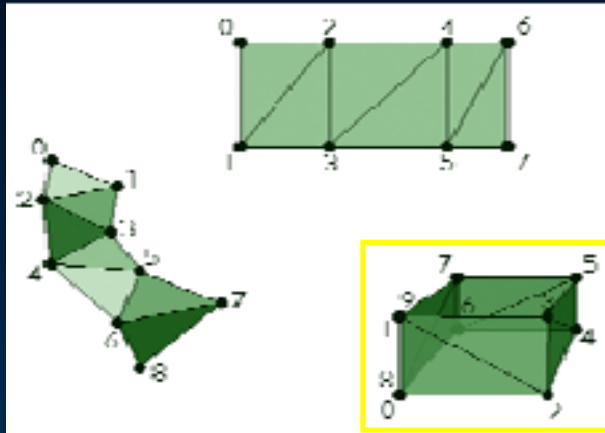
Single Building: MultiPatch Geometry

MultiPatch: a 3D geometry used to represent the outer surface/shell of features that occupy a discrete area/volume in three-dimensional space. MultiPatches comprise 3D rings and triangles that are used in combination to model a three-dimensional shell.

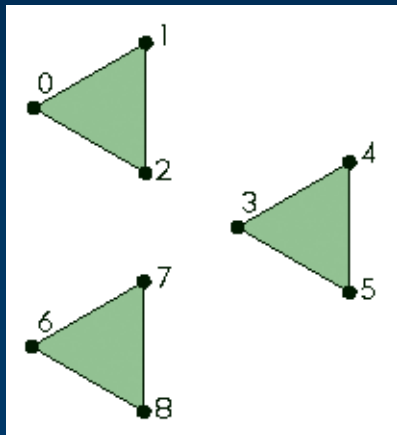
- **One or More Patches**
- **Zero or More Geometry Materials**
- **1 to Many Relationship (Geometry Material : Patch)**



