Report on Weidlinger Simulation

By: Michael H

A more comprehensive version of this document (including anminated GIFs) is available at http://www.geocities.com/throwawayaddy2/report.html .

On March 25th, 2007, I acquired a zipped folder full of World Trade Center blueprints and other files, which was passed from a source (unknown to me) to Dr. Steven Jones. In this zipped folder were zipped files that extracted to files that were 1GB in size. These files contained 3-d models, but nobody knew exactly how to open them despite the fact that they were in a directory named "weidlingerAssoc_WTCtowerCollapseFlex**POVfiles**1780msTO1880ms". (The contents of this folder were included in this torrent file.) Due to the fact that I have no contact with the source that provided these files to Dr. Jones, I can only speculate that "Flex" is in reference to a Newtonian version of Weidlinger Associates' "Flex" simulation software, which was used to simulate the collapse of the World Trade Center towers [1], "POVfiles" is in reference to the fact that the output of this simulation software is in POV-Ray format, and that "1780msTO1880ms" is in reference to the output models being 1780 milliseconds to 1880 milliseconds after the initiation of collapse.

Problems quickly arose because POV-Ray would not open them, and they were too large to be opened in most text editors, but I was eventually able to get them to work. On this page, you will find my report on the model renders, followed by a render walkthrough, should you desire to render these 3-d models yourself.

Report on Renders

It's perhaps important to understand the definition of terms like *simulation*, *animation*, *model*, and *geometry*. In the world of 3-d, a *simulation* is a virtual environment in which objects are defined, and the behavior of those objects is dynamically determined by calculations that are done to accurately simulate effects, which may include things such as gravity, Newtonian mechanics, etc. An *animation* is a virtual environment in which objects are defined, but unlike a simulation, the objects within the environment do not have to adhere to the Laws of Physics. In an animation, a frog can jump to the moon; in a simulation, a frog would jump much more realistically. *Model* and *geometry* are very similar, but there are still subtle differences between the two. A *model* is some sort of entity made up of *geometry*, where *geometry* is a representation of a shape or network of shapes. While they are often used interchangeably, it may help you to know that the model is the represented entity and the geometry is the shape or networks of shapes that make up the model.

The model contained in the files sent to Dr. Jones is quite detailed; the geometry is the output of Weidlinger Associates' Flex simulation of the World Trade Center Two collapse. It should be noted that the Flex simulation software was developed by Weidlinger Associates for the Department of Defense [2, 3]. Though the model is very detailed, limited information is available on the complexity and accuracy of the simulation software, as well as the extent that Weidlinger Associates engineers had to artificially induce the collapse. Weidlinger Associates employs professional engineers, however, so despite the fact that they had to tweak their simulation software to force its output to match real-world behavior [4] (which NIST also did by tweaking variables), their report and simulation is just as valid (or invalid, depending on who you ask) as the NIST report and simulation.

Initial reports stated that the steel trusses and columns melted, which caused the building to collapse. These reports were quickly discarded in favor of theories that said the steel trusses merely softening due to excessive heat, causing them to fail and initiate a pancake collapse. This theory was used by FEMA's Building Performance Study and NOVA's "Why The Towers Fell" documentary, but also faced heavy criticism, as it did not explain why the segments above the collapse initiation failed at the exact same time as the floors began to pancake.

The conclusion of the NIST report (which was released after the Weidlinger Associates simulation was completed and also heavily relied upon an unreleased simulation) was that the trusses were heated, sagged excessively, and subsequently pulled the external columns inward, causing them to buckle, despite providing only questionable evidence to support their theory. The Weidlinger Associates simulation, however, came to the conclusion that not only did the trusses did not fail, but they did not contribute at all to the initiation of collapse [5, 6, 7].

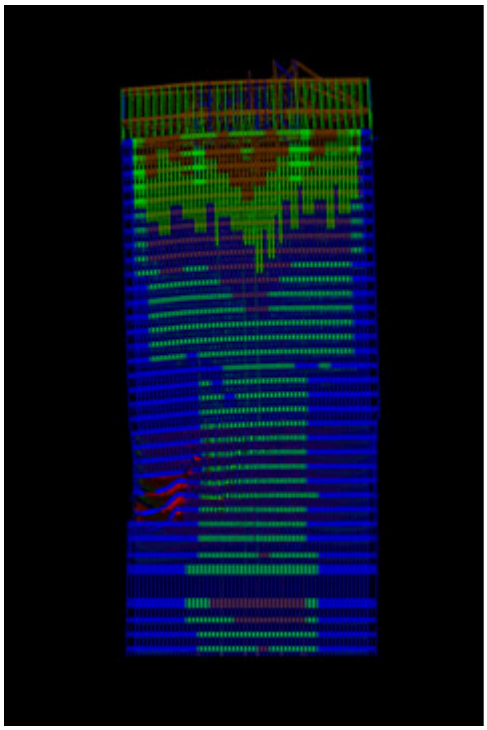
Renders

The contents of the POV files did not specify any sort of color scheme, so I arbitrarily set one. The different colors show either the different components or the different materials in the building's structure. The floors are red, the trusses are light green, the spandrels are primarily blue and light green, the exterior columns are primarily blue, and the interrior columns are brown.

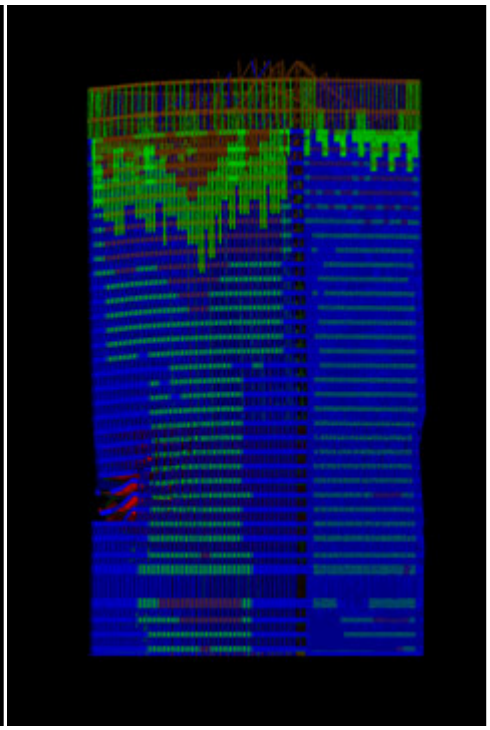
Thumbnails of Orthographic Renders of 1780ms file, rotating 30 degrees with each picture

Hover your mouse over the pictures for render settings.

