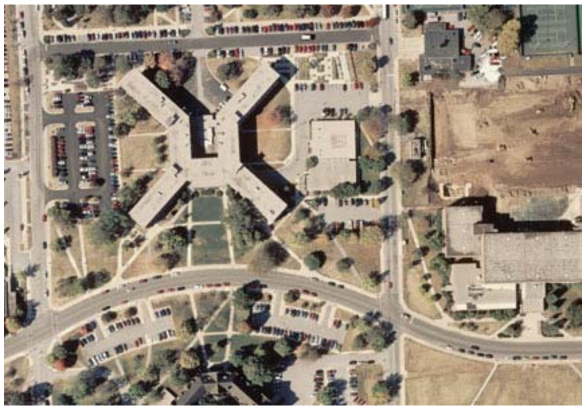
Generation of 3D Models from Stereo-Digitized Vertex Points

CE 503 Photogrammetry I – Fall 2006, Updated Fall 2007



Roof vertices of Meredith Residence Hall (X-shaped building above) were stereo-digitized in Erdas Imagine – Stereo Analyst, using the 3D measure tool found in the Utility menu. Enable collection with the "+" icon, then lock the collection state with the lock icon, collect roof vertices, then save the file. That file can be edited with Wordpad and saved as ascii text, MSDOS format, so that it can be accessed by any application reading text files. The file saved from Erdas, with extraneous verbiage looks like:

```
Point 1. 913619.153003 574814.126241 meters, 202.4839 meters.
Point 2. 913591.378824 574842.250892 meters, 202.3014 meters.
Point 3. 913566.194539 574842.135841 meters, 202.5219 meters.
Point 4. 913538.028017 574813.144398 meters, 202.4903 meters.
Point 5. 913528.484537 574822.608403 meters, 202.4289 meters.
Point 6. 913560.761106 574855.851849 meters, 202.2132 meters.
Point 7. 913586.859185 574856.066539 meters, 202.2118 meters.
Point 8. 913586.266837 574872.226894 meters, 202.1069 meters.
Point 9. 913560.512141 574872.372728 meters, 202.3257 meters.
Point 10. 913527.858928 574904.079755 meters, 202.1199 meters.
Point 11. 913537.025076 574914.180870 meters, 202.0544 meters.
Point 12. 913564.777501 574886.634280 meters, 202.2331 meters.
Point 13. 913590.587228 574886.559973 meters, 202.2336 meters.
Point 14. 913618.685162 574915.173155 meters, 202.0479 meters.
Point 15. 913628.513809 574905.434790 meters, 202.1111 meters.
Point 16. 913600.421289 574876.538109 meters, 202.2986 meters.
Point 17. 913600.597993 574852.435320 meters, 202.4550 meters.
Point 18. 913628.939091 574823.765559 meters, 202.6411 meters.
Point 19. 913619.436619 574814.131658 meters, 202.4839 meters.
```

Edit that to save only the X,Y,Z coordinates, and force common Z for all points:

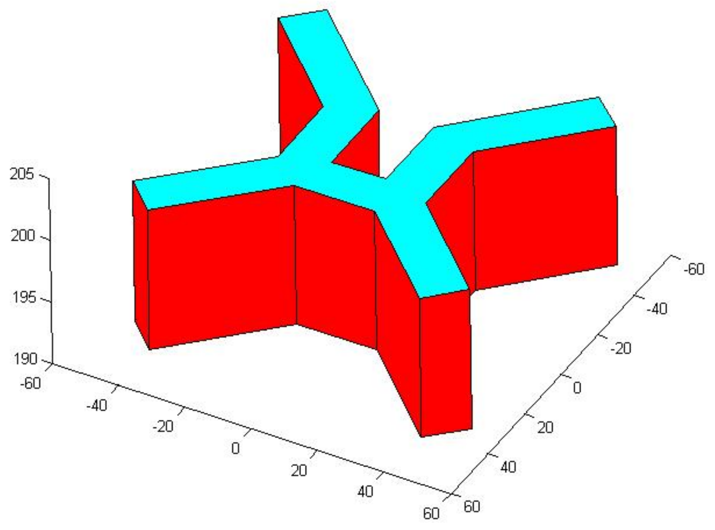
```
 913619.153003 574814.126241 202.30
 913591.378824 574842.250892 202.30
 913566.194539 574842.135841 202.30
 913538.028017 574813.144398 202.30
 913528.484537 574822.608403 202.30
 913560.761106 574855.851849 202.30
 913586.859185 574856.066539 202.30
 913586.266837 574872.226894 202.30
 913560.512141 574872.372728 202.30
 913527.858928 574904.079755 202.30
 913537.025076 574914.180870 202.30
 913564.777501 574886.634280 202.30
 913590.587228 574886.559973 202.30
 913618.685162 574915.173155 202.30
 913628.513809 574905.434790 202.30
 913600.421289 574876.538109 202.30
 913600.597993 574852.435320 202.30
 913628.939091 574823.765559 202.30
 913619.436619 574814.131658 202.30
```