Single Building: Patches

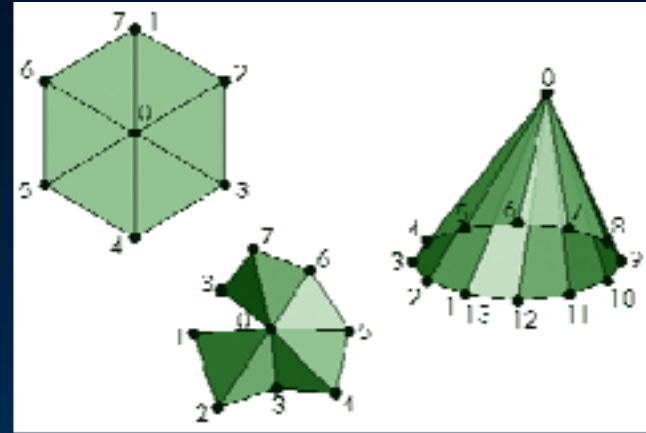


TriangleStrip



Triangles

Building Walls



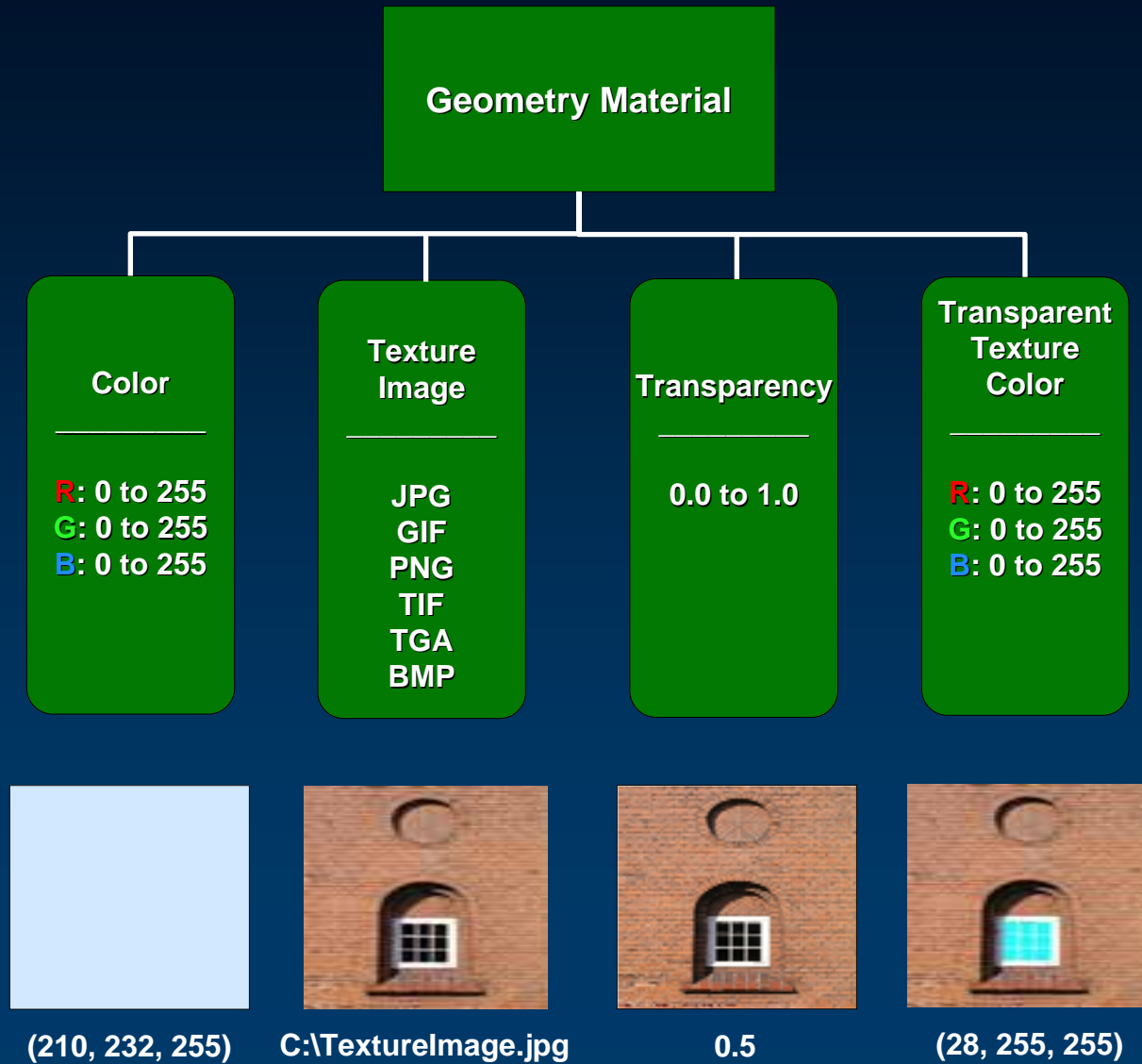
TriangleFan

Building Floors,
Building Roofs



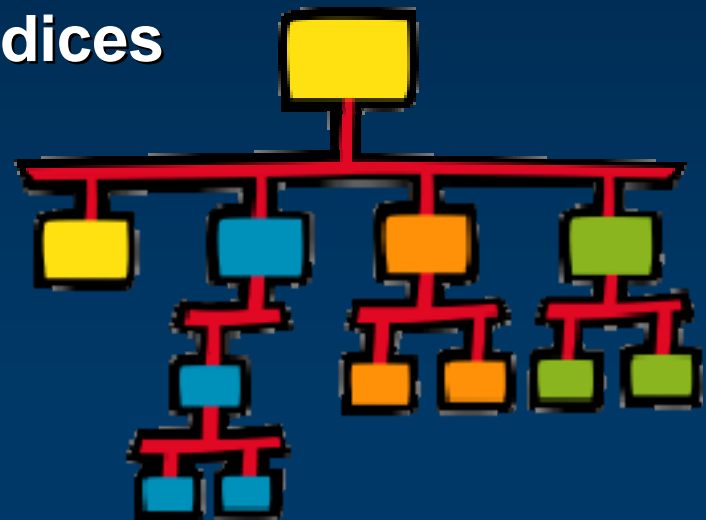
Ring

Single Building: Geometry Materials



Single Building: Steps

1. Define Geometry Material List
2. Initialize General MultiPatch Creator
3. Define Patch Types
4. Define Material Indices
5. Define Point Starting Indices
6. Define Texture Point Starting Indices
7. Define Points
8. Define Texture Points
9. Generate MultiPatch
10. Cleanup Resources



1. Define Geometry Material List

```
//Define Geometry Materials

IGeometryMaterial floorGeometryMaterial =
    new GeometryMaterialClass();

floorGeometryMaterial.TextureImage =
    @"C:\Texture Images\FloorTexture.jpg";

IGeometryMaterial roofGeometryMaterial =
    new GeometryMaterialClass();

roofGeometryMaterial.TextureImage =
    @"C:\Texture Images\RoofTexture.jpg";

IGeometryMaterial wallGeometryMaterial =
    new GeometryMaterialClass();

wallGeometryMaterial.TextureImage =
    @"C:\Texture Images\WallTexture.jpg";

//Add Geometry Materials to Geometry Material List

IGeometryMaterialList geometryMaterialList =
    new GeometryMaterialListClass();

geometryMaterialList.AddMaterial(
    floorGeometryMaterial
);

geometryMaterialList.AddMaterial(
    roofGeometryMaterial
);

geometryMaterialList.AddMaterial(
    wallGeometryMaterial
);
```



