COMPUTER IMAGE GENERATION

FLIGHT AND GROUND SIMULATION UPDATE - 2017

ASSUMPTIONS AND GOALS

- Problem:
  - CGI as used in simulation is a complex technology with many nuances and application specific variations
  - Much more than can be presented in four hours

- Assumptions:
  - Most people here are associated with an organization that purchases and/or operates CGI equipped simulators
  - Or are about to acquire new simulators
  - Few, if any, are about to develop their own image generators

- My goals
  - Allow you to better understand the CGI portion of your simulator
  - Discuss things that are generally available and/or achievable with today's technology and products
  - Explain some of the options available, to include the Database
  - Comment on some possibilities for the future of CGI in simulation

INTRODUCTION

Agenda

- History, types and applications
- 3D Computer Graphics Basics
- Image Generators Basics
- Graphics Shaders
- Features, Functions, Capabilities
- Geo-specific Imagery
- Special Effects
- Hidden Talents

History and Applications

- and some definitions

COMPUTER GENERATED IMAGERY (CGI)

- Computer graphics are everywhere - TV, Movies, Games, etc.
  - We are concerned with 2D images calculated from the geometry of a 3D Synthetic Environment or Database
- Concentrate on simulation applications - primarily training
  - Based on an interactive and moving “eyepoint” from which each new perspective image is calculated in “real-time”
- Provide real-time operation that allows the “man in the loop” to experience “real-world” interaction with the environment
- Key issue - Can the IG provide sufficient content at a sufficient rate?

IMAGE GENERATOR EVOLUTION

1965 — 1990 — 2000 — 2020 — 20XX

Dedicated Hardware

• Custom hardware – designed only for real-time image generation, simulation applications

• ~1,000 polys/chan.
• ~ $1m/channel
• Deterministic update
• Customized for new simulation features
• Cycled every 3-5 years

-Night/Dusk systems, calligraphic lights
-User expectation of stable "real-time"

Graphics Workstation

• High performance graphics hardware designed for multiple applications – some not real-time (e.g. movies)
• ~ 5,000 polys/chan.
• ~ $100k/channel
• Variable frame rate
• Customizations for simulation "considered"
• Cycled every 1-2 years

- Volume of applications led to lower cost
- Harder to control due to variable frame time

Graphics Processors

• Very high performance GPUs added to PCs – designed primarily for gamers
• 1 to 2 GPUs/PC-IGs
• ~ 500,000 polys/chan.
• ~ $2k to $20k/channel
• Variable frame rate
• Never customized for simulation, no volume
• Cycles 6 - 8 months

- Very wide range of applications due to <$
- High performance expectations

Grid Processing

• Scalable performance GPUs added to Servers – designed primarily for distributed computing.
• 5 to 50 GPUs/Servers
• "Limitless" polys/chan.
• ~ $30k/Servers
• latency & bandwidth issues
•customized for gaming
Cycles 8 -12 months

- "Pay-to- Play"
- **IG in the Cloud**
- low performance expectations (initially)

IMAGE GENERATOR EVOLUTION

1965 1990 2000 2020 20XX

Training Applications

- Computer Generated Imagery

CGI TRAINING APPLICATIONS

- Full mission simulators – WST, FMT (aircraft)
  - Train all aspects of vehicle operations (aircraft, tank, ship, etc.)
  - Large field of view – multi channel (getting larger with PCs)
  - Strict Database correlation requirements for all Databases, sensors
  - Cockpit displays, HUD

CGI TRAINING APPLICATIONS

- Full mission simulators – WST, FMT (Ship Handling)
  - Train all aspects of vehicle operations
  - Large field of view – multi channel
  - Strict Database correlation requirements for all Databases, sensors
  - ECDIS, Radar , and Map Display

CGI TRAINING APPLICATIONS

- Full mission simulators (UAV) – Something new (the future?)
  - Simulator image and display very similar to “real-world”
  - Simulating aircraft video and sensor images – no “out the window”
  - Very detailed Database requirements – high magnification

CGI TRAINING APPLICATIONS

- Part Task Trainer (PTT) – Operational Flight Trainers (OFT)
  - Selected or specific operations - landings, refueling, driver trainer
  - Less of the same type hardware – smaller field of view, fewer channels
  - Smaller Databases, may not include sensor channels

CGI TRAINING APPLICATIONS

- Combined or team trainers - Distributed Mission Operations
  - May be lower fidelity simulators - not training basic skills
  - Complex visual systems – Correlation between different platforms
    - Fair fight in “common” environment

CGI TRAINING APPLICATIONS

- Ground vehicles - cars, trucks, armored vehicles, railroads
  - Complex simulation due to ground friction, short transport delay
  - Visual resolution and field of view important – see into turns, read signs
  - Rearview mirrors create some pesky problems

CGI TRAINING APPLICATIONS

- Medical applications – Many CGI simulations
  - Some similar to "Flight and Ground" applications

CGI TRAINING APPLICATIONS

- Desk Top Trainers
- Low Cost Individual or Group Training

CGI TRAINING APPLICATIONS

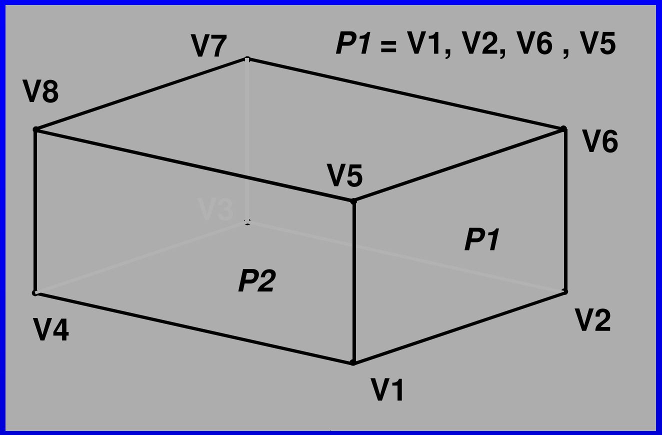
- Head Mounted Wearables
- High Fidelity Immersive Individual or Group Training

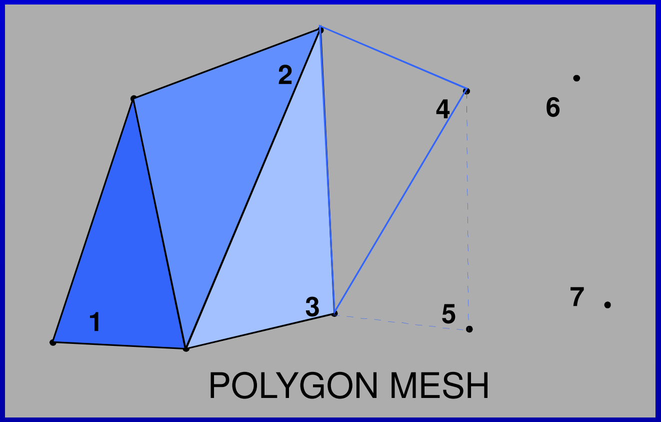
3D Computer Graphics

- Some basics definitions

3D COMPUTER GRAPHICS BASICS

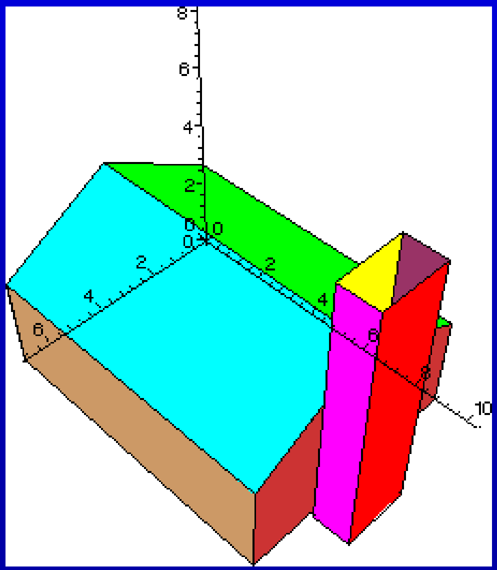
- Vertices
  - X,Y,Z points, no volume
  - All locations defined by vertices
    - Polygons, light points, separating planes, model routes...
- Polygons
  - Planar, convex, single sided, 2D
  - Three or more vertices
    - System may use triangles
  - Order of vertices important
  - Polygons may be “meshed” or T- stripped for efficiency
    - Same color, texture, shading...
- Objects
  - Groups of polygons, lights collected for management purposes
  - Details depend on architecture

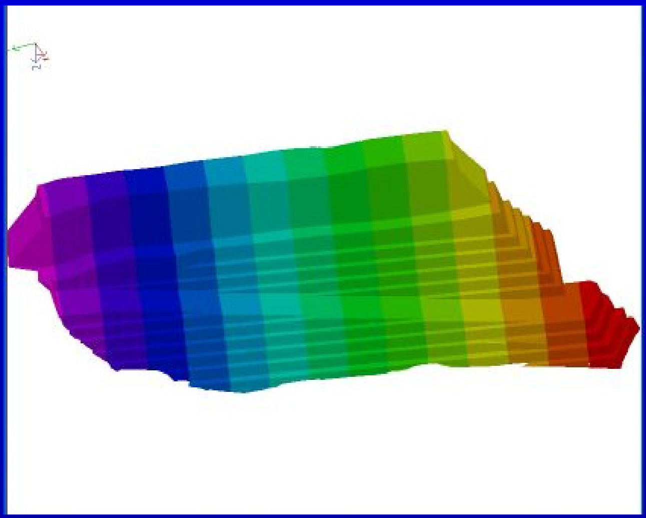




3D COMPUTER GRAPHICS BASICS

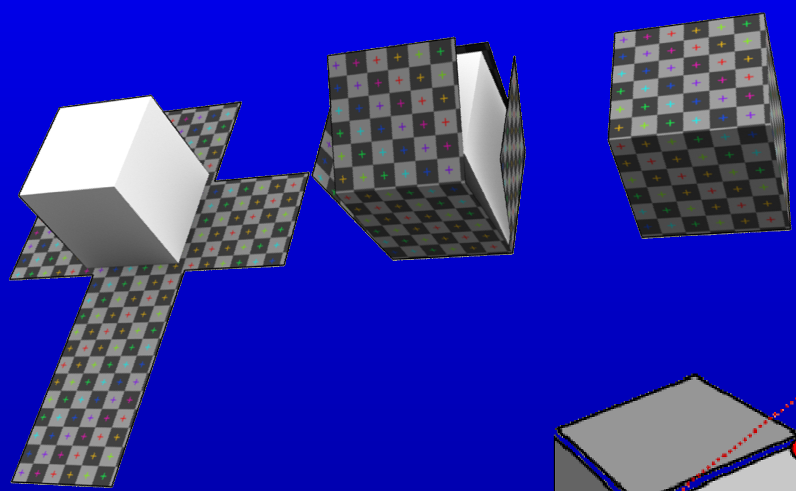
- Color and/or intensity
  - Stored in table(s)
  - Part of polygon definition
- Red, green, blue color components
  - 256 possible values each (8 bits), 16 million combinations
  - 256 location table normally sufficient
    - Sensor applications may require more

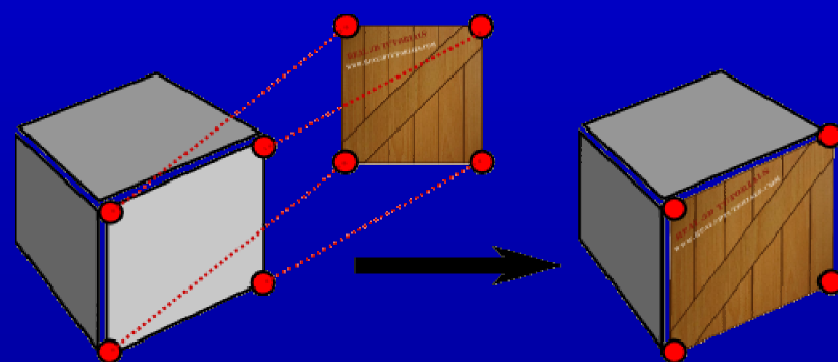




3D COMPUTER GRAPHICS BASICS

- Texture mapping is a way of adding detail to 3D model using a bitmap or raster image by “mapping” the “Texture” to the surface of a object or polygon. Every vertex in a polygon is assigned a texture coordinate (known as UV coordinate) either via explicit assignment or by procedural definition.





Three Types of Graphics “Shaders”

Shaders are small programs that run on the GPU (Graphics Processing Unit) which is normally found on the Graphics Card. The GPU has multiple cores that run the shaders in parallel. This make for much higher throughputs then when these calculation are done on the CPU. Therefore allowing for more continent to be displayed and in turn creating more vibrant scenes.

1. Vertex Shader

    Transform vertices of existing 3D objects.

2. Geometry Shader

    Creates new geometry from existing 3D objects.

3. Pixel Shader

    Manipulate the pixels (texture) applied to existing 3D objects.

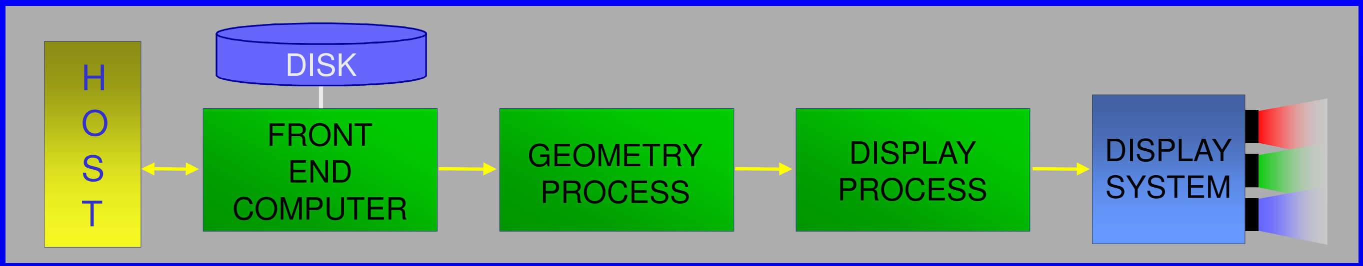
NOTICE ALL SHADERS REQUER SOME GEOMETRY TO EXIST FIRST

“Shaders” RUN-TIME CUSTOMIZABLE

“Shaders” RUN-TIME CUSTOMIZABLE

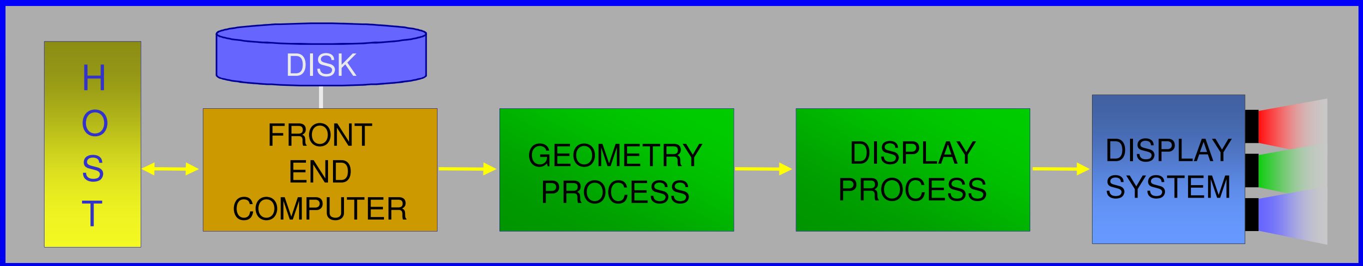
Image Generators (IG)
- Some of the basics

BASIC IMAGE GENERATOR FUNCTIONS



- Similarities in all 3D graphics systems or devices
- Major components and functions - may be hardware and/or software
  - Database - mathematical definition of environment (gaming area)
    - Functions of real-time software and Database called a scenegraph
  - Front end computer
  - Geometry process
  - Display process
- IG calculates a <u>unique</u> image for each position input
  - Speed usually not a factor in resulting image – no motion blur
    - At 1,000 knots eyepoint moves ~28 feet in 1/60th of a second

FRONT END COMPUTER



- Host interface
  - One way originally, now all two way
  - Ethernet, DMA, reflective memory
  - Transfers position and control data from host
  - Common Image Generator Interface (CIGI) - developed by Boeing
- Software:
  - Operating system - unix, linux, Windows
  - Graphics system - OpenGL, DirectX, proprietary formats
  - Real-time application - vendor specific
- Database management and model/feature selection
- System management - overload, special effects, coordination
- Mass storage - Only a problem with large geo-specific imagery Databases

REAL-TIME

- What is "real-time"?
  - "Natural" response to visual cue - “just like in the airplane”
    - Time between input and response referred to as <u>transport delay</u>
    - Higher update rate major factor in shorter transport delay
  - No "objectionable" stepping due to large change in eyepoint location
  - <u>Stable</u> update rate also important – can’t be real-time <u>most</u> of the time
- Capacity and time tradeoff (again)
  - Image will flicker if display <u>refresh</u> is too slow
    - Eye perceives fall off of phosphor intensity
  - Image will step or jump if <u>update</u> is too slow
    - Distance between new images too great for your brain to “smooth”
- "Real-time" is very application sensitive
  - 60hz - fast movers (jets), cars and trucks (closely coupled to the ground)
  - 30hz - civil and military transport aircraft, helicopters (display dependent)
  - 15hz - armored vehicles (early implementations), maritime
- Less of an issue as graphics performance improves – “more then 60hz is faster then you can see it”

DEFINE UPDATE RATE AND REFRESH RATE

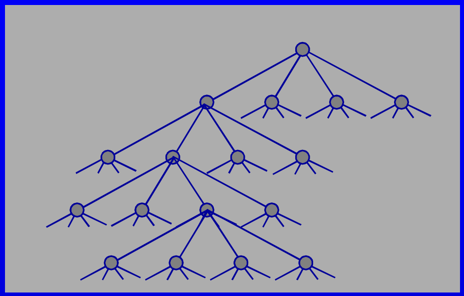
- Update is a new calculated image
  - Normally based on new eyepoint location
- Refresh is a new image drawn on the display
  - No lowwer then 30hz or it will be very noticeable
  - However, some older systems even ran a 15hz and were affective
- Ideally these should be the same – preferably min. at 60hz -120hz
- If update rate is less than refresh rate the image is be drawn more than once
  - Causes double imaging at high angular rates
  - Illusion due to seeing the same picture multiple times
  - Less of a problem with more powerful GBUs
    - Update and refresh rates more likely to mate (i.e. 60hz)
  - With head mounted display (HMD) angular rates may be caused by movement of the head rather than the vehicle

TRANSPORT DELAY

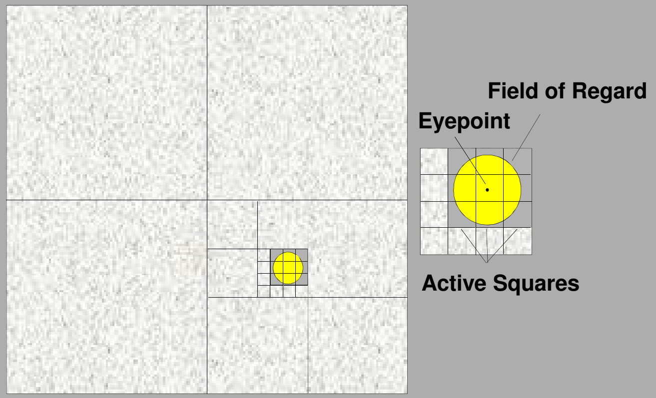
- Image generator delay only
  - Time from receipt of new position data - to display of resultant image
    - Normally measured in milliseconds
  - Theoretical value - dependent on system architecture
    - How much time each process takes – or is allowed
    - How efficiently data is passed from one process to the next
  - Affected by update rate (may be different between day and night)
- Total system delay - with simulator
  - Can be measured with special hardware and Databases
  - Depends on how data is sent to the IG – asynchronous vs. synchronous
  - Prediction algorithms sometimes used to reduce apparent delay
- "Acceptable" delay dependent on application – again
  - Similar to, and related to, update rate
  - Excessive delay is a suspected factor in simulator sickness
- GPU based systems tend to have less delay than historical IGs
  - Everything is normally done within a single processing frame

DATABASE MANAGEMENT

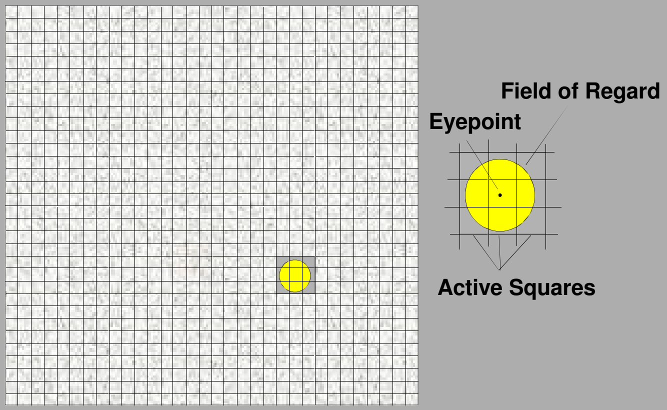
- Available Database (on the disk) or Active Database (in the IG)
  - Required when geographic area is larger than the visibility range
- Hierarchical structure
  - Culls large portions of Database quickly
- "Geographic" approach
  - Compare eyepoint location to known grid structure



- Most systems retrieve the full 360˚ around the eye position

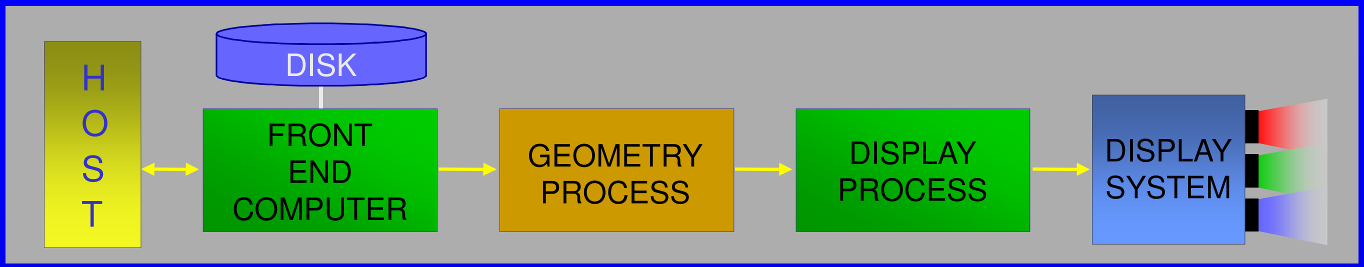


Heirarchical Quad-Tree Structure



Grid Management Structure

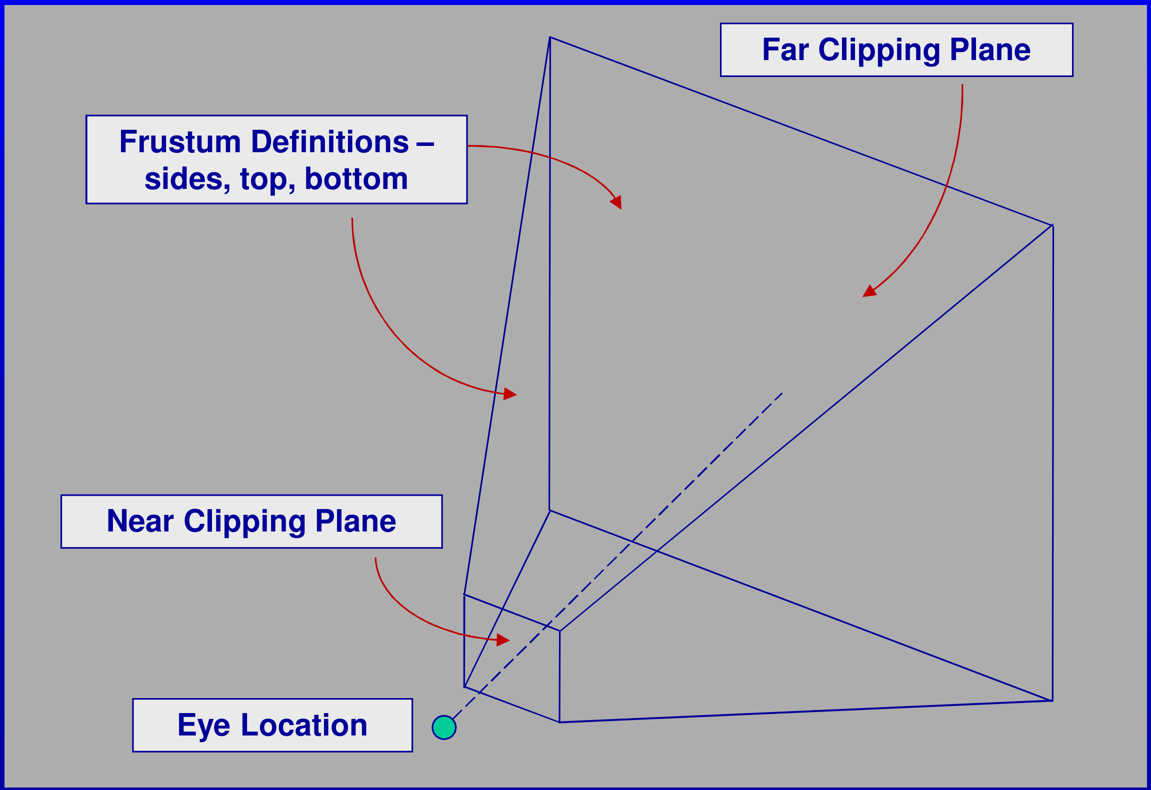
GEOMETRY PROCESS



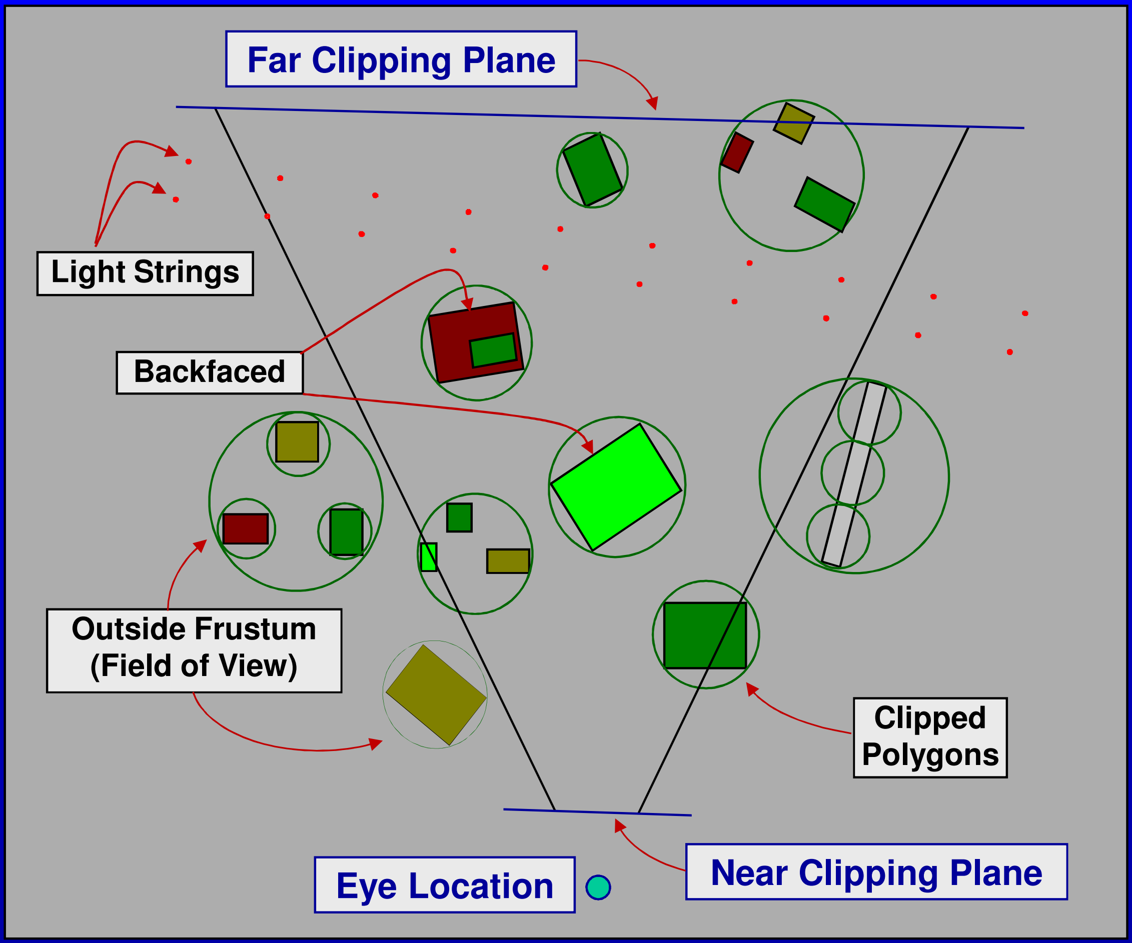
- Converts 3D Database into 2D perspective image based on position
  - Matrix calculations performed on each vertice
- Functions may be performed in software or hardware
  - Was a major discriminator in early PC graphics accelerators
- Culls unnecessary polygons and/or objects
  - Outside viewing frustum (field of view, near and far clipping planes),
  - Database structured to assist process
- Vertex and geometry shaders run here
- Expands light string data into individual light points
- May clip polygons and light strings that extend past edge of image
- Geometry culling efficiency a factor in system polygon/light capacity
- Sometimes referred to as transform, in transform and lighting (T&L)

CHANNEL FRUSTUM – FIELD OF VIEW

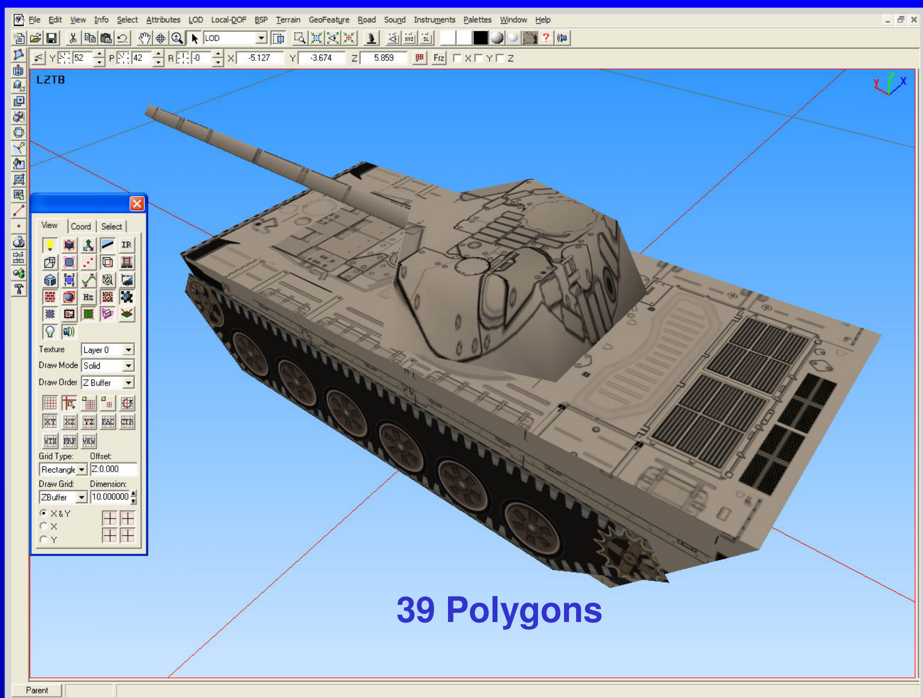
- Channel definition in IG must match Display System design
- Differences will cause distortions – may lead to sim-sickness, headaches
- Position of near clipping plane important to Z-buffer resolution
- Far clipping plane should be just outside visibility range