Matlab Script

Read that file with a Matlab script (m-file), assume a constant ground elevation, construct directly the roof polygon, then the side polygons, and render to a figure using the “patch” command. XY coordinates are shifted to local origin. The Matlab script:

```
% dsp3d.m  2-oct-06
% display 3d patches of a digitized building footprint

zhi=202.3;
zlo=191.0;
load mdth.txt;
[m n]=size(mdth);
X=mdth(:,1);
Y=mdth(:,2);
Z=zeros(m,1);
for i=1:m
  Z(i)=zhi;
  end
Xm=mean(X);
Ym=mean(Y);
X=X-Xm;
Y=Y-Ym;
patch(X,Y,Z,'c');
hold on
for i=1:m-1
  xpat=[X(i);X(i+1);X(i+1);X(i)];
  ypat=[Y(i);Y(i+1);Y(i+1);Y(i)];
  zpat=[zhi;zhi;zlo;zlo];
  patch(xpat,ypat,zpat,'r');
  end
```

The rendered polygons appear as follows, note that the “axis equal” command was not given so the displayed aspect ratio is not correct:



Autocad Script

Use Matlab to read the same file and generate an Autocad script, in which we define the polygon faces using the “3dface” command. When Autocad script is created, execute it by typing “script” at the command line. Note that “3dface” requires input of triangles or quadrilaterals only, so we have to break up the complicated roof polygon into individual faces. That is the “fac” array which are index numbers. They probably assume faces are “convex” as it speeds up the rendering. The Matlab file creating the script:

```
% mk_scr.m  2-oct-06
% make autocad script for input of building faces

zhi=202.3;
zlo=191.0;
load mdth.txt;
[m n]=size(mdth);
X=mdth(:,1);
Y=mdth(:,2);
Z=zeros(m,1);
for i=1:m
  Z(i)=zhi;
  end
Xm=mean(X);
Ym=mean(Y);
X=X-Xm;
Y=Y-Ym;

fac=[1 2 17 18;
3 4 5 6;
2 3 6 7;
17 2 7 8;
16 17 8 13;
13 8 9 12;
12 9 10 11;
14 15 16 13];

fid=fopen('bldg.scr','wt');
for i=1:8
  fprintf(fid,'3dface
%4.2f,%4.2f,%4.2f\n',X(fac(i,1)),Y(fac(i,1)),zhi);
  fprintf(fid,'%4.2f,%4.2f,%4.2f\n',X(fac(i,2)),Y(fac(i,2)),zhi);
  fprintf(fid,'%4.2f,%4.2f,%4.2f\n',X(fac(i,3)),Y(fac(i,3)),zhi);
  fprintf(fid,'%4.2f,%4.2f,%4.2f\n',X(fac(i,4)),Y(fac(i,4)),zhi);
  fprintf(fid,'\n');
  end


for i=1:m-1
  xpat=[X(i);X(i+1);X(i+1);X(i)];
  ypat=[Y(i);Y(i+1);Y(i+1);Y(i)];
  zpat=[zhi;zhi;zlo;zlo];
  fprintf(fid,'3dface %4.2f,%4.2f,%4.2f\n',xpat(1),ypat(1),zpat(1));
  fprintf(fid,'%4.2f,%4.2f,%4.2f\n',xpat(2),ypat(2),zpat(2));
  fprintf(fid,'%4.2f,%4.2f,%4.2f\n',xpat(3),ypat(3),zpat(3));
  fprintf(fid,'%4.2f,%4.2f,%4.2f\n',xpat(4),ypat(4),zpat(4));
  fprintf(fid,'\n');
  end

fclose(fid);
```

And the resulting Autocad script written by the previous program:

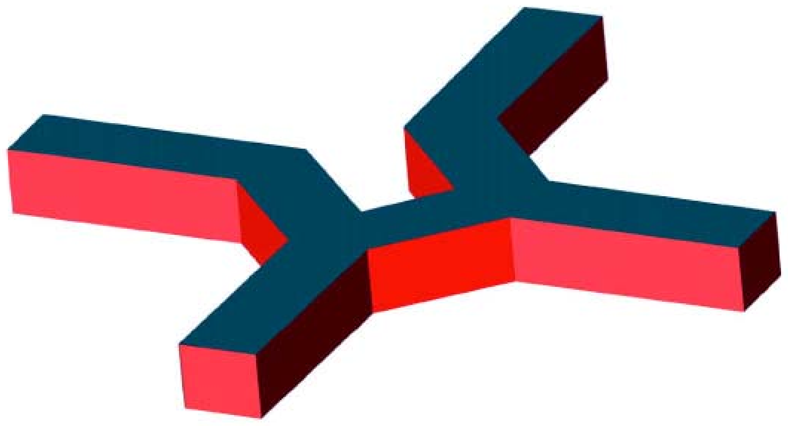
```
layer
make 0
color t 0,255,255 0

3dface 37.34,-47.44,202.30
9.56,-19.31,202.30
18.78,-9.13,202.30
47.12,-37.80,202.30

3dface -15.62,-19.43,202.30
-43.79,-48.42,202.30
-53.33,-38.96,202.30
-21.05,-5.71,202.30

3dface 9.56,-19.31,202.30
-15.62,-19.43,202.30
-21.05,-5.71,202.30
5.04,-5.50,202.30

3dface 18.78,-9.13,202.30
9.56,-19.31,202.30
```

```
5.04,-5.50,202.30
4.45,10.66,202.30

3dface 18.61,14.97,202.30
18.78,-9.13,202.30
4.45,10.66,202.30
8.77,25.00,202.30

3dface 8.77,25.00,202.30
4.45,10.66,202.30
-21.30,10.81,202.30
-17.04,25.07,202.30

3dface -17.04,25.07,202.30
-21.30,10.81,202.30
-53.96,42.52,202.30
-44.79,52.62,202.30

3dface 36.87,53.61,202.30
46.70,43.87,202.30
18.61,14.97,202.30
8.77,25.00,202.30

layer
make 1
color t 255,0,0 1

3dface 37.34,-47.44,202.30
9.56,-19.31,202.30
9.56,-19.31,191.00
37.34,-47.44,191.00

3dface 9.56,-19.31,202.30
-15.62,-19.43,202.30
-15.62,-19.43,191.00
9.56,-19.31,191.00

3dface -15.62,-19.43,202.30
-43.79,-48.42,202.30
-43.79,-48.42,191.00
-15.62,-19.43,191.00

3dface -43.79,-48.42,202.30
-53.33,-38.96,202.30
-53.33,-38.96,191.00
-43.79,-48.42,191.00

3dface -53.33,-38.96,202.30
-21.05,-5.71,202.30
-21.05,-5.71,191.00
-53.33,-38.96,191.00

3dface -21.05,-5.71,202.30
5.04,-5.50,202.30
5.04,-5.50,191.00
-21.05,-5.71,191.00
```