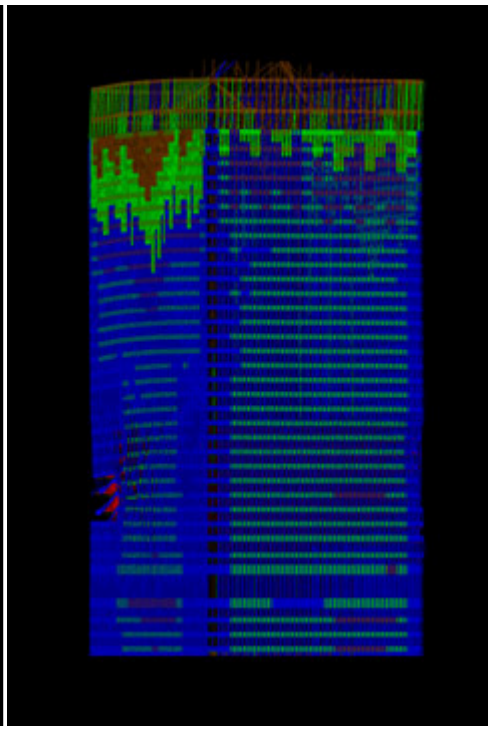
(Note: requires javascript, Rotation in tooltips is based on render settings, not real-world angle of view.)



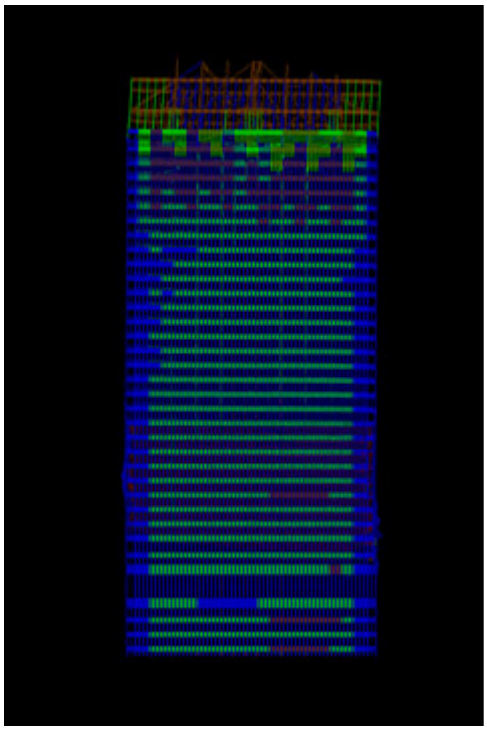
View from 0 degrees (North)



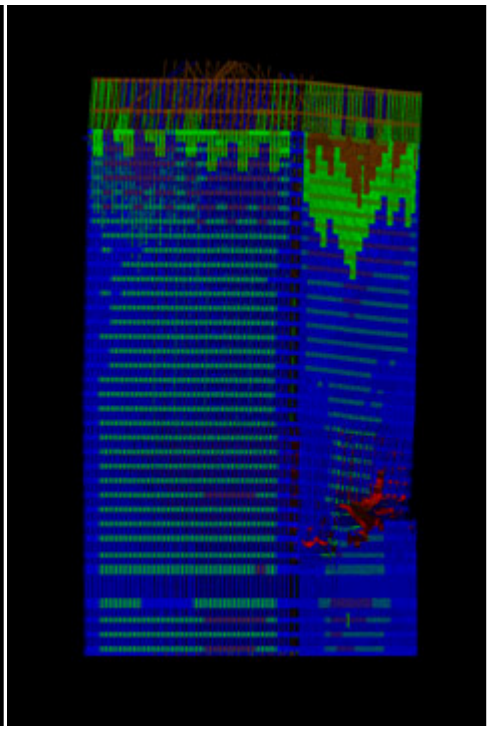
View from 330 degrees



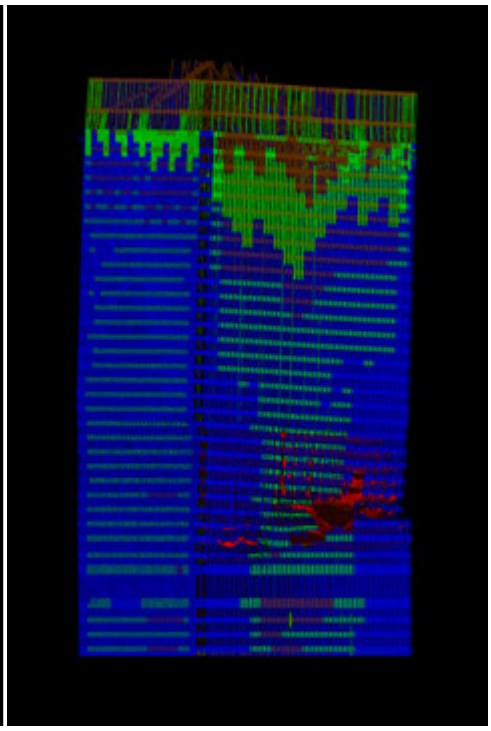
View from 300 degrees



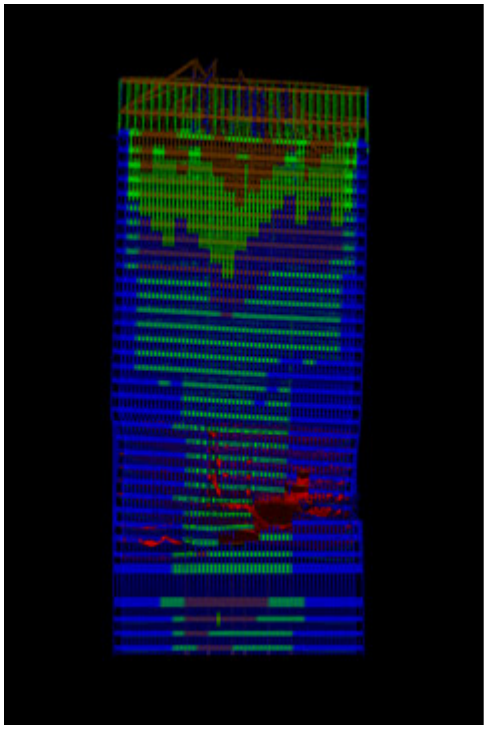
View from 270 degrees (West)