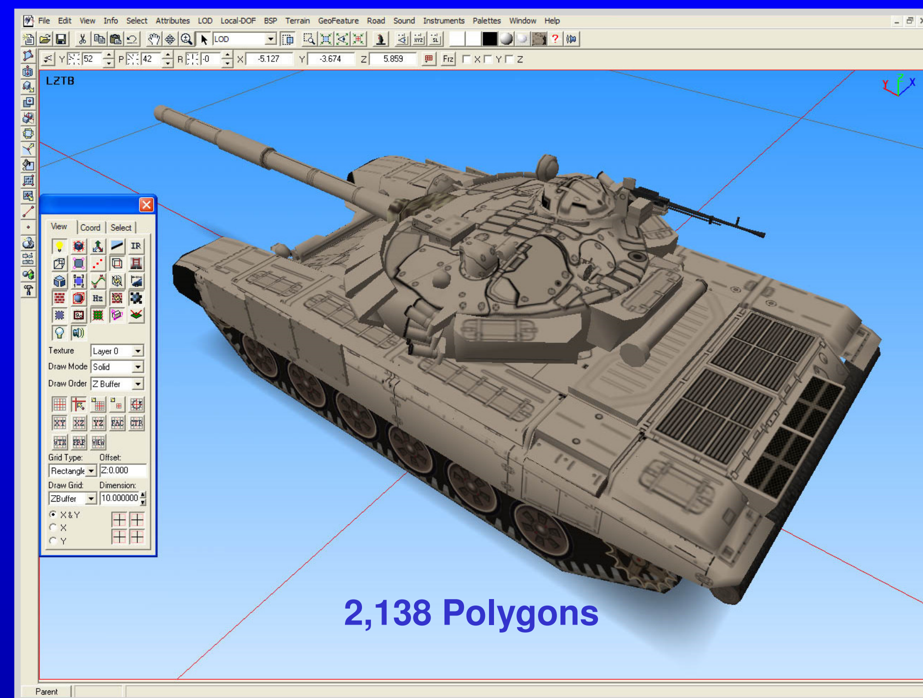


VISIBLE SURFACE DETERMINATION



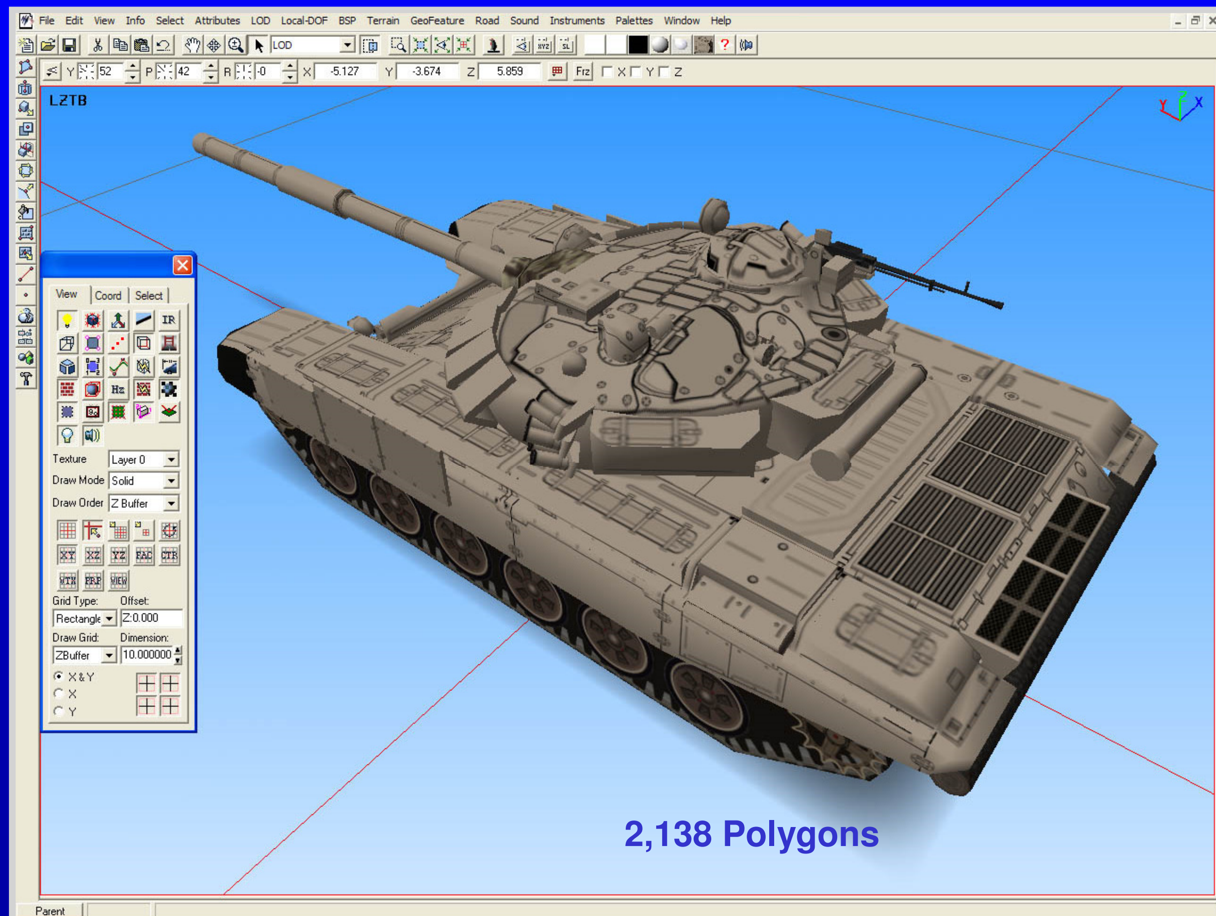
LEVEL OF DETAIL (LOD)





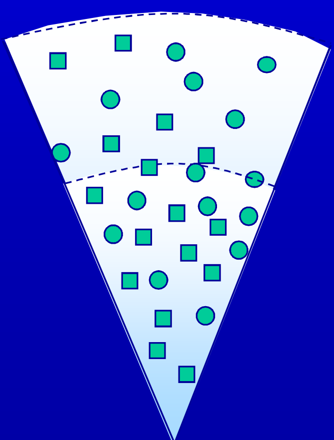
LEVEL OF DETAIL (LOD)

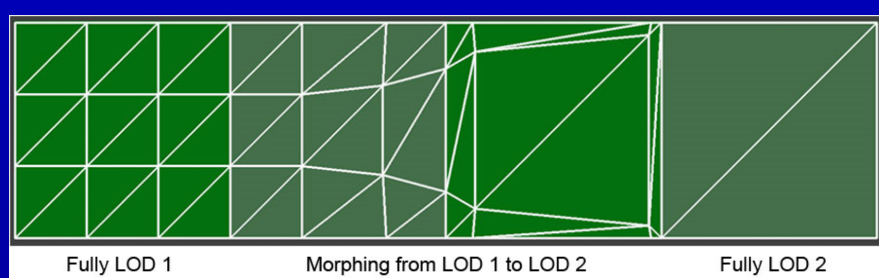
39 Polygons



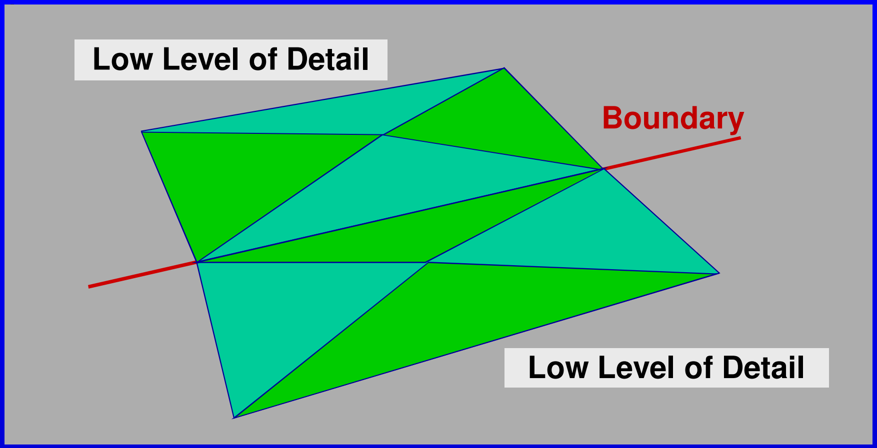
LEVEL OF DETAIL (LOD)

- Add detail as object approaches eye
  - The later the better - increases density
- Fading or blending
  - Uses transparency during swap
- Morphing, continuous LOD
- Enhance distant objects for detection
  - Size, color, contrast
- Different for terrain and feature models

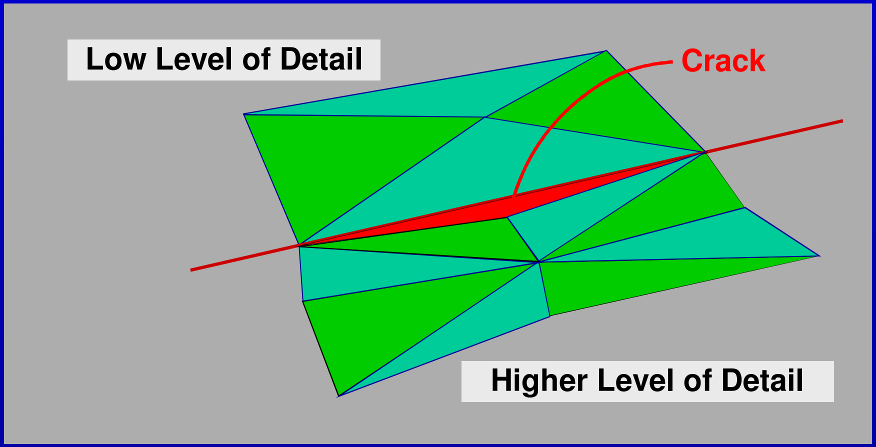




TERRAIN LEVEL OF DETAIL CRACKS



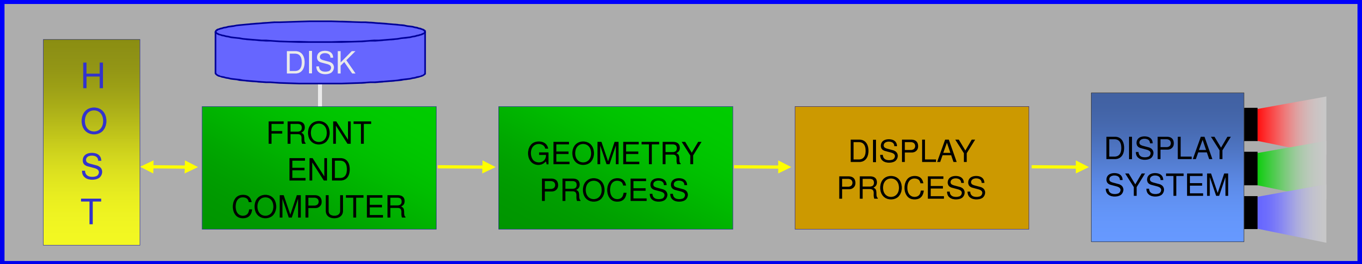
- Models generally change all at once
- Terrain changes tile by tile as eye approaches
- Can result in cracks at the boundaries
- Most terrain tools will handle the problem



SWITCH NODES

EX. DAMAGED, DESTROYED VERSIONS OF MODELS

DISPLAY PROCESS



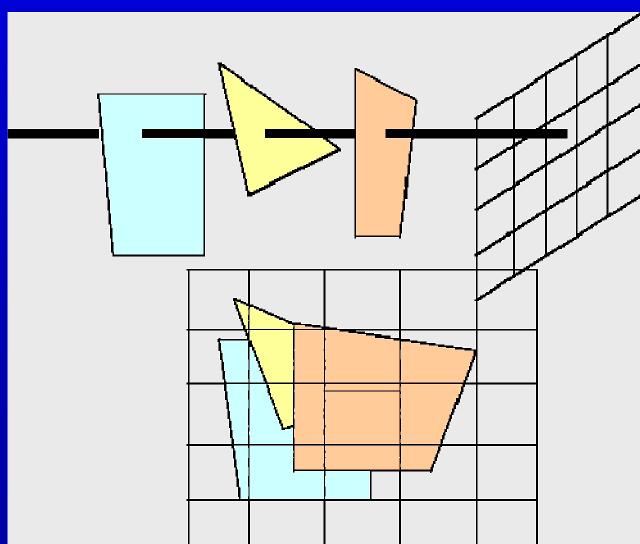
- Maps perspective polygons onto raster structure for current image
  - Many pixels must be filled more than once - depth complexity, pixel fill
  - Computes correct pixel color from multiple data inputs
  - Color calculated from: texture, shading, transparency, fog, anti-aliasing
    - Some of these effects make up a geo-state
  - Pixel/fragment shader effects run here
- Priority implementation in z-buffer systems (more later)
- Dual frame buffer most common approach
  - Assembles complete image in one half of buffer
  - Previous image from other half of buffer is sent to the display
  - Also called a swap buffer

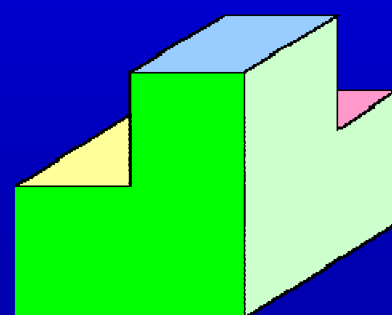
DISPLAY PROCESS

- System must buffer two each of:
  - red, green, and blue pixel components
  - alpha (transparency) component
  - z-depth information for each pixel
  - Some number of anti-aliasing versions
    - Depends on A-A implementation selected
- Available buffer memory shared with texture memory in current GPUs
  - A factor in determining available texture capacity
  - As available memory expands (up to 3 gig now) more can be used for texture and/or shader implementations
  - Buffer requirements only increase with more displayed pixels and/or better anti-aliasing alternatives

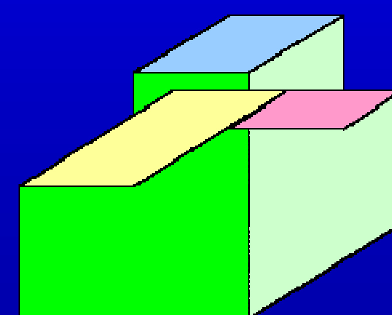
Hidden Surface Removal (HSR)

- Z-buffering or Depth buffet
  - The Z-depth is calculated at each pixel position (x,y)
  - All pixel depths initialize to 1
  - depth is less then 1 with 0 as the closest
  - If the z-value is less that the value in the depth buffer filled with the color
  - Once all the polygons have been processed the buffer contains closer faces correctly overlapping faces that are further away.





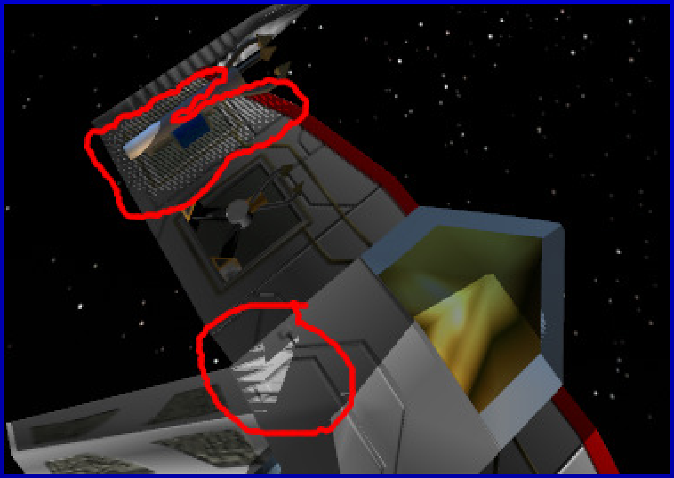
Z-buffered



Not Z-buffered

PRIORITY – Z-BUFFER APPROACH

- Compares distance to each pixel as it is sent to the display processor
  - Content/color of closest pixel is placed in z-buffer
  - More distant pixel is overwritten or discarded
- Requires mechanism to account for co-planar polygons
  - Resolution of z-buffer will determine how close two polygons can be
    - Effected by position of near and far clipping planes – especially near
  - "Z-fighting" at the pixel level results from insufficient resolution
    - Looks much like aliasing – flashing, flickering portions of polygons
- Problems with very close objects
  - Inside near z-clip (e.g. ownship parts)
  - Significant effect on z-buffer
- Issues with transparency, anti-aliasing
- Performance is improved if closer objects can be drawn first
  - Test and replace takes longer than test and discard

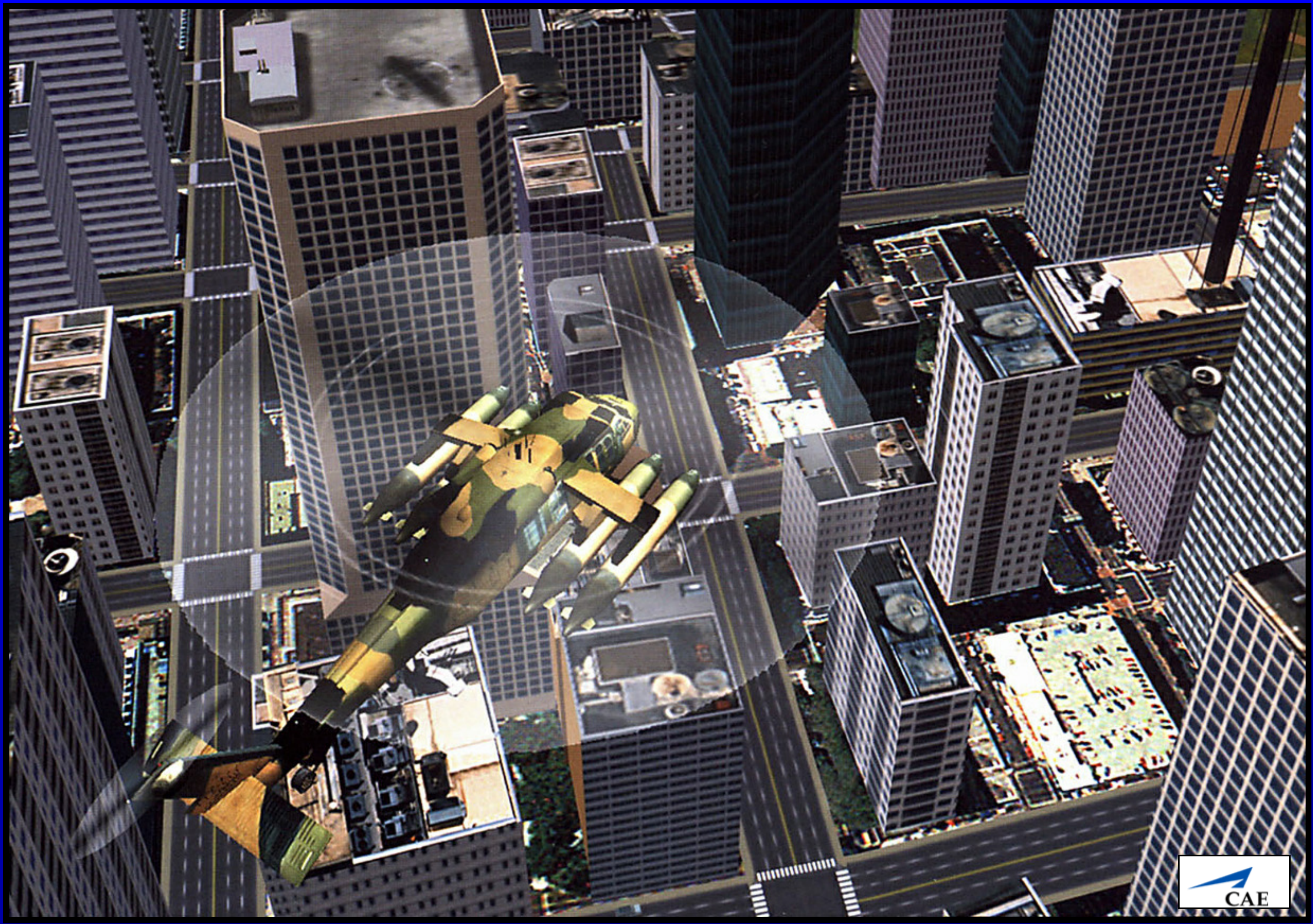


TRANSPARENCY - TRANSLUCENCY

- Alpha blending commonly used
  - Polygons include red, green, blue and alpha (r,g,b,α) values
  - Alpha value is a percentage of opacity scaled from 0 to 1
    - Pixel color blended with background based on percentage
- Polygons require sorting in z-buffer system
  - Original color may be lost otherwise
- Used for canopies, windows, props and special effects
- Also for level of detail changes
  - "Fade in" of new polygons much less distracting than "popping" objects in
  - Allows significant increase in scene density

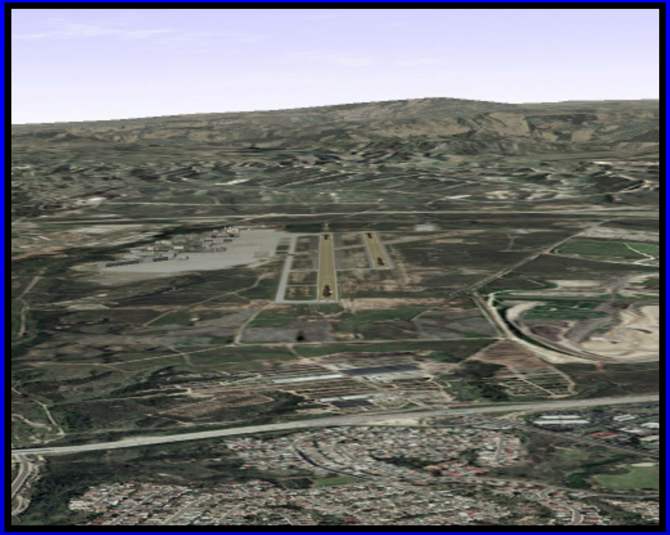
TRANSPARENCY EXAMPLES

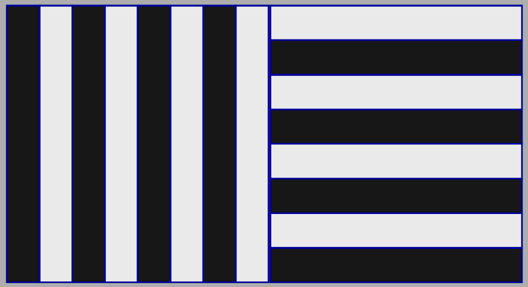
TRANSPARENCY EXAMPLES



RESOLUTION FACTORS

- Multiple display and optics factors
  - Set up, focus, design eyepoint....
  - Number of pixels in image
    - Function of display processor hardware and update rate
  - Degrees subtended per pixel
    - Dependent on field of view size
- Subjective resolution tests
  - What can be usably seen in the image
  - Improved by anti-aliasing
- Objective resolution tests
  - What can be measured
    - 10% modulation transfer function
  - Equally spaced bars, optical line pairs
    - "Defeats" anti-aliasing techniques



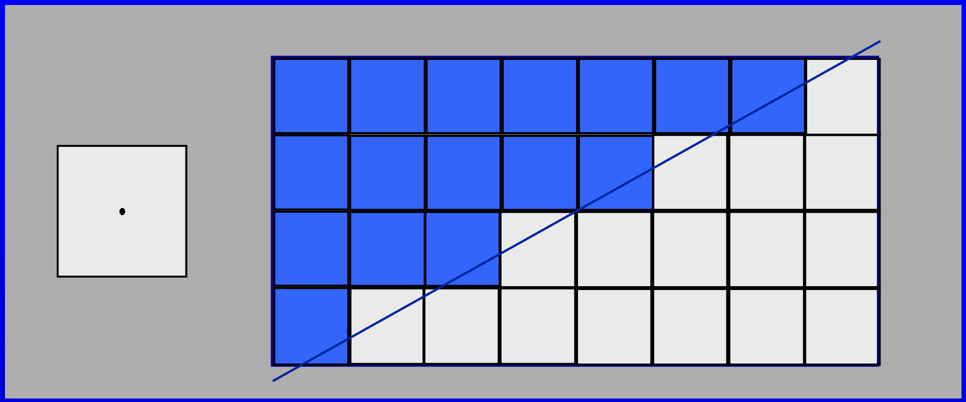


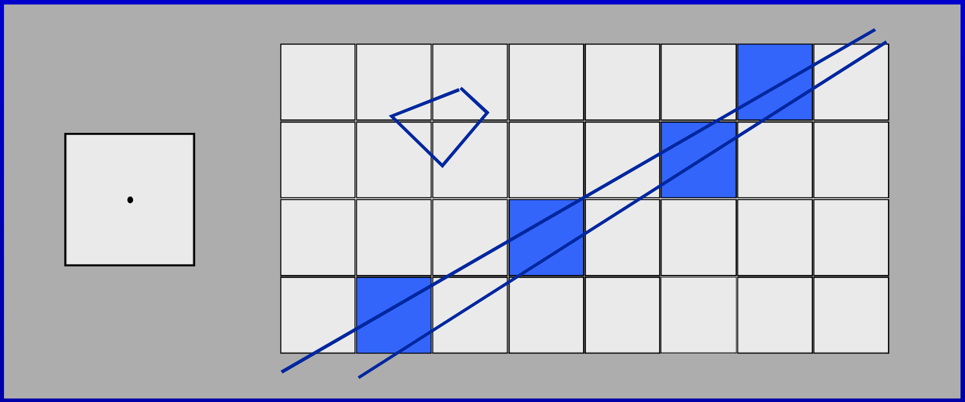
Optical line pair (OLP) test pattern

ANTI-ALIASING

- Aliasing due to pixelization of image – quantization
  - Second generation system very good at anti-aliasing
  - Early PC/GPU systems very weak – lack of gamer expectations
- Most anti-aliasing makes use of multiple samples per pixel
  - Pixel is sampled at multiple points (2, 4, 16…) - the results are averaged
    - Or the percentage of the pixel covered by each polygon is calculated
  - Sometimes by calculating full image multiple times - results averaged
    - Multiple PC graphics cards may be used - one per image
  - Traditional light point anti-aliasing handled differently than polygons
    - Most lights now are small polygons manipulated by vertex shaders
    - Control light point shape and behavior based on brightness, distance
      - Improved anti-aliasing, perspective growth
  - Calligraphic lights - draw technique and addressing resolution reduces or removes aliasing effects
- Significant impact on pixel throughput
  - e.g. Pixel output may be reduced up to 75% in four sample anti-aliasing

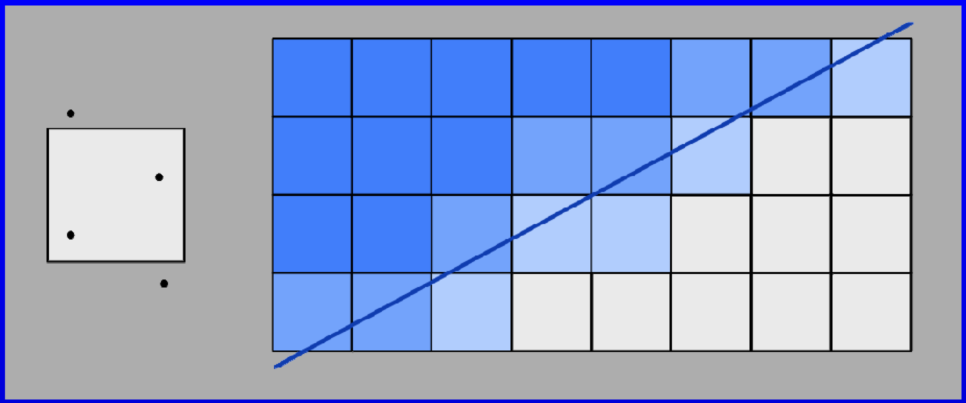
SINGLE SAMPLE ANTI-ALIASING



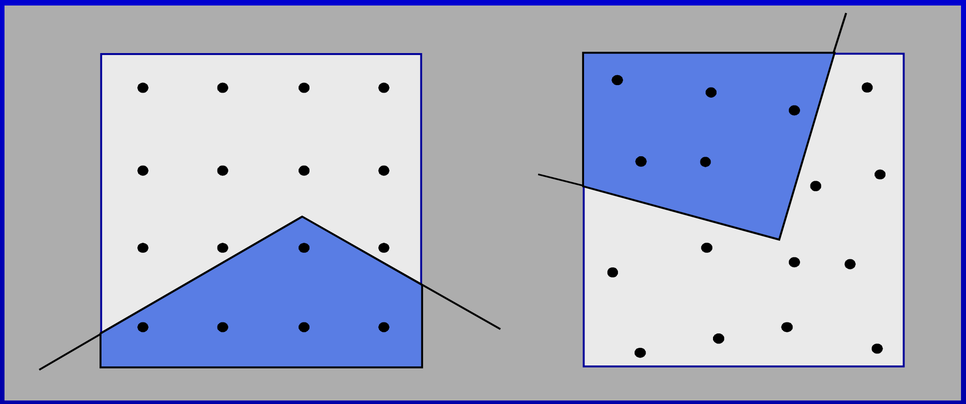


ALIASING EXAMPLES

MULTIPLE SAMPLE ANTI-ALIASING



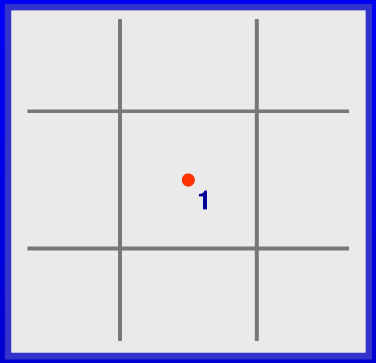
- Average four samples together
- Defocuses edge
- Up to 4x pixel processing



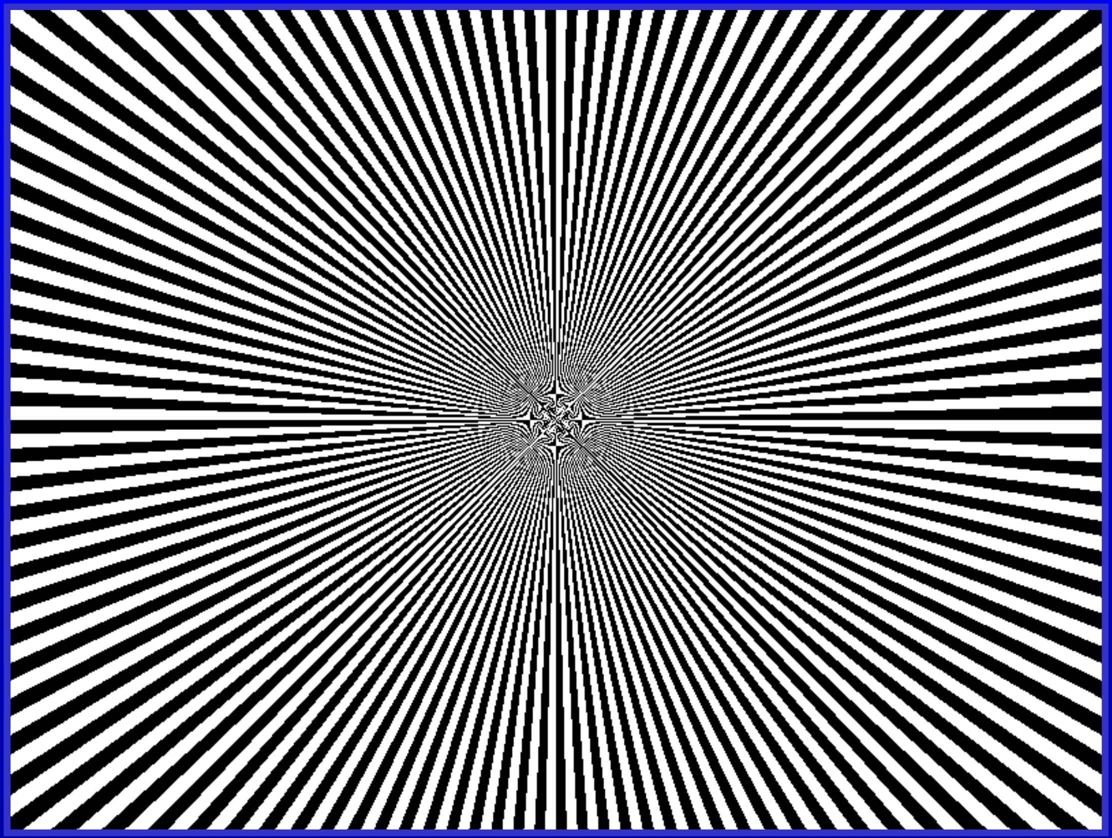
- Average 16 samples together
- Smoother blend
- Much more pixel processing

MULTIPLE SAMPLE ANTI-ALIASING

ONE SAMPLE, ONE TAP (AA TURNED OFF)