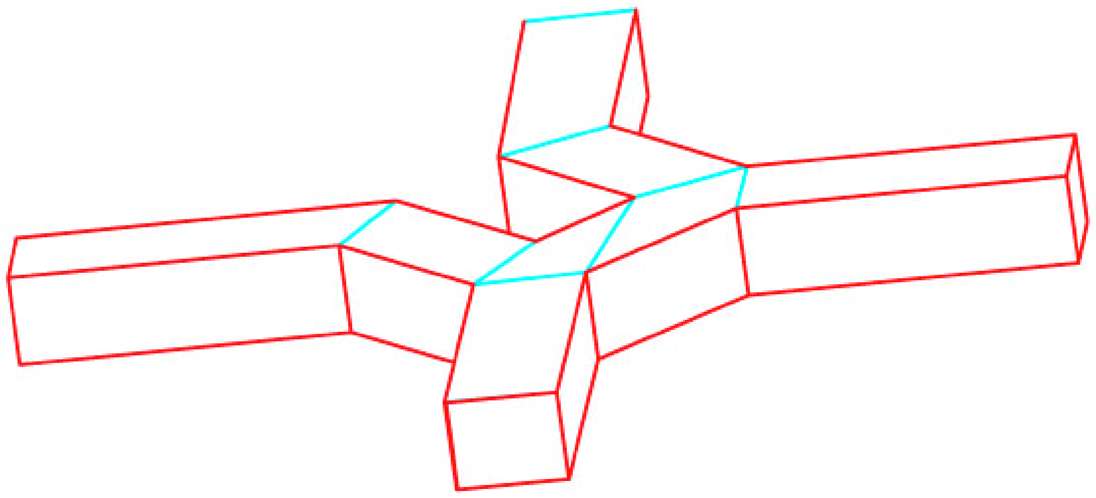
```
3dface 5.04,-5.50,202.30
4.45,10.66,202.30
4.45,10.66,191.00
5.04,-5.50,191.00

3dface 4.45,10.66,202.30
-21.30,10.81,202.30
-21.30,10.81,191.00
4.45,10.66,191.00

3dface -21.30,10.81,202.30
-53.96,42.52,202.30
-53.96,42.52,191.00
-21.30,10.81,191.00

3dface -53.96,42.52,202.30
-44.79,52.62,202.30
-44.79,52.62,191.00
-53.96,42.52,191.00

3dface -44.79,52.62,202.30
-17.04,25.07,202.30
-17.04,25.07,191.00
-44.79,52.62,191.00

3dface -17.04,25.07,202.30
8.77,25.00,202.30
8.77,25.00,191.00
-17.04,25.07,191.00

3dface 8.77,25.00,202.30
36.87,53.61,202.30
36.87,53.61,191.00
8.77,25.00,191.00

3dface 36.87,53.61,202.30
46.70,43.87,202.30
46.70,43.87,191.00
36.87,53.61,191.00

3dface 46.70,43.87,202.30
18.61,14.97,202.30
18.61,14.97,191.00
46.70,43.87,191.00

3dface 18.61,14.97,202.30
18.78,-9.13,202.30
18.78,-9.13,191.00
18.61,14.97,191.00

3dface 18.78,-9.13,202.30
47.12,-37.80,202.30
47.12,-37.80,191.00
18.78,-9.13,191.00
```

```
3dface 47.12,-37.80,202.30
37.62,-47.43,202.30
37.62,-47.43,191.00
47.12,-37.80,191.00
```

The resulting 3D object in Autocad , rendered using the view / flat shading menu item appears as follows (in this case aspect ratio is correct):



To show the subdivision of the roof polygon, the rendering was done using the “hidden” (wireframe) command as shown here:



VRML Script

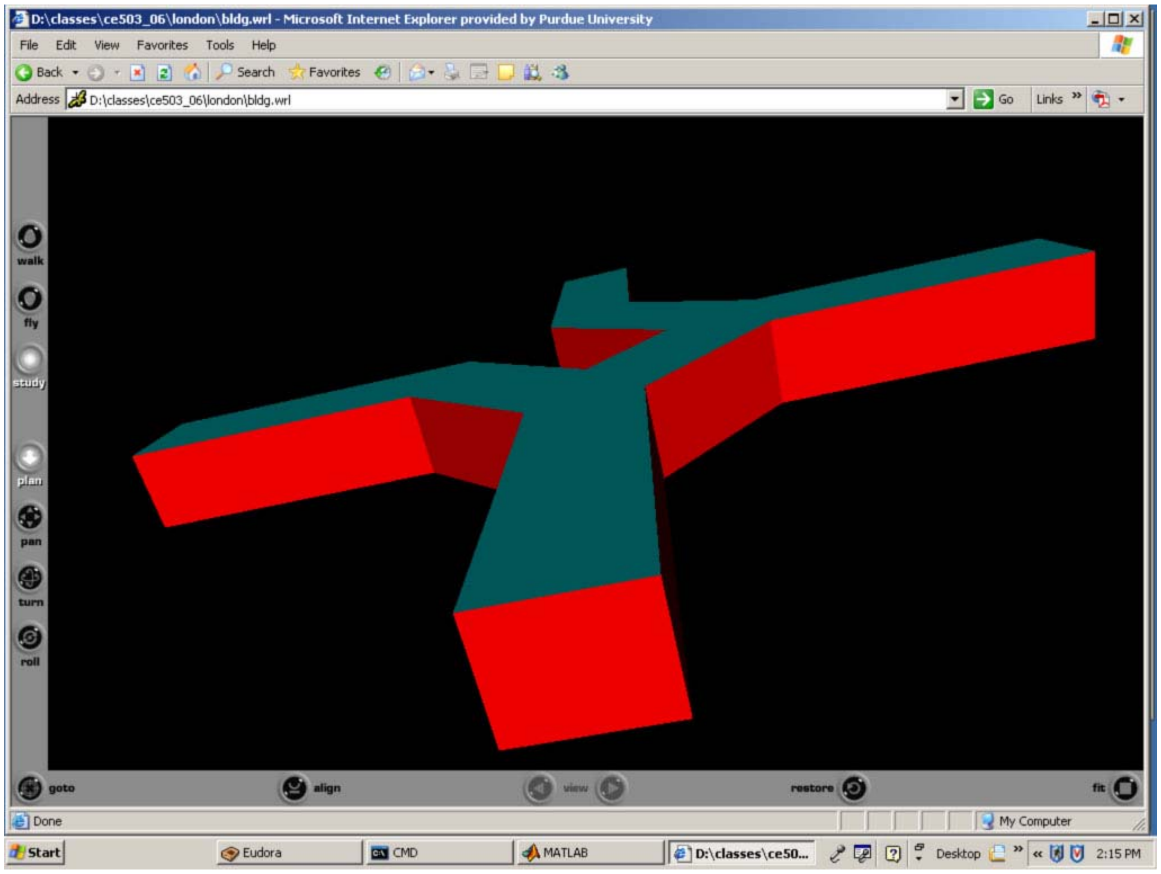
Another Matlab file was created to generate a VRML script. If your web browser has a VRML plugin you can view and navigate around rendered objects. Recommend Cortona VRML client, freeware download at parallelgraphics.com. Good reference for VRML syntax is VRML 2.0 Sourcebook, by Ames, Nadeau, and Moreland, John Wiley & Sons. The Matlab file:

```
% mk_vrml.m  3-oct-06
% make vrml code for input of building faces

zhi=202.3;
zlo=191.0;
load mdth.txt;
[m n]=size(mdth);
X=mdth(:,1);
Y=mdth(:,2);
Z=zeros(m,1);
for i=1:m
  Z(i)=zhi;
  end
Xm=mean(X);
Ym=mean(Y);
X=X-Xm;
Y=Y-Ym;
% these points were originally read in cw-order, let's
% reverse into ccw order
Xtmp=zeros(m,1);
Ytmp=zeros(m,1);
Ztmp=zeros(m,1);
for i=1:m
  Xtmp(i)=X(m-i+1);
  Ytmp(i)=Y(m-i+1);
  Ztmp(i)=Z(m-i+1);
  end
X=Xtmp;
Y=Ytmp;
Z=Ztmp;

% write intro stuff for vrml file
% see page 231 in vrml source book ames, nadeau, moreland

fid=fopen('bldg.wrl','wt');
fprintf(fid,'#VRML V2.0 utf8\n');
fprintf(fid,'Shape {\n');
fprintf(fid,'    appearance Appearance {\n');
fprintf(fid,'        material Material { }\n');
fprintf(fid,'}\n');
fprintf(fid,'    geometry IndexedFaceSet {\n');
fprintf(fid,'        coord Coordinate {\n');
fprintf(fid,'            point [\n');
fprintf(fid,'            # coordinates around the top of building\n');
for i=1:m
  fprintf(fid,'                %10.2f %10.2f %10.2f,\n',X(i),Y(i),zhi);
  end
fprintf(fid,'            # coordinates around the bottom of
building\n');
for i=1:m
  fprintf(fid,'                %10.2f %10.2f %10.2f,\n',X(i),Y(i),zlo);
  end
fprintf(fid,'            ]\n');
fprintf(fid,'        }\n');
fprintf(fid,'        coordIndex [\n');
for i=1:m-1
  j=i-1;
  fprintf(fid,'%3d,',j);
  end
fprintf(fid,'%3d,\n',-1);
for i=1:m-1
  j1=i;
  j2=i-1;
  j3=m+i-1;
  j4=m+i-1+1;
  if(i ~= m-1)
    fprintf(fid,'%3d,%3d,%3d,%3d,%3d,\n',j1,j2,j3,j4,-1);
  else
    fprintf(fid,'%3d,%3d,%3d,%3d\n',j1,j2,j3,j4);
    end
  end
fprintf(fid,'        ]\n');
fprintf(fid,'        convex FALSE\n');
fprintf(fid,'        colorPerVertex FALSE\n');
fprintf(fid,'        color Color {\n');
fprintf(fid,'            color [\n');
fprintf(fid,'                0.0 1.0 1.0, # cyan\n');
fprintf(fid,'                1.0 0.0 0.0  # red\n');
fprintf(fid,'            ]\n');
fprintf(fid,'        }\n');
fprintf(fid,'        colorIndex [\n');
fprintf(fid,'            0, # top\n');
for i=1:m-1
  if(i ~= m-1)
    fprintf(fid,'%2d,',1);
  else
    fprintf(fid,'%2d # sides\n',1);
    end
  end
fprintf(fid,'        ]\n');
fprintf(fid,'    }\n');
fprintf(fid,'}\n');
fclose(fid);
```