FloorTexture.jpg



RoofTexture.jpg



WallTexture.jpg

2. Initialize General MultiPatch Creator

```
//Define Initialization Parameters

int partCount = 3; //Total # of patches participating in MultiPatch

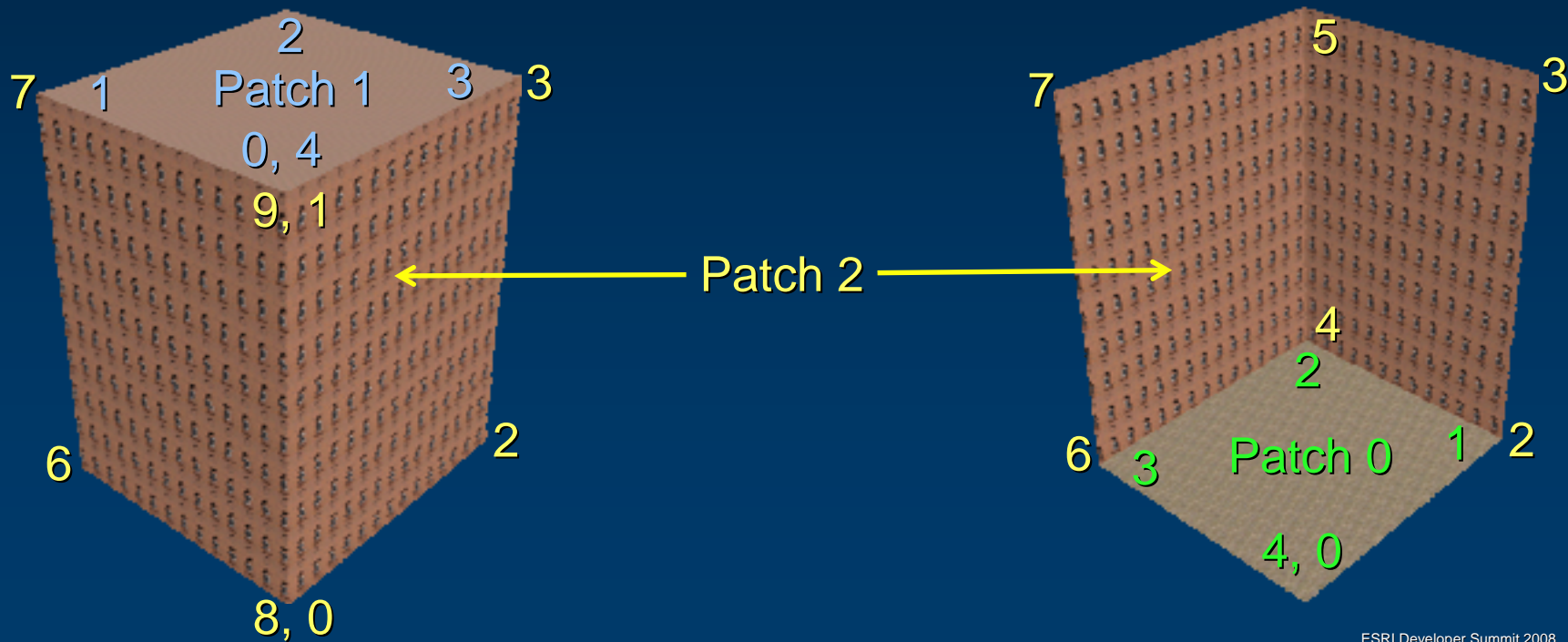
int pointCount = 5 + 5 + 10; //Total # of geometry vertices in all patches participating in MultiPatch
int texturePointCount = pointCount; //Total # of texture vertices in all patches participating in MultiPatch

bool hasMs = false; //Should geometry vertex Ms be honored, if set?
bool hasIDs = false; //Should geometry vertex IDs be honored, if set?
bool hasNormals = false; //Should geometry vertex lighting normal vectors be honored, if set?

//Initialize General MultiPatch Creator

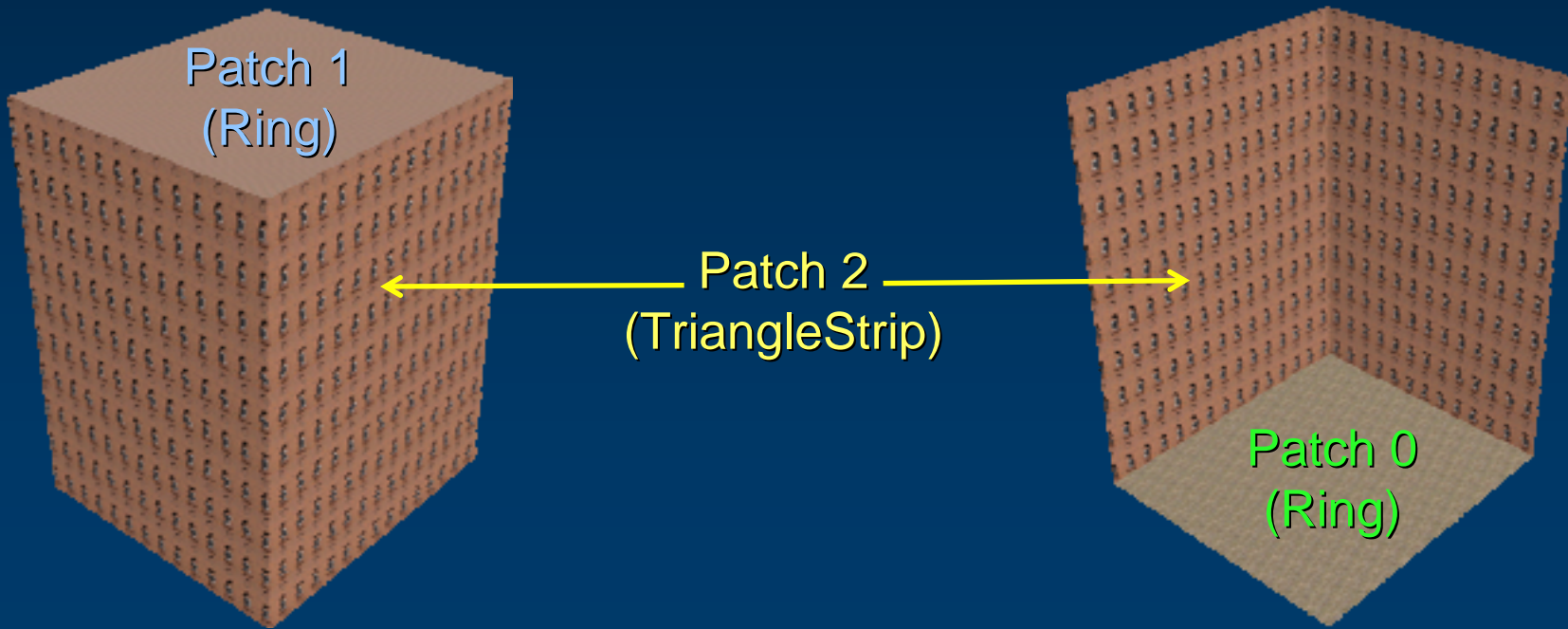
IGeneralMultiPatchCreator generalMultiPatchCreator = new GeneralMultiPatchCreatorClass();

generalMultiPatchCreator.Init(pointCount, partCount, hasMs, hasIDs, hasNormals, texturePointCount, geometryMaterialList);
```



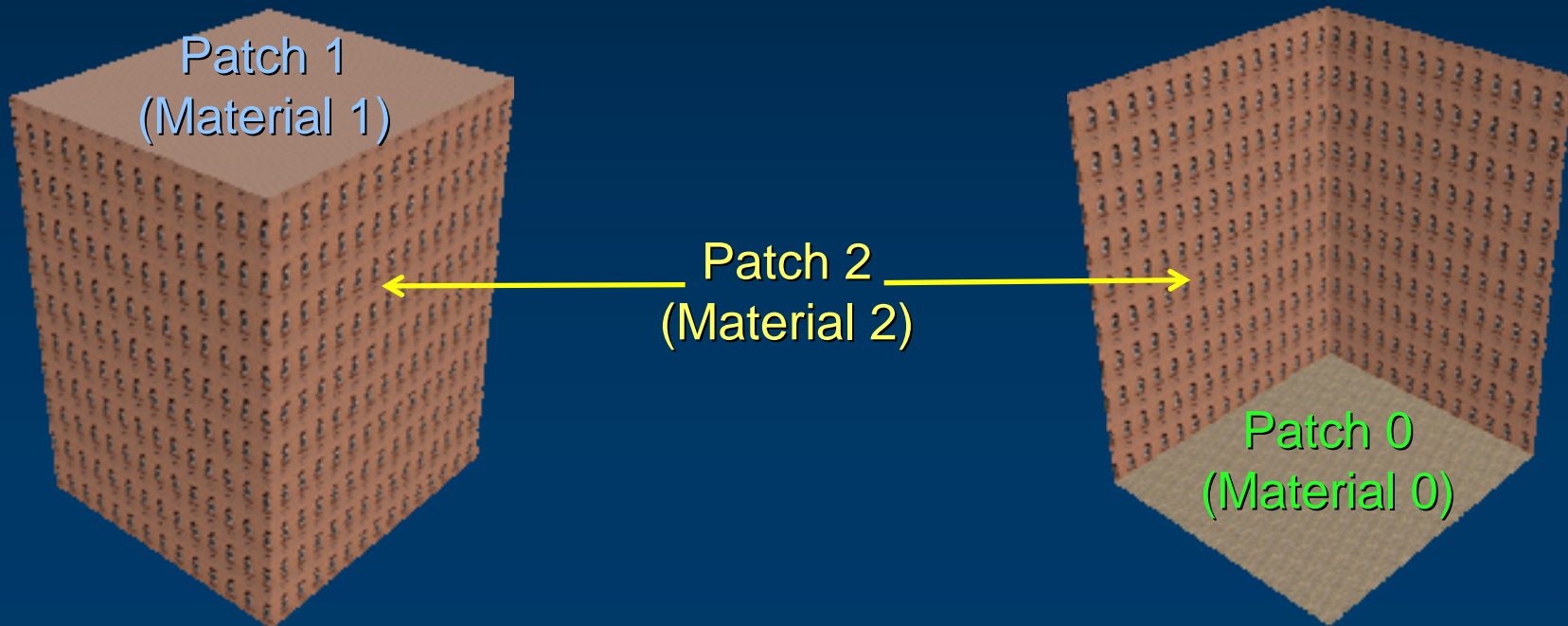
3. Define Patch Types

```
//Define Patch Types  
  
int patchIndex;  
  
generalMultiPatchCreator.SetPatchType(patchIndex = 0, esriPatchType.esriPatchTypeRing);  
generalMultiPatchCreator.SetPatchType(patchIndex = 1, esriPatchType.esriPatchTypeRing);  
generalMultiPatchCreator.SetPatchType(patchIndex = 2, esriPatchType.esriPatchTypeTriangleStrip);
```