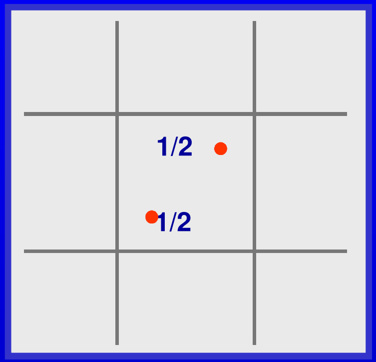


- Sample from pixel

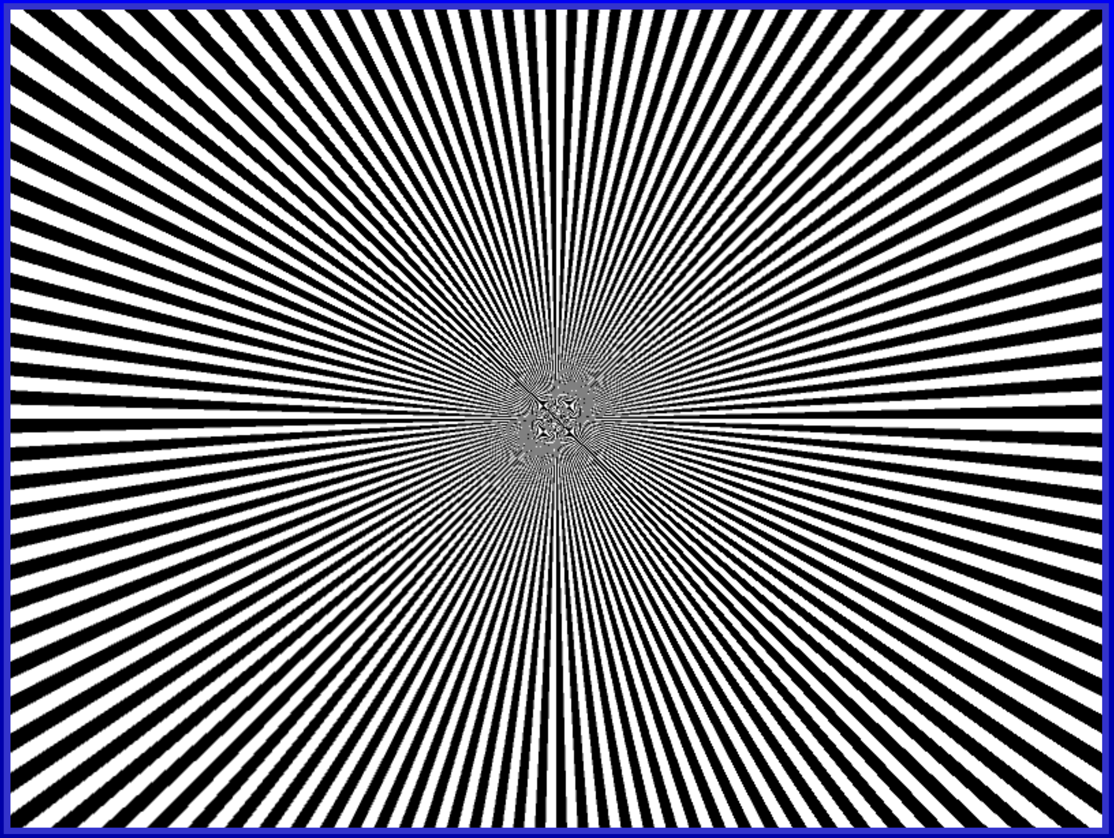


MULTIPLE SAMPLE ANTI-ALIASING

TWO SAMPLE, TWO TAP

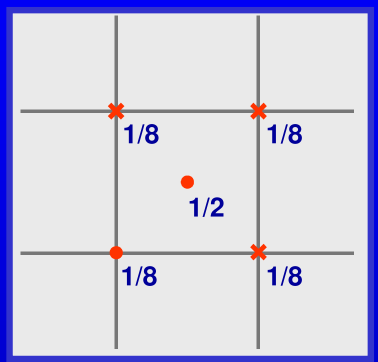


● Sample from pixel

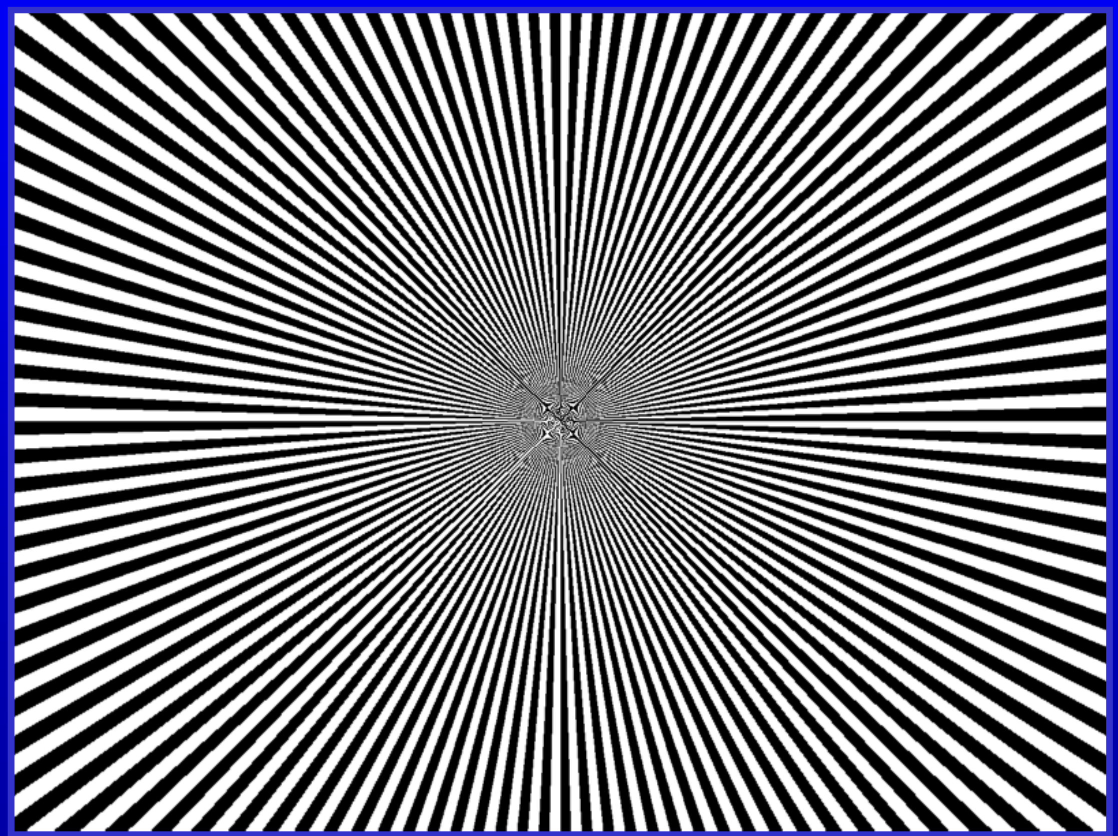


MULTIPLE SAMPLE ANTI-ALIASING

TWO SAMPLE, FIVE TAP (QUINCUNX)

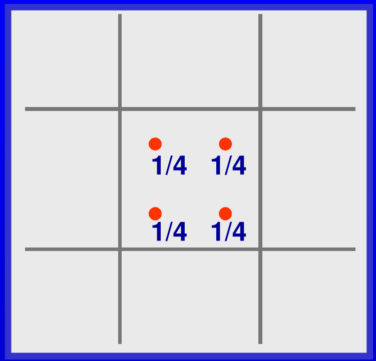


- ● Sample from pixel
- ✕ Sample from adjacent pixels

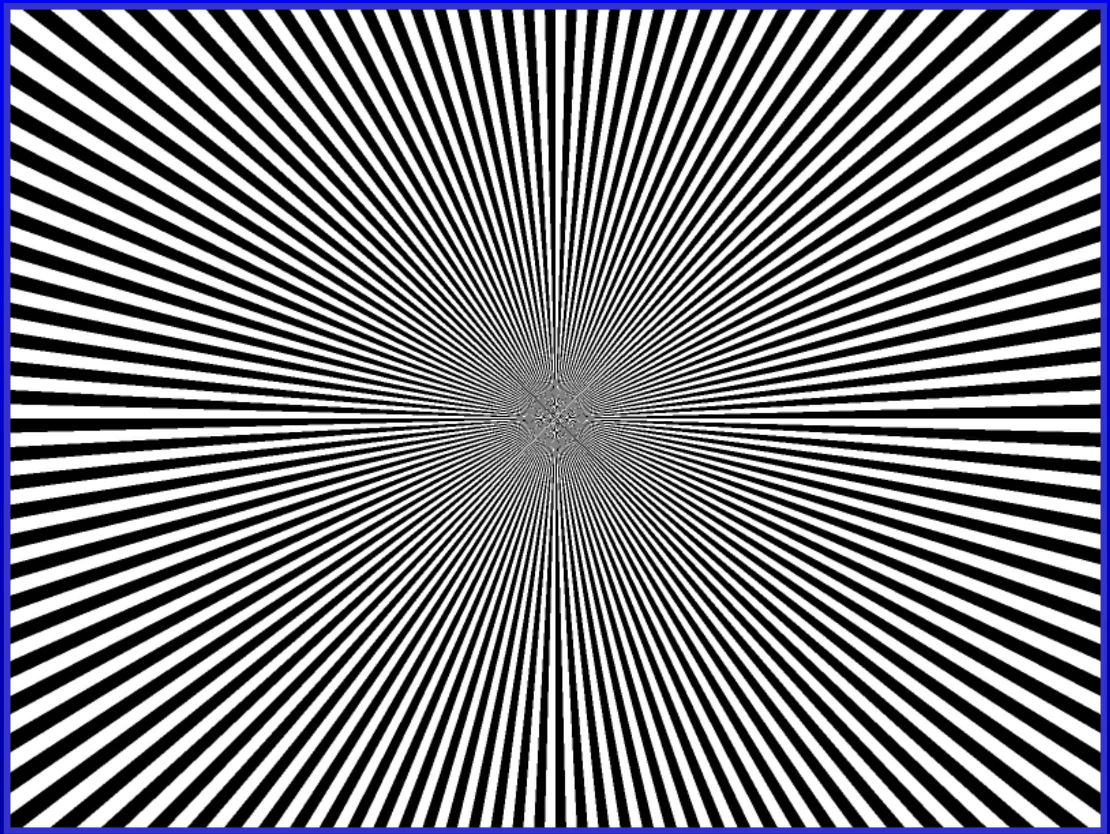


MULTIPLE SAMPLE ANTI-ALIASING

FOUR SAMPLE, FOUR TAP

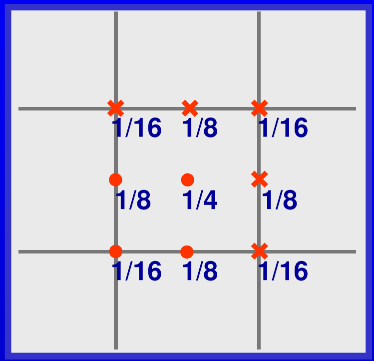


● **Sample from pixel**



MULTIPLE SAMPLE ANTI-ALIASING

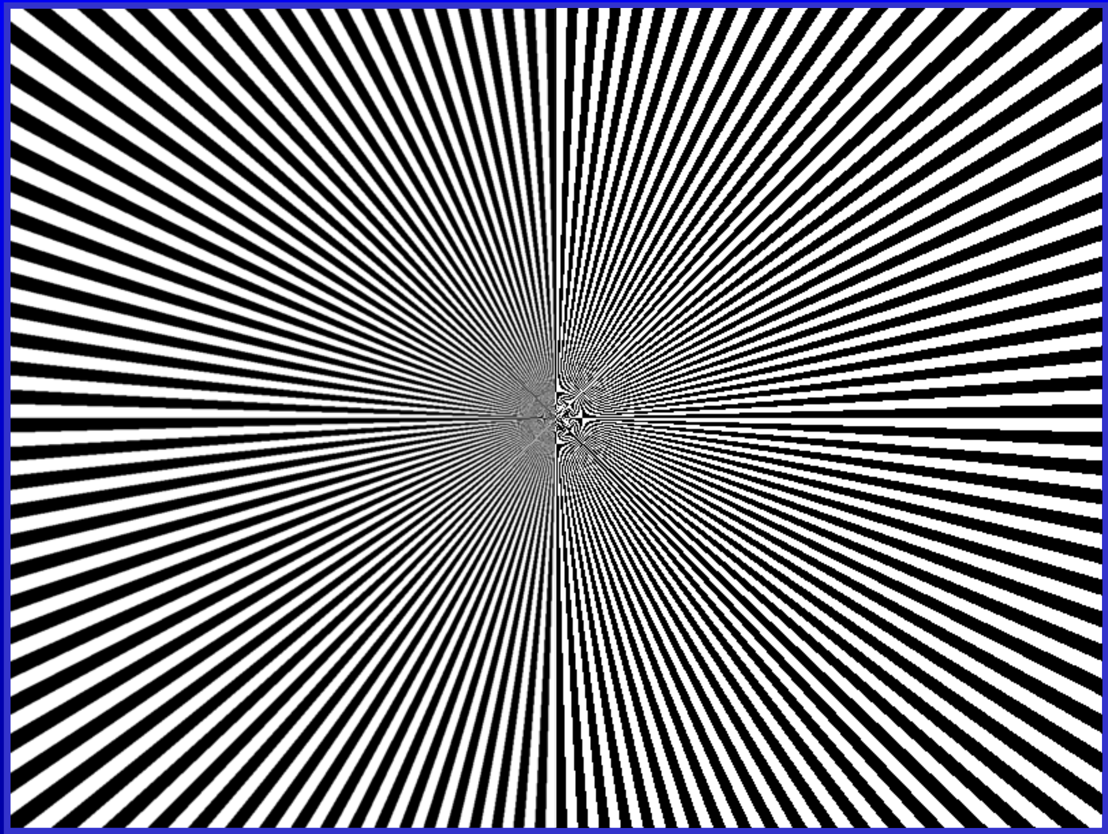
FOUR SAMPLE, NINE TAP



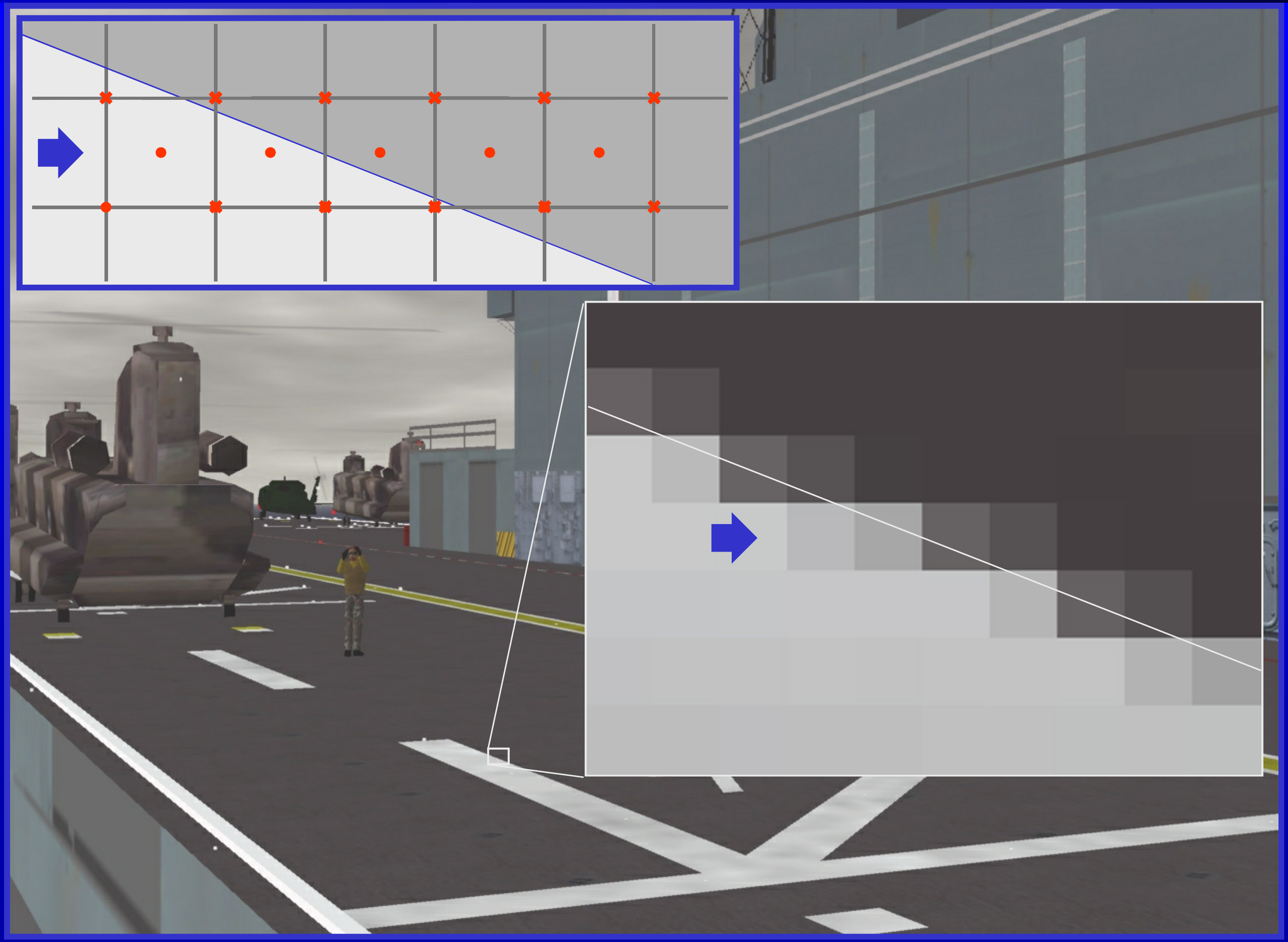
● Sample from pixel

✕ Sample from adjacent pixels

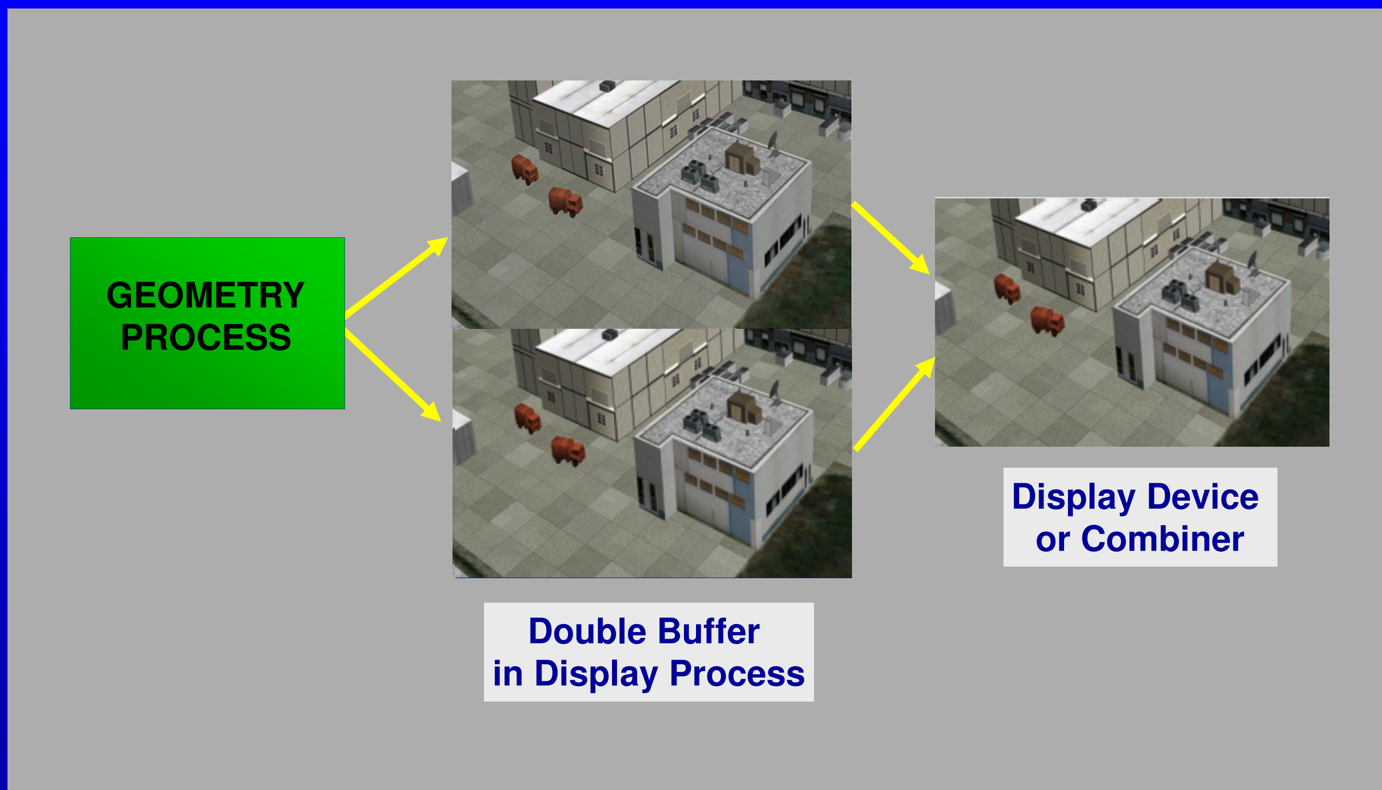
More complex versions of this method in later graphics devices



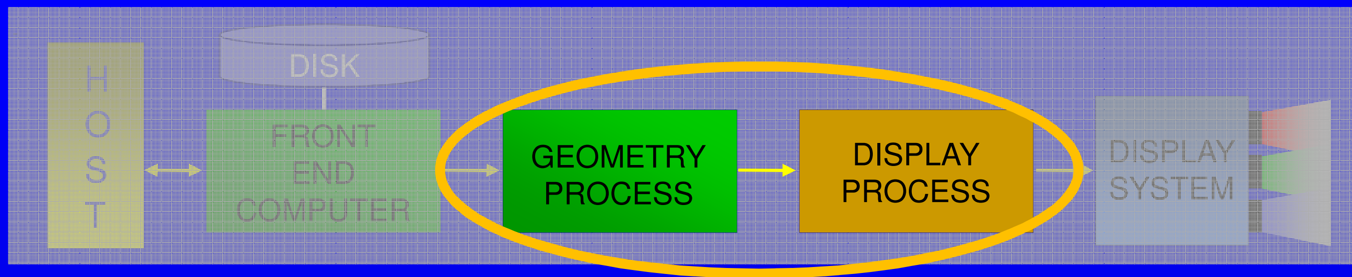
MULTIPLE SAMPLE ANTI-ALIASING



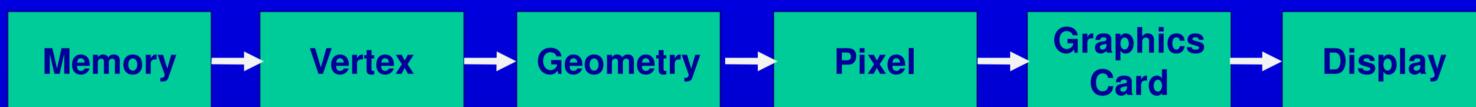
DISPLAY PROCESS – SWAP BUFFER



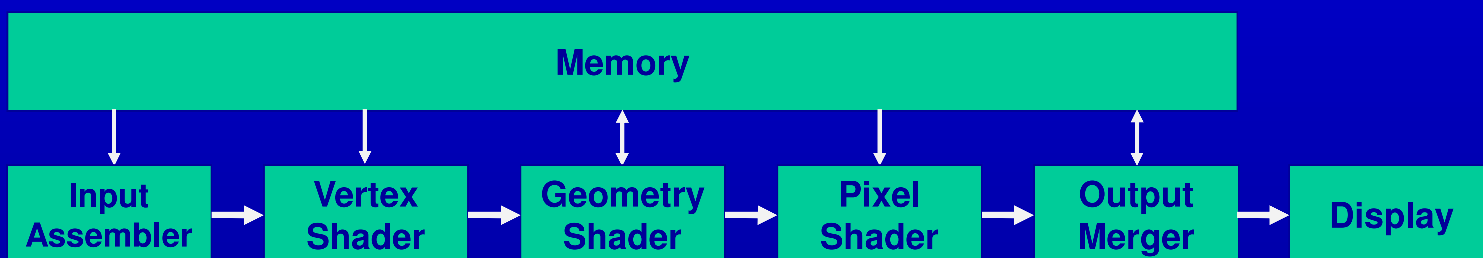
GRAPHIC PIPELINES THEN AND NOW



Fixed-Function Pipeline (on CPU)



Shader Pipeline (on GPU)



NOTICE THE SIMILARITY BETWEEN THE STEPS IN THE PROCESSES

THE MADERN GRAPHICS PIPELINE IN PRACTICES

Fixed-Function Pipeline

Shader Pipeline

THE MADERN GRAPHICS PIPELINE IN PRACTICES

Fixed-Function Pipeline

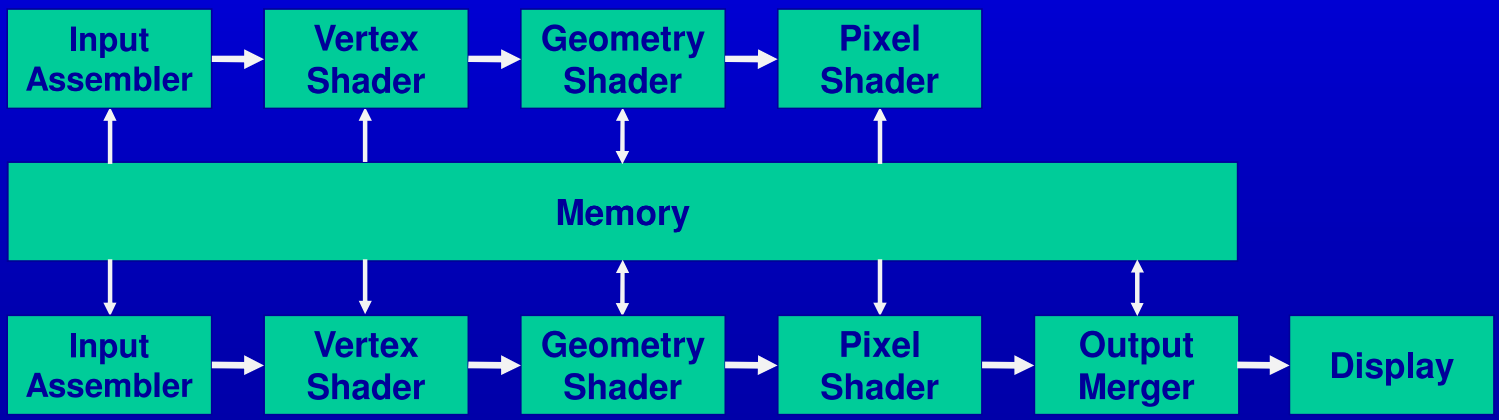
Shader Pipeline

THE BENEFIT OF SHADERS IS PARALLEL PROCESSING

The more GPU cores the more content that can be displayed.
Each Shader runs independently in it own thread on the core and most systems have multiple cores running multiple threads each. The Shaders tend to be so fast that you can also run them for several passes per-frame. This allows for effectively endless combinations per-frame creating boundless flexibility.

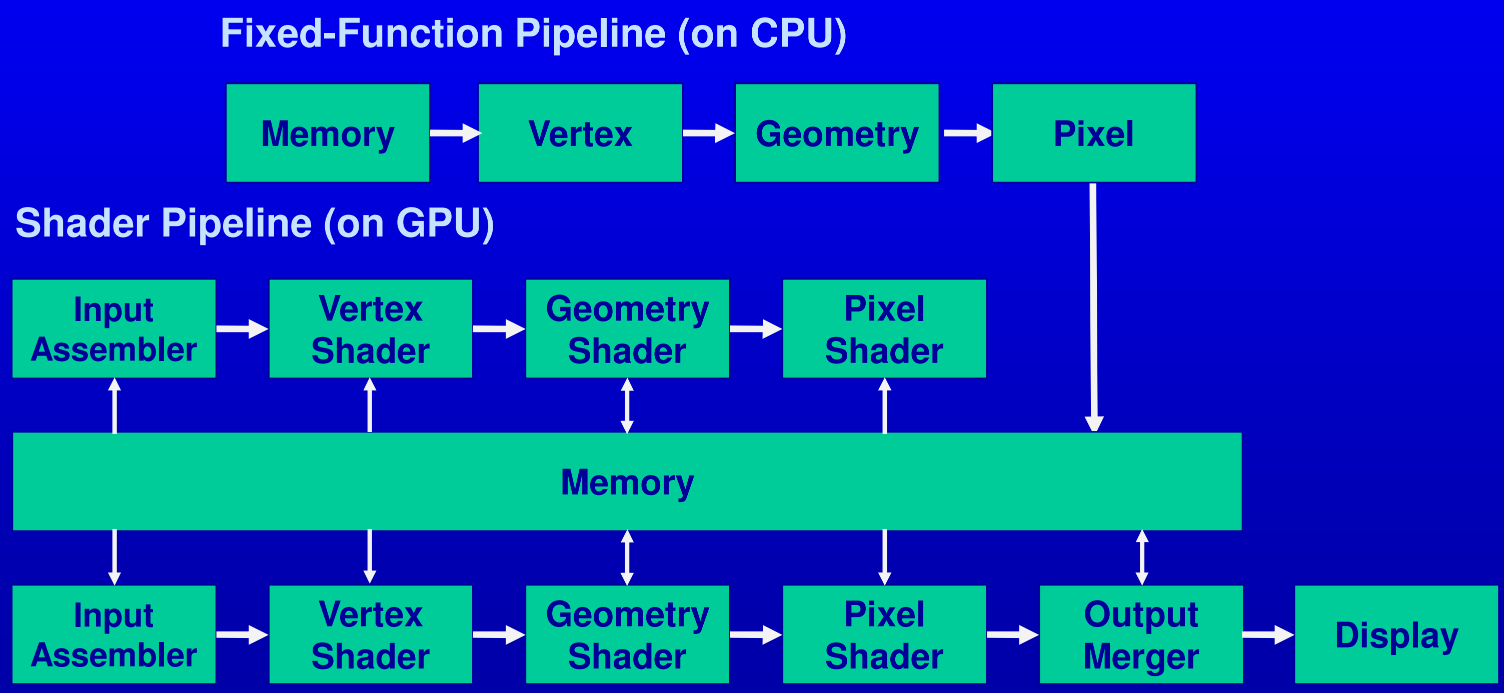
Hence why people like to say "Shaders are free!"

Shader Pipeline (on GPU) Run in Parallel



THE GRAPHICS PIPELINE IN PRACTICES

Most of todays IGs use a combination of the two pipelines. The Fixed-Function portion is used to define only the basic explicate geometry in the form of pre-modeled objects. They are then send them to the Shaders to be used as there “existing” geometry to enhanced.



SO EFFICIENT there "FREE" ?