The generated VRML code, note convex assumption is disabled,

```
#VRML V2.0 utf8
Shape {
    appearance Appearance {
        material Material { }
}
    geometry IndexedFaceSet {
        coord Coordinate {
            point [
            # coordinates around the top of building
                     37.62     -47.43     202.30,
                     47.12     -37.80     202.30,
                     18.78      -9.13     202.30,
                     18.61      14.97     202.30,
                     46.70      43.87     202.30,
                     36.87      53.61     202.30,
                      8.77      25.00     202.30,
                    -17.04      25.07     202.30,
                    -44.79      52.62     202.30,
                    -53.96      42.52     202.30,
                    -21.30      10.81     202.30,
                      4.45      10.66     202.30,
                      5.04      -5.50     202.30,
                    -21.05      -5.71     202.30,
                    -53.33     -38.96     202.30,
                    -43.79     -48.42     202.30,
                    -15.62     -19.43     202.30,
                      9.56     -19.31     202.30,
                     37.34     -47.44     202.30,
            # coordinates around the bottom of building
                     37.62     -47.43     191.00,
                     47.12     -37.80     191.00,
                     18.78      -9.13     191.00,
                     18.61      14.97     191.00,
                     46.70      43.87     191.00,
                     36.87      53.61     191.00,
                      8.77      25.00     191.00,
                    -17.04      25.07     191.00,
                    -44.79      52.62     191.00,
                    -53.96      42.52     191.00,
                    -21.30      10.81     191.00,
                      4.45      10.66     191.00,
                      5.04      -5.50     191.00,
                    -21.05      -5.71     191.00,
                    -53.33     -38.96     191.00,
                    -43.79     -48.42     191.00,
                    -15.62     -19.43     191.00,
                      9.56     -19.31     191.00,
                     37.34     -47.44     191.00,
            ]
        }
        coordIndex [
  0,  1,  2,  3,  4,  5,  6,  7,  8,  9, 10, 11, 12, 13, 14, 15, 16,
17, -1,
  1,  0, 19, 20, -1,
  2,  1, 20, 21, -1,
  3,  2, 21, 22, -1,
  4,  3, 22, 23, -1,
  5,  4, 23, 24, -1,
  6,  5, 24, 25, -1,
  7,  6, 25, 26, -1,
  8,  7, 26, 27, -1,
  9,  8, 27, 28, -1,
 10,  9, 28, 29, -1,
 11, 10, 29, 30, -1,
 12, 11, 30, 31, -1,
 13, 12, 31, 32, -1,
 14, 13, 32, 33, -1,
 15, 14, 33, 34, -1,
 16, 15, 34, 35, -1,
 17, 16, 35, 36, -1,
 18, 17, 36, 37
        ]
        convex FALSE
        colorPerVertex FALSE
        color Color {
            color [
                0.0 1.0 1.0, # cyan
                1.0 0.0 0.0  # red
            ]
        }
        colorIndex [
            0, # top
 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1 # sides
        ]
    }
}
```