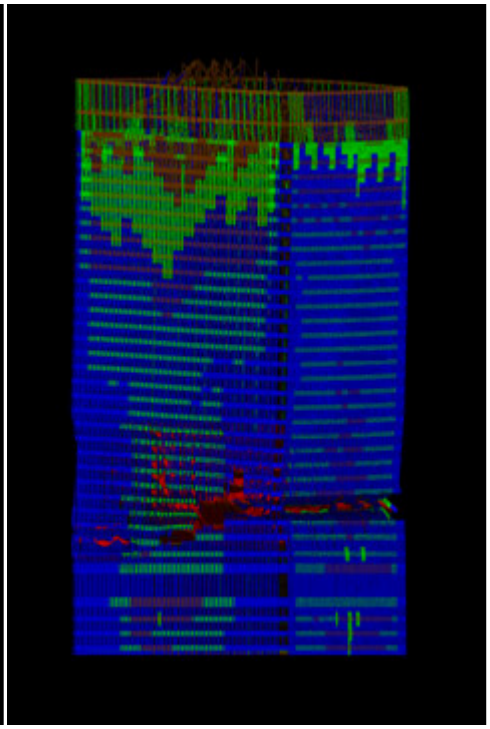
View from 240 degrees



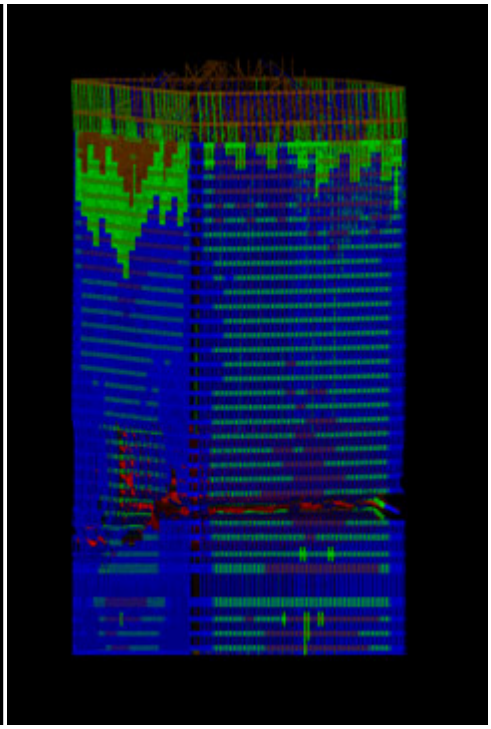
View from 210 degrees



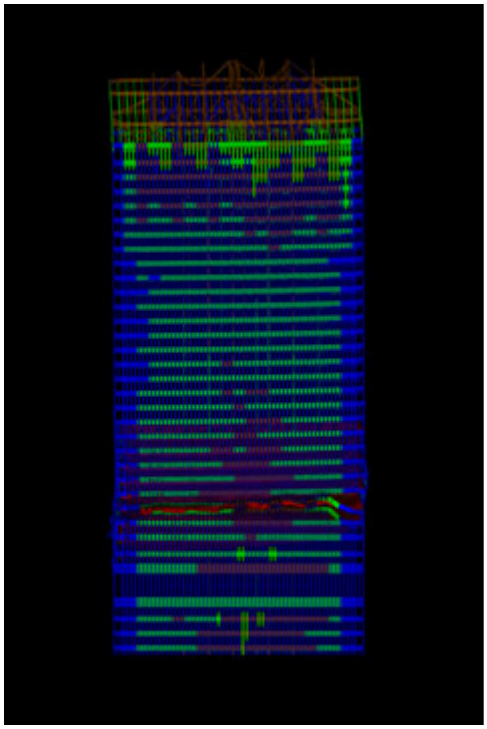
View from 180 degrees (South)



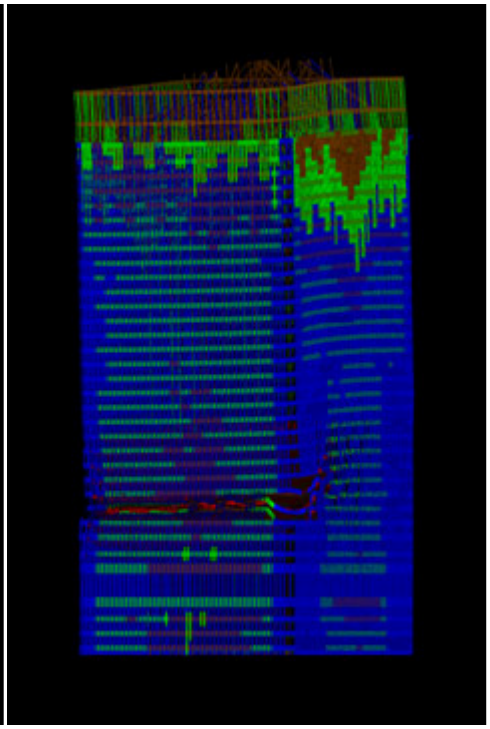
View from 150 degrees



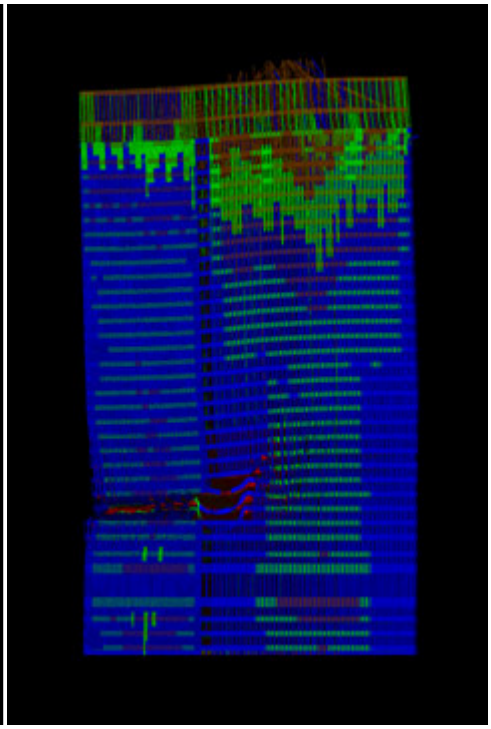
View from 120 degrees



View from 90 degrees (East)