OVERLOAD

- Each image generator process can (and will) overload
- Front end computer or CPU overload
  - Too many effects, dynamic models, calculations (e.g. Ballistics, scoring)
  - Loading of Database, particularly texture maps
  - Solution - do less important functions in background, reduce traffic
- Geometry process
  - Too many polygons, light points and/or shaders to test and process
  - Solution - remove polygons/lights from scene - reduce detail
- Display process
  - Pixel fill rate, depth complexity - too many polygon overlays,
    - Too many pixel shader effects, geo-state changes
  - Solution - reduce pixel calculations - difficult without visible artifacts
- PC systems provide very little performance information from GPU
  - Everything done in single frame - every function dependent on all others
  - Overload best prevented with careful Database design, implementation

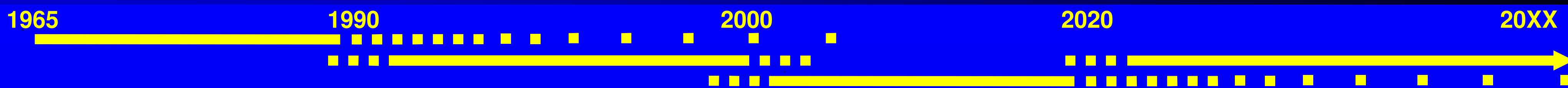
Large System IGs

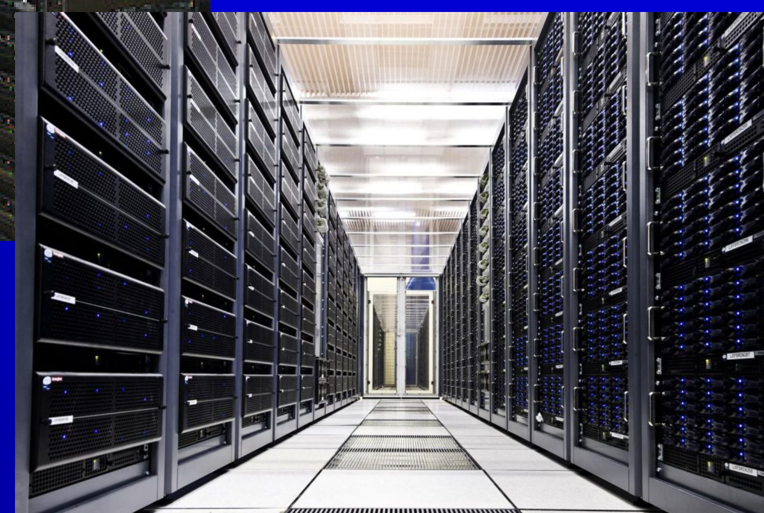
- A Roomful of Computers

IMAGE GENERATOR EVOLUTION

1965 1990 2000 2020 20XX

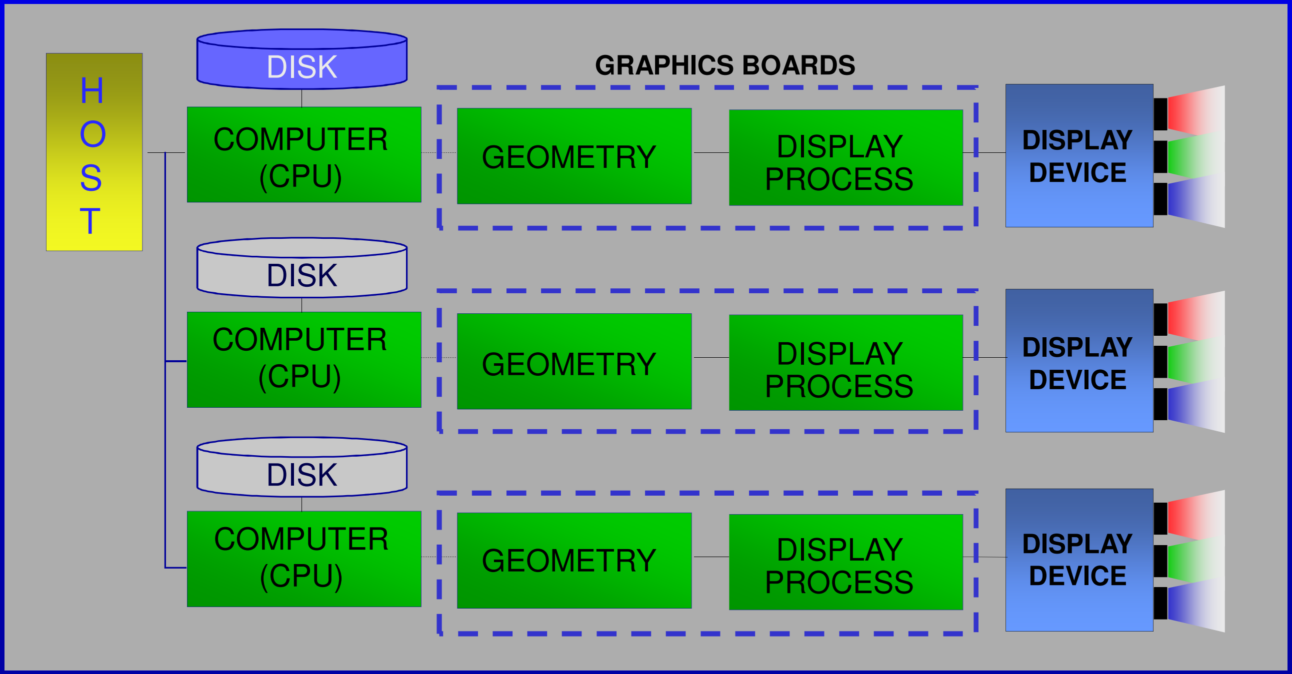
IMAGE GENERATOR EVOLUTION





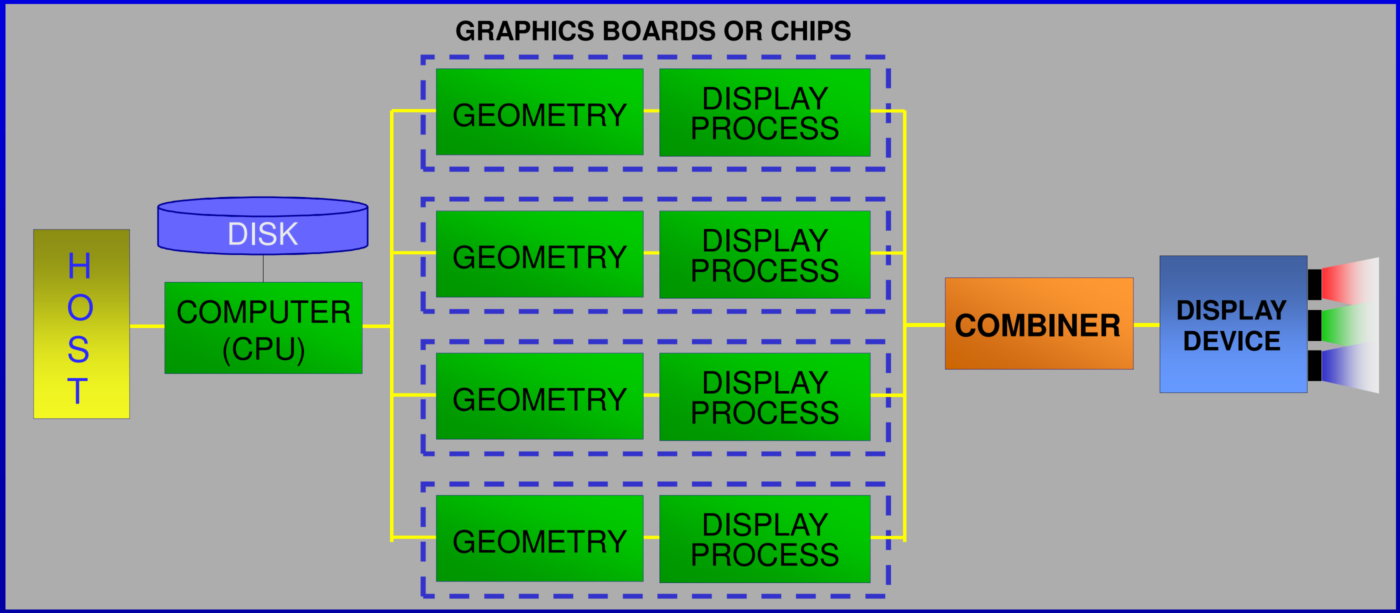
TYPICAL PC IMAGE GENERATOR (PC-IG)

- Architecture for simple multi channel system – One PC per channel
  - One PC, or additional PC may be master
    - Communicate with host, coordinate system functions
  - May use single disk to distribute Database, system software
  - May be necessary to synchronize channels for image stability
  - Different viewing frustum (field of view) for each channel

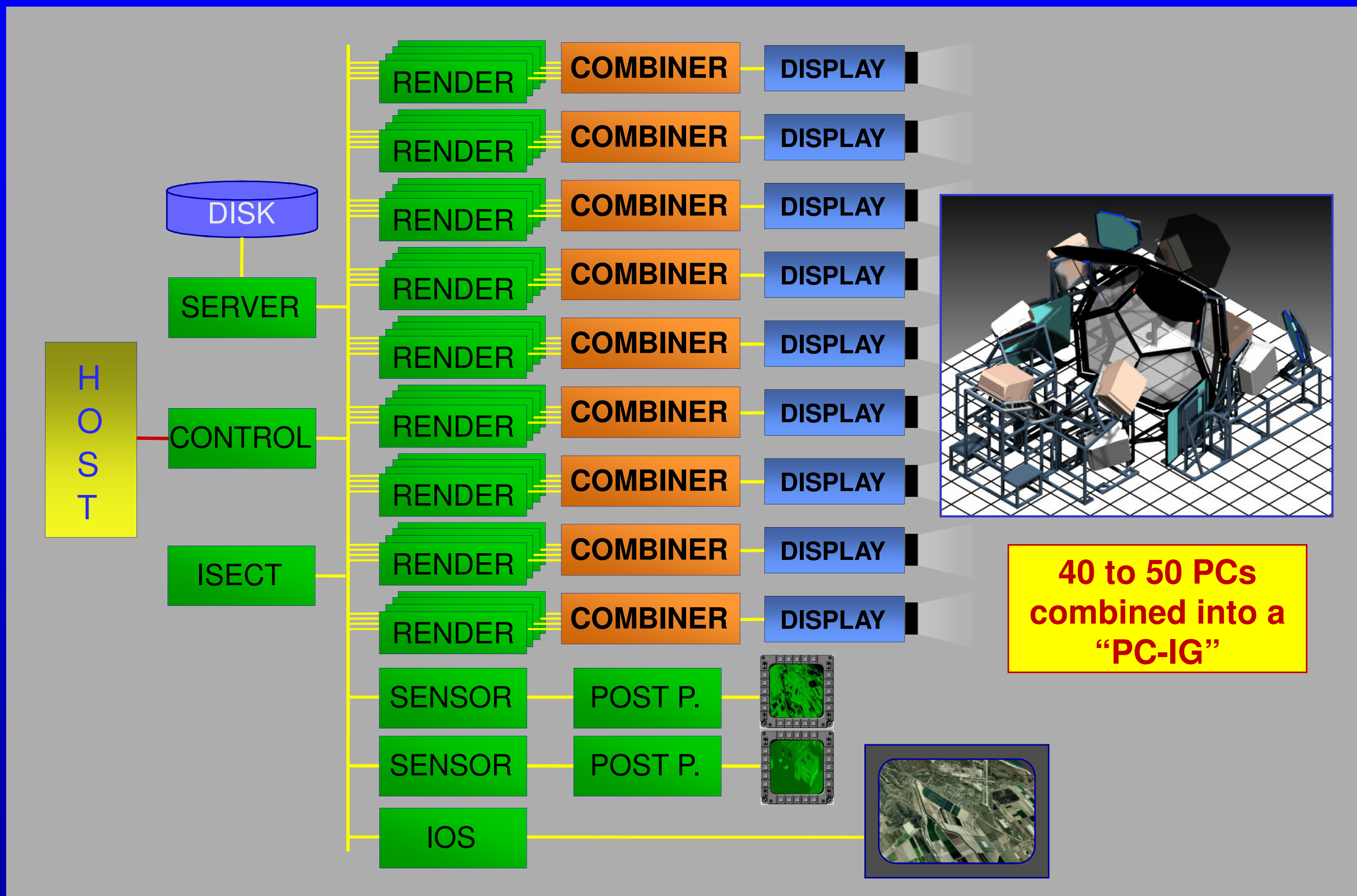


TYPICAL PC “BASED” IMAGE GENERATOR

- Architecture for multi – PC and/or GPU channel
  - Images from multiple PCs, boards or chips combined for single image
    - Increased pixel and polygon capacity, improved anti-aliasing, both
    - GPUs should be gen locked in order to synchronize the images
  - Possible/practical due to low cost of PC commodity components
  - Current designs include multiple GPUs in single graphics board (e.g. SLI)
  - Combiner function may be included in new high resolution projectors



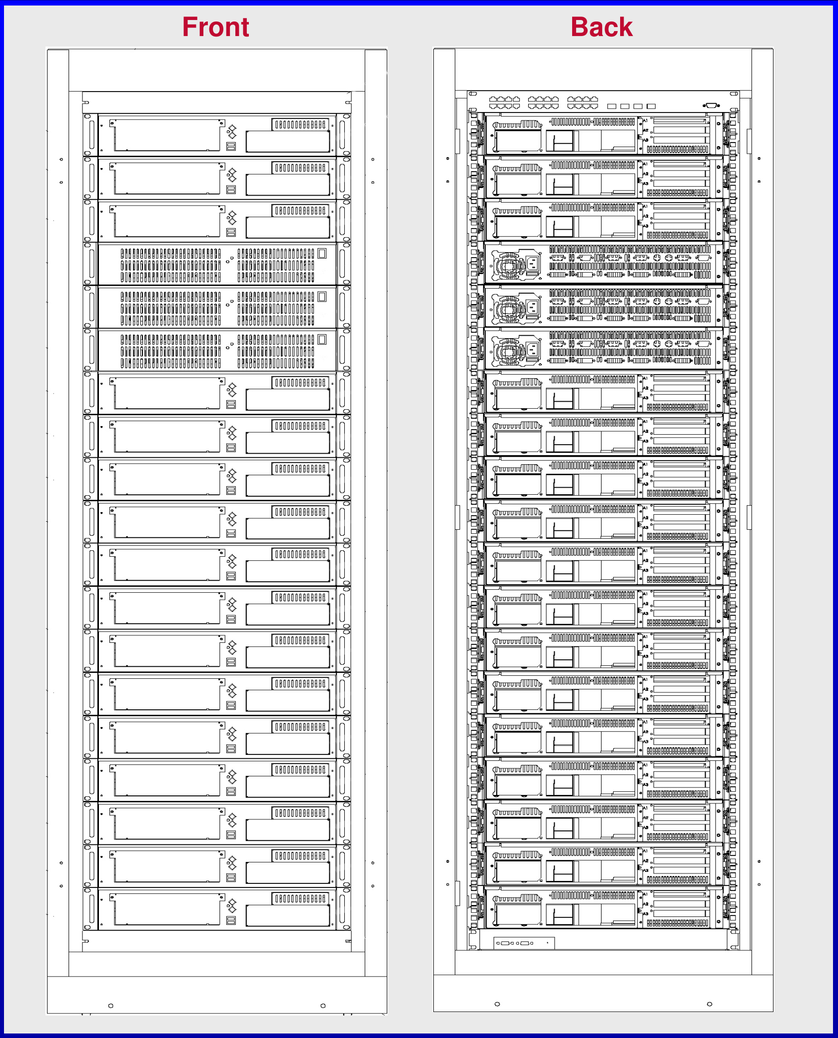
NOTIONAL SYSTEM CONFIGURATION



NOTIONAL IG CABINET LAYOUT

- Not typical desk side PCs
- Standard 40u, 19” rack
  - All components 2u
  - May be more or less
- Three or four cabinets for a full up system
- This configuration has up to 16 computers, 3 video combiners in each rack

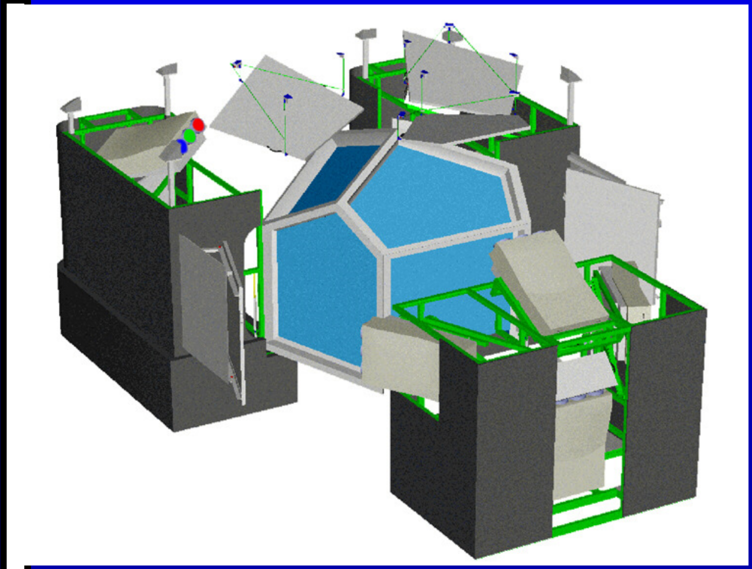
- Less hardware footprint with greater performance than previous generations



SYSTEM IMPLICATIONS – MULTI CHANNEL

- “Channel” is generally the image from a single display device
- Airlines generally require three or five channel systems
  - FAA requires 75° per pilot for Level C and D (fixed wing)
  - 220° x 60° for helicopter simulation
- Military may require eight or more - large field of view and sensors
  - Systems are getting larger as IG per channel costs come down

3 OR 5 CHANNEL SYSTEM



8 CHANNELS + SENSORS

TYPICAL MULTI-CHANNEL APPLICATIONS

TYPICAL MULTI-CHANNEL APPLICATIONS

TYPICAL MULTI-CHANNEL APPLICATIONS

TYPICAL MULTI-CHANNEL APPLICATIONS

TYPICAL MULTI-CHANNEL APPLICATIONS

Available Features, Functions, Capabilities

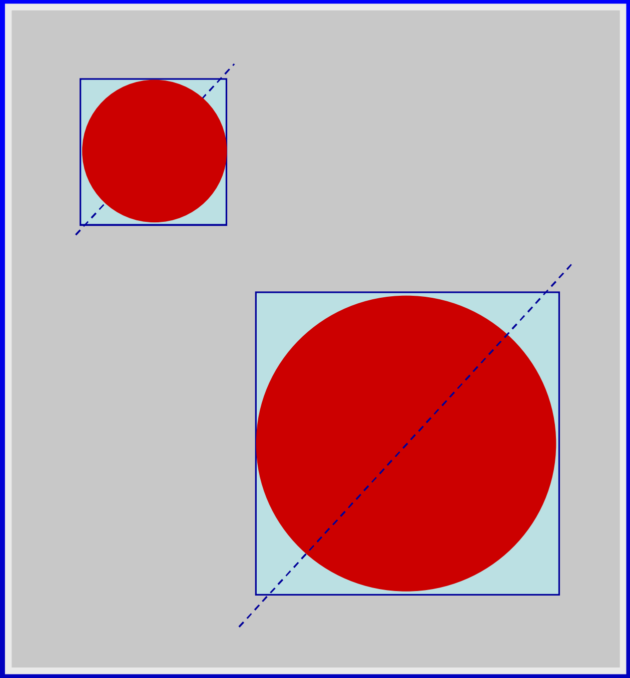
- a Progression in Fidelity

Light Point

- In the beginning…..

LIGHT POINTS TYPES

- Calligraphic drawn separately
  - Electron beam allowed to dwell on each lightpoint
  - Sharper and brighter
  - Serious display impacts, costs
  - Requires separate processing and buffering in IG
- Raster lights – point features
  - Included in pixel fill
  - Much less dynamic range
- Raster lights – shader generated
  - Square polygons with texture circles
  - Size of light controlled by vertex shaders based on distance
  - Brightness of light controlled by fragment shader based on distance and other parameters
  - May be implemented with geometry shader
    - Single point sent to GPU – polygon formed and controlled by shader



LIGHT POINTS (CONT.)

- Color and intensity
  - Raster light colors same as polygons
  - Calligraphic light colors may be limited by display
- Switch control for user interaction
- Directionality - angle and width
  - Horizontal and vertical
- Special effects required for complex airport features, other applications
  - Flashing and rotating strings
  - Sequenced flashing lights (strobes)
    - Same point hit several times to create bright spot (calligraphic)
  - Vertical approach slope indicator (VASI)

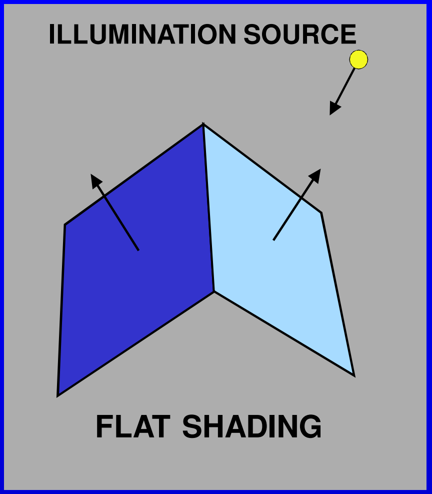
VISIBILITY (FOG) EXAMPLES

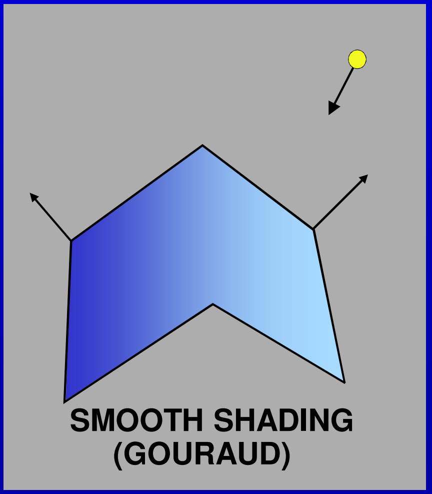
Polygon Shading

- not “Shaders” yet

POLYGON SHADING

- Sun shading (flat)
  - Polygon normal compared to vector from "sun"
  - Entire polygon same intensity
  - Minimum intensity limited - “ambient” level
- Smooth shading (Gouraud)
  - Normal vectors are compared with illumination vector to compute color at the vertices
  - Pixel color interpolated from vertice color during pixel fill process
  - Vertice colors interpolated across adjacent polygons
  - Objects made to appear rounded





EXAMPLES OF FLAT SHADING

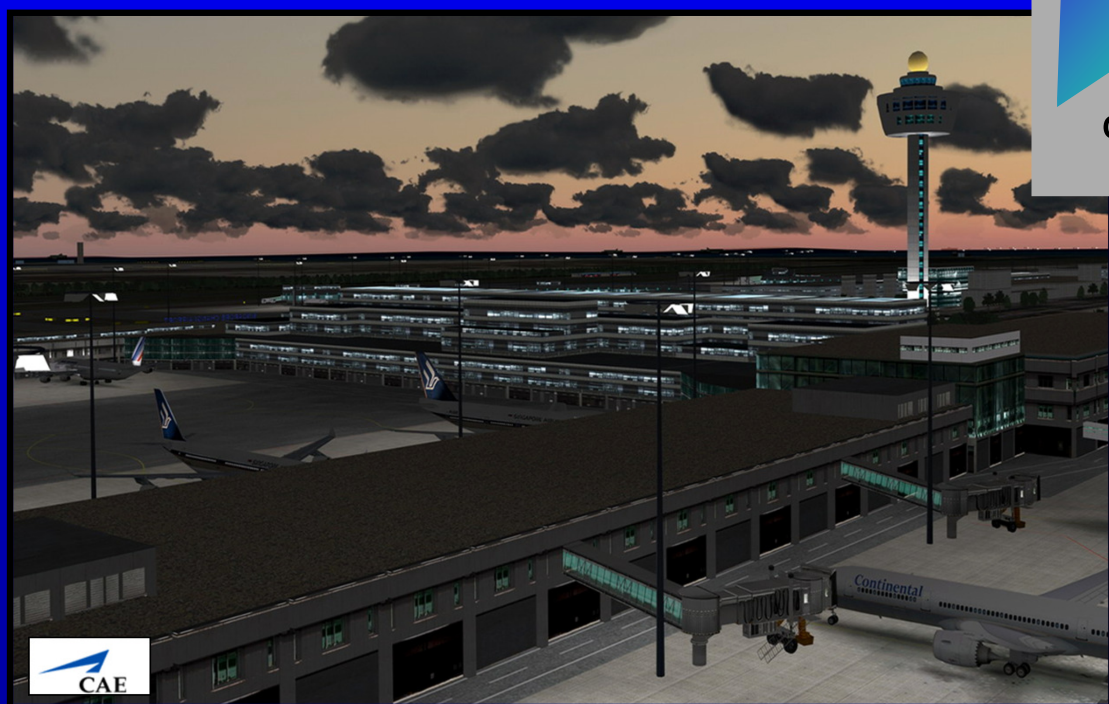
EXAMPLES OF SMOOTH SHADING

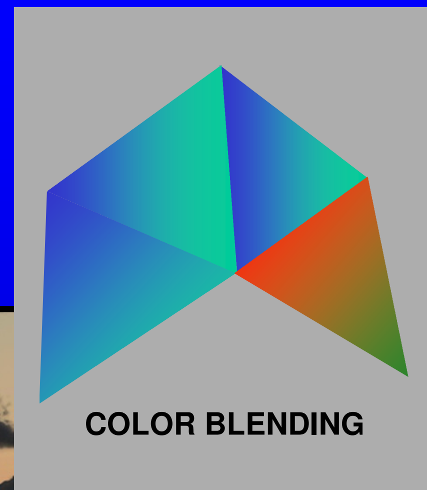
EXAMPLE OF DYNAMIC SMOOTH SHADING

EXAMPLE OF SPECULAR SHADING

POLYGON SHADING

- Color blending
  - Color defined for each polygon vertex
  - Blends color across a single polygon
- Luminous, emissive polygons
  - “Day" intensity at dusk or night





90

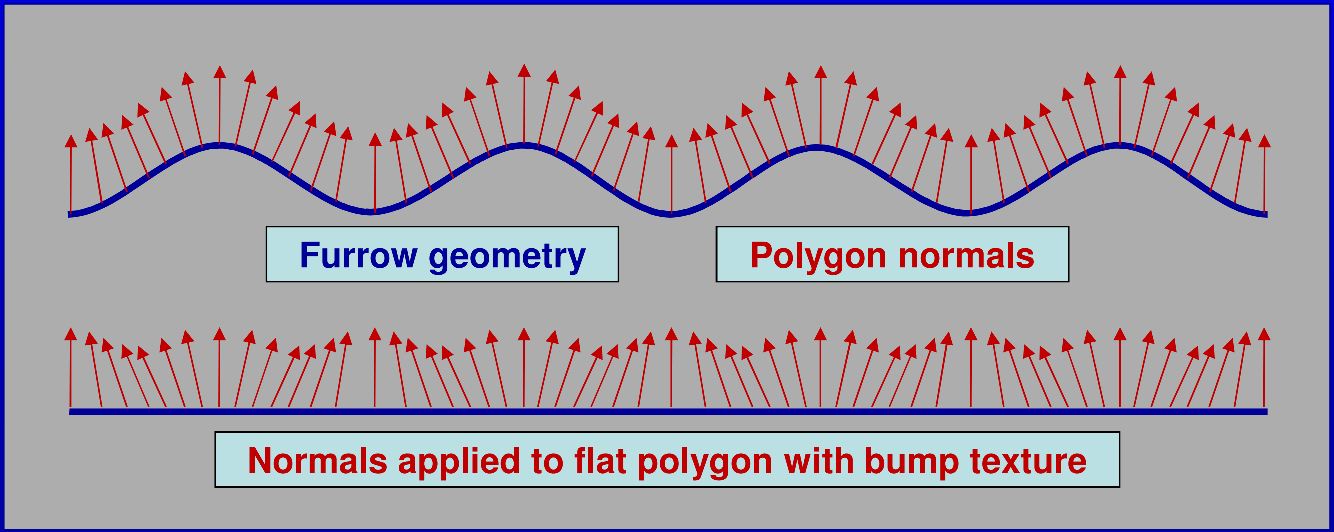
EXAMPLES OF LUMINOUS POLYGONS

BUMP MAPPING

- Few “real” implementations
  - Lots of emulations, tricks
  - Only work in special cases
- Requires Phong shading
- Map data bends surface normals for each pixel
  - Surface responds to changes in illumination angle
- Now being emulated in shaders

BUMP MAPPING EFFECT

- Plowed field example
- Geometry of furrows
- Polygon normals if field created with large number of small polygons
  - Interact with illumination source(s) for shading
- Apply same normals to flat polygon
  - Align with texture image of plowed field
- Normals interact with illumination to create appearance of furrows



BUMP MAPPING OFF

BUMP MAPPING ON

BUMP MAPPING ON - DIFFERENT SUN ANGLE

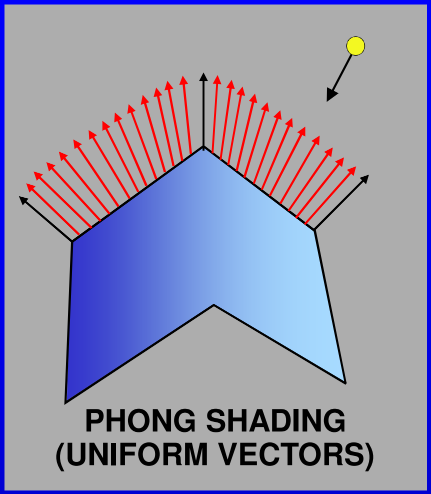
BUMP MAPPING ON, SPECULAR SHADING OFF

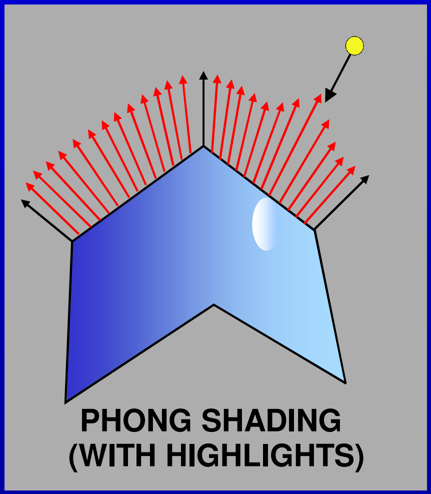
BUMP MAPPING ON, SPECULAR SHADING ON

BUMP MAPPING - CONTAMINATED RUNWAY

PHONG SHADING

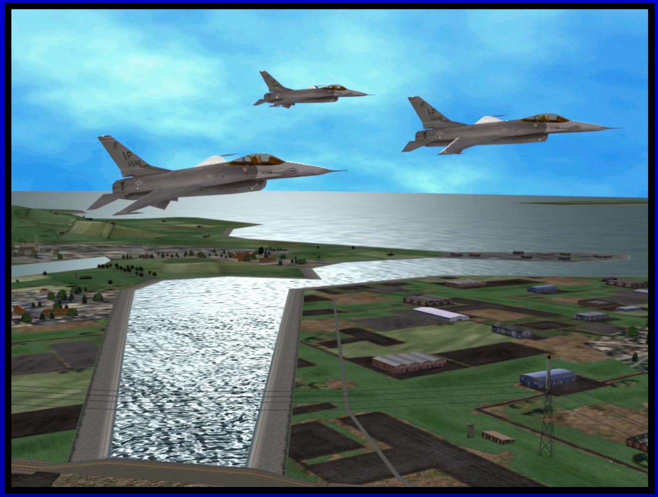
- Phong shading
  - Normal vectors averaged across adjacent polygons
  - Per pixel normal vectors are interpolated from vertice normals
  - Each pixel normal individually compared with illumination vector to determine pixel color
  - Additional calculations can exaggerate illumination based on desired material type
    - Matte surfaces will have small highlight effect
    - Shinny surfaces will have highlights
      - Glint from canopies, water surface
    - Effect created by modifying the length of the vectors relative to sun angle
  - Bump texture modifies the angle of these normals to texture the surface
    - Gives 3D effect to flat polygon



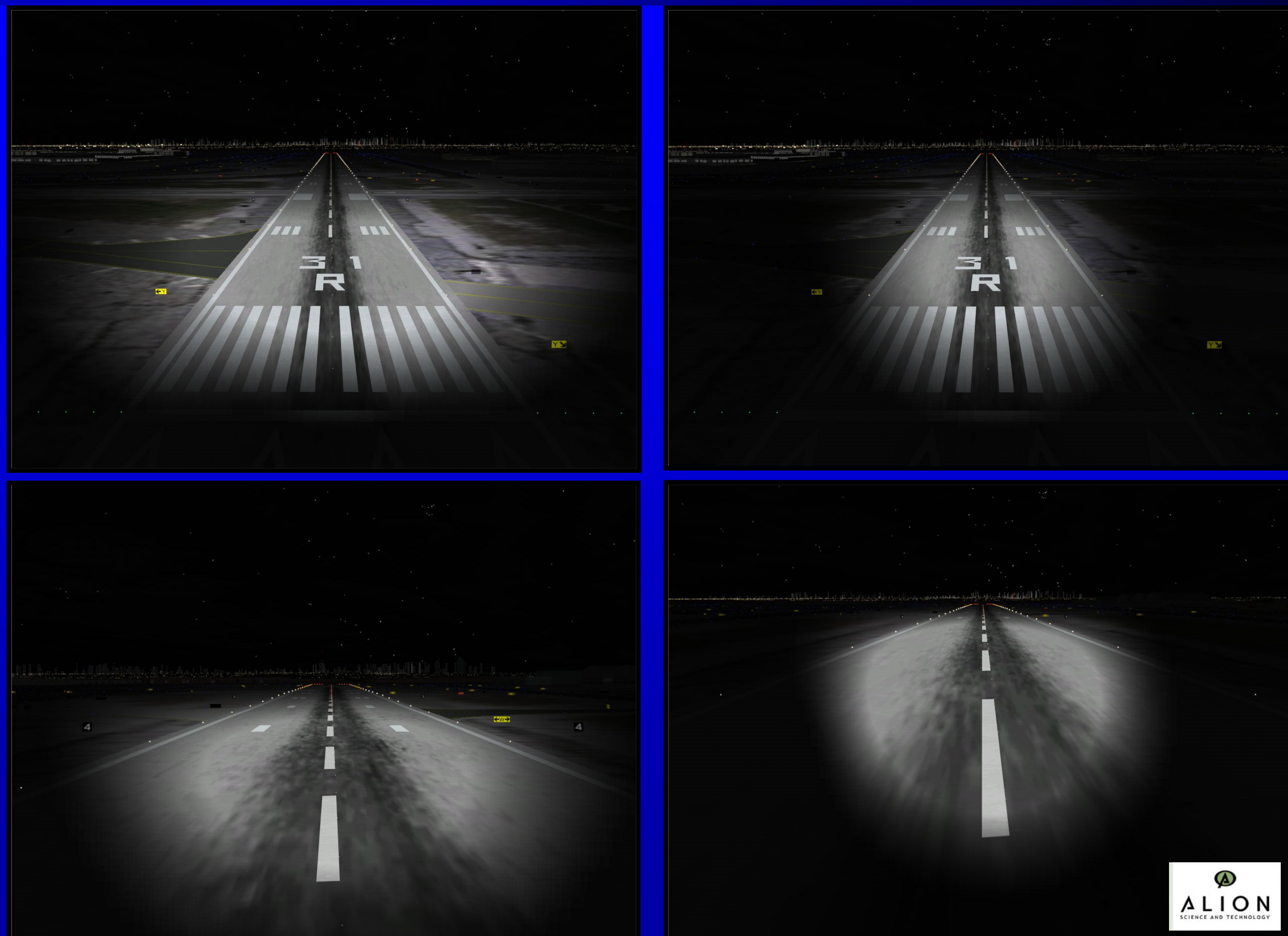


PHONG SHADING

- Can be used in place of other shading methods for more realistic and dynamic lighting reflections



PHONG SHADING - LANDING LIGHTS



LANDING LIGHTS, SEARCH LIGHTS

- Illumination of Database terrain or features is difficult – i.e. expensive
- Fixed landing lights can be done as a screen space increase in intensity
  - Steerable lights more difficult
- Older systems included special hardware to support illumination
- Other (smaller) systems will redraw the illuminated area to increase the brightness
  - Pixel fill impact - effected by size of illumination area
- Light sources now included as IG feature in most systems
  - Still expensive but much more effective

