



# **Integrating an Oblique Airphoto Library from Pictometry Directly Into ArcMap**

**Dr. Charlie Richman**

**Associate Director and  
Chief Information Officer**

**DC Office of Planning**

# About the DC Office of Planning

- We're the municipal planning office for DC
- Recently regrown from less than a dozen to about 75 staff
- Around 40 ArcMap users
- Full range of planning responsibilities



# GIS at DCOP

- We've built a centrally managed, centrally monitored ArcGIS infrastructure for the OP
- Core GIS staff provide application development, data development and support
- Front-line planners make most of their own maps using our tools
- Many of those maps are very site-specific

# Aerial Imagery at DCOP

- We use orthos routinely, typically the NGA (NIMA) 133 Cities 30cm imagery [available online at [gisdata.usgs.net](http://gisdata.usgs.net)]
- We also use an oblique aerial library licensed for DC government use from Pictometry International
- See [www.pictometry.com](http://www.pictometry.com) for more on their products



# About the Pictometry Images

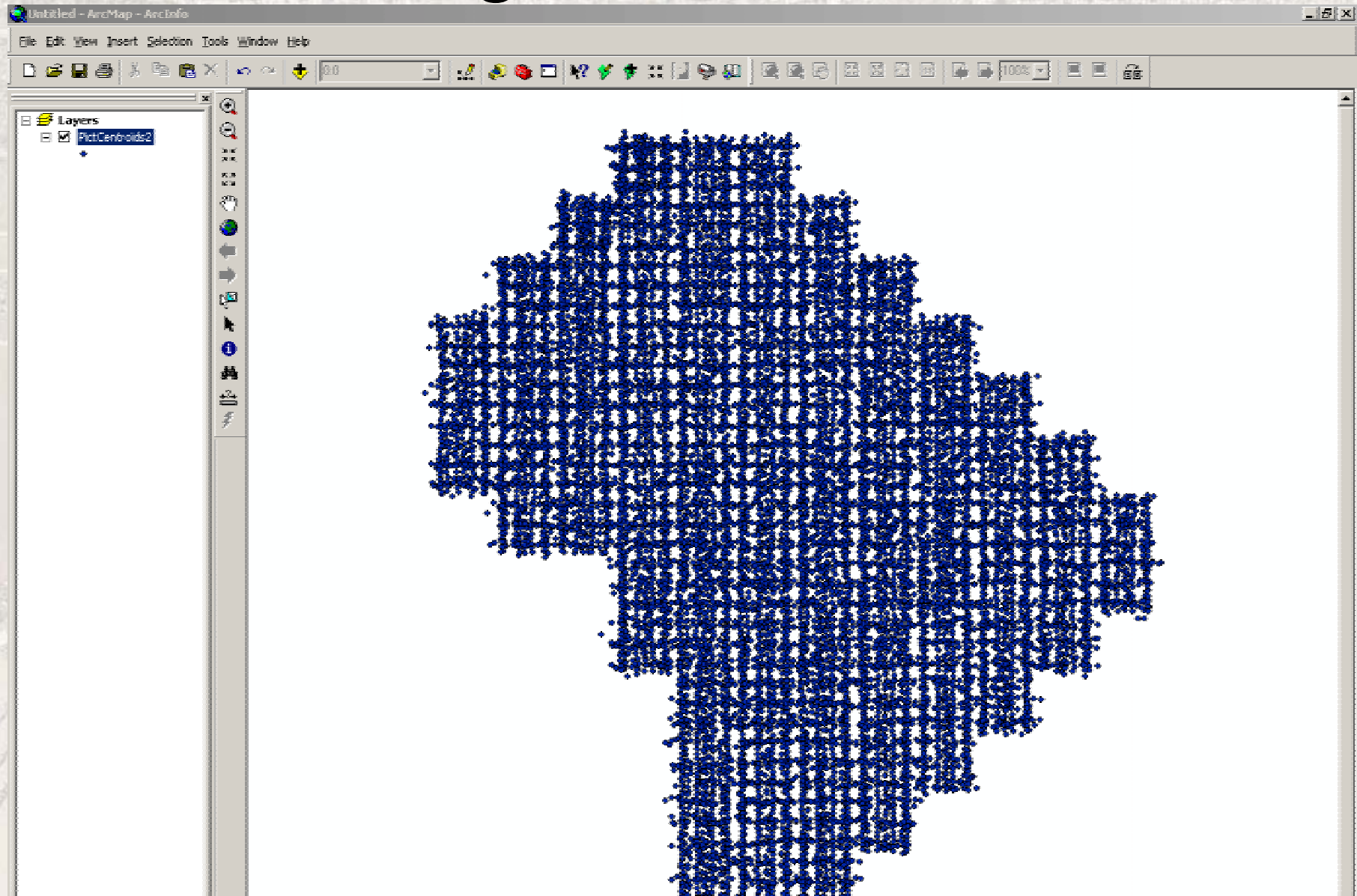
- Detailed imagery flown with sensors pointing straight down and also down at about 40 degree angles towards the North, South East and West
- Two series of flights at different elevations
- About 50 gigabytes of imagery
- Delivered by the vendor to be used with their Electronic Field Study application