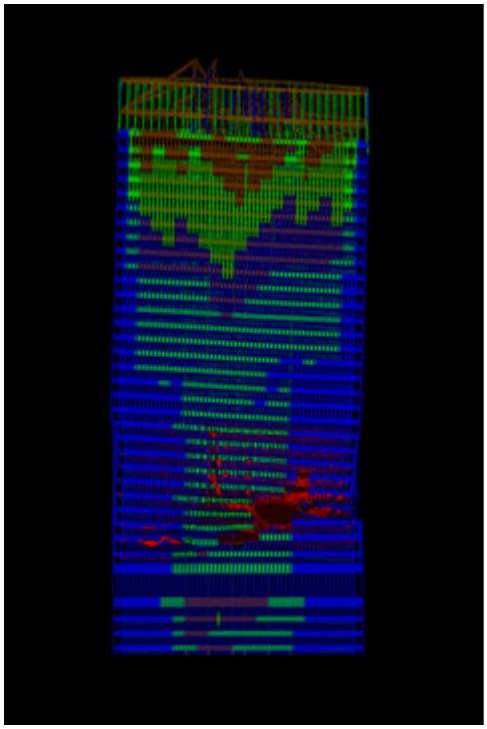


View from 60 degrees

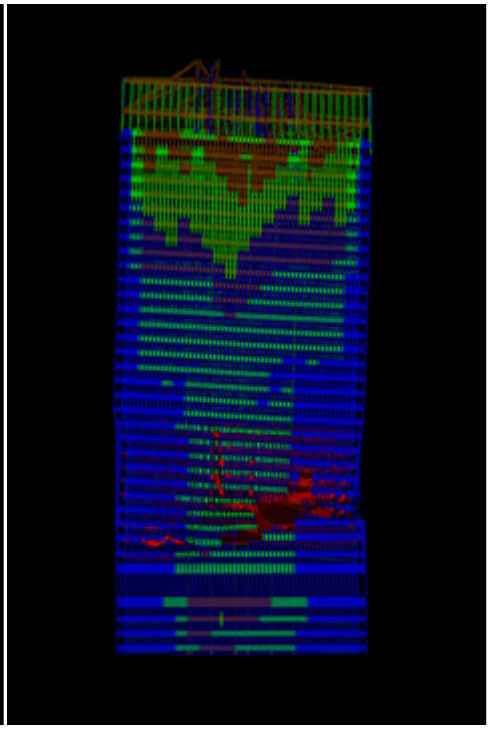


View from 30 degrees

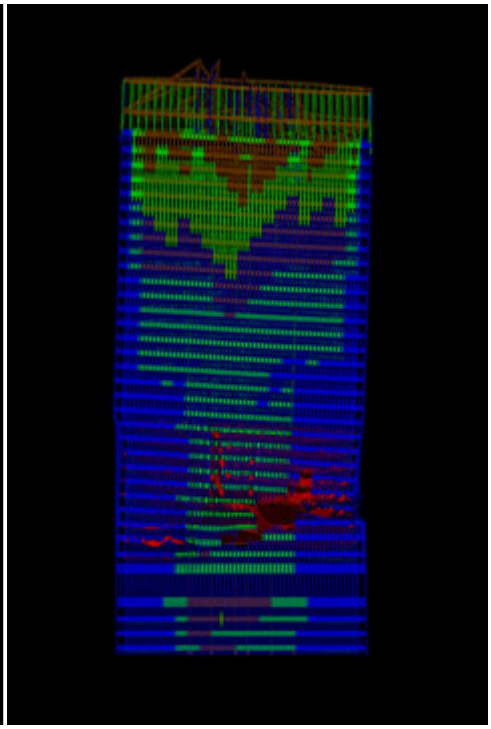
Thumbnails of Orthographic Renders of 1780ms to 1880ms files, rotated 180 degrees