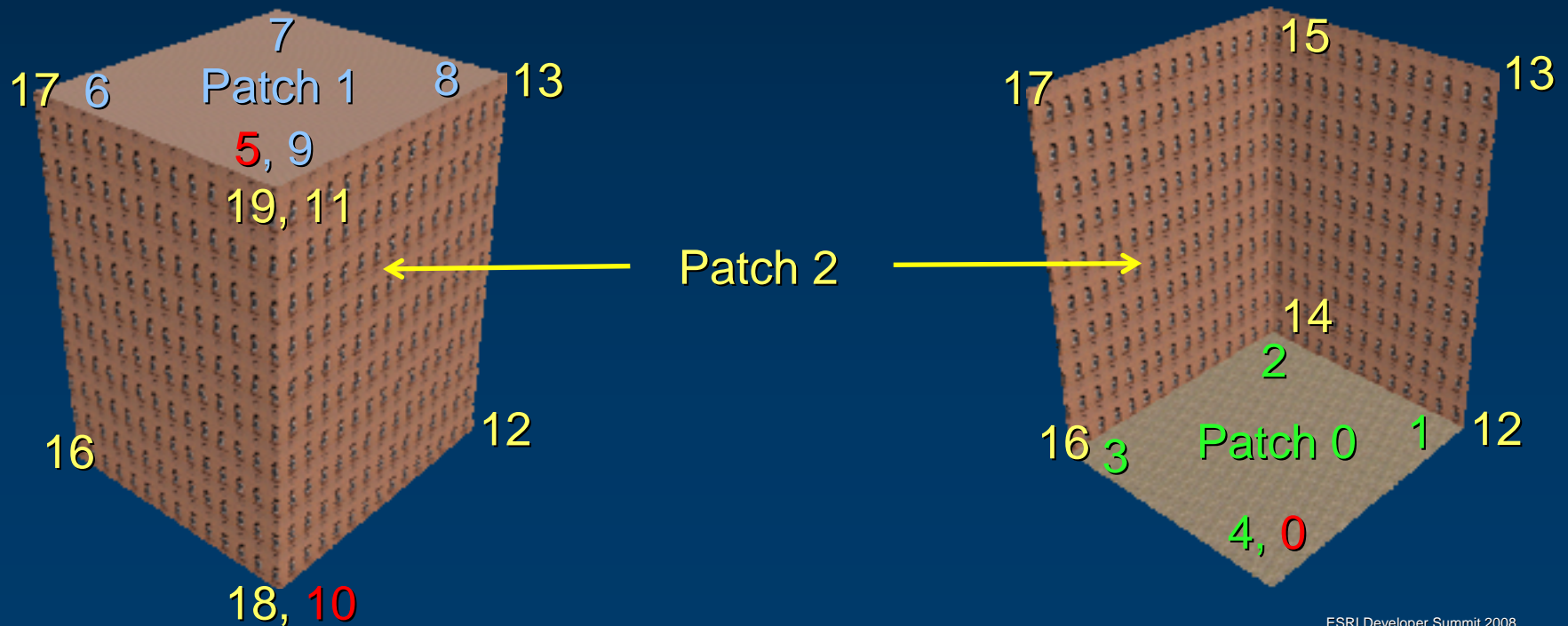
4. Define Material Indices

```
int patchIndex, materialIndex;  
  
//Define Material Indices  
  
generalMultiPatchCreator.SetMaterialIndex(patchIndex = 0, materialIndex = 0);  
generalMultiPatchCreator.SetMaterialIndex(patchIndex = 1, materialIndex = 1);  
generalMultiPatchCreator.SetMaterialIndex(patchIndex = 2, materialIndex = 2);
```



5-6. Define Point & Texture Point Starting Indices

```
int patchIndex, patchPointIndex, patchTexturePointIndex;  
  
//Define Patch Point Starting Indices  
  
generalMultiPatchCreator.SetPatchPointIndex(patchIndex = 0, patchPointIndex = 0);  
generalMultiPatchCreator.SetPatchPointIndex(patchIndex = 1, patchPointIndex = 0 + 5);  
generalMultiPatchCreator.SetPatchPointIndex(patchIndex = 2, patchPointIndex = 0 + 5 + 5);  
  
//Define Patch Texture Point Starting Indices  
  
generalMultiPatchCreator.SetPatchTexturePointIndex(patchIndex = 0, patchTexturePointIndex = 0);  
generalMultiPatchCreator.SetPatchTexturePointIndex(patchIndex = 1, patchTexturePointIndex = 0 + 5);  
generalMultiPatchCreator.SetPatchTexturePointIndex(patchIndex = 2, patchTexturePointIndex = 0 + 5 + 5);
```