Textures

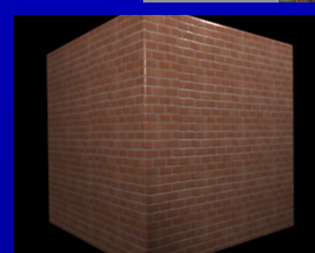
- The First Big Jump

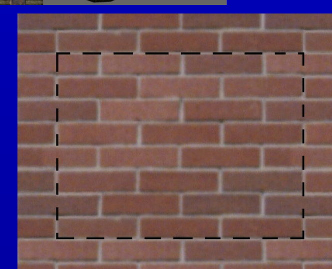
TEXTURE

- The first Major change in IG technology and applications
  - Hard to imagine systems without texture now
- Texture is one or more images applied to a polygon surface
  - Photographs, drawings, synthetic images
  - Increases detail, cues, realism...
- Array of texture elements (texels)
  - Powers of two, usually up to 4096^2
    - Not always square, 64 x 256
- Sensitive to aliasing if texel is allowed to be smaller than pixel
  - "Level of detail" maps, mip maps
    - Multum in parvo
  - Different approach sometimes used for geo-specific texture
    - “clip texture” or “global texture”

TEXTURE MAP PLACEMENT/CAPACITY

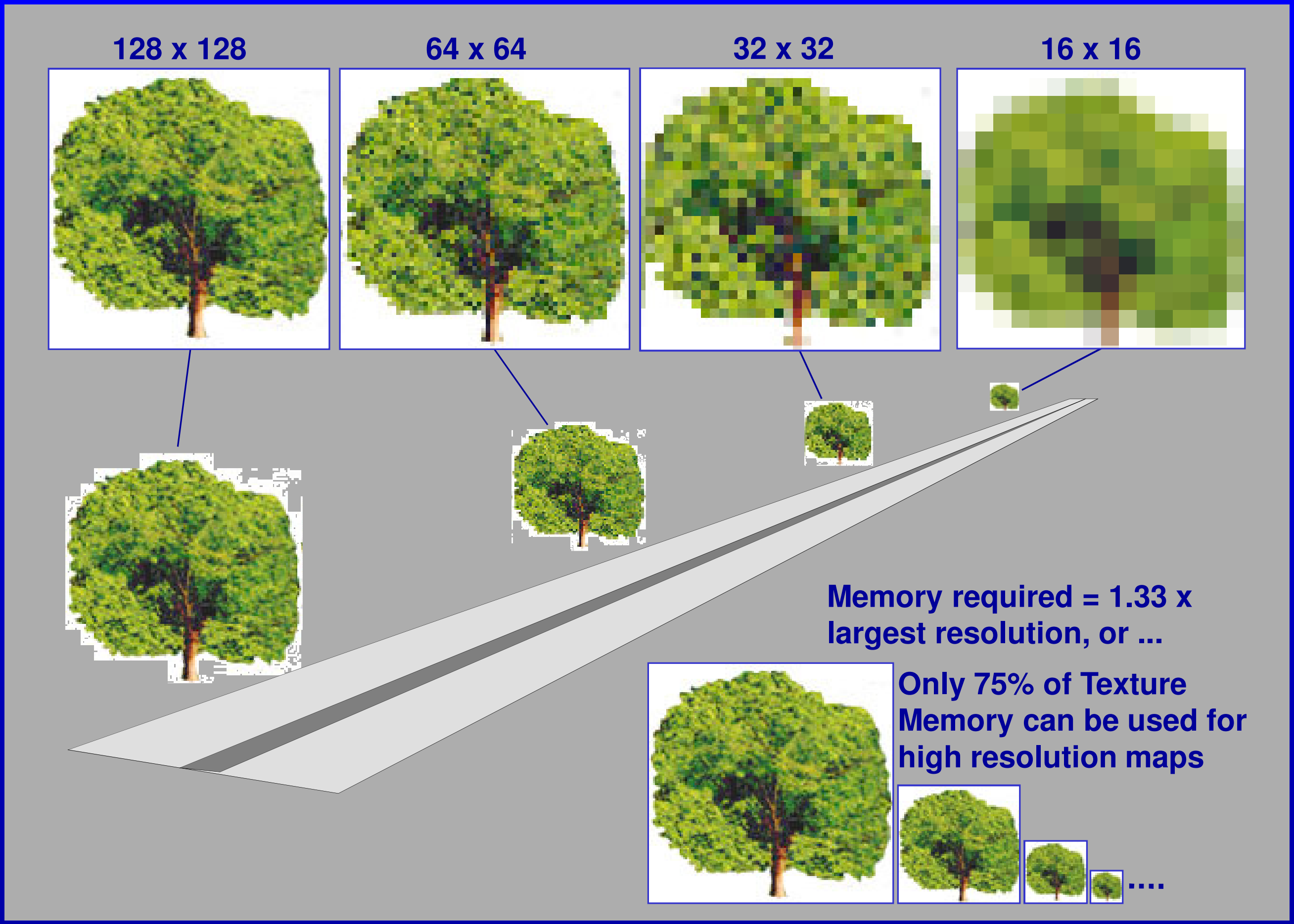
- Techniques for applying texture
  - Per polygon texture
  - Wrapped texture
  - Projected texture
  - Out of plane, reflective texture
- Capacity/capability considerations
  - How many maps per polygon?
  - How many maps/texels on line?
    - Mgbytes of texture memory?
    - Complicated by compression techniques
  - Can new maps be paged in?
    - Needed for geo-specific imagery
  - Texture motion
  - Bi-linear, tri-linear blending





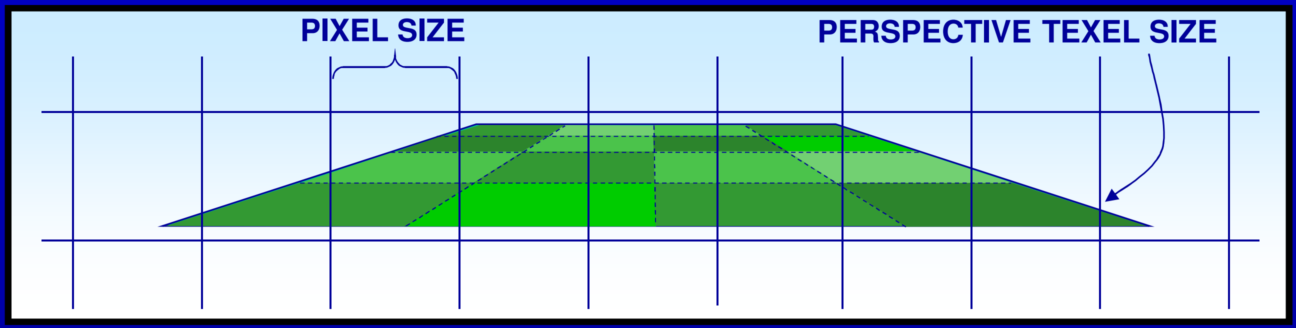
106

MIP TEXTURE MAPS



ANISOTROPIC FILTERING

- Available in most graphics boards and real-time systems
  - Substitute for anti-aliasing in some PC implementations
- Compensates for perspective flattening of texel
- Filters mip levels to smallest pixel dimension - takes multiple samples
  - Averages results for final pixel value
- Computationally expensive
  - Number of additional levels may be selectable
- Typically defined per polygon and/or texture map



ANISOTROPIC FILTERING

ANISOTROPIC FILTERING

ANISOTROPIC FILTERING

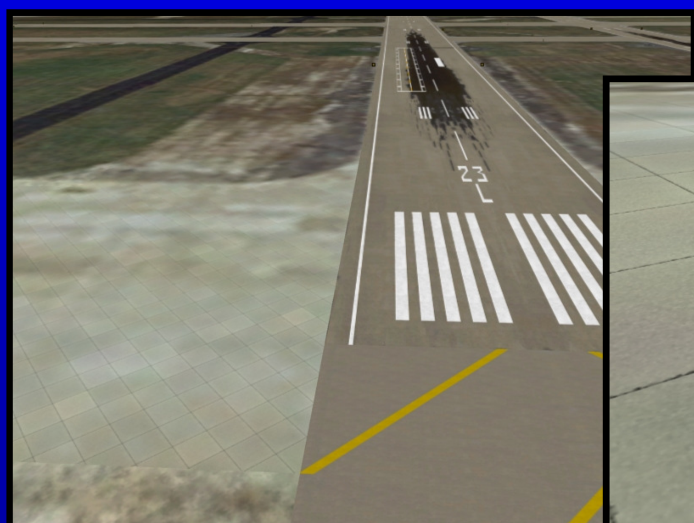
"REFLECTIVE" TEXTURE

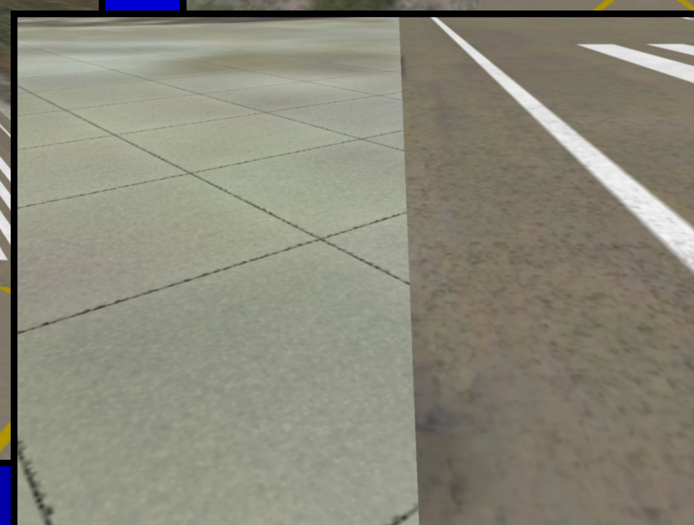
APPLICATIONS OF TEXEL DATA

- Geo-specific texture
  - Uses satellite/aerial photos as source
    - Photo edges must be blended
    - Orthorectification required
    - Artifacts may have to be removed
  - Very large texture capacity required
    - Real-time paging of texture data
  - Provides a great deal of realism for daytime out the window scenes
  - Problems with correlated sensors
    - Little or no feature data may be used to augment content
    - Few polygons to provide material information
    - Texture maps must be encoded with material data per texel
  - Content for FLIR, radar, and other sensors derived from imagery
    - Typical feature/culture data (e.g. DFAD, VMAP) does not fully align with imagery
    - Features simplified and/or out of date

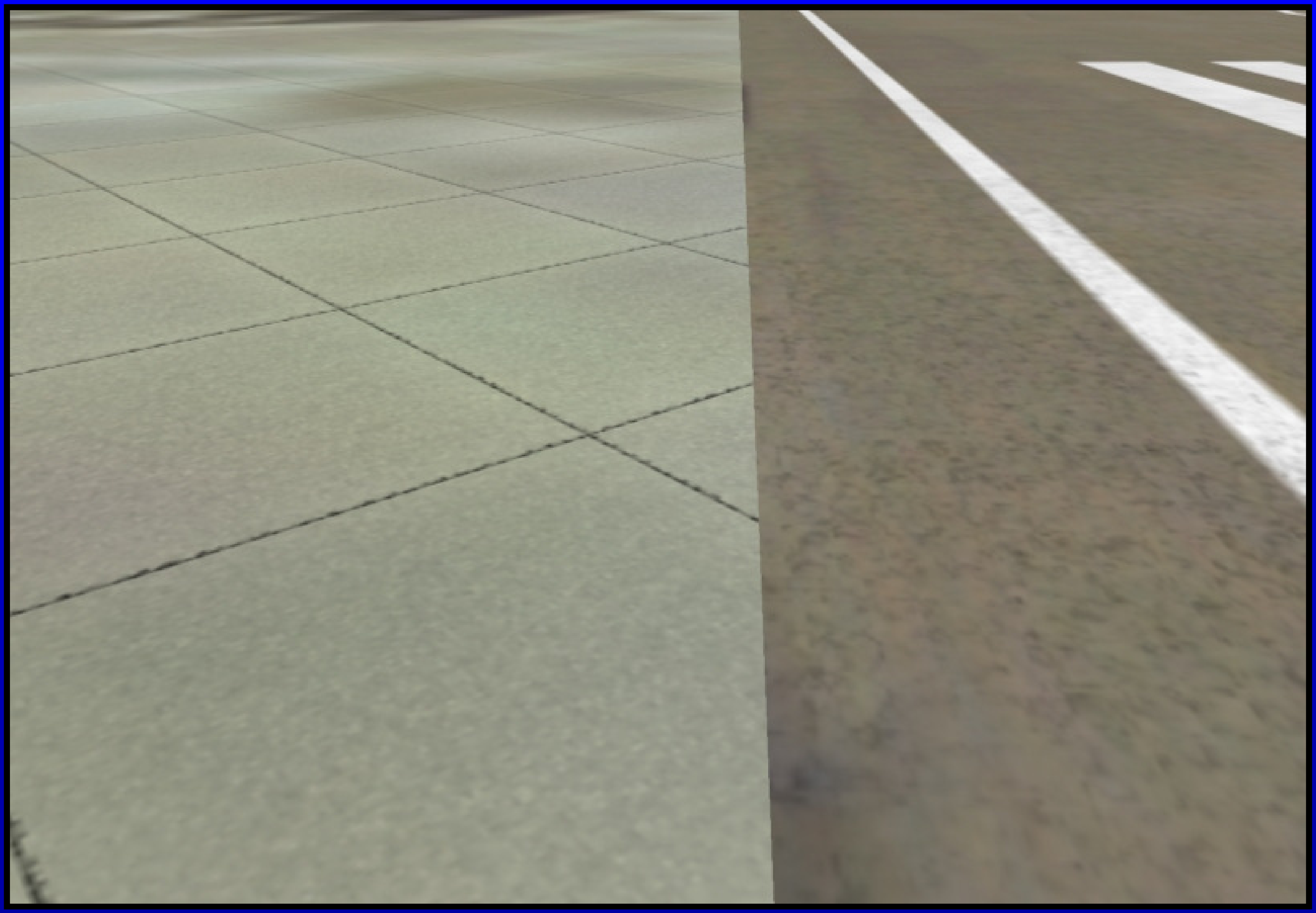
APPLICATIONS OF TEXEL DATA

- Modulation of polygon...
  - Intensity, color, transparency
  - Tessellation should not be obvious
- Full color (photo) texture
  - Replace or augment polygon color
- Micro or detail texture allows detail when close to surface





MICRO OR DETAIL TEXTURE



INTENSITY MODULATION

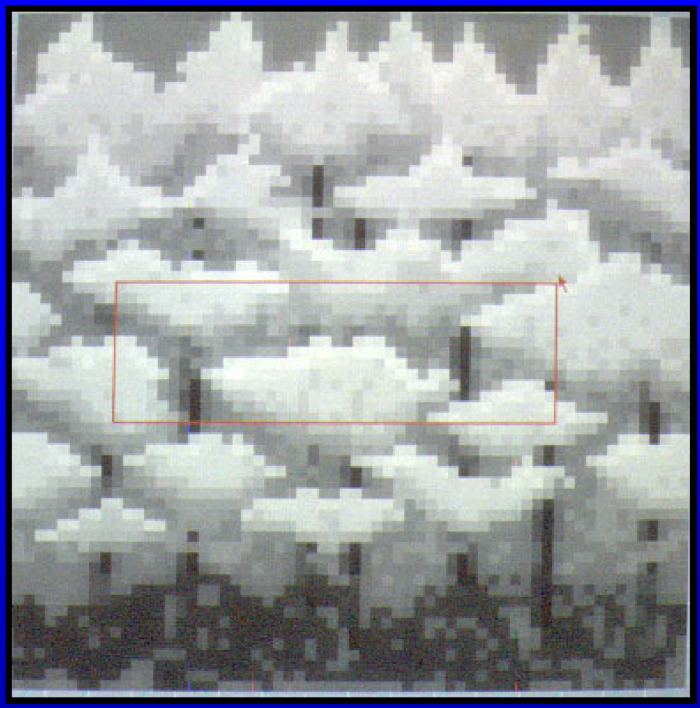
TRANSPARENCY MODULATION

TRANSPARENCY MODULATION

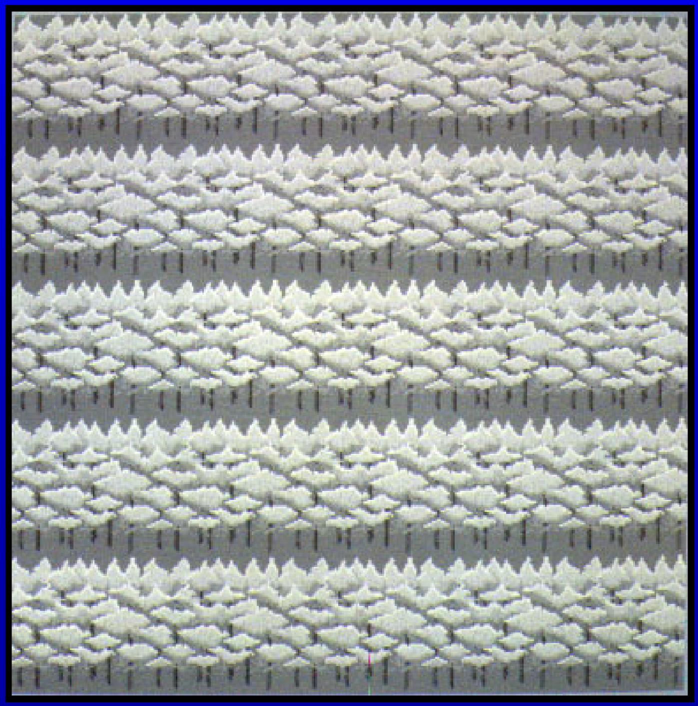
TRANSPARECY MODULATION

- 2D in the distance – 3D when close

TESSELLATION EXAMPLE

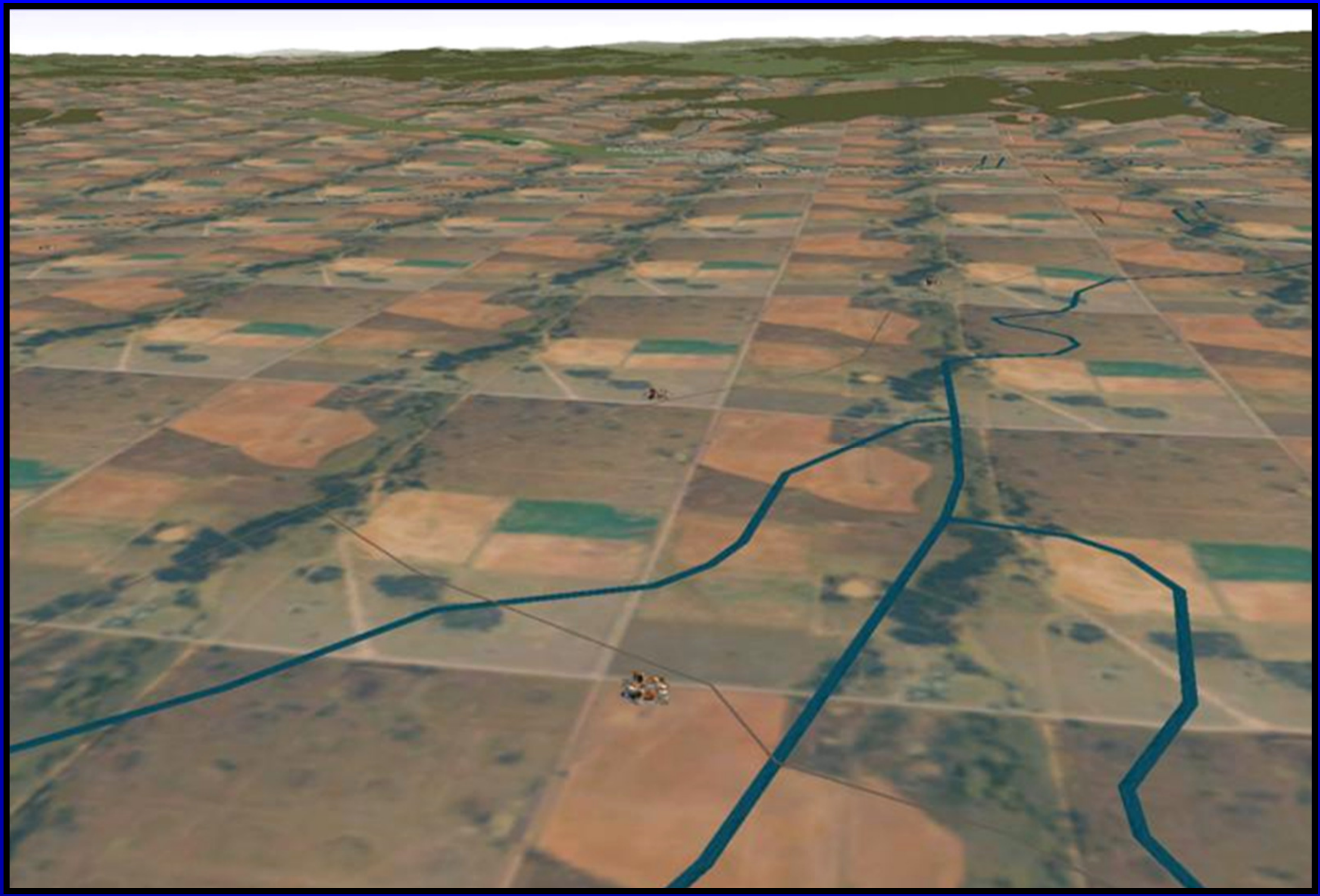


Single instance of texture pattern



Pattern is designed to tessellate horizontally but not vertically

TESSELLATION OF GEO-TYPICAL TERRAIN



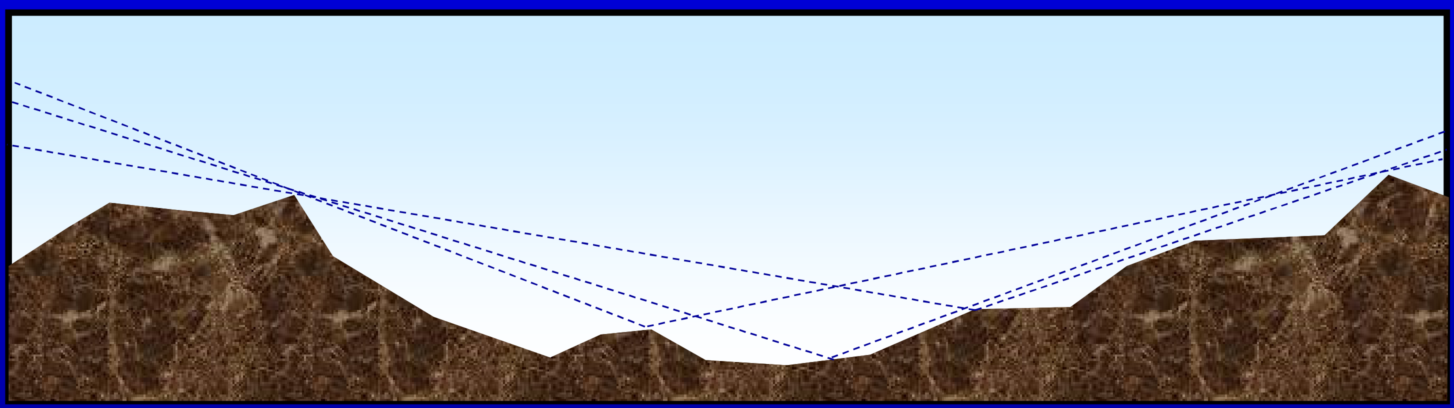
FULL COLOR OR “PHOTO” TEXTURE

FULL COLOR OR "PHOTO" TEXTURE

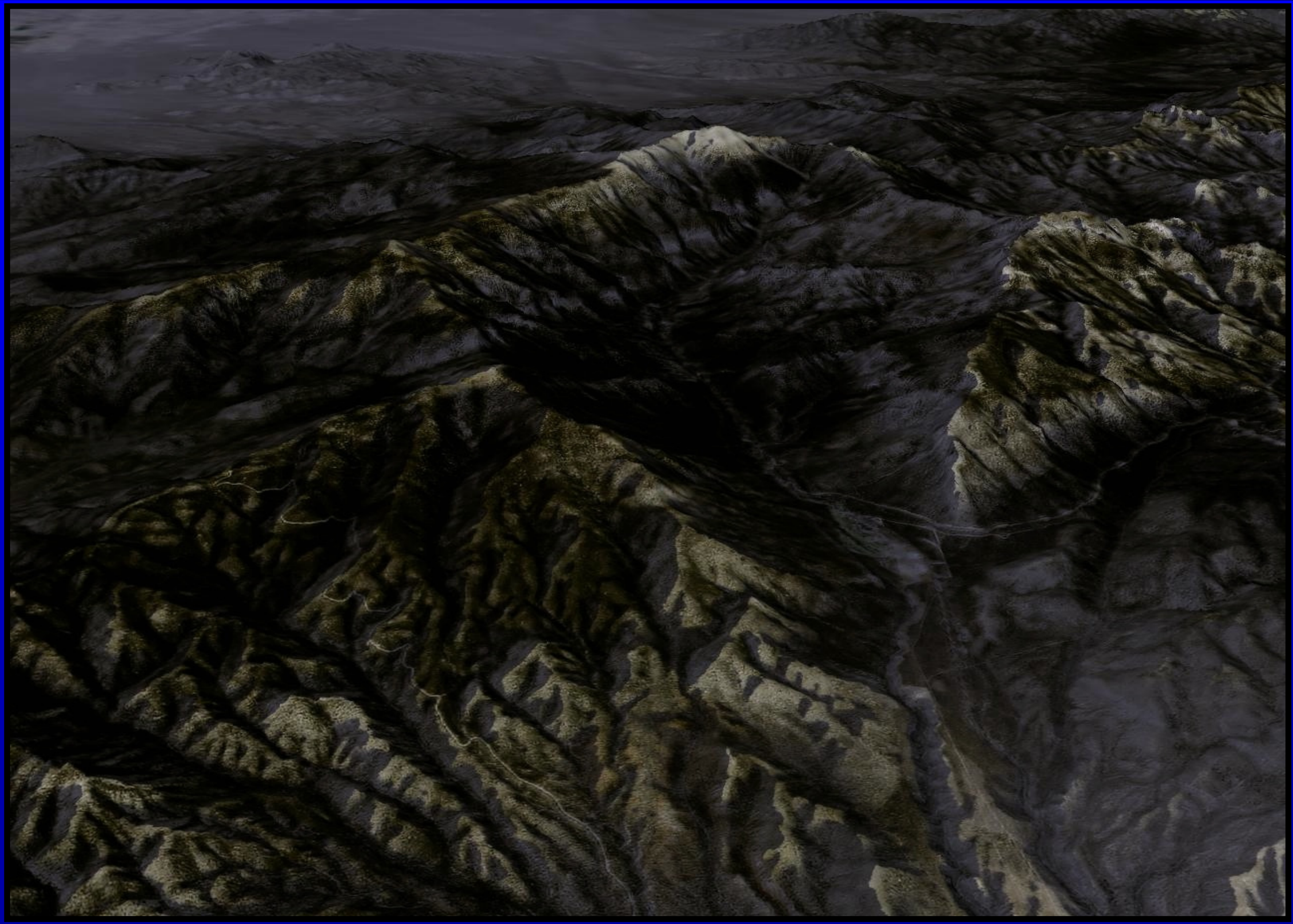
FULL COLOR OR “PHOTO” TEXTURE

“SHADER” BASED SHADING – This is why they are called “SHADERS”

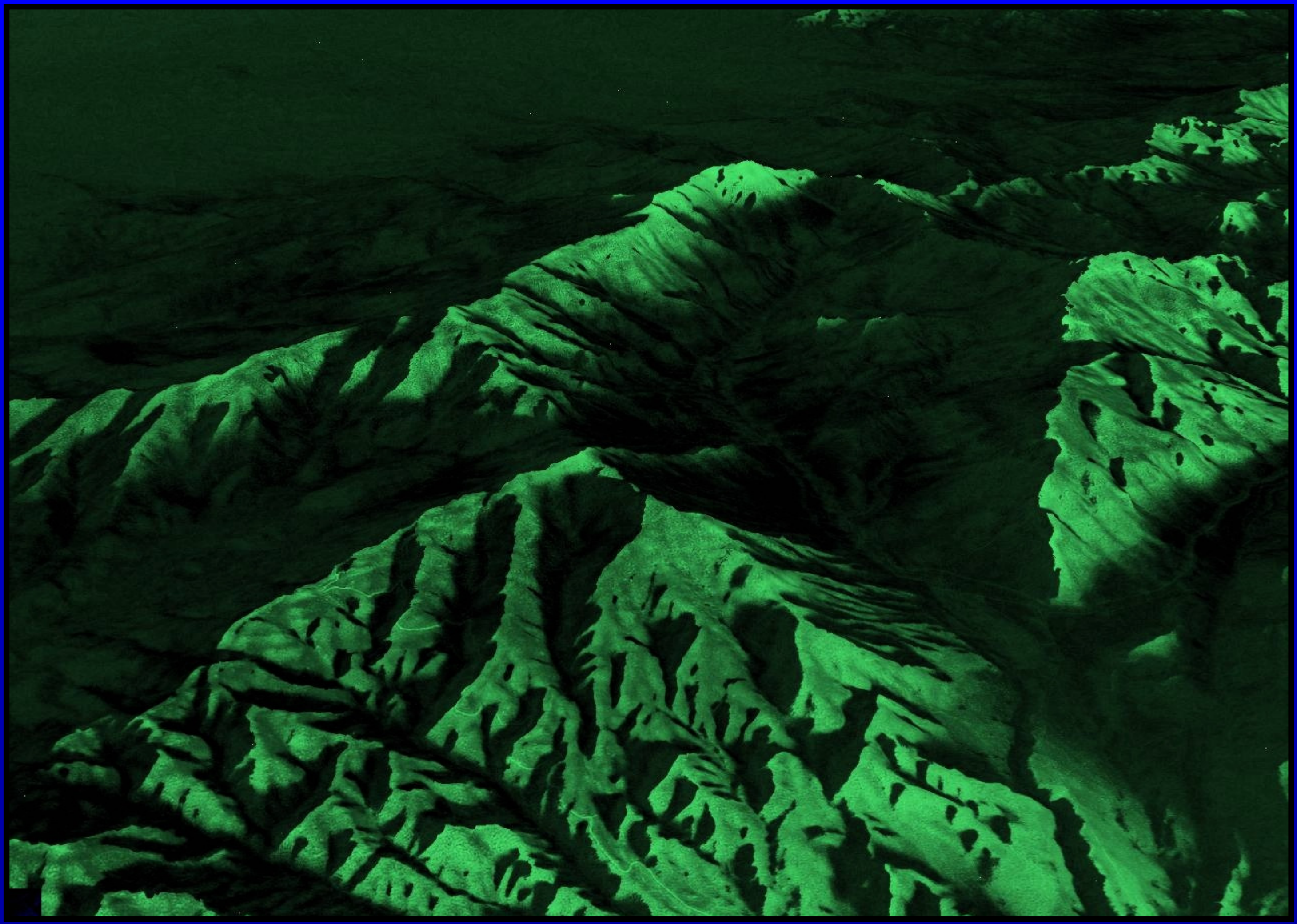
- “Texture map” stores angels to local horizon for each texel
  - Shader compares angels to illumination source
  - Darkens texel when “sun” passes below horizon
  - Used primarily for NVG simulation
  - Works for terrain, not so good for features
  - Air Force including in standard “STAGE” source data sets