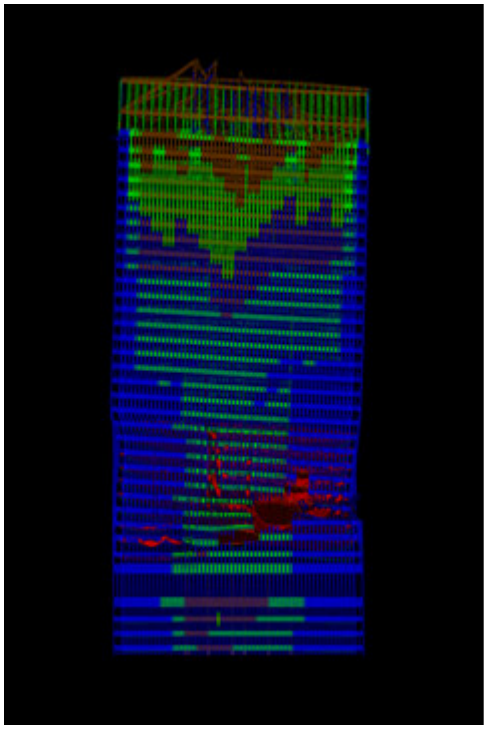
1780ms



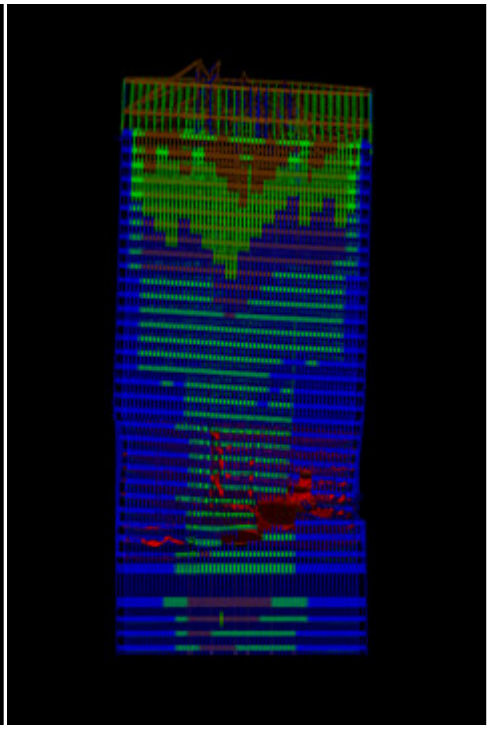
1790ms



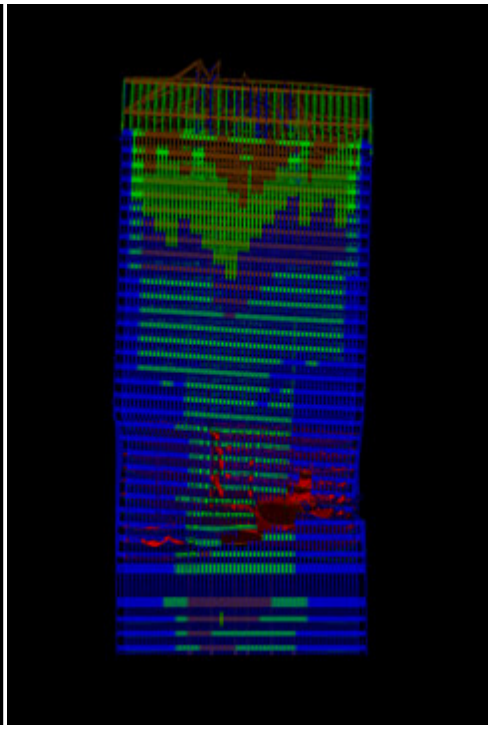
1800ms



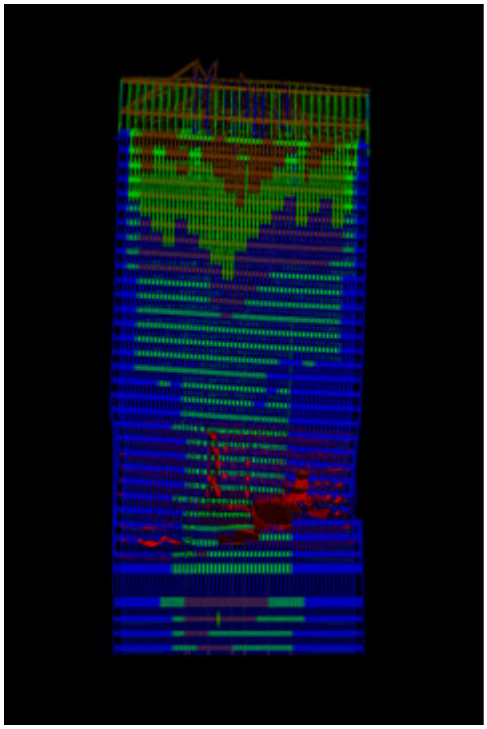
1810ms



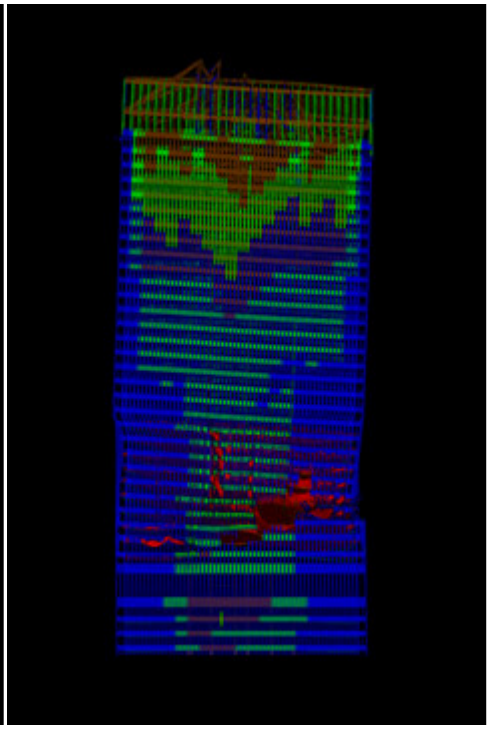
1820ms



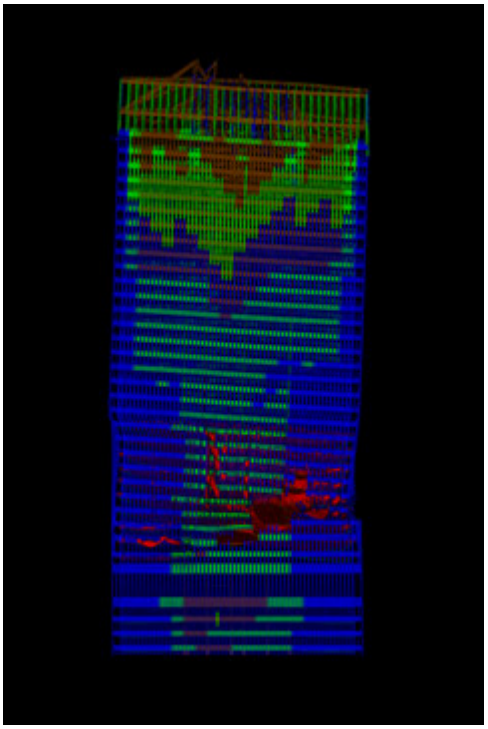
1830ms



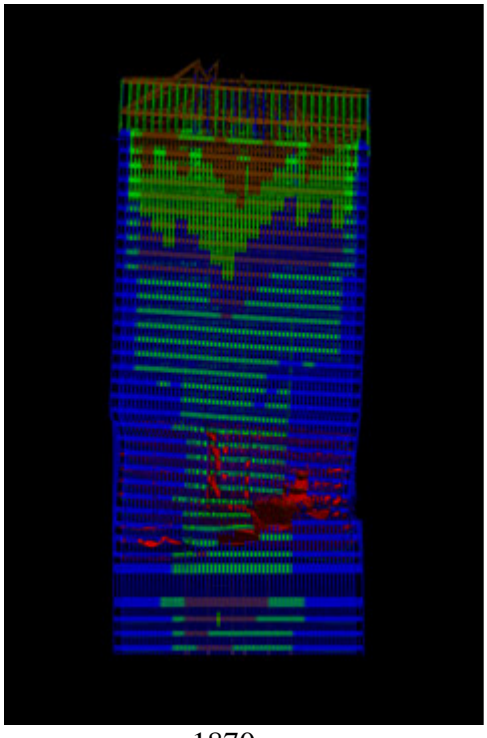
1840ms



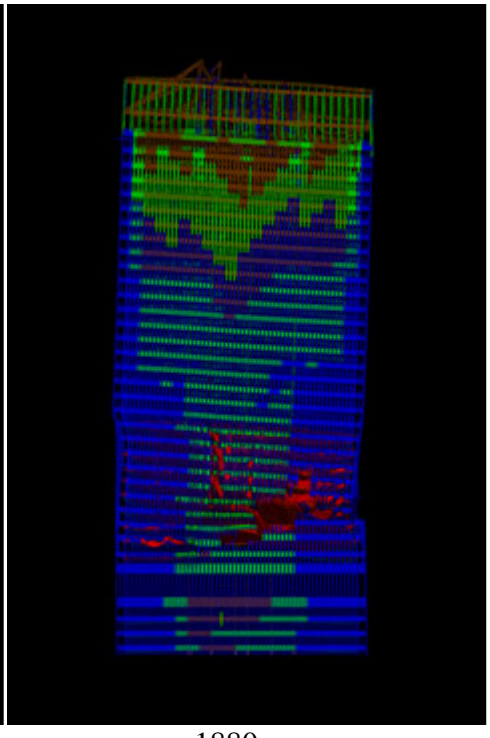
1850ms



1860ms



1870ms



1880ms

These renders bear some resemblance to what happened on September 11th, which is to say that the collapse was initiated primarily by a failure of the columns (significant and sudden floor failure would have pushed smoke and fire out the sides of the buildings, as multiple floors failed, for many seconds prior to the potential failure of the entire structure), and the first few moments of collapse were very asymmetrical in nature... but there are still significant problems here.

Issues with Weidlinger Simulation Results

No Details on Simulation Programming. Was this a simulation that was continuous in nature? Did it, without any variable tampering, include the impact of the aircraft, the fires (thermodynamic calculations, including conduction of heat), and the collapse? Or, as the evidence would indicate, was it a simulation that was run after the model was set up to give the desired results? Or, worse still, was this simply an animation?

Leaked Sample Spans Short Duration. The time elapsed during these different frames is very small -- only 1/10th of a second. Why did Dr. Jones' source not send the simulation output at the time of collapse and further into the collapse -- perhaps 5, 10, and 15 seconds into collapse? Did they not run the simulation further, or did they run the simulation further and did not like the results? If they did not like the results, did they feel that their simulation software would not accurately simulate the collapse of a building that was heavily deformed?

Questionable Collapse Initiation. What caused the failure of the columns? NIST did laboratory tests to see how a fully loaded test floor would react to an office fire, but their test floor only showed minimal signs of sagging and did not collapse [8]. They ended up having to use a computer simulation, which apparently did not collapse without using "severe" parameter values as opposed to "realistic" parameter values [9]. How did Weidlinger initiate this collapse? Their statements in the press indicate fire caused column failure, but this is extremely questionable, as the core columns would have conducted and dispersed the heat along the height of the structure. The NIST report, which tested some of the steel from the World Trade Center towers, says that not a single column sample showed signs of the temperatures that are associated with significant weakening of steel (600ºC is steel's half-strength temperature):