# Two Levels

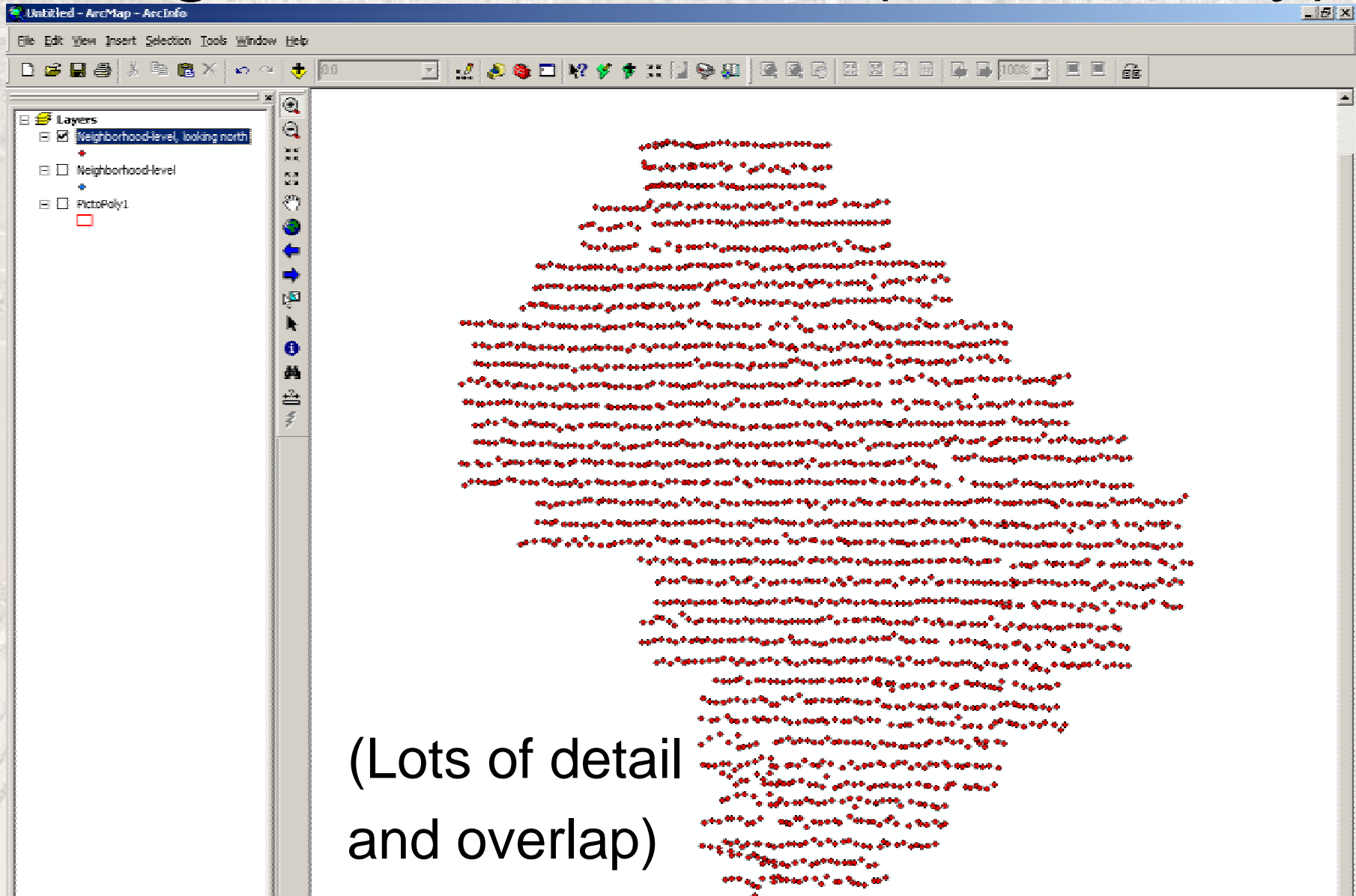
- “Community” level oblique images
  - Ours from around 4000 feet
  - Resolution varies with distance to the camera, around 1 foot GSD
- “Neighborhood” level oblique images
  - Ours from around 2500 feet
  - Approximately 0.4 foot GSD
- Two levels of orthos as well



# Lots of Images...

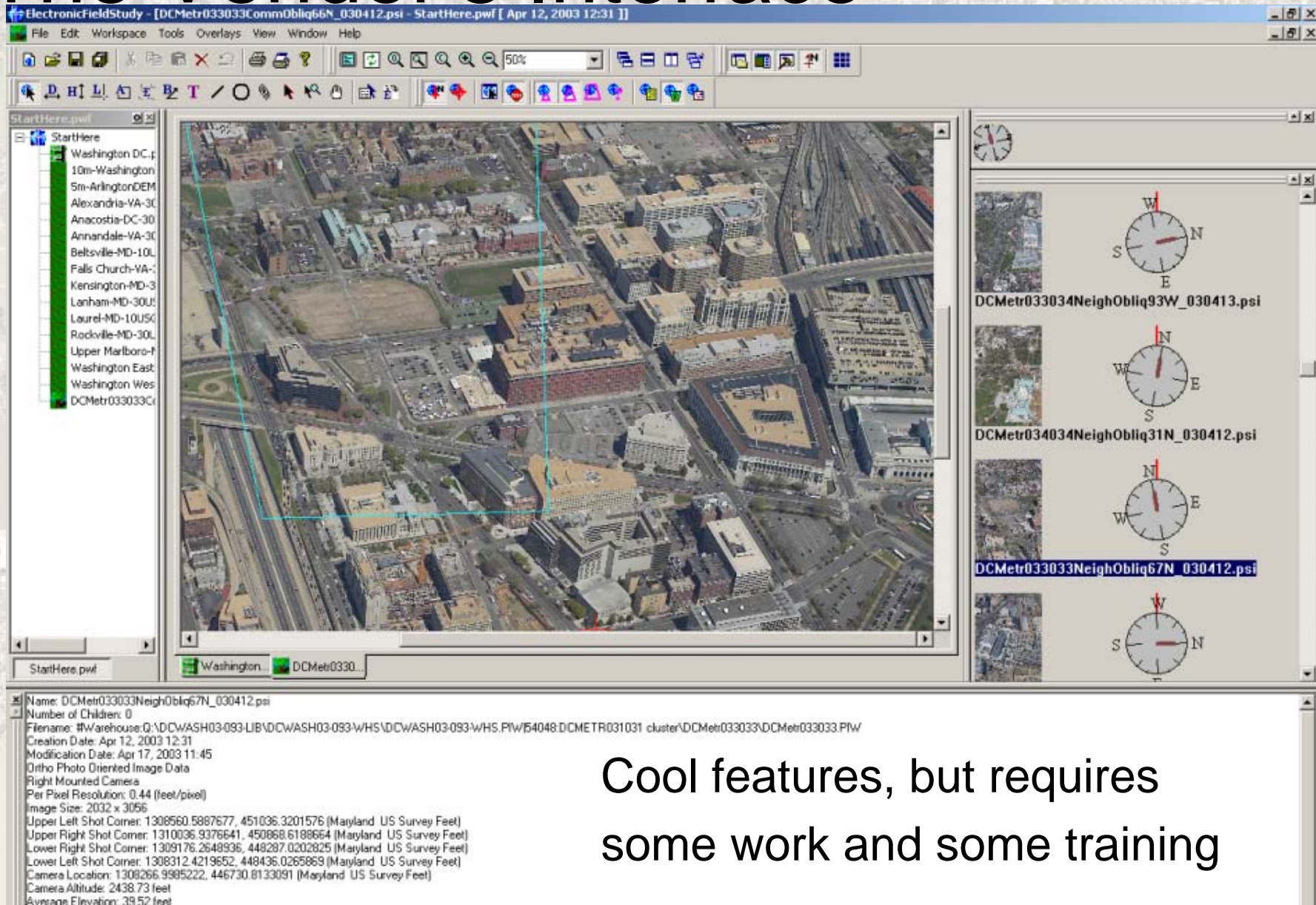


# Neighborhood Shots (North Only)





# The Vendor's Interface



Cool features, but requires  
some work and some training



# Goals For Our Interface

- Keep it in ArcMap
- Keep it simple – one button, nothing fancy
- Display the best shot for the area we're already mapping automatically
- Approximations are OK