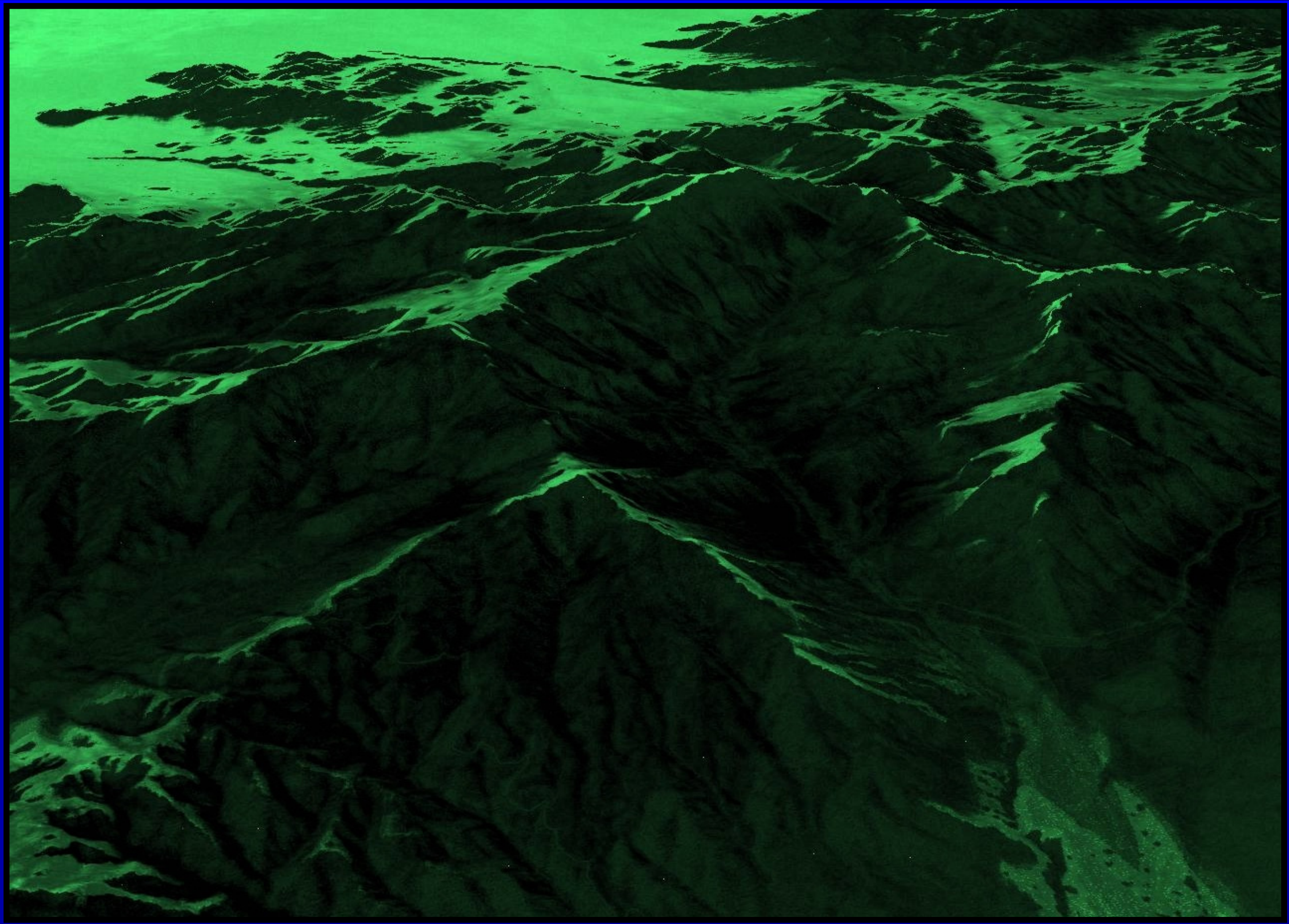
“SHADER” BASED - HORIZON MAPS



“SHADER” BASED - HORIZON MAPS



“SHADER” BASED - HORIZON MAPS



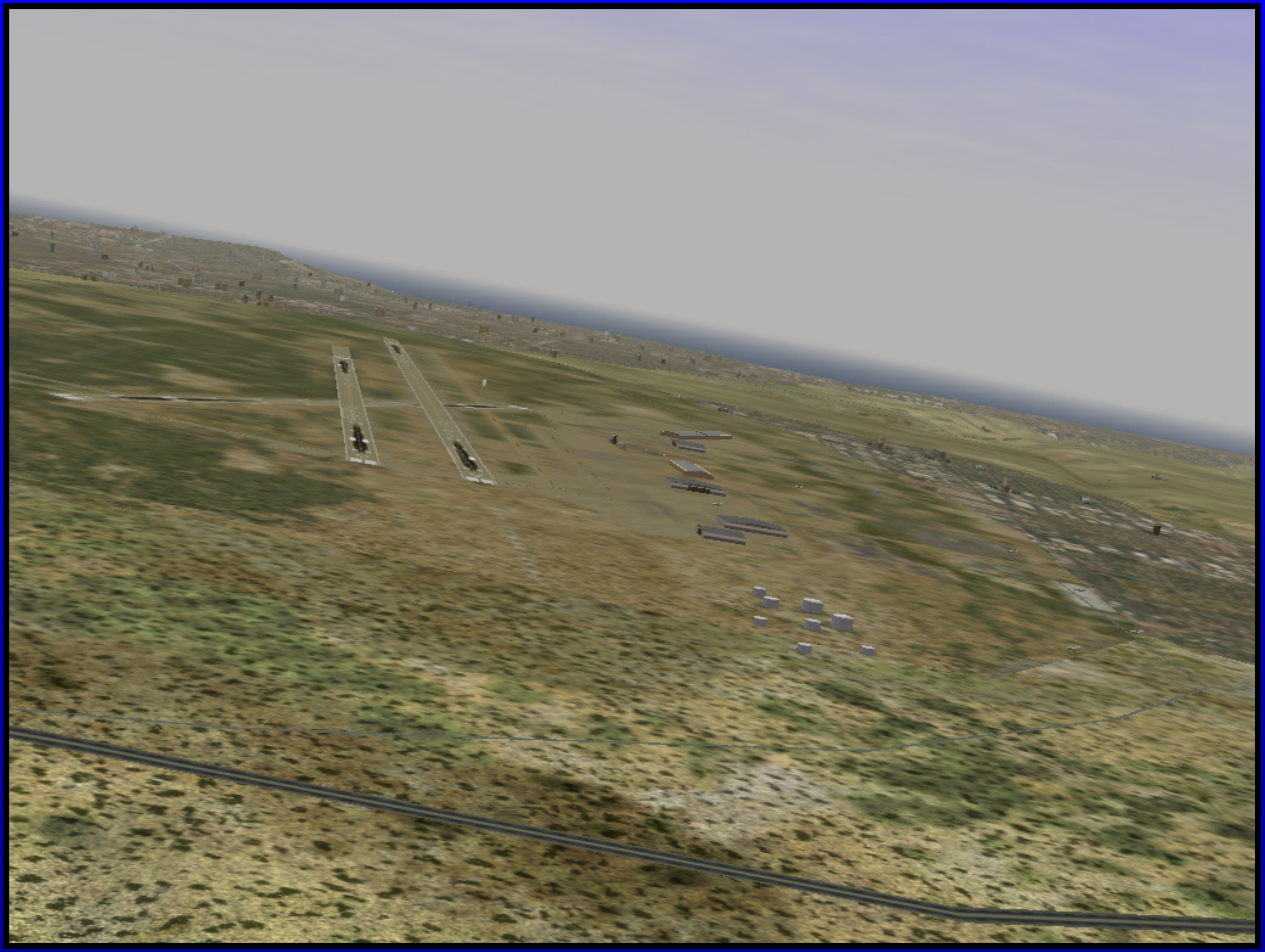
SHADER BASED - SHADOWS

Geo-specific Imagery

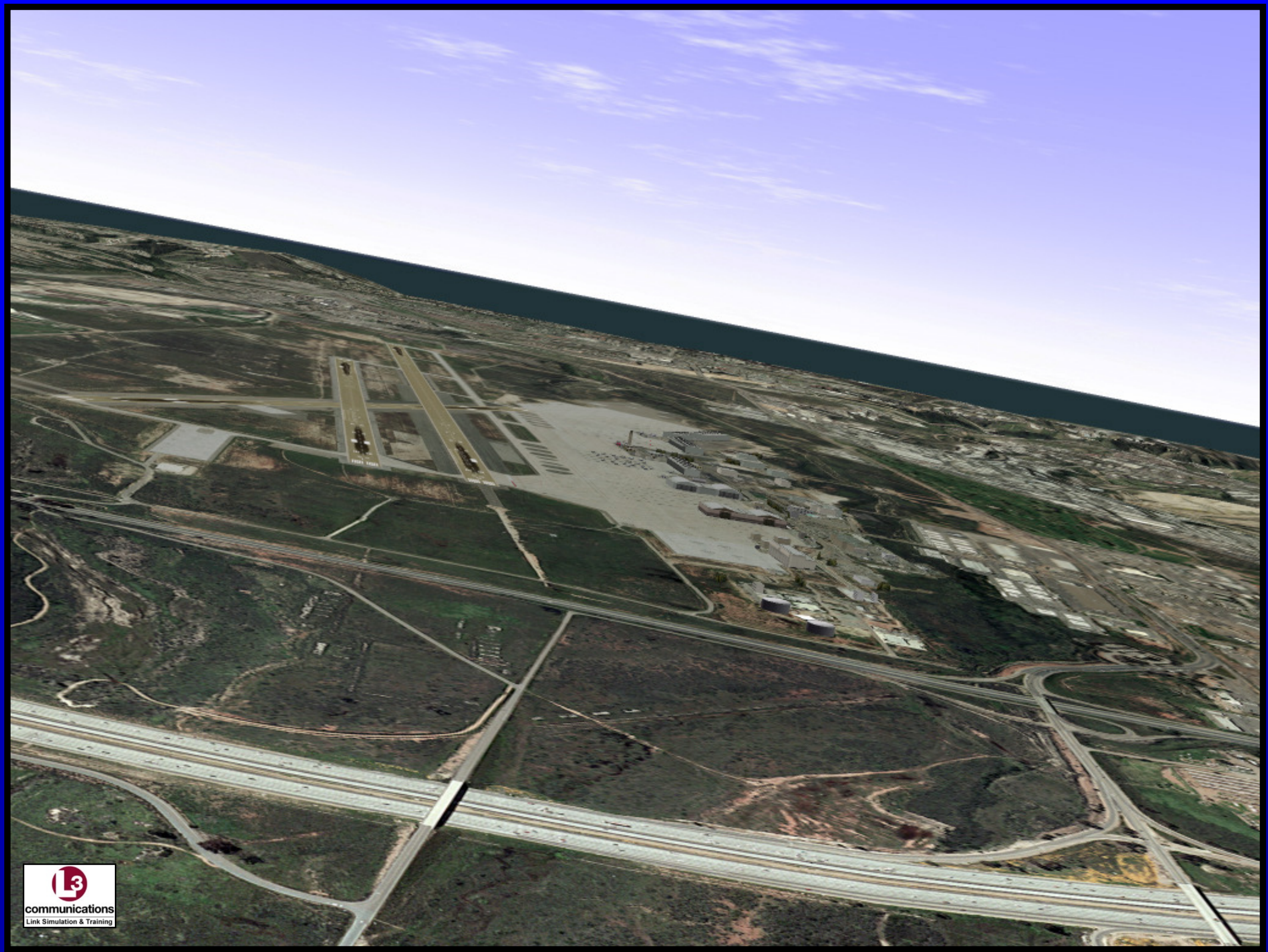
- Good news, bad news

EARLY APPLICATION OF GEO-SPECIFIC TEXTURE

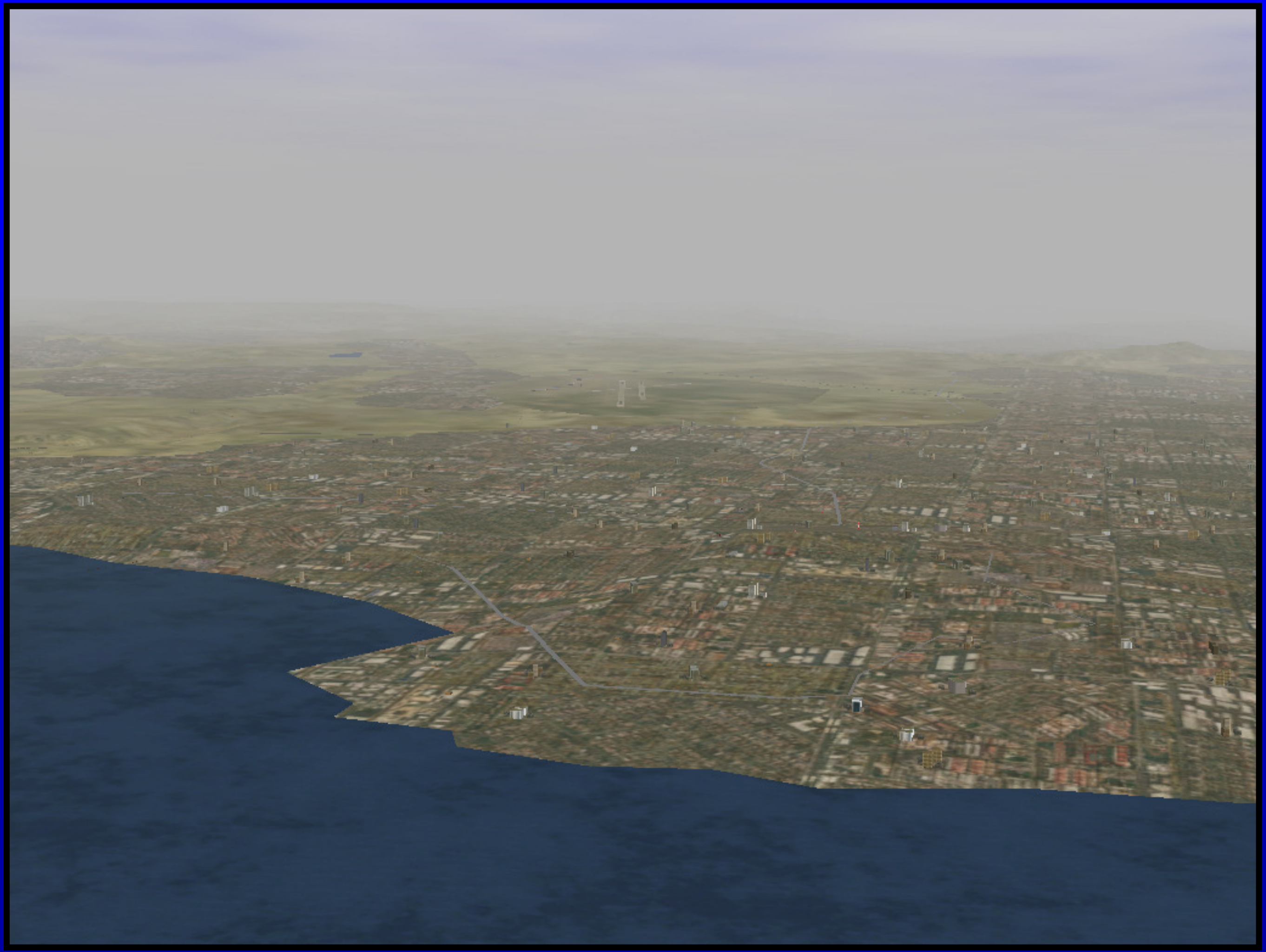
GEO-TYPICAL TEXTURE OF AIRFIELD APPROACH



GEO-SPECIFIC TEXTURE OF SAME APPROACH

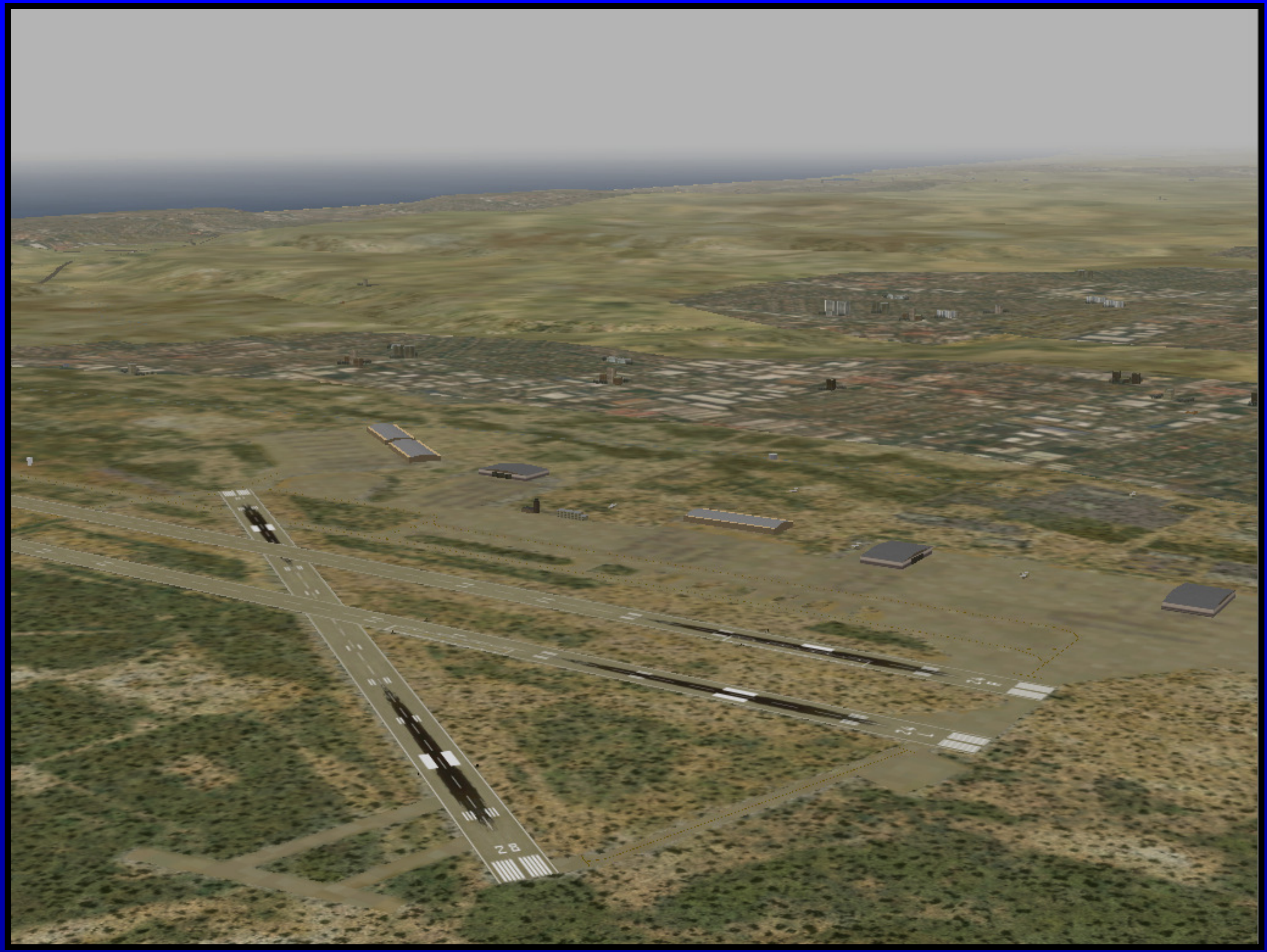


GEO-TYPICAL TEXTURE ON AIRFIELD APPROACH

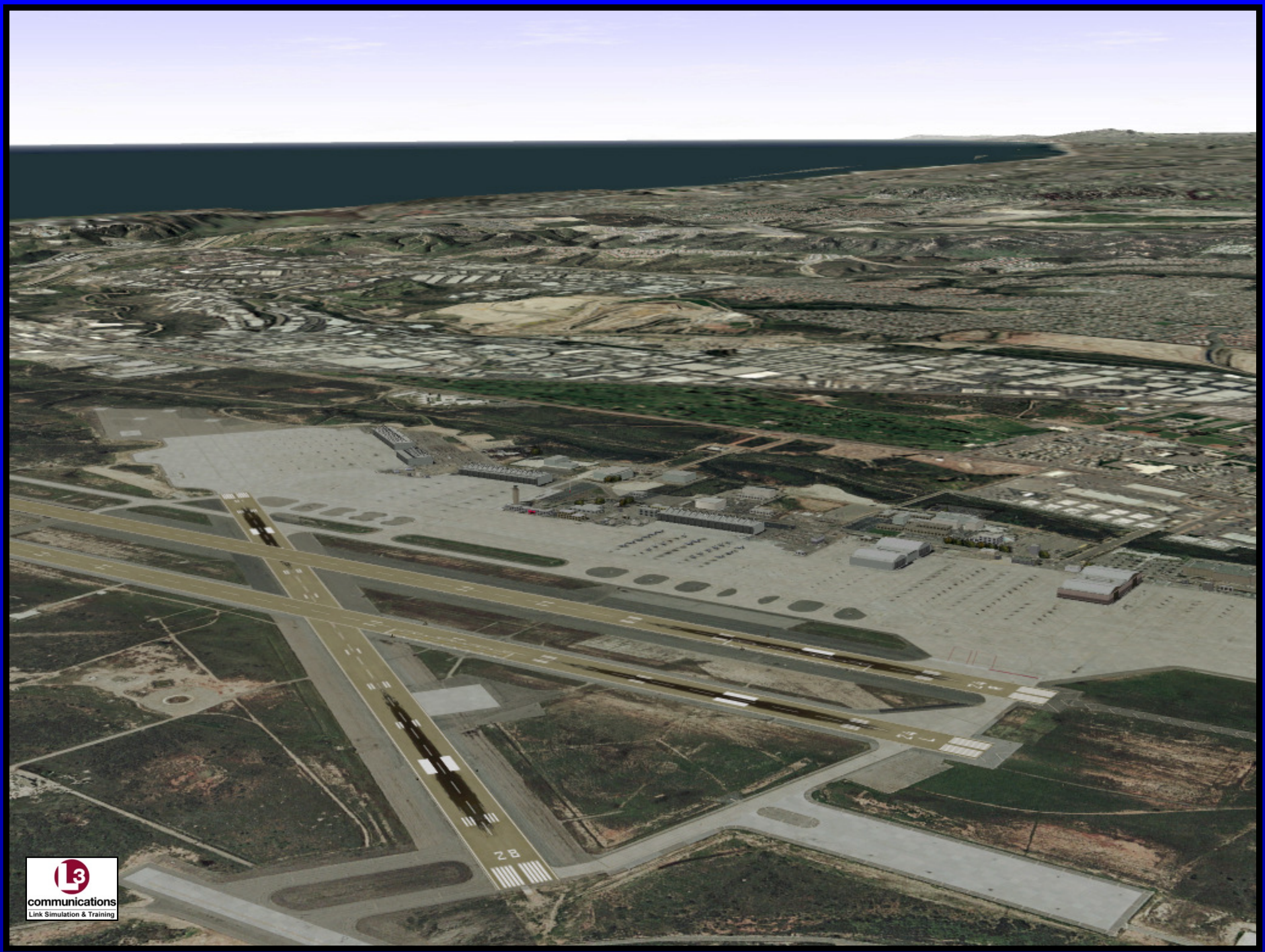


GEO-SPECIFIC TEXTURE OF SAME APPROACH

GEO-TYPICAL TEXTURE OF AIRFIELD



GEO-SPECIFIC TEXTURE OF SAME AIRFIELD



SATELLITE AND AERIAL IMAGERY CHARACTERISTICS

- Satellite imagery is a combination of monochromatic and color images
  - Monochromatic (panchromatic) is typically higher resolution
  - Color images are lower resolution
- "Best" resolution available for most of earth is a combination of:
  - .6 meter monochrome imagery merged with 2.4 meter color imagery
  - Typically captured at the same time
  - New imagery can be found for "Hot Spots" at sub-meter resolution
- Older imagery may not be as well matched
  - Larger differences in monochrome and color resolution
- Imagery in other bands also available – particularly IR
  - Can be evaluated for material encoding
- Aerial photography can be higher resolution - ≤ 1 foot
  - "Normal" color photography - not blended with monochrome
- In order to be used with typical tools imagery must be geo-referenced
  - Geo-tiff or similar file format
  - Includes position and size data – e.g. lat./long. of corner, number of texels

1 METER MONOCHROME (PANCROMATIC)

15 METER COLOR IMAGERY

“HIGH RESOLUTION” PAN SHARPENED

IMAGERY RESOLUTION ISSUES

- Resolution is the geographic size of each pixel/texel on the ground
  - For 1 meter imagery each pixel/texel is a “picture” of 1 meter of surface
- The “correct” resolution is very dependent on the typical altitude of the eyepoint above the terrain
  - Standard bi-linear blending will cause larger texels to blur when the eyepoint is too close
  - Micro-texture can help – very difficult to apply correctly
  - Using higher resolution than required adds to cost and storage
- Normal implementation is to use lower resolution in most areas
  - Higher resolution surrounding areas of interest – e.g. airfields, targets
  - Highest resolution at the area of interest – combined with 3D models
  - Blend across transitions
- Imagery is of very limited use for ground applications

15 METER IMAGERY FROM ALTITUDE

15 METER IMAGERY OF SAME AREA

5 METER IMAGERY OF SAME AREA

1 METER IMAGERY OF AIRFIELD

ONE METER WITH FIVE METER IMAGERY

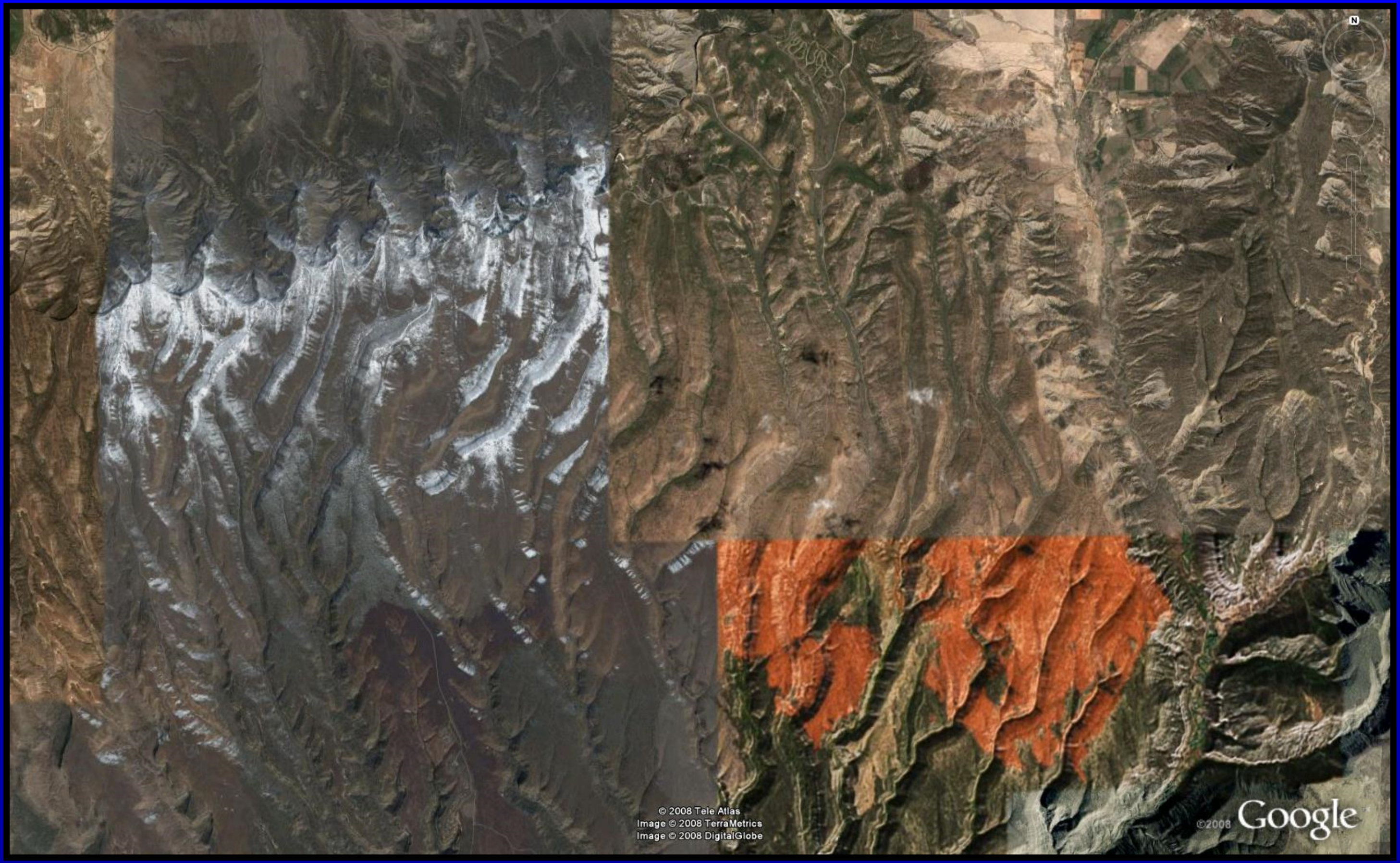
ONE METER BLENDED INTO FIVE METER

CONTENT ISSUES WITH SOURCE IMAGERY

- Color match
- Cloud cover

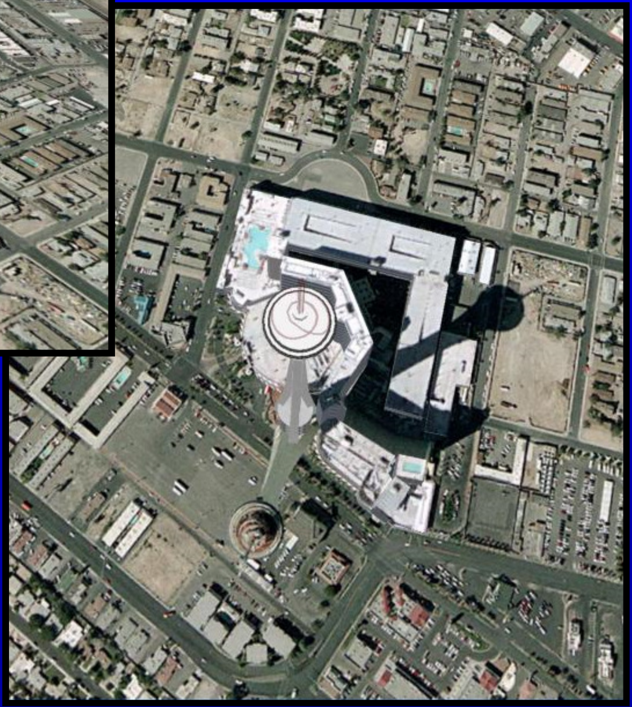
CONTENT ISSUES WITH SOURCE IMAGERY

- Color match
- Seasonal differences



DIFFICULTIES WITH IMAGERY

- Image of tall feature
  - Camera off nadir
  - Worse for tall features
  - Hard to remove
    - Fill in missing data



- Shadow of feature(s)
- Good for one time of day
- Conflicts with generated shadow
- Dusk, sensor issues
  - Shadow should be removed
- Labor intensive to remove

“Shaders”, the “Game” Changer

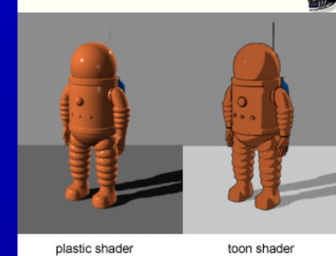
"SHADERS"

- The next evolution (revolution) in PC based graphics
  - Is having more impact than the addition of texture in the '80s
  - Shader programs are replacing original "built in" shaders
- Small programs that affect how the image is rendered
  - Operate on individual vertices, pixels/fragments or objects
  - Mathematical operations run once for each pass to add "special effects" in 3D space .
  - Control data can be passed into the shaders at run-time
- Important for shader effects to be the same in all PCs or channels
  - Want to see the same thing (e.g. explosion) from all PCs
  - Appear the same across image/channel boundaries
  - This can be difficult due to the "Dumbness" of most shaders
- They allows the GPU to have higher throughputs (number of triangles that can be processed per second).

Vertex Shaders

- Vertex shaders can manipulate properties such as position, color, and texture coordinate. they can also be used for lighting and shadow calculations.
    - A vertex shader program does not add new geometry to a 3D object , but can be used to transform existing vertices or animate them. Examples include :Transforms : translate , rotate and/or scale a object from its initial position to the another position in the scene. movement of plants due to windFlowing Water or dynamic terrain skins morphing or moving objects
- Projections : transform the vertices of all the 3D objects in a scene to a position that is relative to the camera view, perspective view, or isometric view.Properties : on-the-fly lighting, color, and shadow calculations. Used to create light points, sensor effects (e.g. Blooming in NVG)

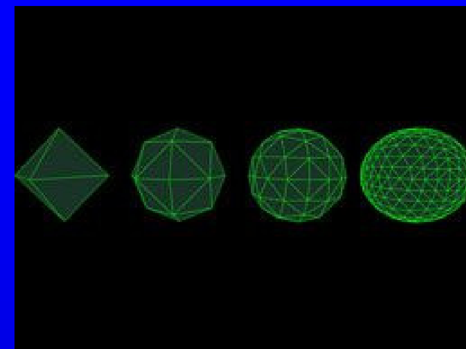


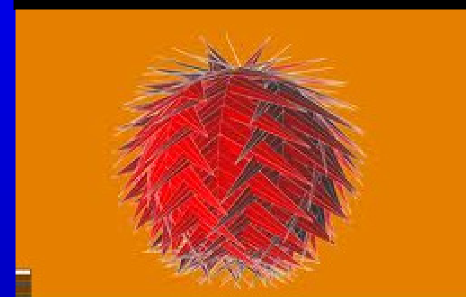



Geometry Shaders

- Geometry shades make it possible to create new geometry from existing geometry on the GPU. In most cases this geometry will be determined by a procedural algorithm.
- Examples include :
  - Grass : using a piece of “land” the shader generates triangles to simulate the blades.
  - Terrain: This shader can calculate the necessary geometry to simulate additional terrain fidelity not found in the source data.
  - Fur : creates too much strain for the classic 3D pipeline to make it look real. geometry shaders do a very convincing job with little noticeable effect on performance.
- When a lot of vertices are needed and they can be created from a procedural algorithm it is advantageous to use a geometry shader program.