Over 170 areas associated with the 21 exterior panels were analyzed, and... only three locations showed evidence of [the steel possibly exceeding 250ºC]. Four of the core columns with known as-built locations were examined... for [two of the columns], sufficient paint for analysis was not available. For [the other two columns, a] few areas of paint were observed, [showing] no indication indication of temperatures over 250ºC [10].

In the several columns with known pre-collapse fire exposure, metallographic analysis provided no conclusive evidence that the steel exceeded 625ºC [11].

Awkward Buckling. In addition to the 800x1000 resolution renders, I also made an 8,000x10,000 resolution render (a cropped version of it is available here), which very clearly shows odd buckling within the structure. What would cause the SouthWest corner to buckle this way? Moments created by the severed support columns on East half of the building should have created forces that would have caused the columns to buckle in the other direction.

Inexplicable Floor Sag. The large resolution image also shows what can only be described as very extreme floor sag. According to NIST's lab tests, the floors should not have sagged more than a few inches [12], but

multiple floors that were not struck by the aircraft were sagging well over 10 feet. Was this the case prior to collapse? What made them conclude that this did not contribute to the initiation of collapse?

Press Statements on Weidlinger Report

Computer analysis by Weidlinger showed further that the planes also destroyed or disabled approximately 20 out of 47 columns in the center of the core of Tower 1 and approximately 5 out of 47 columns in the southeast corner of the core of Tower 2 [13, 14].

Twenty core columns were destroyed or disabled in the North tower? This seems very extreme.

In both towers, the damage to those columns was severe — so severe, in fact, that the simulations predict that the south tower should have, by this calculation, collapsed immediately... John Osteraas, director of civil engineering practice at Exponent Failure Analysis, who has been retained by the insurance companies, said that the incorrect result cast doubt on some of its predictions [15].

John Osteraas makes a very legitimate argument. If the Weidlinger Flex simulation shows that the building should have collapsed immediately after the impact of the plane, then they need to reanalyze their software or their 767 and WTC2 models. What changes did they make in order to fix this problem, or did they just make a quick, artificial fix and move onto the simulation of the fire?

In fact, the team concluded that, absent further fire damage, the Towers would not have collapsed [16].

The Weidlinger analysis created a series of diagrams for the towers, showing how stresses were distributed before they were struck, then after. Immediately after impact, the stress on remaining columns shot up, over a butterfly-shaped pattern around the impact zone on the facade and throughout the core. But none of the columns were stressed to the breaking point [17].