7. Define Points

```

int patchPointIndex;
IPoint point;

//Define Patch Points
//Floor

point = new PointClass(); point.PutCoords(299918.004, 58577.206); point.Z = 0;
generalMultiPatchCreator.SetPoint(patchPointIndex = 0, point);

point = new PointClass(); point.PutCoords(299958.021, 58665.284); point.Z = 0;
generalMultiPatchCreator.SetPoint(patchPointIndex = 1, point);

...

point = new PointClass(); point.PutCoords(299918.004, 58577.206); point.Z = 0;
generalMultiPatchCreator.SetPoint(patchPointIndex = 4, point);

//Roof

point = new PointClass(); point.PutCoords(299918.004, 58577.206); point.Z = 15;
generalMultiPatchCreator.SetPoint(patchPointIndex = 5, point);

point = new PointClass(); point.PutCoords(299958.021, 58665.284); point.Z = 15;
generalMultiPatchCreator.SetPoint(patchPointIndex = 6, point);

...

point = new PointClass(); point.PutCoords(299918.004, 58577.206); point.Z = 15;
generalMultiPatchCreator.SetPoint(patchPointIndex = 9, point);

//Wall

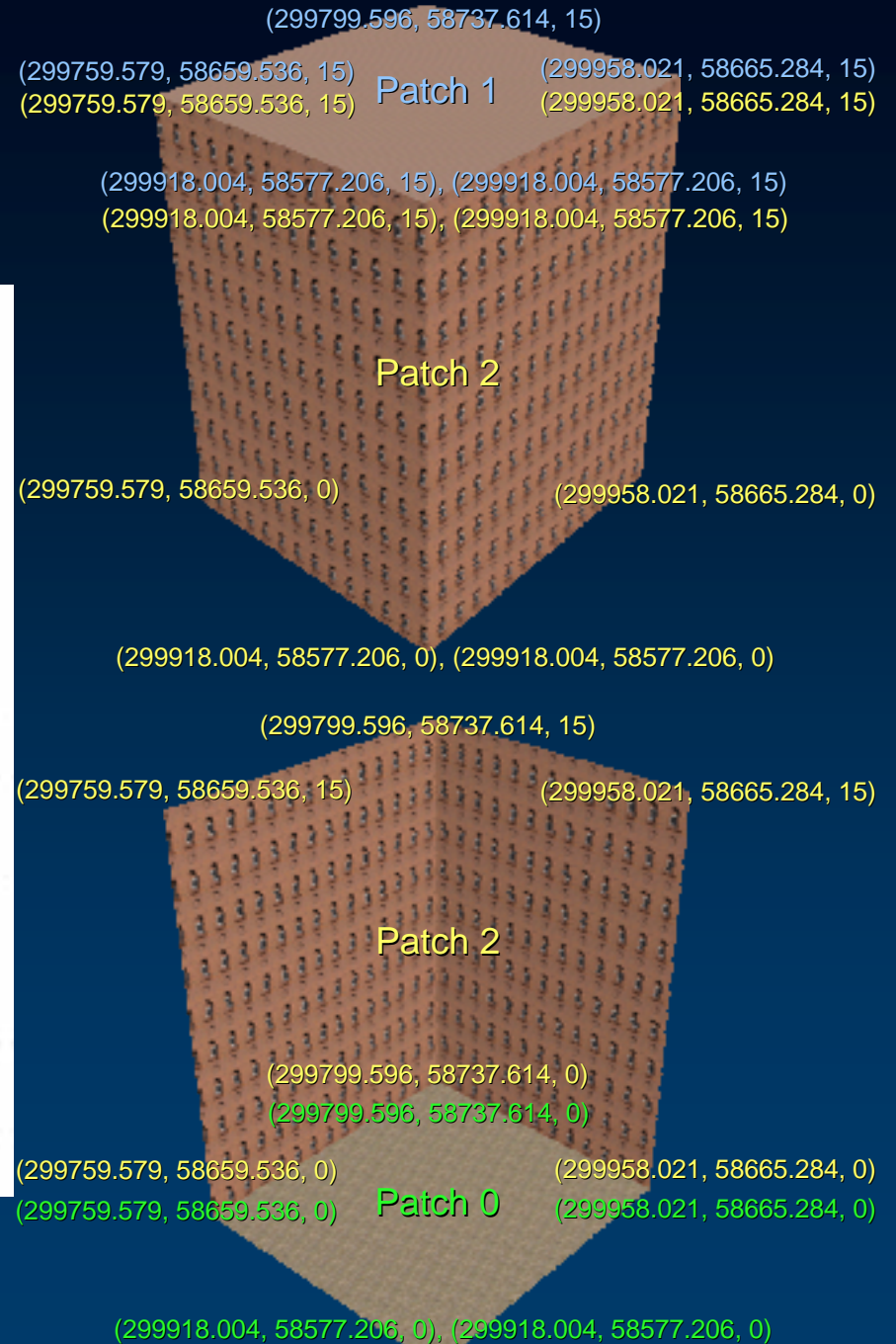
point = new PointClass(); point.PutCoords(299918.004, 58577.206); point.Z = 0;
generalMultiPatchCreator.SetPoint(patchPointIndex = 10, point);

point = new PointClass(); point.PutCoords(299918.004, 58577.206); point.Z = 15;
generalMultiPatchCreator.SetPoint(patchPointIndex = 11, point);

...

point = new PointClass(); point.PutCoords(299918.004, 58577.206); point.Z = 15;
generalMultiPatchCreator.SetPoint(patchPointIndex = 15, point);