- Geometry Shaders
  - Can create vertices and polygons to form objects
  - Allows a large amount of information to be passed into the GPU
  - These are somewhat new and not yet fully exploited
    - May by used to increase complexity in objects or terrain





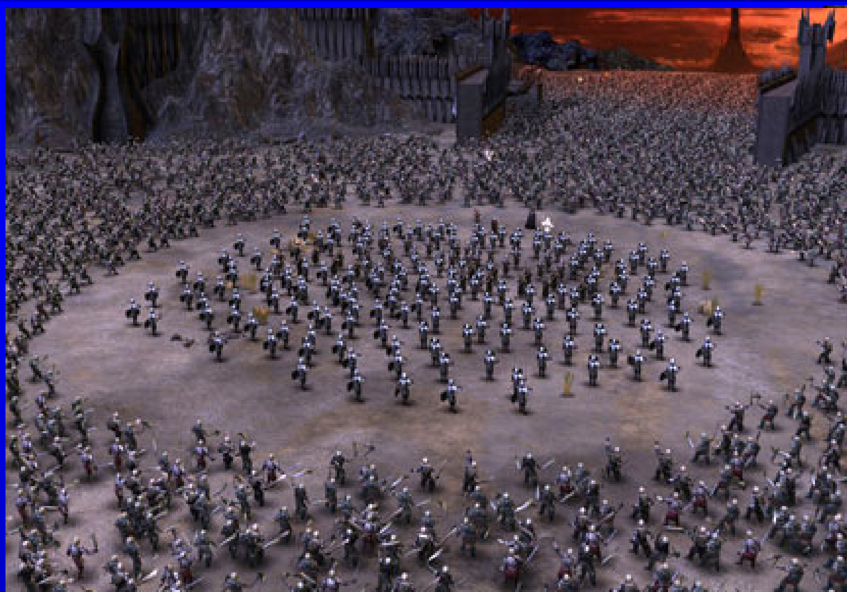
Pixel Shaders

- Also referred to as Fragment shaders
- Pixel shaders can Dynamically effect texture, shading, color, and z-depth but cannot create transparency, anti-aliasing, or depth tests.
- Examples include :
  - Simple texture switching (e.g. seasonality)
  - Normal mapping
  - Particle effects (e.g. dust trail)
  - Alpha blending
  - Advanced shading tricks (e.g. Phong-like shading)
- Pixels receive no information from their neighboring Pixels. This makes them fast and efficient but not very “smart”

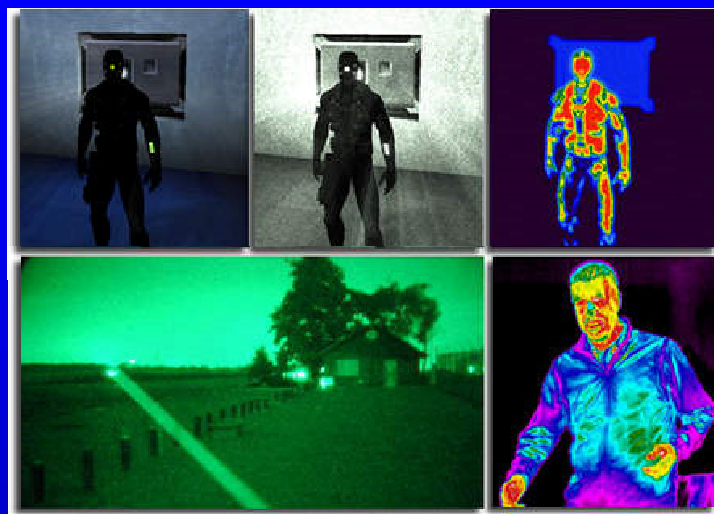
SHADER BASED - special effects

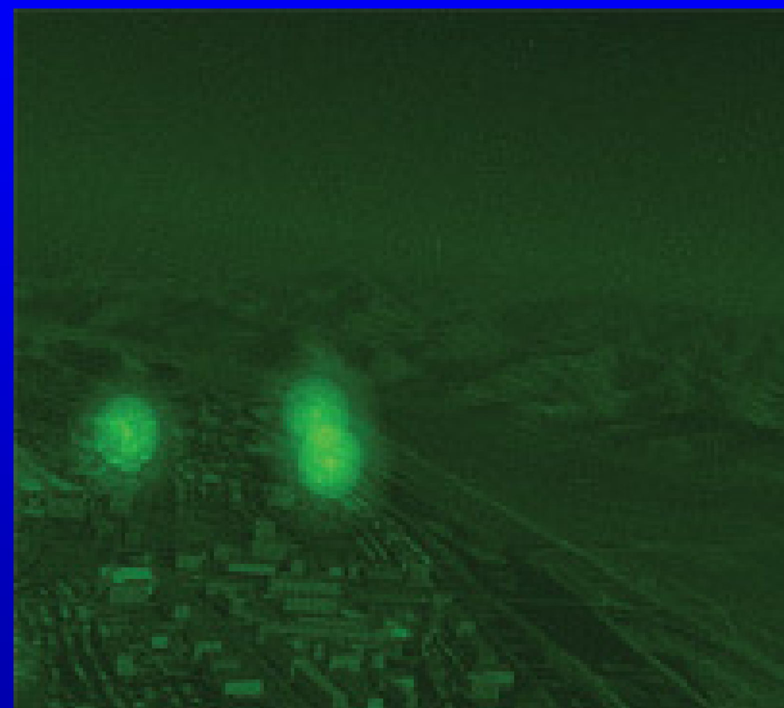
SHADER BASED - Customized Look and feel

SHADER BASED - Multiples Of The Same Model



SHADER BASED - Sensor effects





160

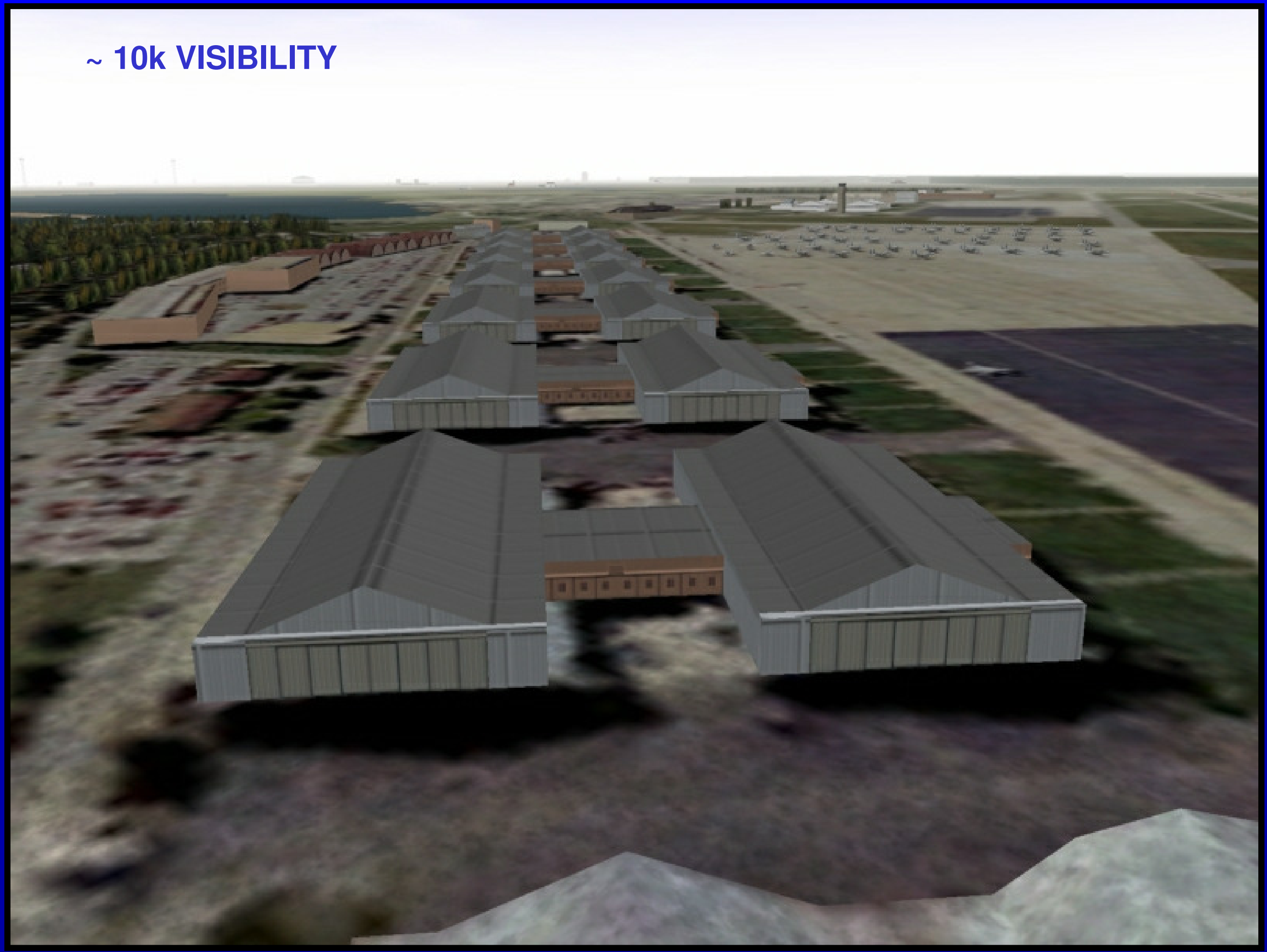
Special Effects

- Advanced Topics

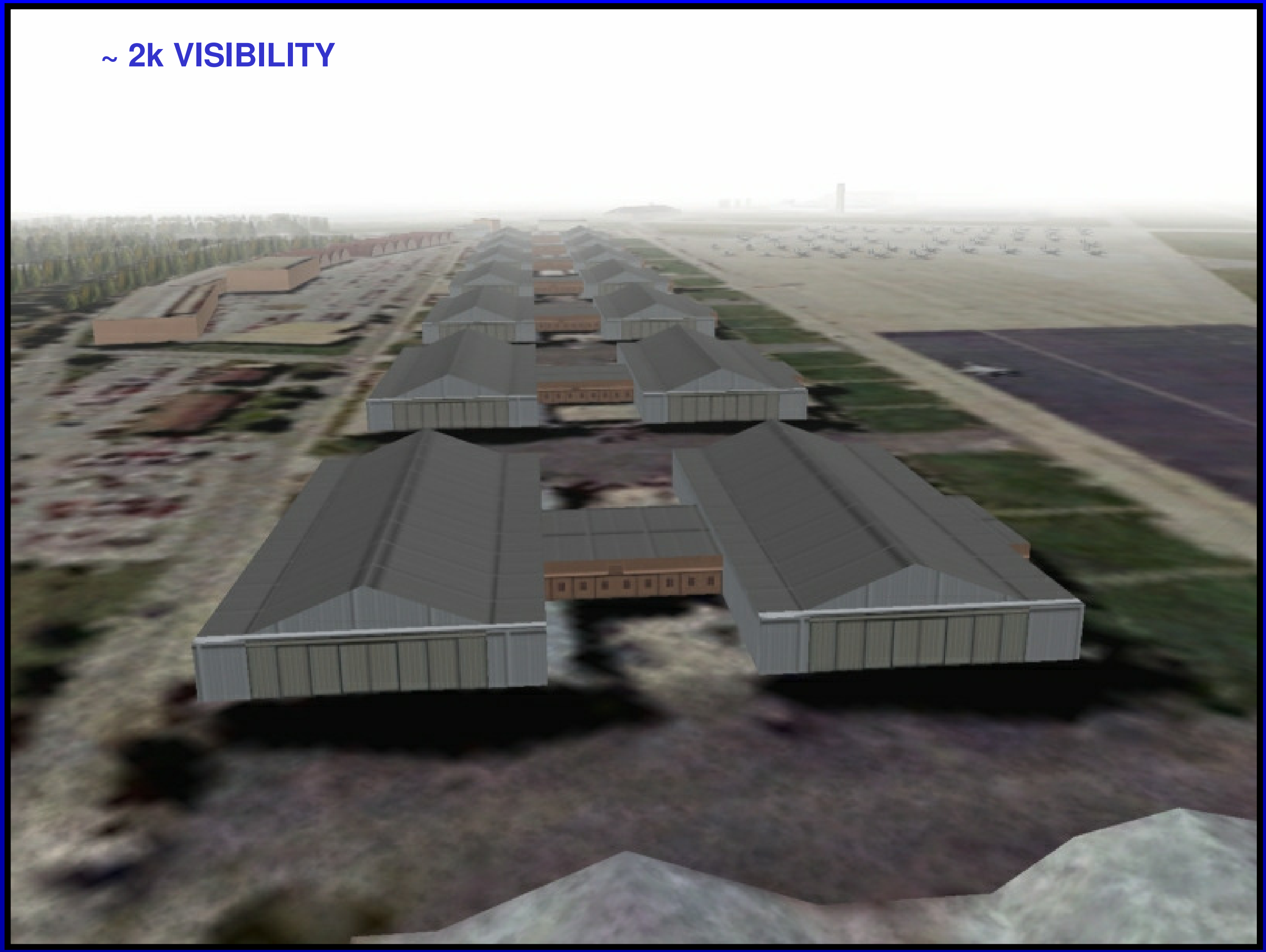
VISIBILITY EFFECTS

- Fog created by image generator
  - No direct interaction with Database
  - Visibility function does not delete polygons or lights
  - Database content and/or far z-clip can be managed to match visibility range
- Combines fog color with pixel color
  - Utilizes range (z-depth) of each pixel
  - Fog color may be changed in real-time
- Light points, luminous polygons
  - Should have separate fog function
  - Must be visible at greater distance
- Fog function should look right rather than be right
  - Flexibility, tuneability important

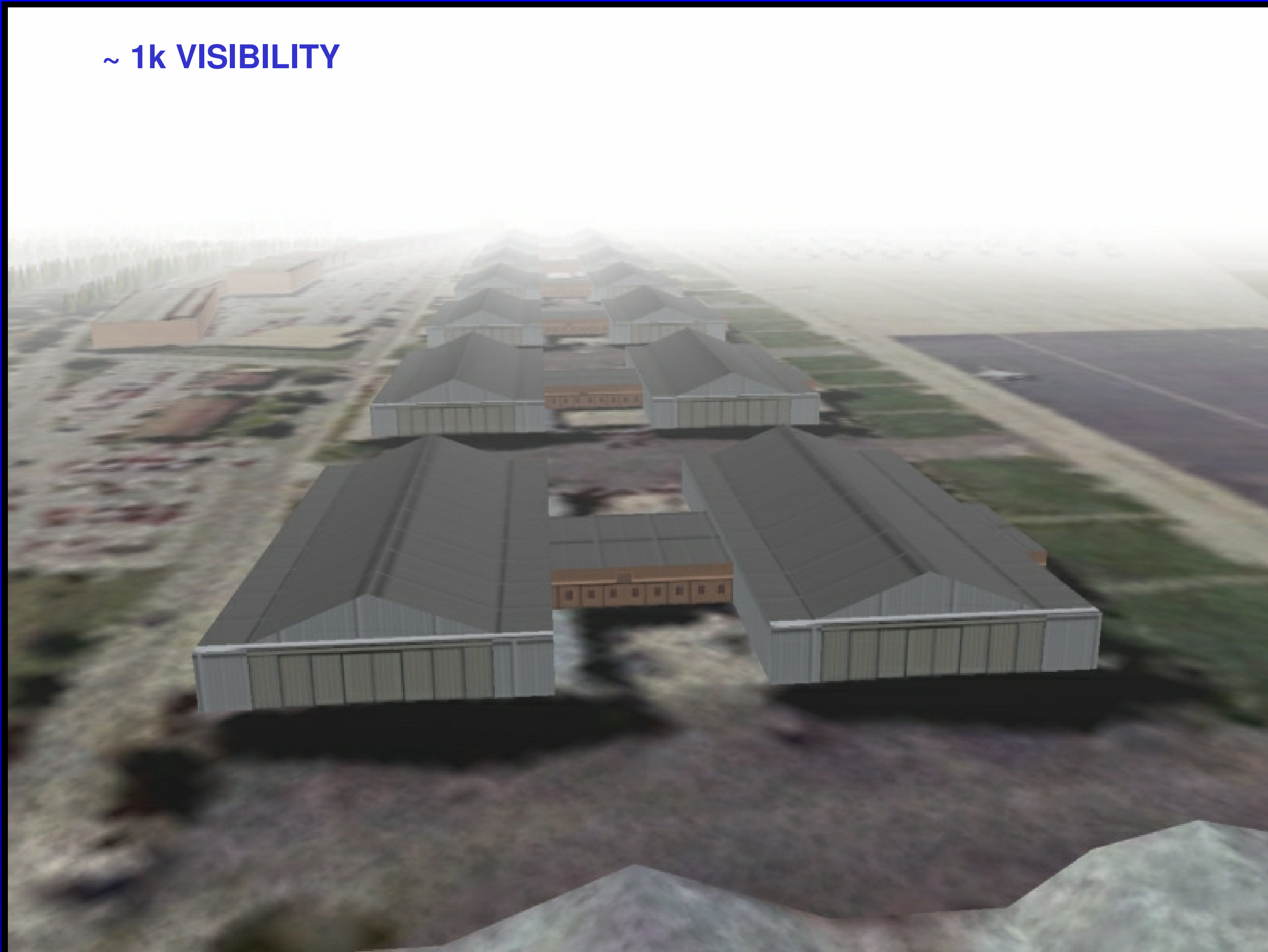
VISIBILITY (FOG) EXAMPLES



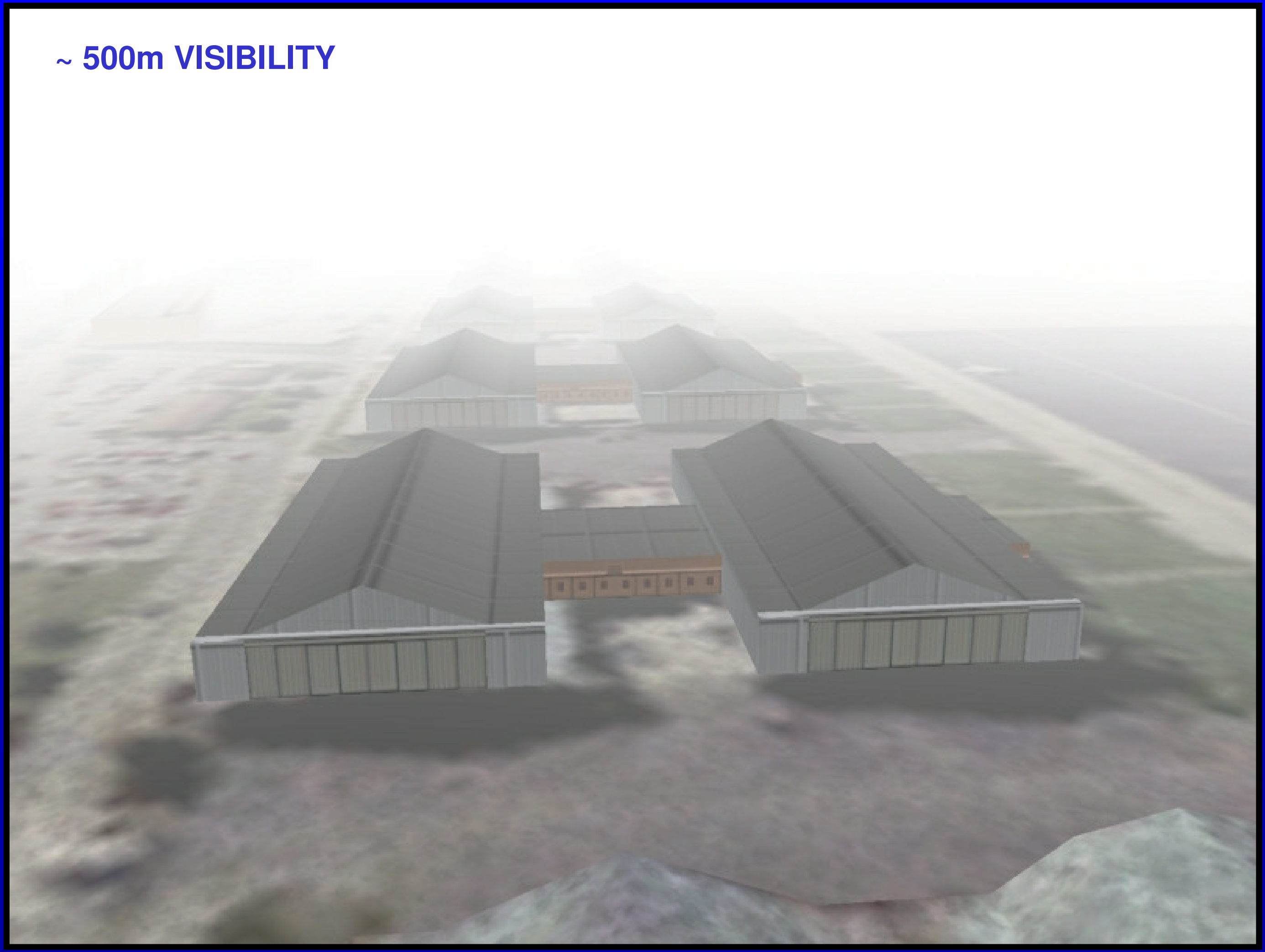
VISIBILITY (FOG) EXAMPLES



VISIBILITY (FOG) EXAMPLES



VISIBILITY (FOG) EXAMPLES



VISIBILITY (FOG) EXAMPLES

SKY AND HORIZON "The Sky Box"

ANIMATIONS

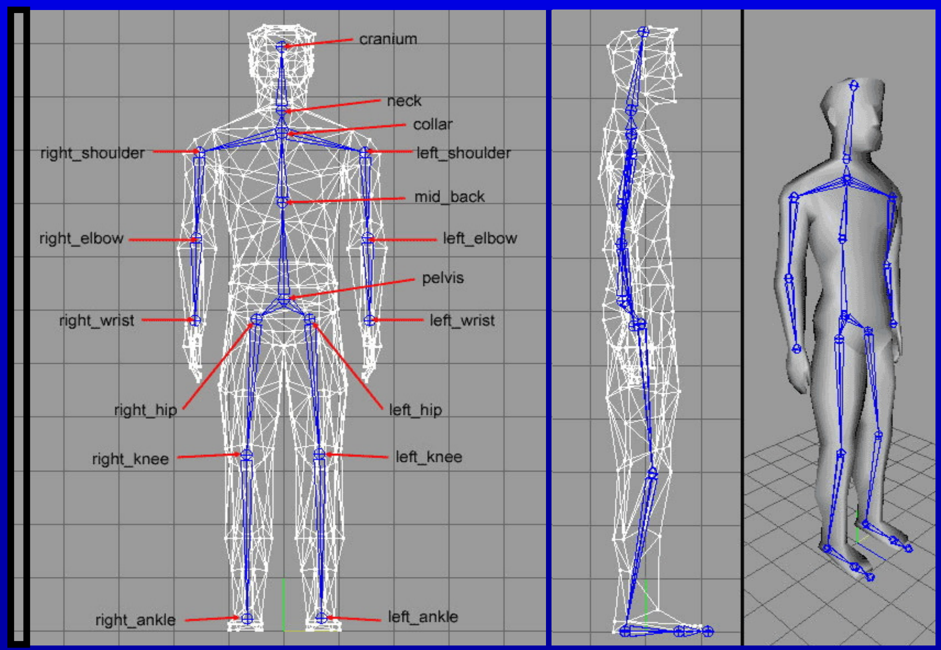
- "Flip Book" Animations
  - Cycles through multiple versions of feature to create apparent motion or activity
  - Often positioned as a dynamic model
  - Can cycle once or repeat until turned off from host
  - Host can control speed and direction of animation
  - Individual cells may be selectable

EXAMPLES OF ANIMATIONS - CONTRAILS

- Segments of contrail created in real-time
  - Polygon geometry based on flight path of aircraft
  - Time between samples and total number active is programmable
  - Polygons at far end fade out
- Can also be used for ship wakes and dust clouds behind vehicles
- Requires real-time processing to create polygons and manage them
  - Polygons and pixels are a capacity considerations
- Typical of animations that are now done in vertex shaders
  - Geometry shaders next?

ANIMATIONS

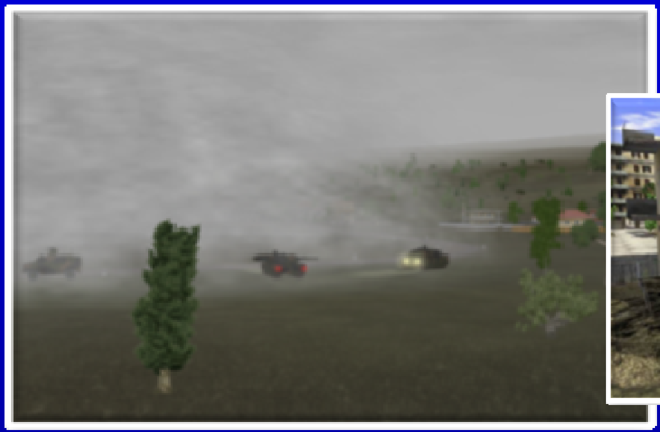
- “Skeletal” animation
  - Each limb or joint is controlled
  - Works well with motion capture
  - Same skeleton and animation can be used with multiple “skins”
  - Much more control, flexibility
  - Also smoother transitions between animations



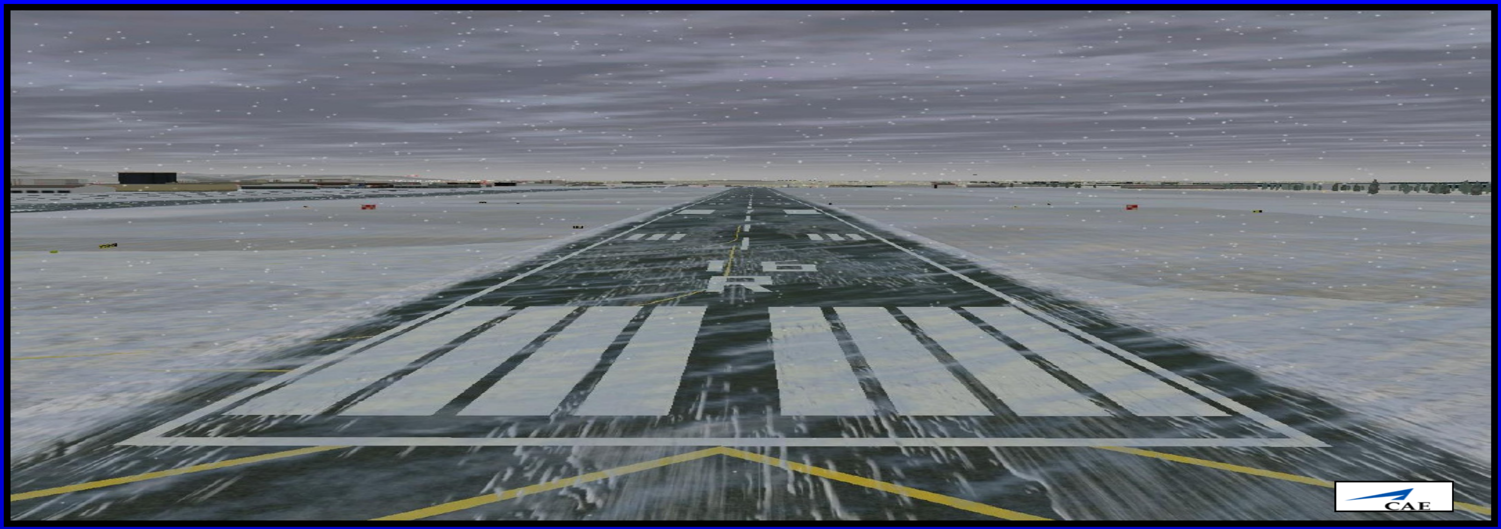
EXAMPLES OF SKELETAL ANIMATION

BATTLEFIELD EFFECTS

- Smoke Screens and Dust Clouds
- Smoke Signals
- Field Artillery, Mines, IEDs
- Buildings and Bridges
  - Hit Marks and Damage
    - For small caliber impact marks displayed



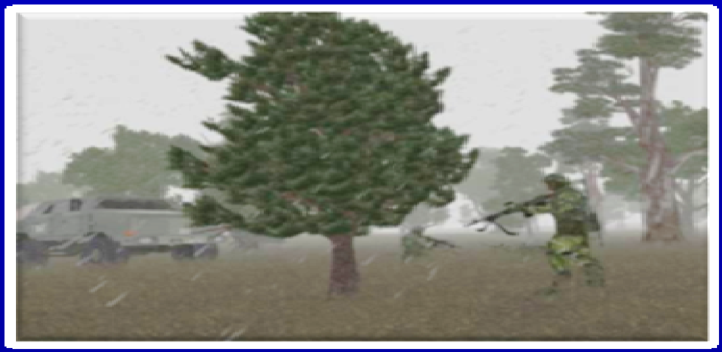
WEATHER EFFECTS



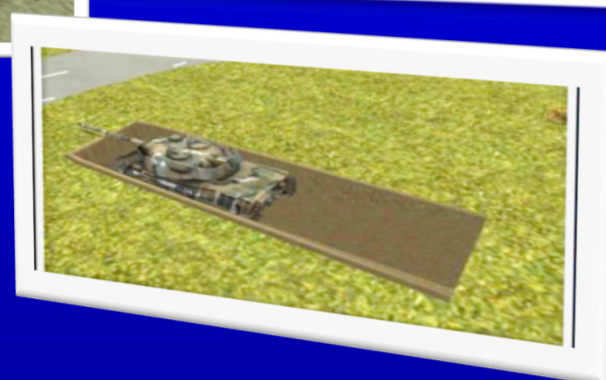
- Weather effects are combination of fog color, the Sky Box, and animations

ATMOSPHERE EFFECTS

- Precipitation (Rain, Snow)
- Global Fog/Haze
- Moon, Stars, Sun, and Glare
- Verifying Cloud Coverage
- Wind Direction & Strength Affects:
  - Dust and smoke
  - Cloud Movement
  - Vegetation Movement
    - Trees and grass sway in the wind
    - Trees fall when "damaged"



DYNAMIC TERRAIN



- The debris of destroyed houses will block roads
- Trenches and ditches will provide protection to infantry troops
- The craters of artillery shells will hide infantry groups
- Tracked vehicles are laying tracks on the ground
- Fences, tank barriers, barrels etc. can be placed by the instructor as dynamic object
- Trees which can be run down will form obstacles for wheeled vehicles

DYNAMIC WATER & AMPHIBIOUS EFFECTS

Visualizing a highly detailed surface such as water in computer graphics requires a large number of very small polygons.

- Tessellation
  - Uses modern GPUs to generate required polygons
  - Runtime Calculated LOD transitions
- Shading
  - Reflection and refraction simulated using an approximate Fresnel factor where the Fresnel term depends on the surface normal (N) and the vector between the camera and the current pixel (V).