# What We Have To Work With

- Thousands of “.psi” files in a tiered directory structure
- The vendor application can make a polygon shapefile listing shot types, orientations, corner coordinates, locations
- For single images it can display camera elevation, sensor size, and focal length

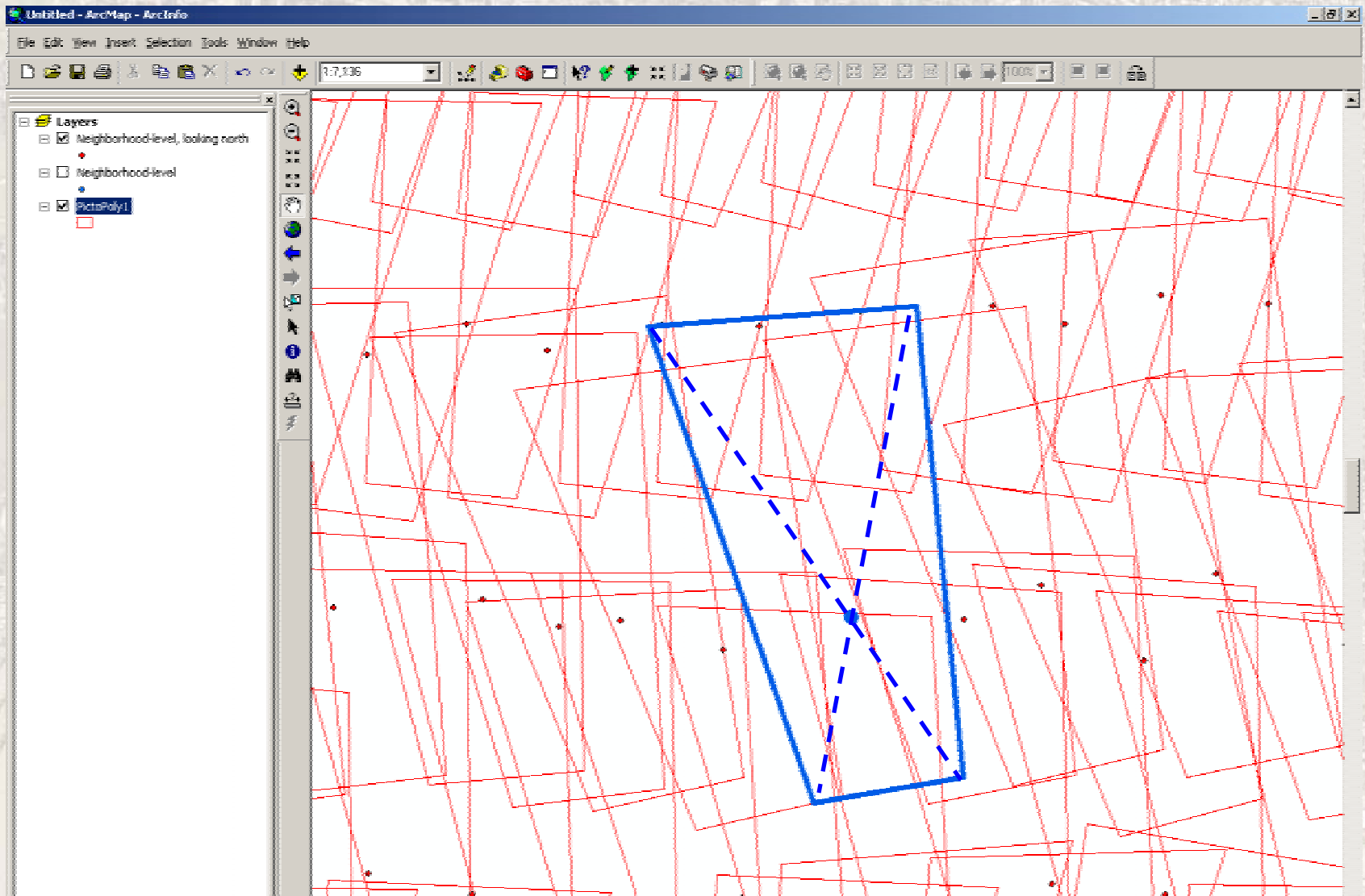
# Finding The Right Shot

- It's too slow to compare each image footprint polygon with our area of interest
- Centroids don't work for oblique shots
- Find the closest “Visual Centroid” to the center of our area of interest





# Creating Visual Centroids



# Confirming The Right Shot

- Do an initial search on the more detailed “neighborhood” image visual centroids
- Examine the footprint of the best image
- If it covers our area of interest we’re done
- If not, try again with the less detailed “community” images and offer the user a choice of full coverage or full resolution



# Oblique Image Display Issues

- We can't add it to the same data frame
- Scale varies substantially from the foreground to the background
- It doesn't have a defined coordinate system
- We need to find the area we care about  
(transform ground coordinates to pixel coordinates)