How can this possibly be true if, as the previous quote pointed out, the simulation showed that WTC2 should have collapsed immediately from the impact of the plane?

If the floor trusses had collapsed first, there would have been a mass of smoke as opposed to differentiated smoke, floor by floor, [Matthys P. Levy, chairman of Weidlinger Associates Inc. and New York City-based engineer that led the study,] adds [18].

While I believe Matthys Levy's statements on the smoke expulsion are somewhat incorrect, I agree that the expulsion of smoke would've been different. Had the floors collapsed first, large amounts of smoke would have been expelled out the windows for seconds prior to any significant movement in the exterior frame.

"Failure of the floors...was shown not to have had any significant role in the initiation of the collapses," says the report. Levy describes the floor truss system as "not unsubstantial," acting more like a membrane than a one-way system. "There was nothing wrong with it," he says [19, 20, 21].

Matthys Levy states that the floor supports did not contribute to collapse, but this is the very basis on which NIST explains the collapse!

It says that debris and dust distributed by the plane crashes inhibited the fires, such that the average air temperatures on the impact floors were between 400 and 700°C (750-1,300°F): significantly lower than those associated with typical "fully developed" office fires. However, says Matthys Levy, "By the time the temperature inside the buildings reached 400 degrees, the steel would have lost approximately 50% of its strength [22].

Levy claims that the steel would've lost half of its strength at 400°C, but the half-strength of steel is actually 600°C. Also, as stated before, NIST found no evidence suggesting that the steel reached such temperatures, and in the case of the North Tower, a woman was filmed waving to onlookers from the hole in the North Tower [23].

"The collapse of the towers was really a minor event in terms of impact on the adjacent buildings," Levy said [24].

How can Levy say this when WTC7 (350 ft away from WTC1) allegedly collapsed because of damage and fires caused by debris that fell from WTC1, and debris from both towers also damaged WTC3, WTC4, WTC5, and WTC6 well beyond repair? Banker's Trust, WFC3, and other buildings were damaged as well, but did not show signs of possible collapse.

The author of this report has a background in Computer Engineering and Mathematics (as well as some experience in Computer Graphics and Physics).

[1] http://www.encyclopedia.com/doc/1G1-125917313.html
[2] http://www.construction.com/NewsCenter/Headlines/ENR/20021104d.asp
[3] http://web.archive.org/web/20040707001222/http://newsmine.org/archive/9-11/questions/collapse/wtc-collapse-secret.txt
[4] http://web.archive.org/web/20040824123629/http://newsmine.org/archive/9-11/questions/collapse/graphic-look.txt
[5] http://www.construction.com/NewsCenter/Headlines/ENR/20021104d.asp
[6] http://www.engr.psu.edu/ae/WTC/SilversteinWTCReleaseStatement.pdf, pg. 1 & 4
[7] http://www.construction.com/NewsCenter/Headlines/ENR/20021025b.asp
[8] NIST NCSTAR 1-6B
[9] NIST NCSTAR 1-6D, pg. 12, footnote pg. 34, nearly all diagrams are labeled "severe", "upperbound", "Case B", or "Case D"
[10] NIST NCSTAR 1-3, Section 6.6.1, pp. 94-95
[11] NIST NCSTAR 1-3, Section 6.6.3, pg. 99
[12] NIST NCSTAR 1-6B
[13] http://www.engr.psu.edu/ae/WTC/SilversteinWTCReleaseStatement.pdf , page 2
[14] http://www.construction.com/NewsCenter/Headlines/ENR/20021025b.asp
[15] http://web.archive.org/web/20040824123629/http://newsmine.org/archive/9-11/questions/collapse/graphic-look.txt
[16] http://www.engr.psu.edu/ae/WTC/SilversteinWTCReleaseStatement.pdf , page 2
[17] http://web.archive.org/web/20040824123629/http://newsmine.org/archive/9-11/questions/collapse/graphic-look.txt
[18] http://www.construction.com/NewsCenter/Headlines/ENR/20021104d.asp
[19] http://www.engr.psu.edu/ae/WTC/SilversteinWTCReleaseStatement.pdf, page 4
[20] http://www.construction.com/NewsCenter/Headlines/ENR/20021025b.asp
[21] http://www.construction.com/NewsCenter/Headlines/ENR/20021104d.asp
[22] http://www.findarticles.com/p/articles/mi_m3601/is_39_49/ai_101679160/
[23] http://www.prisonplanet.com/articles/january2005/170105womanwaving.htm
[24] http://www.findarticles.com/p/articles/mi_m3601/is_39_49/ai_101679160

Render Walkthrough

In order to make these renders, you will need a pretty good computer. I suspect that 2GB of RAM is required (I'm not sure because I have 3.25GB), and it will take quite a bit of time to parse the file, even on a good computer.

I am not responsible for any problems that you may have with your computer, should you decide to download any of the software in this guide or mess up your boot.ini file.

Acquiring the Files

First, you will need the files. Right now, they are only available in torrent format. Here are two torrent links you can get them from: Loose Change 911, Conspiracy Central (click "DOWNLOAD TORRENT")

(If you've never used torrents before, they're pretty easy. µTorrent and Azureus are popular and easy-to-use torrent clients. Once you have a torrent client, you just open the torrent up, and it will start downloading.)

The torrent will download a .zip file. After unzipping it, there should be a folder named "weidlingerAssoc_WTCtowerCollapseFlexPOVfiles1780msTO1880ms", which contains multiple zip files. Each one of these contains the geometry needed to make these renders. Unzip the ones you want to render.

Once unzipped, the files will be 1GB in size. You cannot open them in notepad or wordpad. If you have a text editor that can open them, you can use that, but I used the trial version of Ultra-Edit.

You will also need to download POV-Ray, a 3-d rendering program.

Setting Up the Files

Go to the directory in which the files are stored and make a copy of the file. You will use this file for the changes you're about to make without messing up the original file. You could edit the original file, but it you mess it up, you'll have to overwrite it by unzipping it again.

If you open the renamed file up in your text editor, it should look something like this:

triangle { < 0.10414E+05, 0.20983E+02, 0.62247E+02>,
< 0.10416E+05, 0.22940E+00, 0.83008E+02>,
< 0.10414E+05, 0.20964E+02, 0.82993E+02>
texture { TX9 } }
...
triangle { <x1, y1, z1>,
<x2, y2, z2>,
<x3, y3, z3>
texture { TX } }*
...