```



8. Define Texture Points

```
int patchTexturePointIndex;
IPoint point;

//Define Patch Texture Points
//Floor
point = new PointClass(); point.PutCoords(78.238, 108);
generalMultiPatchCreator.SetTexturePoint(patchTexturePointIndex = 0, point);

point = new PointClass(); point.PutCoords(98, 48.698);
generalMultiPatchCreator.SetTexturePoint(patchTexturePointIndex = 1, point);
...

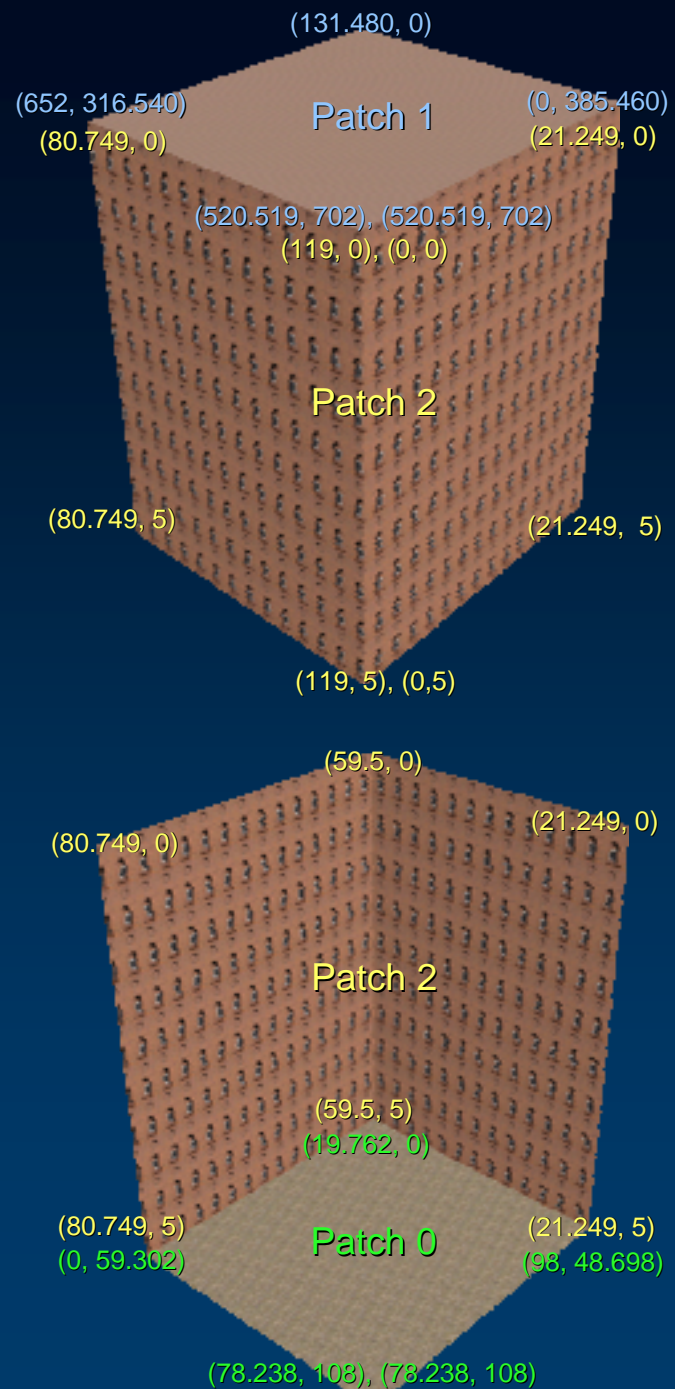
point = new PointClass(); point.PutCoords(78.238, 108);
generalMultiPatchCreator.SetTexturePoint(patchTexturePointIndex = 4, point);
//Roof
point = new PointClass(); point.PutCoords(520.519, 702);
generalMultiPatchCreator.SetTexturePoint(patchTexturePointIndex = 5, point);

point = new PointClass(); point.PutCoords(0, 385.460);
generalMultiPatchCreator.SetTexturePoint(patchTexturePointIndex = 6, point);
...

point = new PointClass(); point.PutCoords(520.519, 702);
generalMultiPatchCreator.SetTexturePoint(patchTexturePointIndex = 9, point);
//Wall
point = new PointClass(); point.PutCoords(0, 5);
generalMultiPatchCreator.SetTexturePoint(patchTexturePointIndex = 10, point);

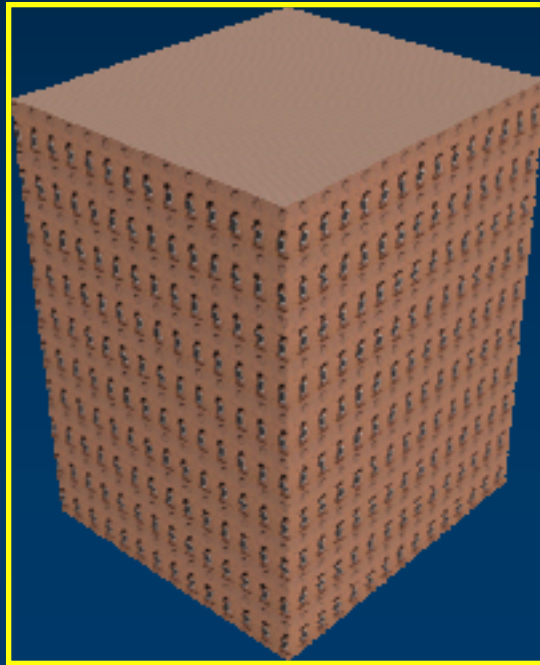
point = new PointClass(); point.PutCoords(0, 0);
generalMultiPatchCreator.SetTexturePoint(patchTexturePointIndex = 11, point);
...

point = new PointClass(); point.PutCoords(119, 0);
generalMultiPatchCreator.SetTexturePoint(patchTexturePointIndex = 19, point);
```



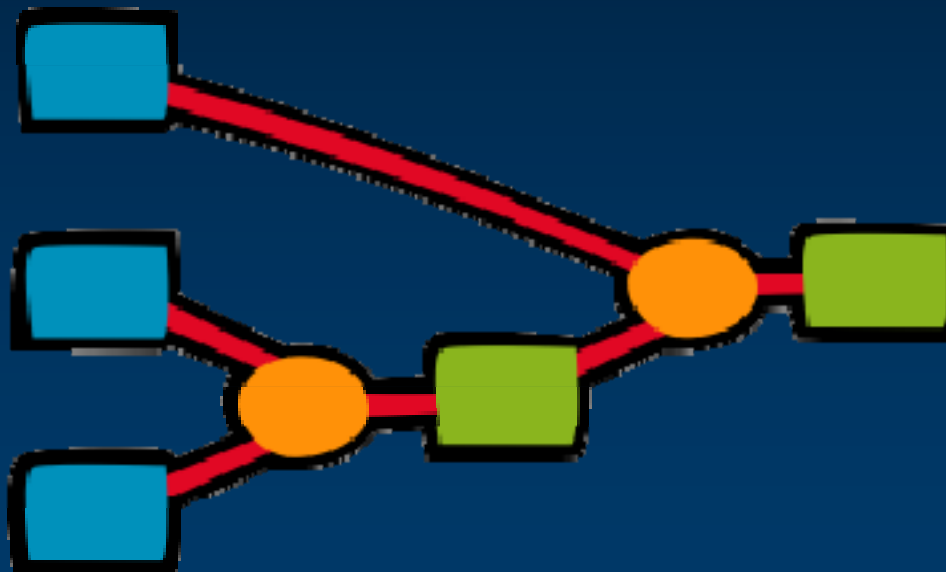
9-10. Generate MultiPatch & Cleanup Resources

```
//Generate MultiPatch Geometry  
IGeometry geometry = generalMultiPatchCreator.CreateMultiPatch();  
  
//Clean Up Resources  
generalMultiPatchCreator.ClearResources();
```