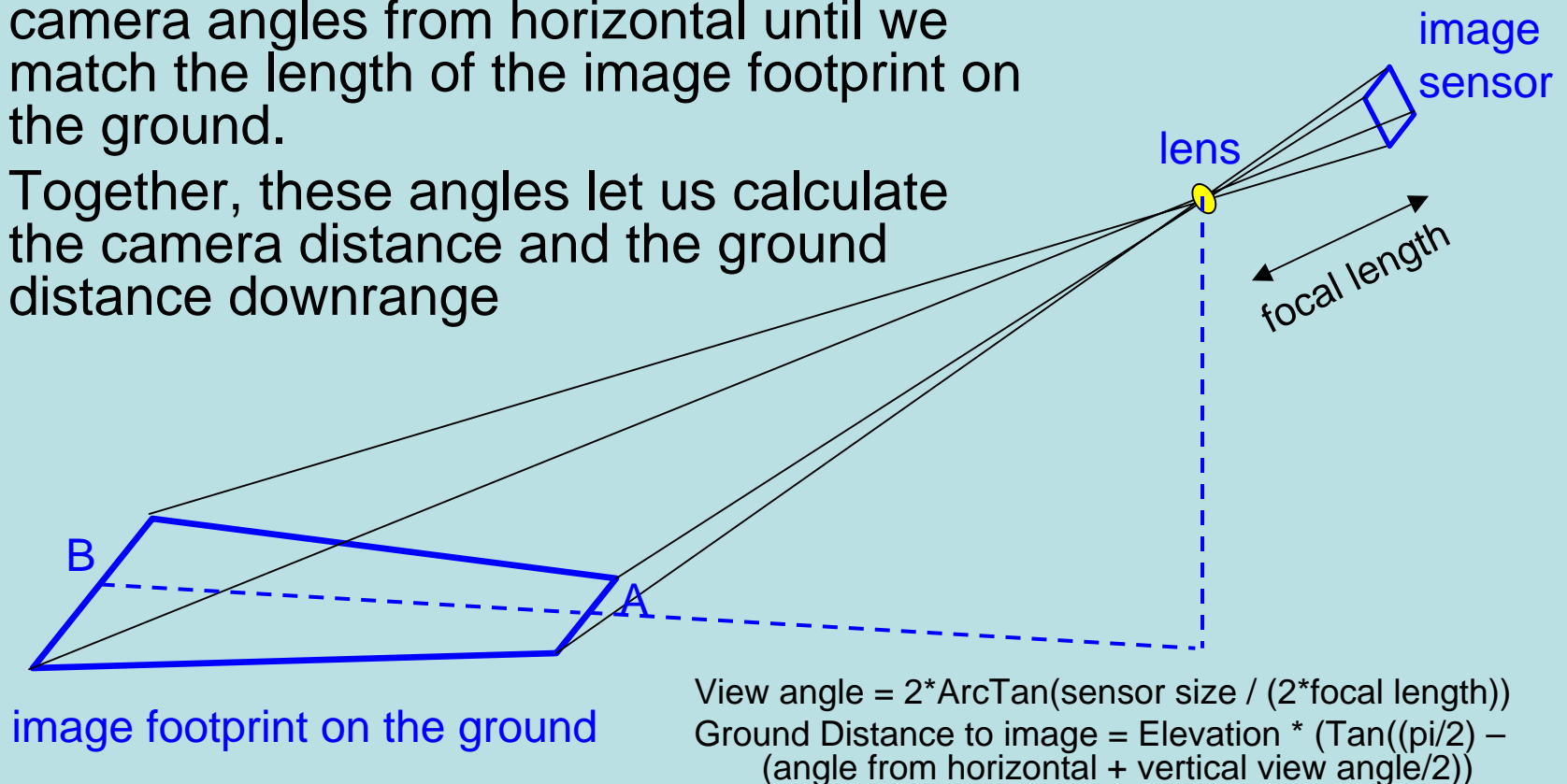
# Where Was The Camera?

- We don't know yet
- By inspecting detailed information for sample images we can learn
  - The sensor size (24 x 36 mm, 2k x 3k pixels)
  - The camera focal length (45 or 80 mm)
  - An approximate altitude
  - An approximate camera angle from horizontal
- We do know the ground coordinates of the corners of each image



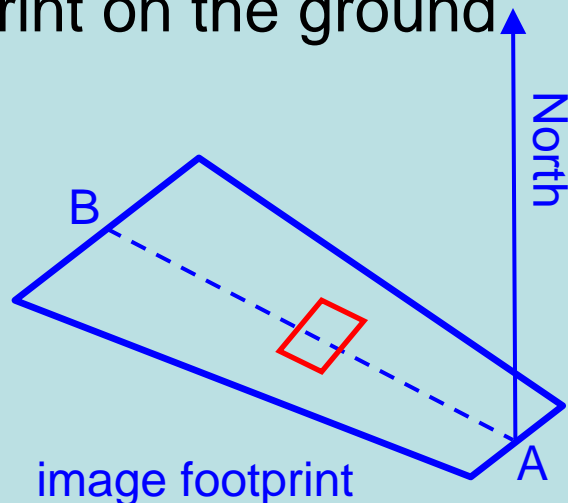
# Finding The Camera

- Camera focal length and sensor size define the angle of view
- For an estimated elevation, test possible camera angles from horizontal until we match the length of the image footprint on the ground.
- Together, these angles let us calculate the camera distance and the ground distance downrange



# State Plane Coordinates and Image Ground Coordinates

- We know the State Plane coordinates for the image corners
- Use them to calculate the angle between the axis of the image and the State Plane system
- Basic trigonometry lets us calculate coordinates for the corners of the area of interest as distances from and along the principal axis of the image footprint on the ground



(remember SOH CAH TOA?)



# Calculating Pixel Coordinates

- The angle from the camera axis at the lens to any corner of the area of interest on the image sensor is the same as the angle from the camera axis to that corner on the ground
- This lets us calculate distances from the axis on the camera sensor -- in pixels
- This tells us the pixel coordinates we need to zoom our image