As you can tell, the file is very long: 24 million lines long.

(I'm not going to explain in great detail what you're about to do; if you want to understand it, try to figure it out for yourself or by asking friends.)

This file is just raw geometry data, which POV-Ray will not render unless you've set up the "scene" (camera, lights, textures, etc). In order to do that, paste this at the very beginning of the file:

```
#version 3.1;
global_settings {
ambient_light <2,2,2>
max_trace_level 1
}

#include "colors.inc"
#include "shapes.inc"

camera {
orthographic // set up camera
up <0,10000,0>
right <-8000,0,0>
direction <0, 0, 1>
angle 70

rotate <0,0,-90> // face origin
location <0, 0, -8000> // displace from origin
rotate <0,0,0> // swing around origin
translate <0.1325E+05, 1180, 1180> // center model in view
}

light_source {<-1E+8, -1E+8, -1E+8> color LightGray}
light_source {<-1E+8, -1E+8, 1E+8> color LightGray}
light_source {<-1E+8, 1E+8, -1E+8> color LightGray}
light_source {<-1E+8, 1E+8, 1E+8> color LightGray}
light_source { <1E+8, -1E+8, -1E+8> color LightGray}
light_source { <1E+8, -1E+8, 1E+8> color LightGray}
light_source { <1E+8, 1E+8, -1E+8> color LightGray}
light_source { <1E+8, 1E+8, 1E+8> color LightGray}

#declare TX1 = texture { pigment {color rgb <0, 0.0, 0.5>} }
#declare TX2 = texture { pigment {color rgb <0, 0.0, 0.8>} }
#declare TX3 = texture { pigment {color rgb <0, 0.4, 0.6>} }
#declare TX4 = texture { pigment {color rgb <0, 0.6, 0.2>} }
#declare TX5 = texture { pigment {color rgb <0, 0.8, 0.0>} }
#declare TX6 = texture { pigment {color rgb <0.2, 0.6, 0>} }
#declare TX7 = texture { pigment {color rgb <0.4, 0.4, 0>} }
#declare TX8 = texture { pigment {color rgb <0.4, 0.2, 0>} }
```

#declare TX9 = texture { pigment {color rgb <0.8, 0.0, 0>} }

mesh {

Also, put a closing curled bracket at the very end of the file (usually control+end will get you there):

}

Save the file.

Setting Up Your Computer

The file is now set up, but your computer likely is not. You can attempt to render the file, but it will probably not work just yet if you're using Windows XP, MCE, or any earlier version. If you want to attempt rendering the file without setting the 3GB switch (esp if you are using a 64-bit OS), go ahead and see if it works.

Due to limitations of 32-bit operating systems, Microsoft (and perhaps other OS makers) decided to limit how much RAM a user applications can use. Without setting the 3GB switch, user applications can never use more than 2GB of RAM. Although I never saw the render take up over 1.2GB of RAM, it didn't work without setting the 3GB switch, and it worked afterwards, so... uhm... there you have it.

In order to set the 3GB switch, follow these instructions given at Microsoft's web site. Basically, all you need to do is append "/3GB" onto a line in your boot.ini file, save the changes, and reboot.

Setting Up the Render

You should definitely be able to render the file now, but will need to change the render settings to make it render properly, so go ahead and open up POV-Ray. You will need to change the render settings to something like 400x500, 800x1000, 1600x2000, etc., since I defined the "up" and "right" vectors as having that aspect ratio. To define one of these resolutions, go to the Tools menu and select "edit resolution INI file". Once the file opens, go to the bottom and add this:

[800x1000, AA 0.1]
Width=800
Height=1000
Antialias=On
Antialias_Threshold=0.1

Save the file and tell POV-Ray to use this setting by selecting it in the dropdown combobox under the File menu.

Rendering the Scene

To save RAM, you should make your renders from the command prompt or from the GUI, but with the file not opened for editing within the GUI.

To render from within the GUI without having the file open, open up POV-Ray, close out all of the windows within it, and then, from the Render menu, select "File Queue". Press the "Add File" button, select the file(s) you want to render, and press the "OK" button.

Alternatively, you can do a command prompt render, go into your Start menu, select Run, and put this in the textbox:

"**[POV-Ray direcotry]**\bin\pvengine.exe" /render "**[file to render]**"

where "**[POV-Ray direcotry]"** is the directory you installed POV-Ray to and "**[file to render]"** is the directory & name of the file that you edited and saved (make sure you use the correct extension: none, .txt, .POV, etc.)

At this point, the POV-Ray GUI should load, and it should begin parsing the file. You should see something like "_____ tokens parsed." in the status bar. It starts off pretty fast, but it slows down as it goes, ending at around 180,000,000 tokens. On my computer, it took 37 minutes to process the file and about 30 seconds to render, using a 800x1000 setting. Your results may vary!

After it has reached 180,000,000 tokens parsed, it should display the render within a few minutes, and it will automatically save this render to the directory that contains the file being rendered.

Changing the Scene

If you make changes to the camera and do not know what you're doing, you may find yourself very confused very quickly! I would not suggest making any changes to the camera outside of what is suggested.

This is a list & description of what everything that you might want to change within the scene file:

ambient_light <2,2,2>	This is the value of ambient light. Ambient lights are not raytraced.
max_trace_level 1	This is the maximum raytracing depth that the rendering engine will calculate. I'm not familiar with POV, but since the textures seem to have no reflectivity by default, I am not sure if this setting matters.
orthographic	This is the type of projection. You may want to change this to "perspective".
rotate <0,0,0> // swing around origin	Changing the "x" value (the first 0) will rotate the camera around the model about an axis that goes through the height of the building. (0 = view from North, 90 = view from West, 180 = view from South, 270 = view from East)
#declare TX1 = texture { pigment {color rgb <0, 0.0, 0.5>} }	This is the declaration of texture "TX1", which you can change the RGB value of, but feel free to change the RGB value of the any of the textures. Although I'm not familiar with POV-Ray, I suspect the range for appropriate values is either 0 to 1 or 0 to 255.