Entire City: Workflow

1. Construct Material Properties Table
2. Construct Building Properties Table
3. Construct Textured Buildings Feature Class



1. Construct Material Properties Table



Material
Properties
File



Material
Properties
Table

Attributes of MaterialPropertiesTB

OBJECTID	MaterialID	MaterialLocation	HasTextureImage	TextureImageFileName	TextureImageHeight	TextureImageWidth	TextureImageUnits
1	1	0	1	Floorred_brick_wall_4193171	4.89	6.67	3
2	2	0	1	Floordeity_cream_coloured_bricks_021782	6.22	7.67	3
3	3	0	1	Floordagonal_patterned_long_bricks_011708	6.89	8.67	3
4	4	1	1	Roofmarble_6110070	0.75	1	3
5	5	1	1	Roofspecked_marble_9261515	0.75	1	3
6	6	1	1	Roofwhite_marble_4131373	0.75	1	3
7	7	1	1	Roofrose_coloured_marble_8181131	0.75	1	3
8	8	2	1	Wallrepetitive_office_windows_23734	73.33	100	3
9	9	2	1	Wallsmall_square_hotel_windows_040521	40	53.33	3
10	10	2	1	Walloffice_windows_1987123	30	40	3
11	11	2	1	Wallwindows_011710	30	40	3
12	12	2	1	Wallanonymous_glass_metal_building_40519	50	50	3
13	13	2	1	Wallsquare_modern_windows_9291381	20.56	23.33	3
14	14	2	1	Wallsquare_traditional_office_windows_9261486	50	66.67	3
15	15	2	1	Wallmodern_windows_021791	26	33.33	3
16	16	2	1	Wallbalcony_deco_motif_4152396	20	15	3
17	17	2	1	Wallsmall_balconied_window_4152364	9	12	3
18	18	2	1	Wallwindow_ornate_brickwork_38923	11.25	15	3
19	19	2	1	Wallscandinavian_window_with_circle_904345	13.07	7	3
20	20	2	1	Walltraditional_window_23932	15	11.25	3

Record: 1 | Show: All Selected | Records (0 out of 20 Selected) | Options

2. Construct Building Properties Table



Building
Footprints
Feature Class



Material
Properties
Table



Building
Properties
Table
Generator



Building
Properties
Table

Attributes of BuildingPropertiesTB

OBJECTID*	BuildingID	Elevation	ExteriorRingID	FloorMaterialStyle	FloorMaterialID	RoofMaterialStyle	RoofMaterialID	WallMaterialStyle	WallMaterialIDString
1	1	14.9	0	2	3	2	4	7	10
2	2	14.1	0	2	2	2	6	7	18
3	3	38.71	0	2	3	2	4	7	14
4	4	53.47	0	2	2	2	5	7	19
5	4	50.69	0	2	1	2	6	7	12
6	4	74.75	0	2	2	2	7	7	8
7	5	27.88	0	2	1	2	4	7	20
8	6	16.67	0	2	3	2	4	7	9
9	7	56.04	0	2	1	2	5	7	10
10	8	19.7	0	2	2	2	6	7	17
11	9	30.99	0	2	3	2	6	7	9
12	10	14.92	0	2	2	2	6	7	15
13	11	58.31	0	2	2	2	5	7	16
14	11	69.49	0	2	3	2	6	7	13
15	12	19.2	0	2	1	2	7	7	15
16	13	13.47	0	2	2	2	7	7	10
17	14	15.02	0	2	1	2	6	7	19
18	15	38.36	0	2	1	2	4	7	13
19	16	39.91	0	2	3	2	7	7	8
20	17	39.31	0	2	1	2	6	7	9

Record: 1 Show: All Selected Records (0 out of 107133 Selected) Options

3. Construct Textured Buildings Feature Class



Building Footprints Feature Class



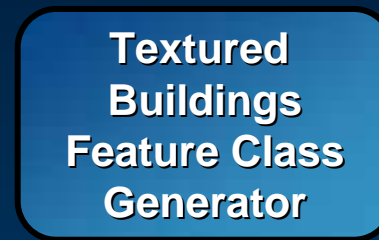
Texture Images



Material Properties Table



Building Properties Table



...

Entire City: Output



Custom Solution: Results

- 58,085 Textured Buildings
- 4.6 GB Feature Class
- Average Feature Size: 79 KB

- ArcObjects C# Solution
- Source Code Provided
- MultiPatch Whitepaper
 - > 150 Pages of Fully Illustrated Code Samples
 - Tips and Best Practices
 - In-Depth Walkthrough of Solution



Demo

Outline: Authoring and Serving the 3D View

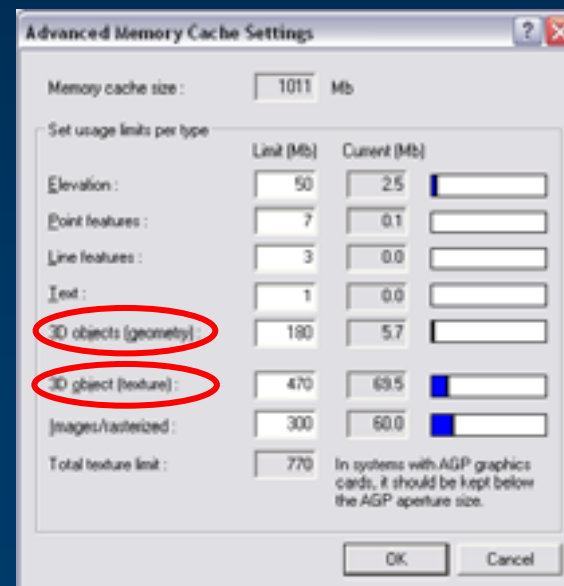
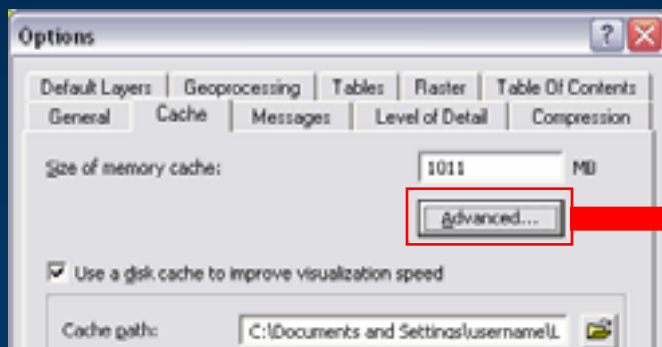
- **Authoring the 3D View**
 - **Globe Data Caching**
 - **Texture Management**
 - **Using Multiple Representations**
 - **Demo**
- **Publishing to ArcGIS Server**
 - **Demo**
- **Questions**