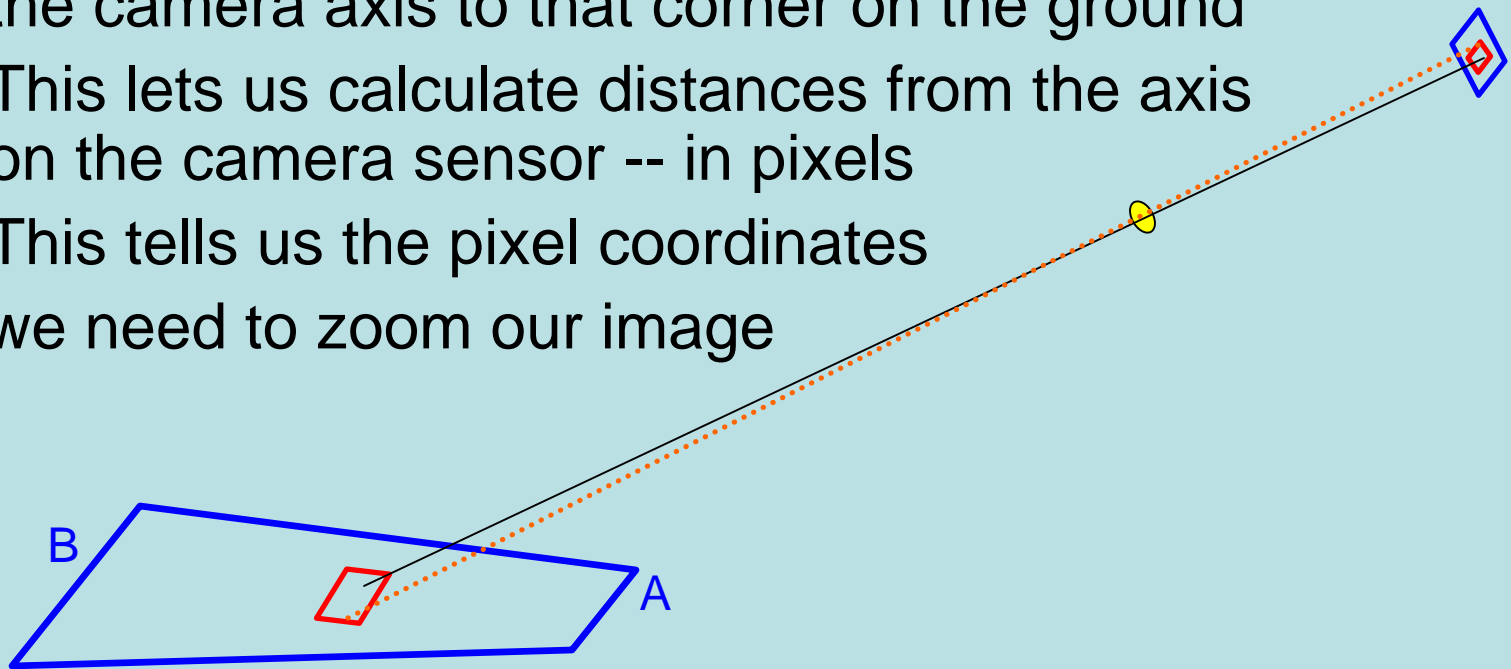


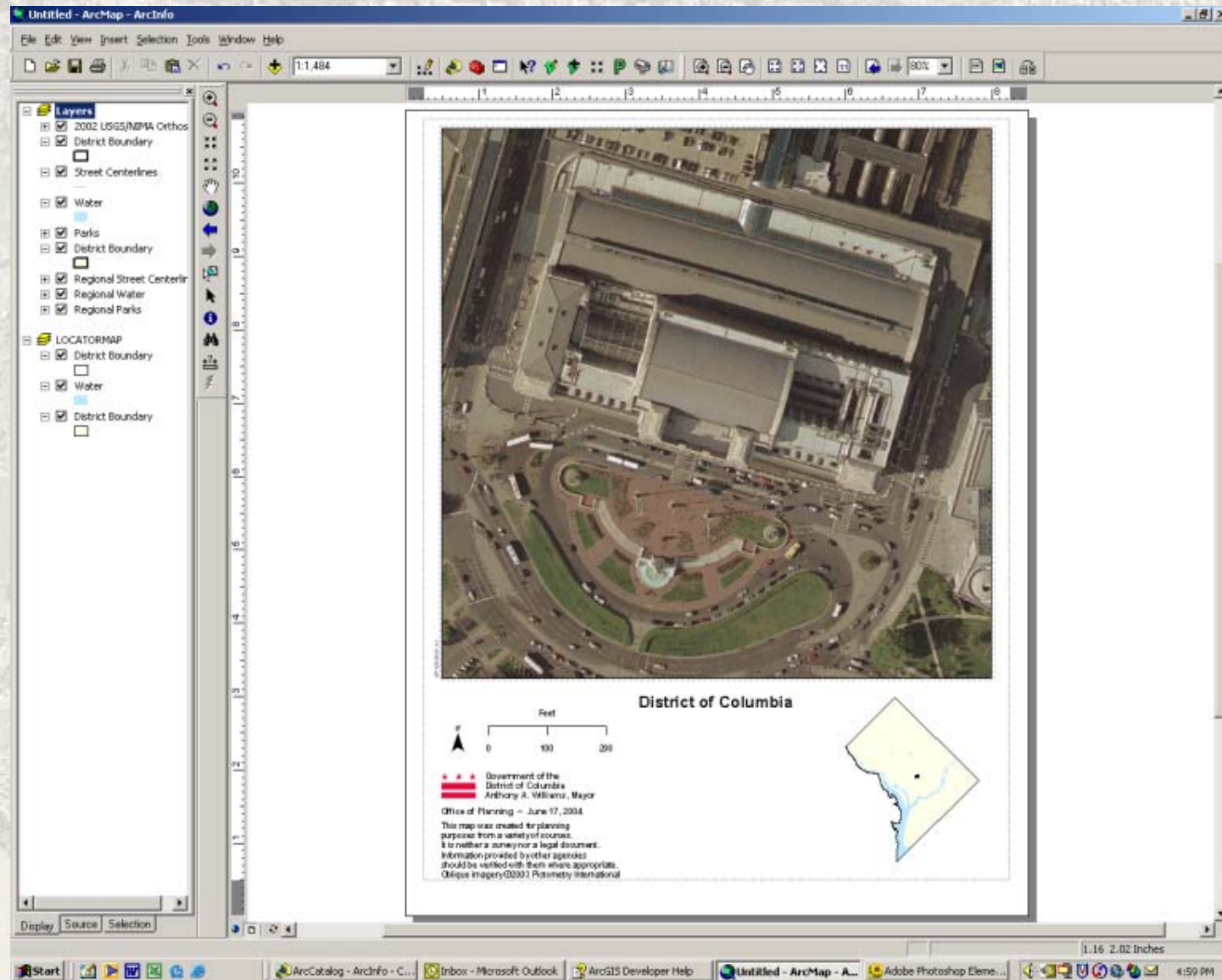
image footprint on the ground

# Ready To Go

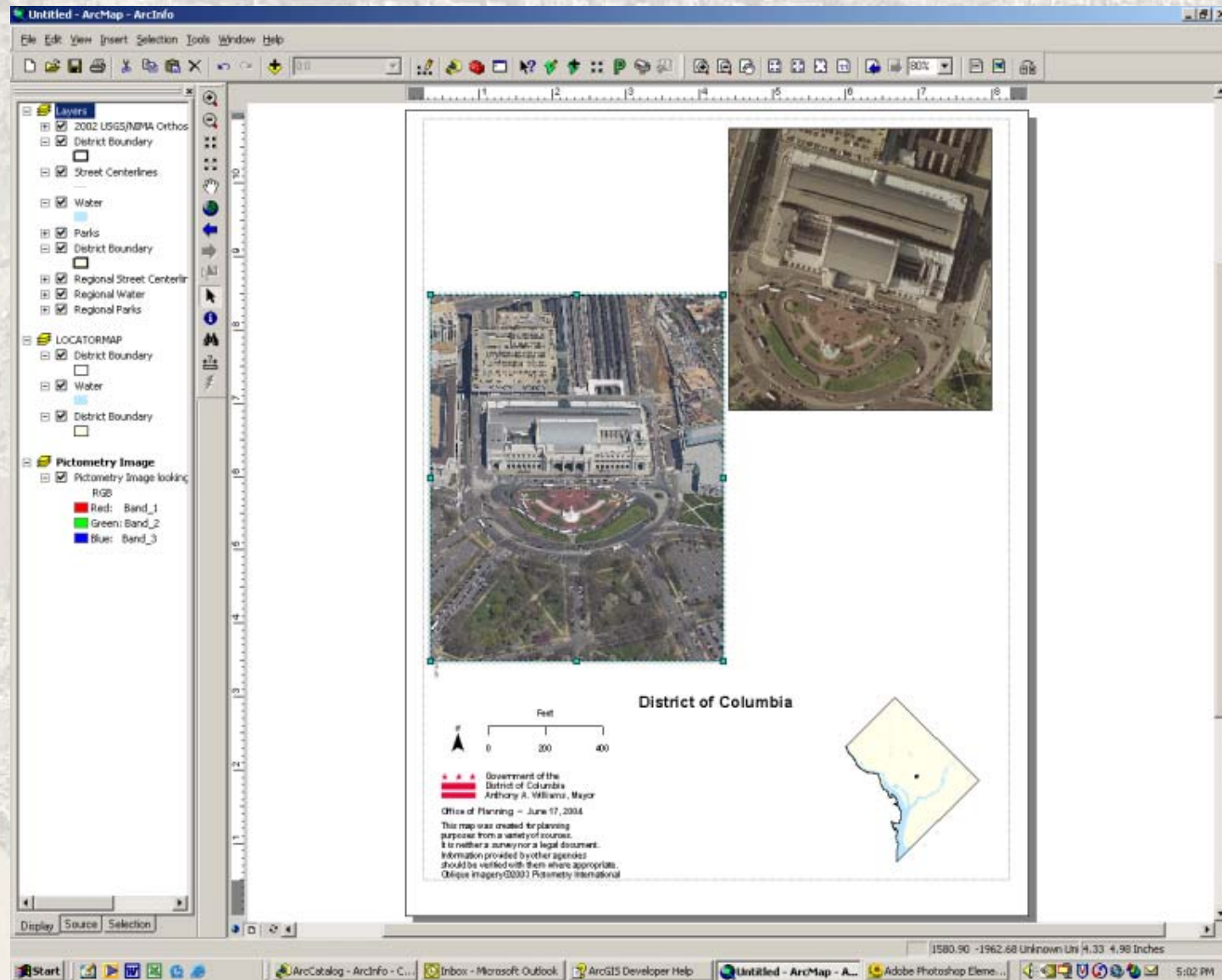
- ArcMap's interface won't open a .psi file
- ...but ArcObjects recognizes it as a .jpeg!
- Create a new Data Frame
- Add it to the layout
- Open the .psi file there as a raster layer
- Zoom to the calculated pixel coordinates
- Done!