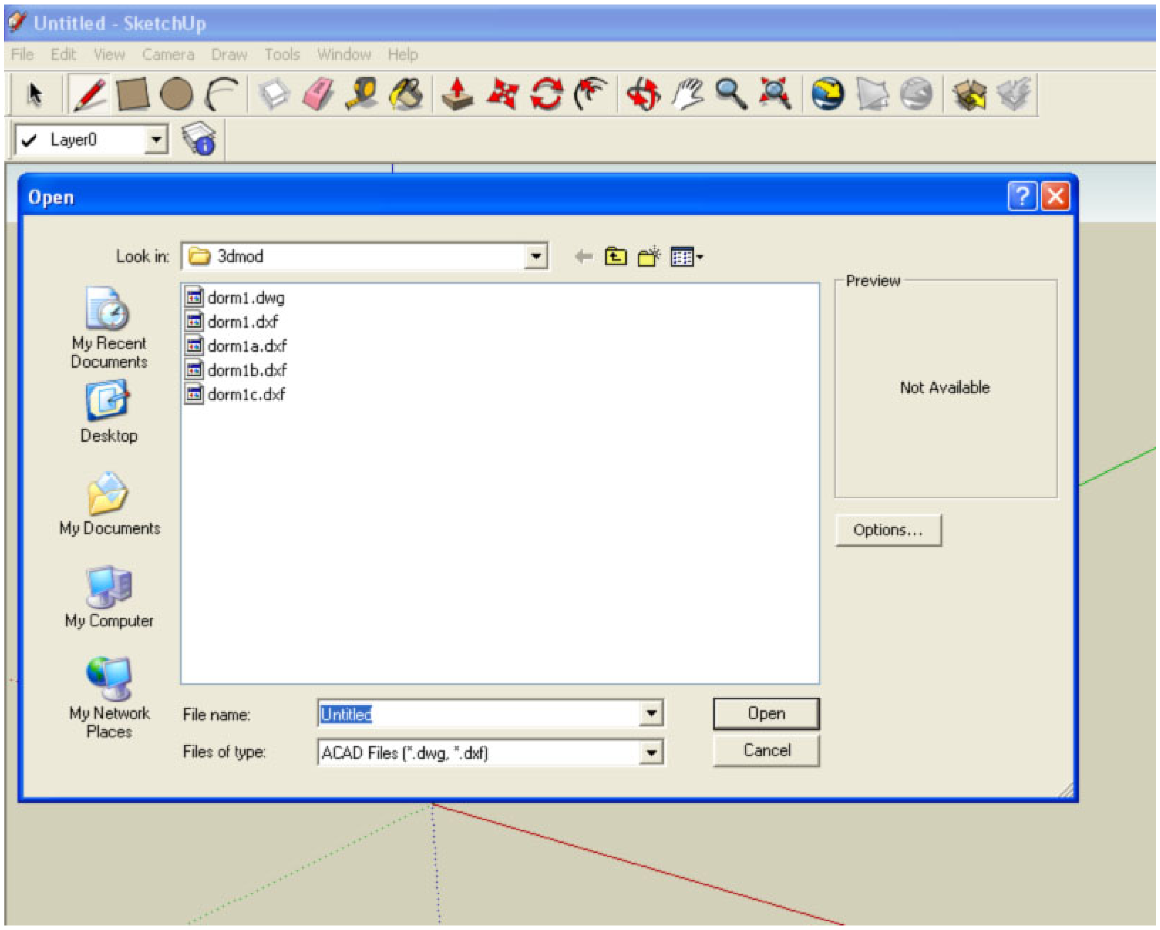
And the building object rendered by the web browser plugin:


D:\classes\ce503_06\london\bldg.wrl - Microsoft Internet Explorer provided by Purdue University
File Edit View Favorites Tools Help
Back
Search
Favorites
Address D:\classes\ce503_06\london\bldg.wrl
Go
Links
walk
fly
study
plan
pan
turn
roll
goto
align
view
restore
fit
Done
My Computer
Start
Eudora
CMD
MATLAB
D:\classes\ce50...
Desktop
2:15 PM

Google Sketchup

Data entry for Google Sketchup is primarily via a manual "sketch" mode, although you can also import a limited set of CAD or geometry files. These include Autocad .dwg and .dxf, 3DStudio Max .3ds, and USGS DEM files. The current data import tool in sketchup seems to only recognize older Autocad files. Fortunately Autocad 2007 allows you to save data in earlier formats, the oldest being R12. The USGS DEM format is ascii, also recognized is the SDTS DEM file. In the documentation for sketchup it says that they support the RUBY API to allow users to customize and extend the menus and functionality of sketchup. I would recommend somebody create an ascii face geometry importer. By the way, you can download the vanilla version of sketchup for free from www.earth.google.com .

Anyway, to demonstrate rendering capability in sketchup the previously shown Autocad data was saved to an old .dxf format. This was imported via the following screen:



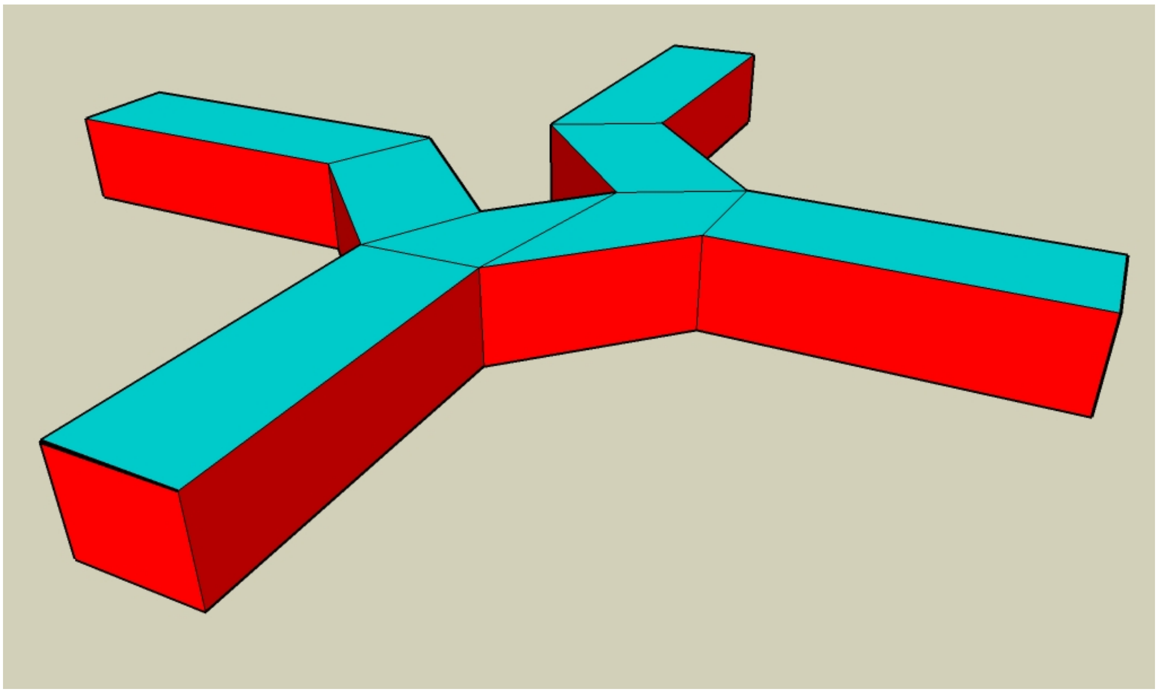
Look in: 3dmod

dorm1.dwg
dorm1.dxf
dorm1a.dxf
dorm1b.dxf
dorm1c.dxf

File name: Untitled

Files of type: ACAD Files (*.dwg, *.dxf)

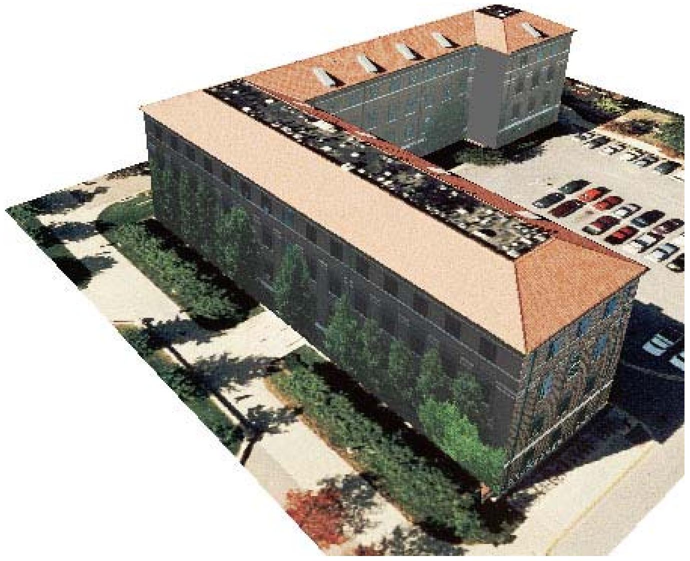
After toggling the “color by layer” switch, the following view was produced in the sketchup display



It appears that such CAD models can be easily merged into Google Earth, and also exported as KML files. Some of the export options are only available with the PRO version which costs a little money, not much.

Image Based Rendering

There are a number of environments in which one can render and view 3D objects. Described above are Matlab graphics, Autocad, and VRML. Omitted but needing inclusion are ESRI 3D shape files and ESRI & Erdas rendering tools, although they seem to be more proprietary and less open than the techniques presented here. Also relevant to this topic are image-based rendering and projective texture mapping for realistic looking displays. That is, don't just paint the polygons with a generic color, but rigorously project photo pixels onto the polygons. To do that you need accurate 3D wireframe feature geometry and triangulated imagery from a variety of viewpoints. Then you get some more challenges like hidden faces, and the inconsistency of the 3D model and the projected image / texture. Commercial tools to automate projective texture mapping are not common. Example from the "old" Chem-E building, note the flattened cars and trees where no model geometry exists.



Photogrammetry is one of the essential tools for creating 3D models. Other alternatives include (a) (for small objects) a 3D coordinate measuring machine, (b) optical surveying techniques with multiple theodolite stations, (c) creation of models from design plans, large scale maps, or as-built drawings, and (d) laser scanning. Photogrammetry is often the easiest, lowest cost, and in many cases the only practical approach to generation of accurate 3D models.