NIGHT VISION GOGGLE STIMULATION

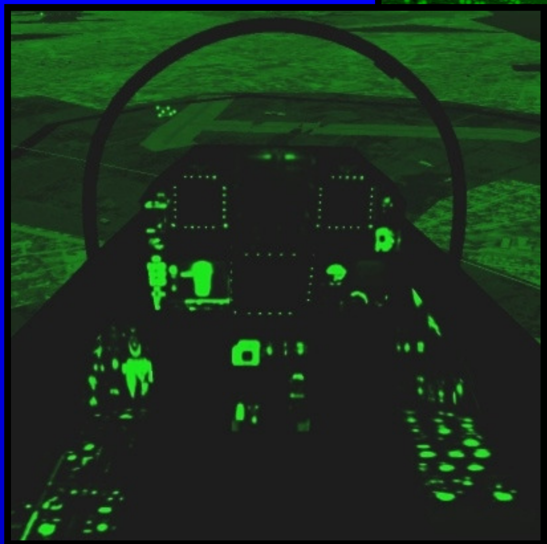
- Stimulates actual goggles
  - Pilot can use personal helmet
  - No simulator specific hardware
- Goggles react to displayed image
  - Display black levels important
  - Lack of dynamic range a problem
  - Different problems with LCD & LCOS
    - May require mechanical filters to achieve black level
    - Much larger dynamic range

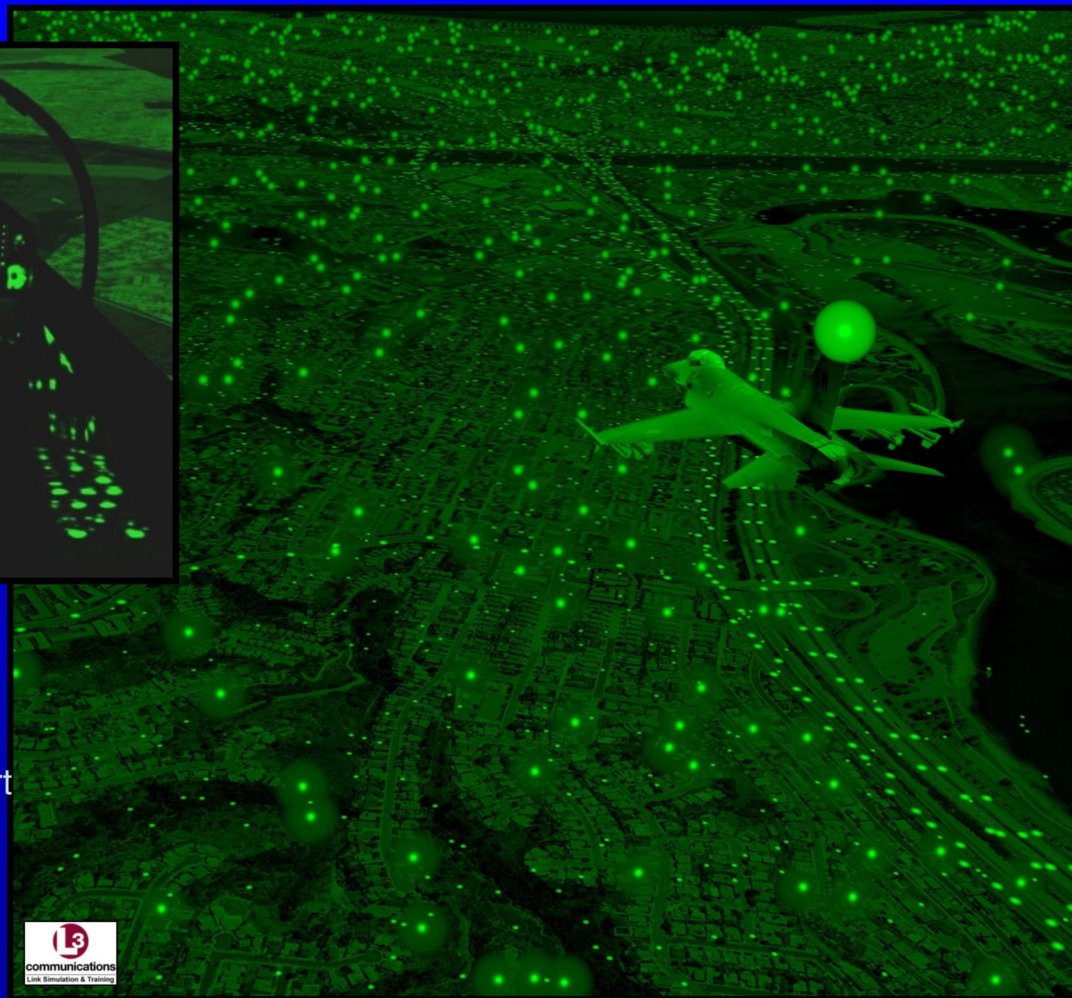
- May require color and/or gamma adjustments to the visual scene
  - Goggles more sensitive to red
  - Can be an issue if one crew will view the night scene without goggles
  - Or crew expects to look under goggles to see outside scene
- Correlation not a problem
  - Typically uses version of visual Database

NIGHT VISION GOGGLE SIMULATION

- HMD that looks and feels like NVG's
  - Should fit on pilot's personal helmet
  - Video, head tracker cables a problem
- May require additional IG hardware
  - For simultaneous out the window scene
- All NVG effects must be simulated as part of scene rendering
  - Blooming of bright lights, weapons effects
  - May require post processor
  - Another shader application
- Cockpit issues
  - Mask of cockpit structure must be modeled
  - Instruments will not appear in simulation
    - Normally an out of focus blur
    - Can be included as part of mask
- Correlation
  - Also uses version of normal visual Database

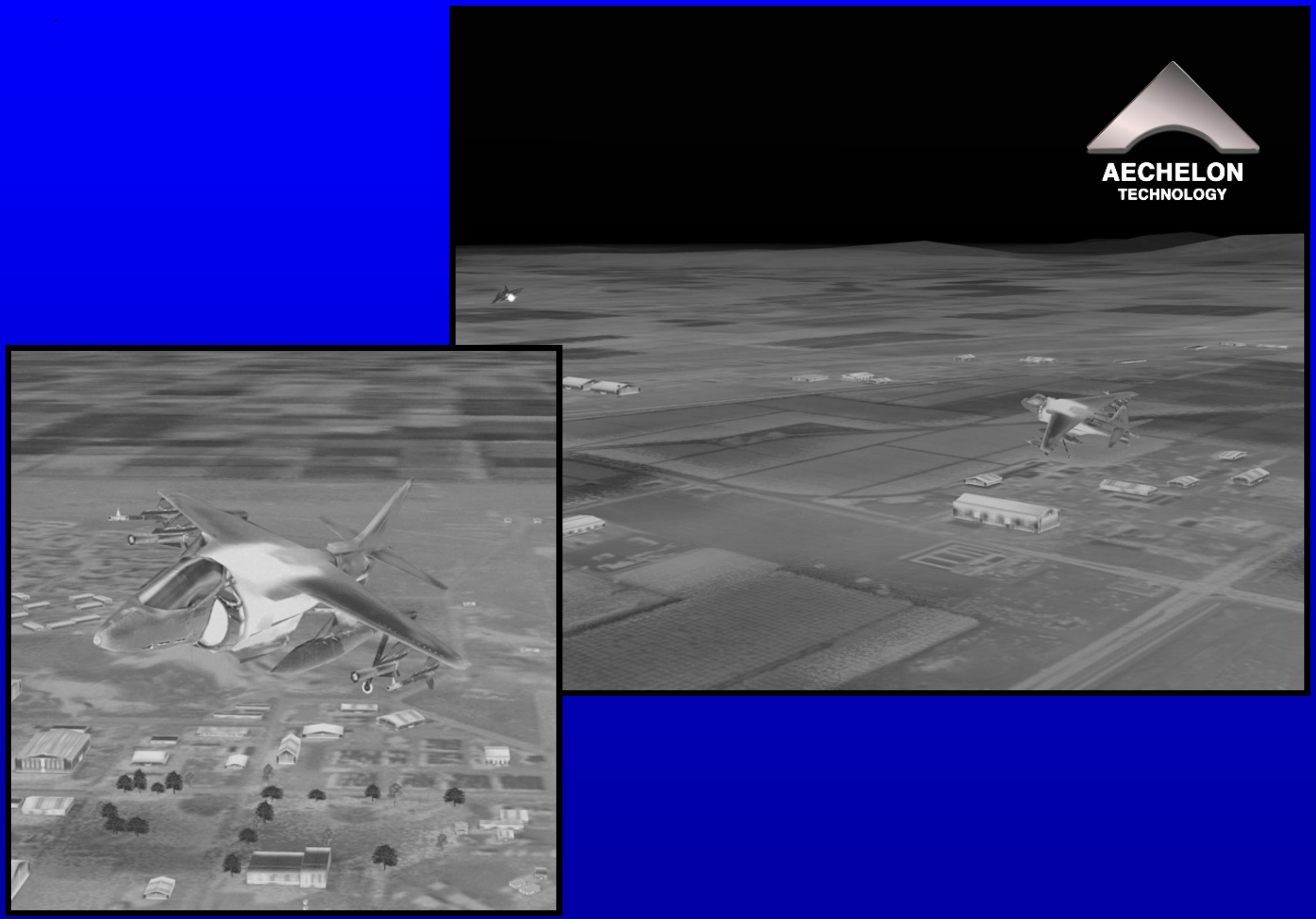
NIGHT VISION GOGGLE SIMULATION



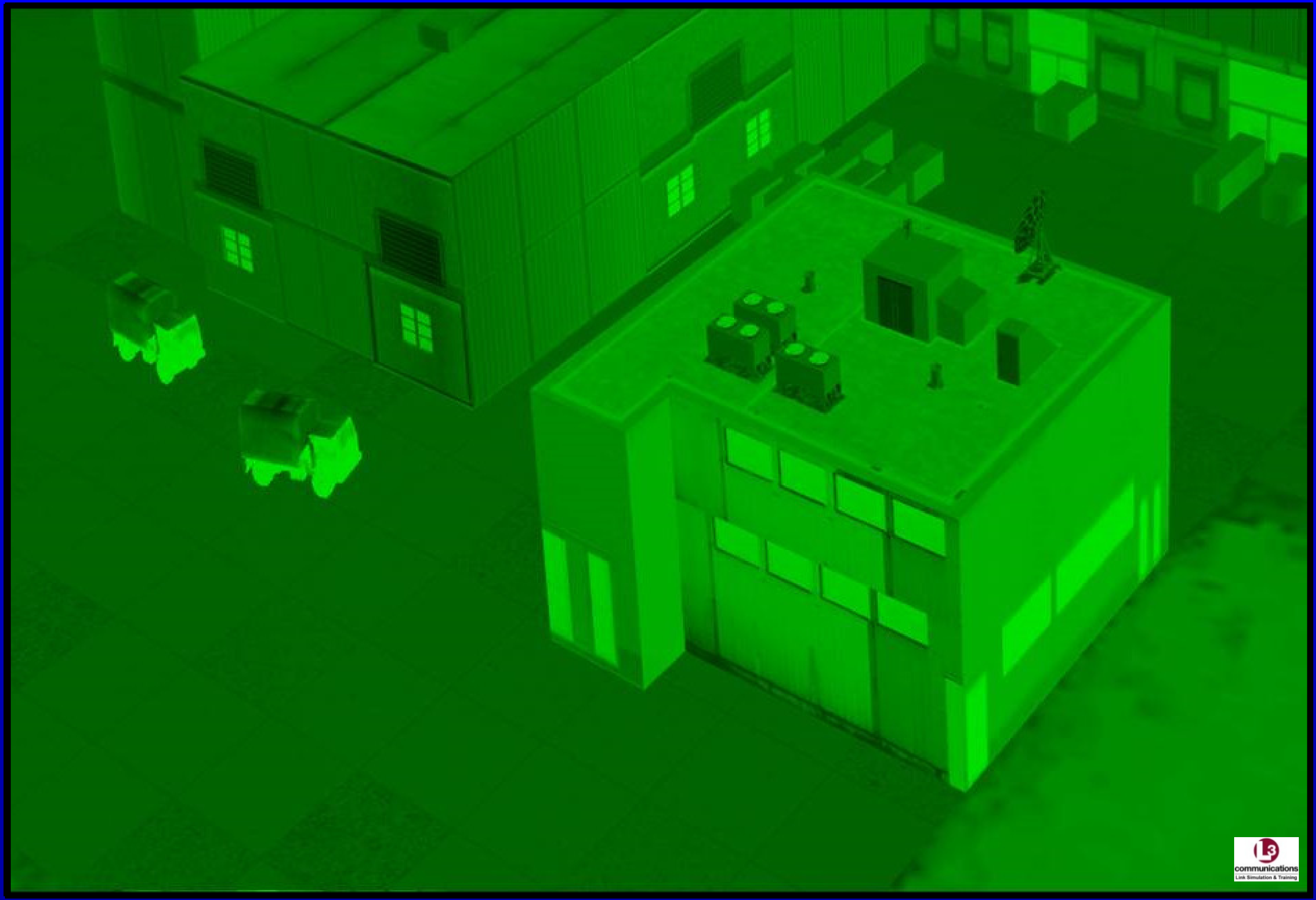


In cockpit effects must also be provided as part of simulated NVG image

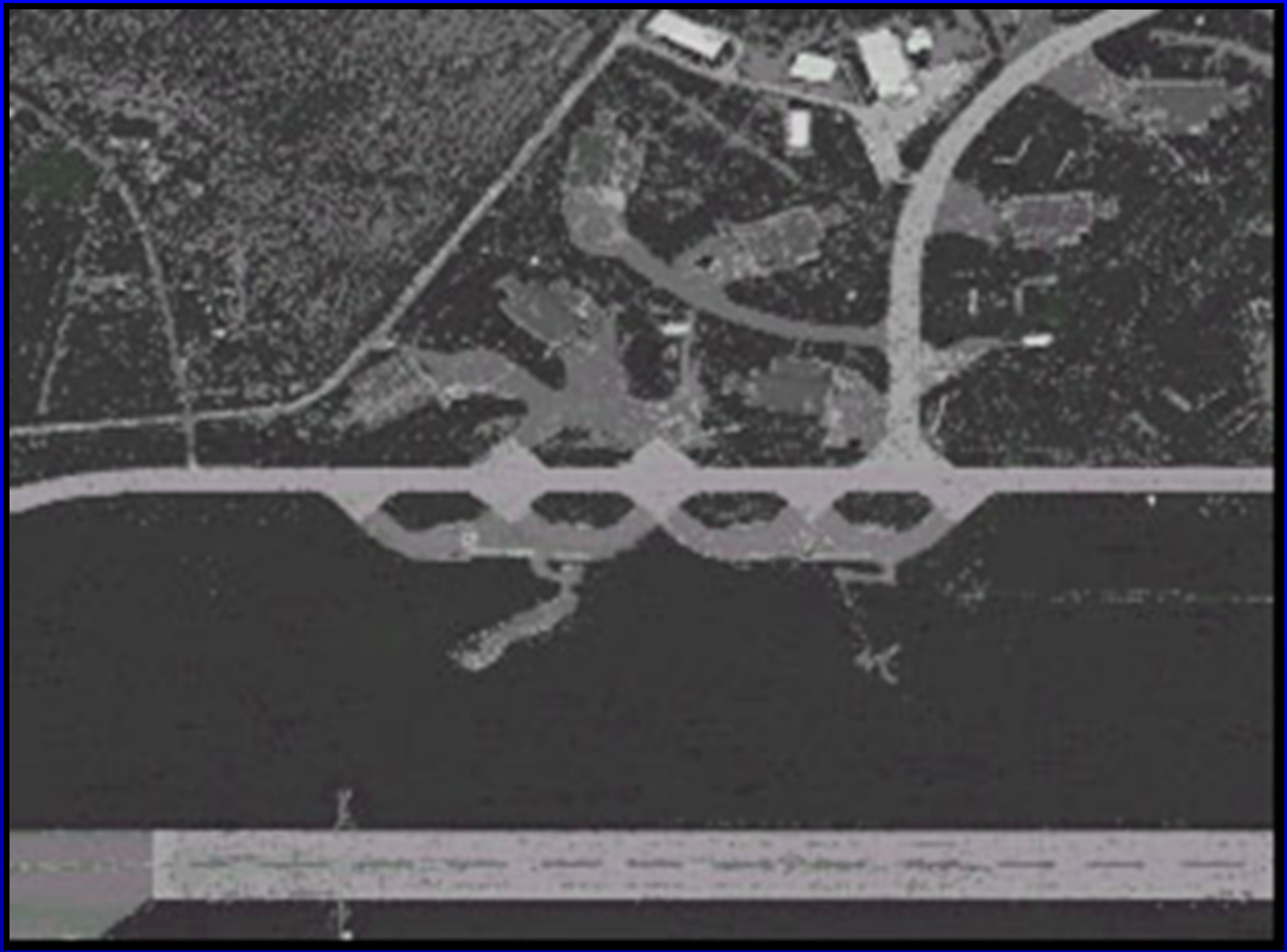
INFRARED SENSOR EFFECTS



CORRELATED SENSORS – IR TEXTURE



INFRARED CROSSOVER EFFECT

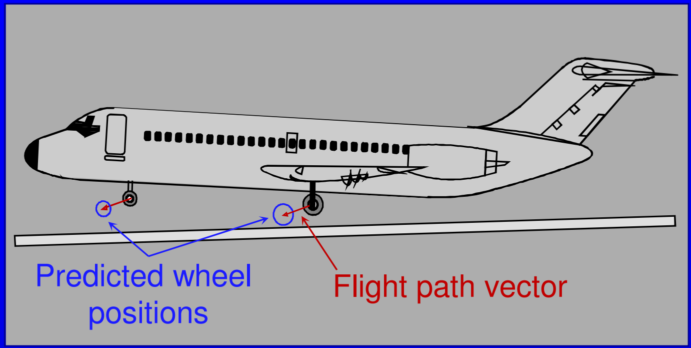


Hidden Talents

- Things the IG is good at that you don’t “see”

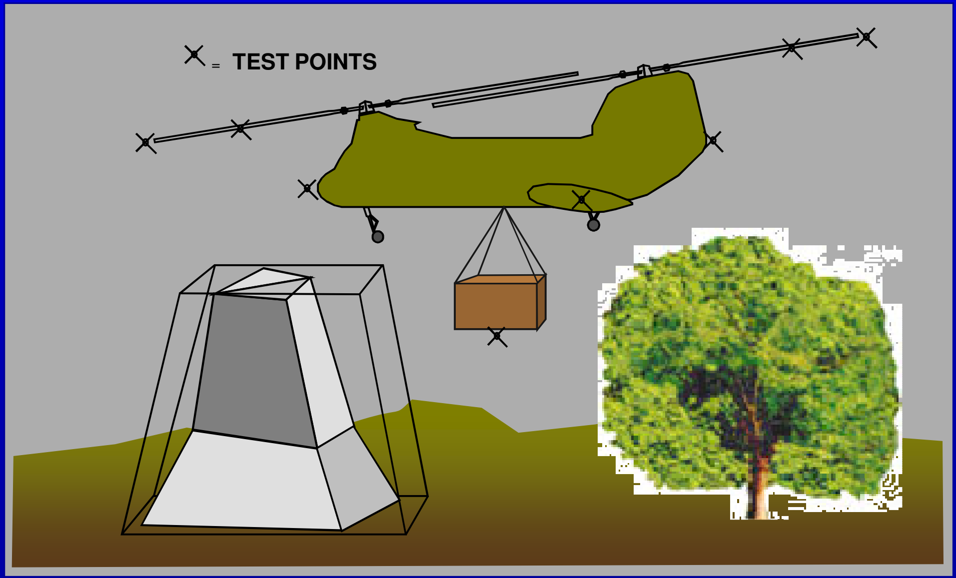
MISSION FUNCTIONS

- Height Above Terrain (HAT)
  - Calculates height of <u>test points</u> above terrain polygons
    - May provide plane equation, surface type
    - Results sent to host
    - Supports landing on non-flat surfaces, radar altimeter
  - Also used for ground clamping of moving models



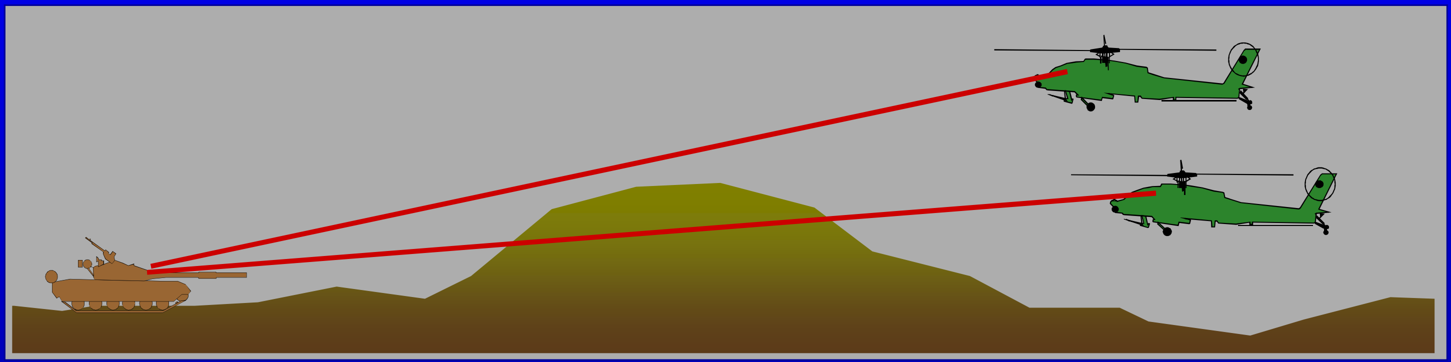
MISSION FUNCTIONS

- Collision detection
- Test points (or vectors) approximate ownship volume
- Tested against Database collision volumes or bounding volumes
    - If positive, tested against individual polygons
    - If positive, collision and polygon material code sent to host
- IG should only calculate collision for ownship features
    - Host decides to initiate a crash, or other failure



MISSION FUNCTIONS

- Line of sight, “laser range finder”
- What can be seen?
- Important for synthetic forces, DIS/HLA network applications
- Difficult for image generator to calculate for other vehicles or eyepoints
    - Database area, features to be tested may not be in active Database



- Mission Functions sometimes referred to as ISECT
- Most of these functions are intersection tests
- Often performed in a separate computer from the render process

DYNAMIC COORDINATE SYSTEMS (DCS)

- Independently moving objects
  - Positioned relative to origin
  - Positioned relative to another DCS
    - Chained coordinates
  - Positioned relative to the eyepoint
- Stamps (or Sprites)
  - Turn to face the eyepoint
  - Can be positioned as a model
  - Can include an animation (e.g. Explosion)
- Controlled through or by the host
  - Host programs or scripts
  - Other simulator(s), DIS/HLA network
- Controlled from visual real-time
  - Programs or scripts
  - Path contained in Database
    - Routed/conflicting traffic

DYNAMIC COORDINATE SYSTEMS

DYNAMIC COORDINATE SYSTEMS

- Tanker (lat,long,alt,h,p,r)
- ~5 hose segments (h,p,r)
- Basket (h,p,r)

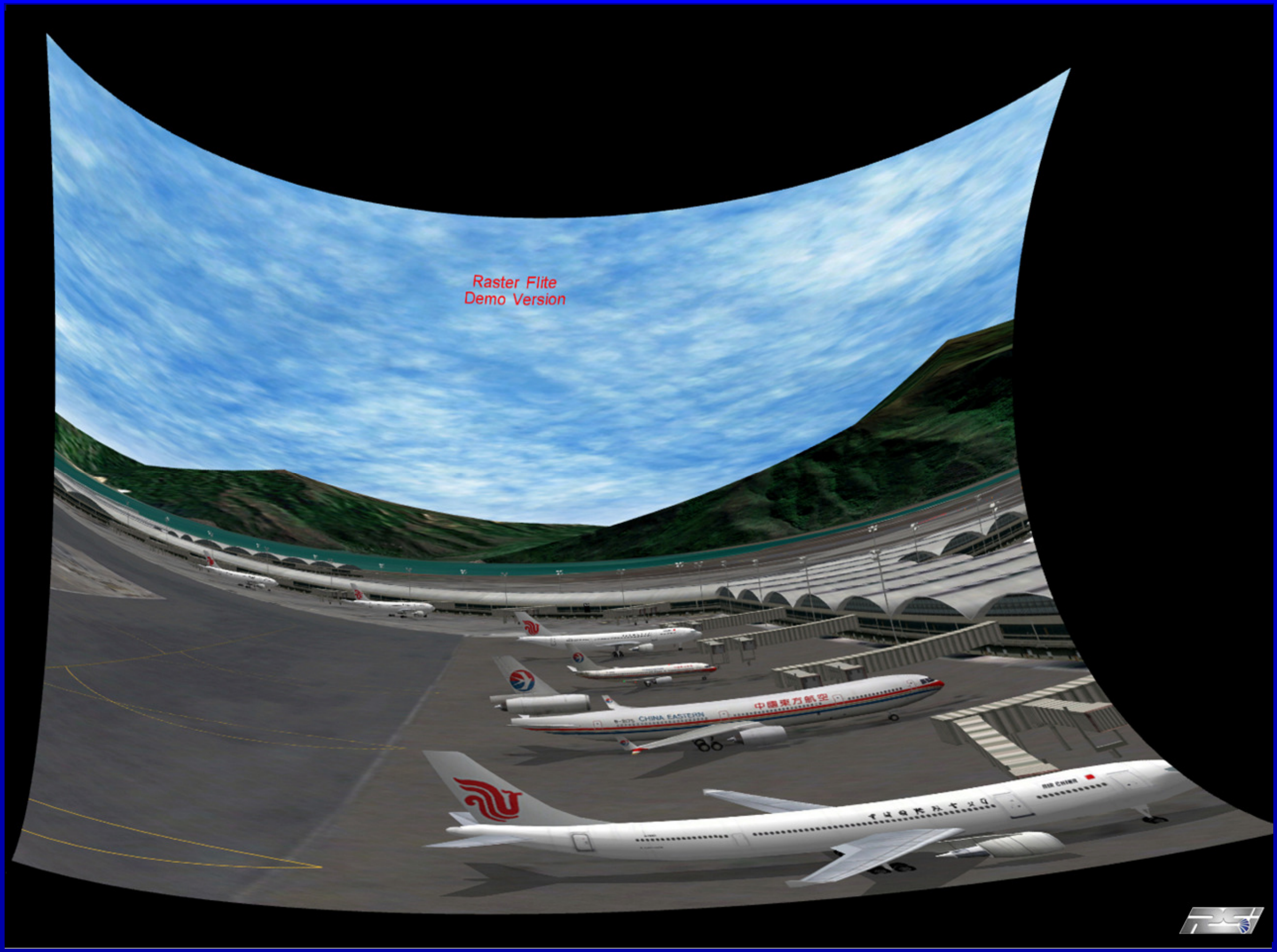
DYNAMIC COORDINATE SYSTEMS

DISPLAY INTERFACE – DISTORTION, BLENDING

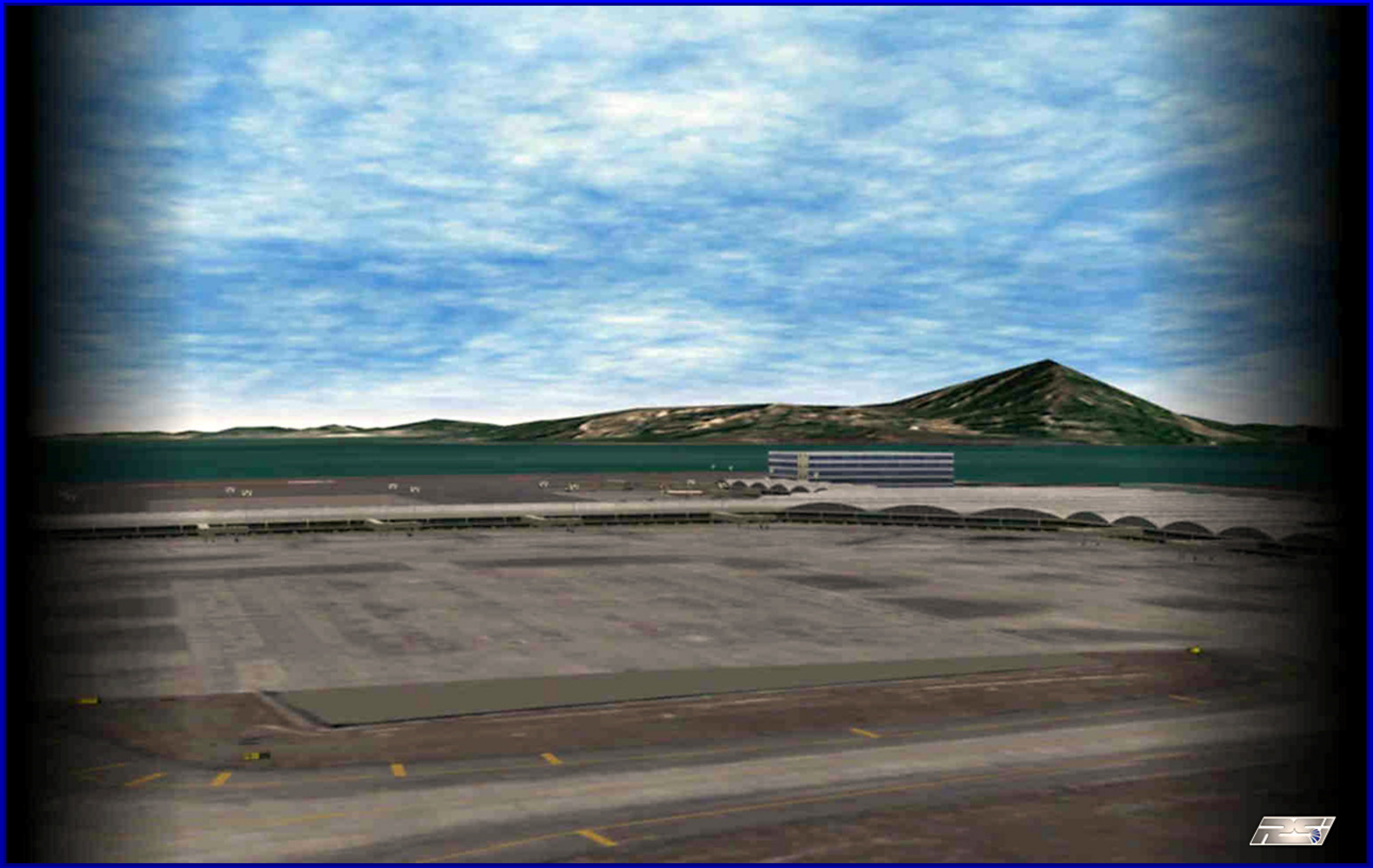
- Distortion - IG normally renders a rectangular, flat image
  - Curved screens require that image be pre-distorted
    - Previously performed in IG – impacted scene content
  - More recently handled in projector electronics and/or lens
    - Or scene processor between the IG and the projector
  - New displays limit amount of distortion possible
    - May have to be done in the IG – again
    - Adds to the draw time – amount depends degree of distortion

- Blending between channels
  - Also previously done in the projectors
  - Edges of image rendered with graduated transparency
    - Requires overlapped channel boundaries – loss of available image
    - Resolution in blend area may be reduced due to pixel misalignment

DISPLAY INTERFACE – DISTORTION, BLENDING



DISPLAY INTERFACE – DISTORTION, BLENDING



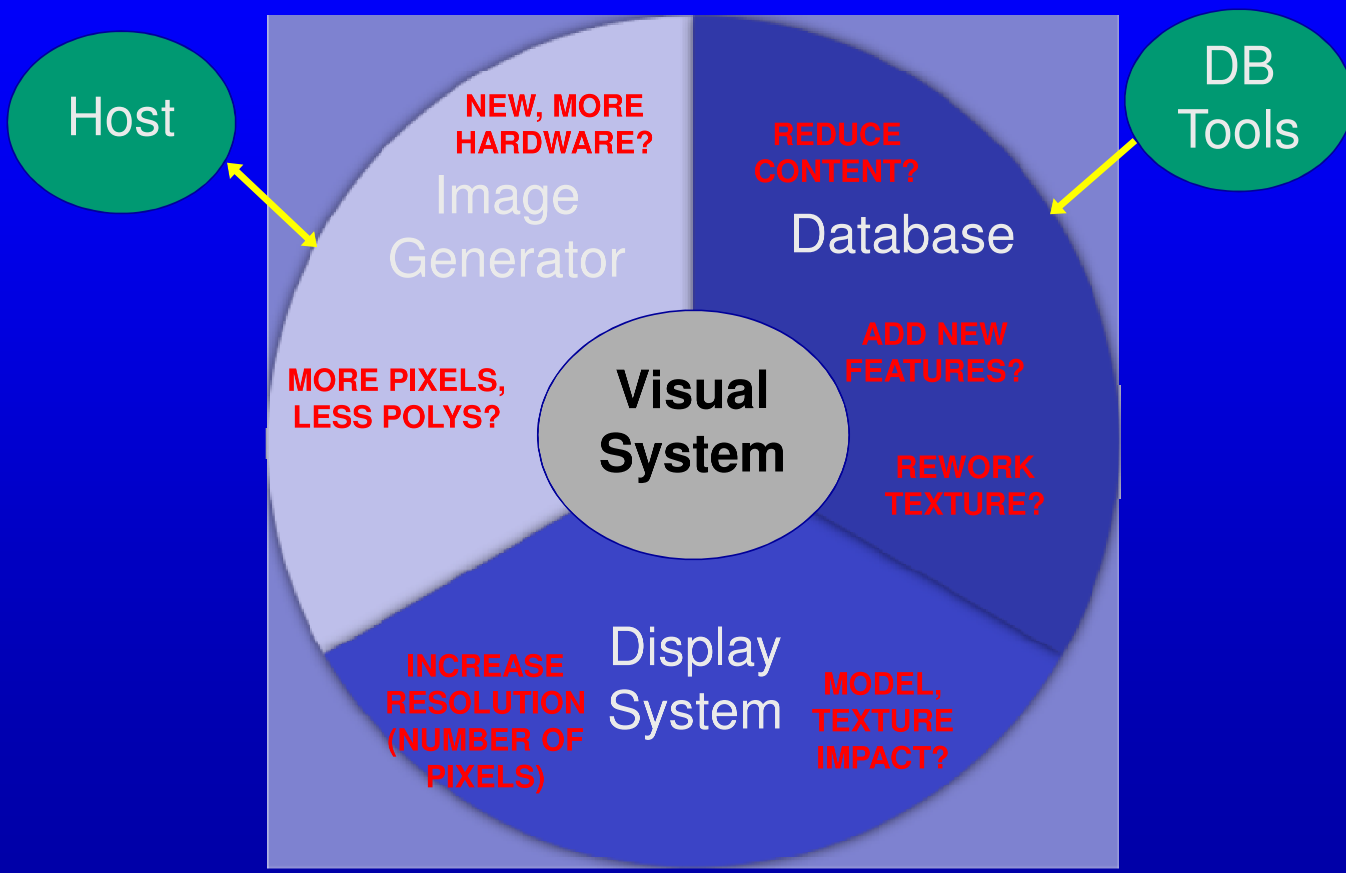
Wrap UP –
Back to System Considerations

Training Effectiveness vs. the "But it Would be cool!" Effect

- FAA Level D Case Study
  - Visual content, configuration standardized due to FAA, et al, regulations
  - Emphasis on weather, landing maneuvers
  - Highest standards allow zero flight time in aircraft

VISUAL SYSTEM COMPONENTS

- Changing one part of system will effect others



DONE!!

Questions??

Thank You!!

COMPUTER IMAGE GENERATION

FLIGHT AND GROUND SIMULATION UPDATE - 2017

199