# Example -- Ortho Only

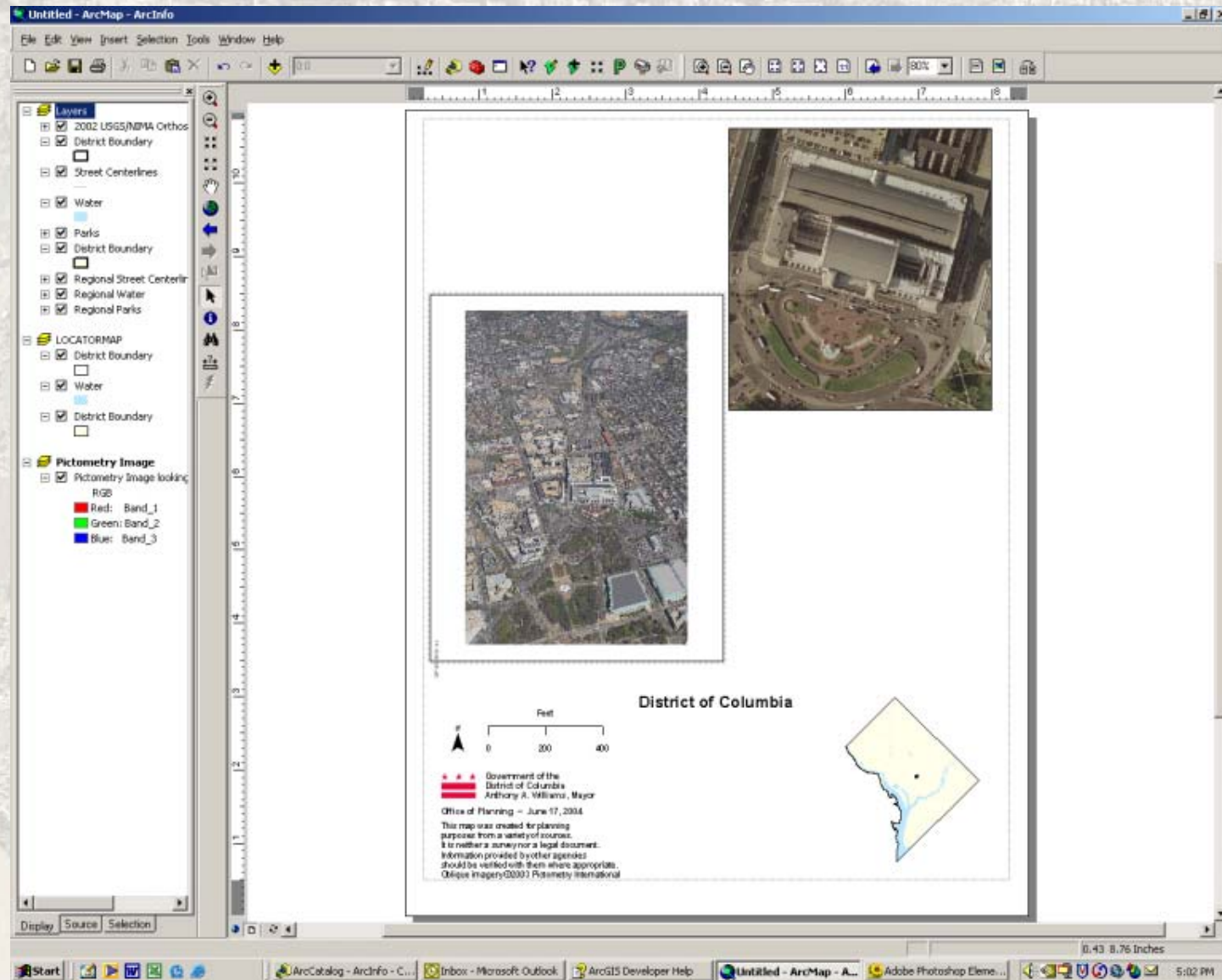


# Example – Adding View North

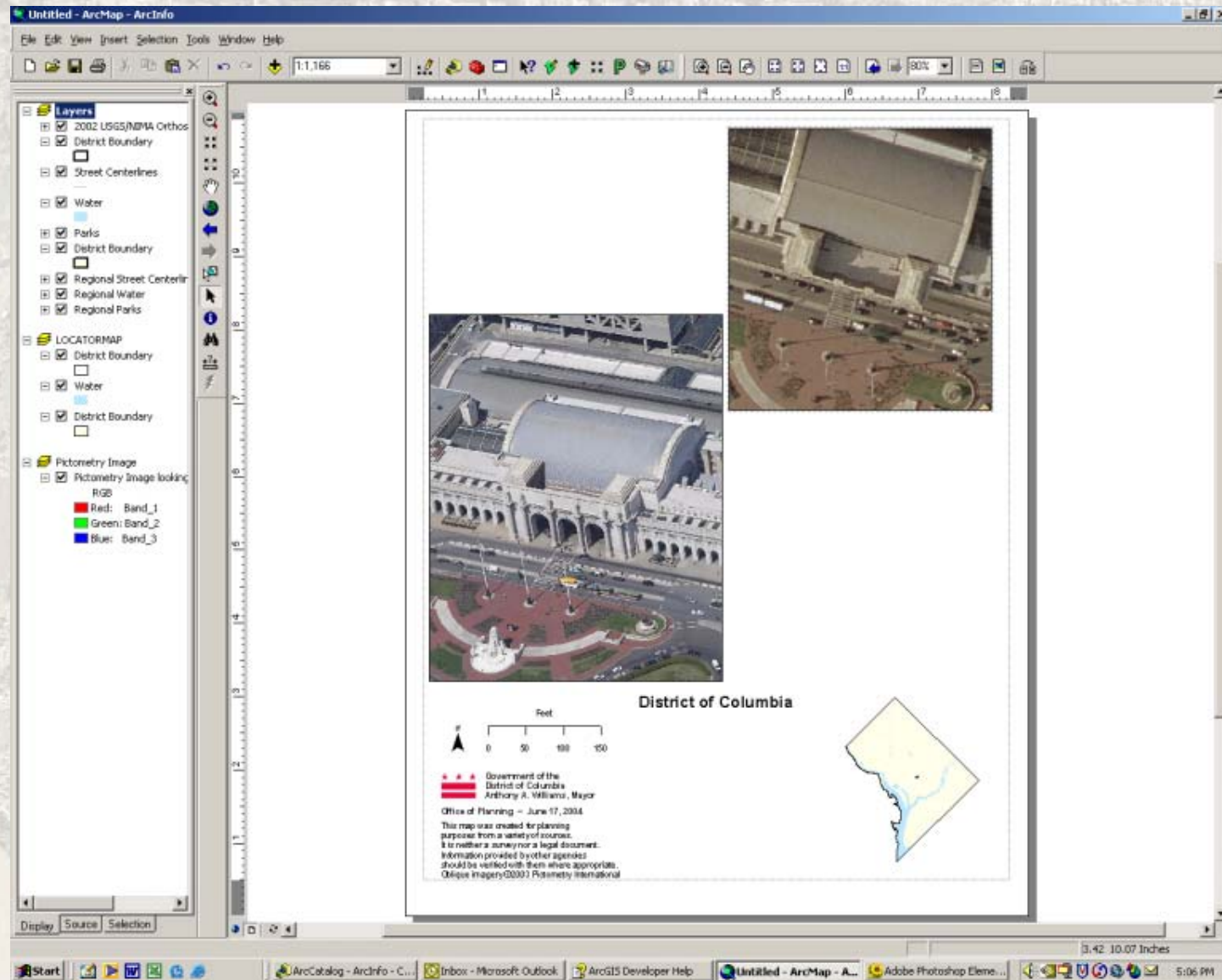




# Example – Why We Zoom In

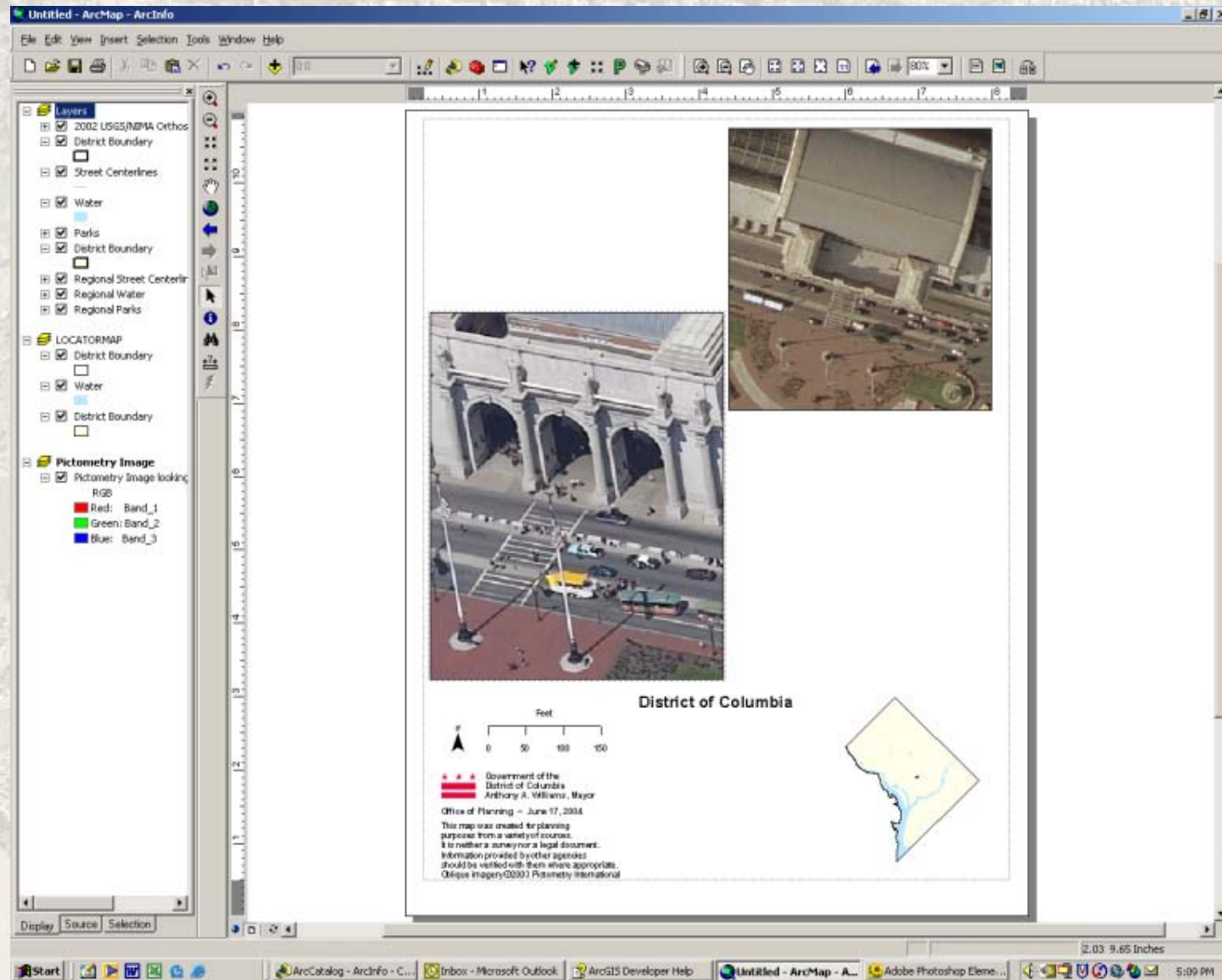


# Example – A Closer View

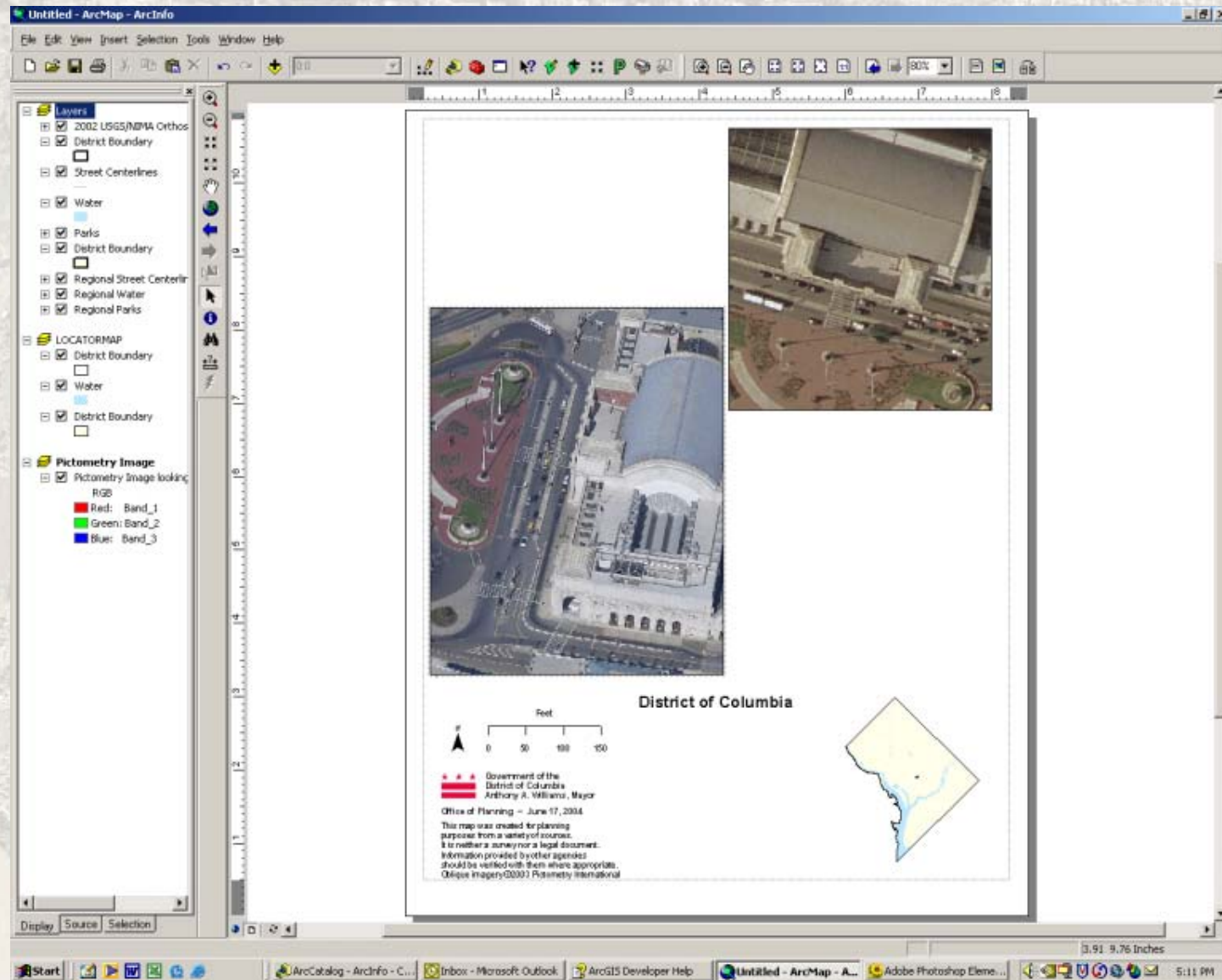




# Example – More Detail



# Example – View Looking West





# How Are We Using It?

- To show site descriptions clearly to our Board of Zoning Adjustment, community groups, and others
- To reduce the need for site visits for permit approvals
- To explore quickly what's actually going on at a wide variety of locations

# Conclusion

- It can be humbling to revisit high school trigonometry  
(but rewarding once you work it through)
- Simple tools actually get used  
(our planners use it routinely)
- Adding this new perspective to ArcMap is very effective in describing the landscape



# Thank You

- Feel free to contact me with comments or questions

Charlie.Richman@dc.gov

<http://planning.dc.gov>