Globe Data Caching

- Mechanism for managing large amounts of data
- To improve display performance, ArcGlobe keeps a cache of tiles for each layer
- Two types of caches
 - **Memory Cache**
 - For immediate use
 - **Disk Cache**
 - Tiles are stored on disk for later use
- Tiles are swapped between disk cache and memory cache based on available resources and the current view

Memory Cache

- Refers to assigning an amount of the computer's RAM for use by ArcGlobe
- Memory can be assigned by data type
- Significant improvement in performance with proper settings



```
IGlobeAdvancedProperties2  
.GetTileMemory(type, limit, current);  
.SetTileMemory(type, limit);
```

Disk Cache

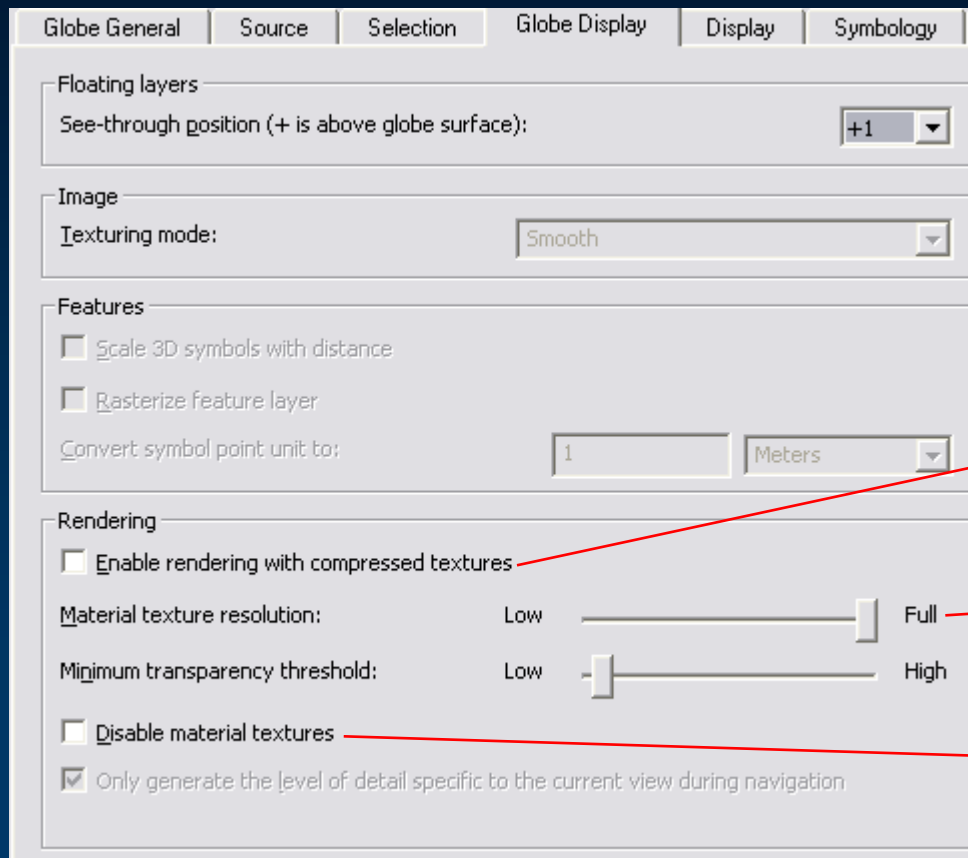
- Tiles are stored on disk for fast retrieval
- Layer disk cache is temporary unless the ArcGlobe document or a layer file is saved
- Two methods of generating the cache:
 - **On-demand**: tiles are created and stored as needed
 - **Explicit**: cache all areas
 - **Partial**: generate cache for specified level of detail
 - **Full**: all levels of detail
- Feature data in ArcGlobe has only one level of detail
- **Tip**: For best performance generate full caches for your 3D objects (buildings etc.)

Texture Management

- Textures on 3D objects can be memory intensive
- Data can take longer to display
- Navigation can be sluggish
- Performance degradation depends upon available physical memory as well as memory cache settings
- Three texture-management options are available:
 - Disable textures
 - Apply DXT compression
 - Downscale textures

Texture Management

• Layer Properties → GlobeDisplay



DXT compression

Texture downscaling

Disable textures

Using Multiple Representations

- Feature data in ArcGlobe has only one level of detail
- If you have large number of features then
 - Create multiple representations of your data
 - Use GP tools in ArcToolbox
 - Use texture management techniques
 - Use different visibility ranges

Far
(less detail)



Near
(more detail)



Using Multiple Representations

Example: Multiple representations for buildings

- Aggregated building footprints
- Individual building footprints
- Extruded footprints
- Buildings without textures
- Buildings with downscaled textures
- Buildings with full resolution textures

Coarsest

Finest

Using Multiple Representations

- For detailed information on which GP tools to use and how to optimize your ArcGlobe document please see:

<http://support.esri.com/index.cfm?fa=knowledgebase.documentation.viewDoc&PID=54&MetaID=1298>

Demo

Publishing to ArcGIS Server

- Publish the ArcGlobe document as a GlobeService
- Supports Local-area (LAN) and Web-based access
- Consume in ArcGIS Explorer, ArcGlobe, Globe Control
- All ArcGlobe supported data types can be served
- Supports Identify, Search and Find features
- Animation is also supported

Demo

Questions?

Thank You