

Improving Blender Data Loading

Brian R. Kent, NRAO

<http://www.cv.nrao.edu/~bkent/computing/kentPASP.html>

We use simple object duplication in the catalog and n-body tutorials - this is an easy way to introduce the interface and concepts to a new user. For large N datasets, this can significantly increase loading time. An improved approach is to use vertices in the same object. This can be accomplished in a number of ways:

```
import bpy
import math
import bmesh
import csv

#Use any primitive object and reduce it to a single vertex
obj = bpy.data.objects['Circle']

if obj.mode == 'EDIT':
    bm = bmesh.from_edit_mesh(obj.data)
    for v in bm.verts:
        if v.select:
            print(v.co)
else:
    print("Object is not in edit mode")

bm = bmesh.from_edit_mesh(obj.data)

#Read in data
filename = 'data.csv'
fields = ['Name', 'X', 'Y', 'Z']
reader = csv.DictReader(open(filename), fields, delimiter=',')

dicts = []

#Skip first header line
next(reader)
for row in reader:
    dicts.append(row)

#Add in vertex elements with XYZ coordinates at each row
for row in dicts:
    xpos = float(row['X'])
    ypos = float(row['Y'])
    zpos = float(row['Z'])
    bm.verts.new((xpos, ypos, zpos))

bmesh.update_edit_mesh(obj.data)
```

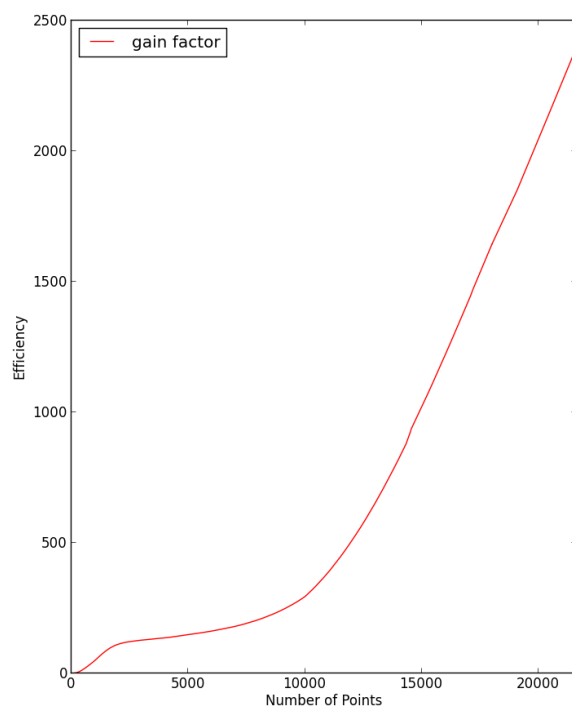
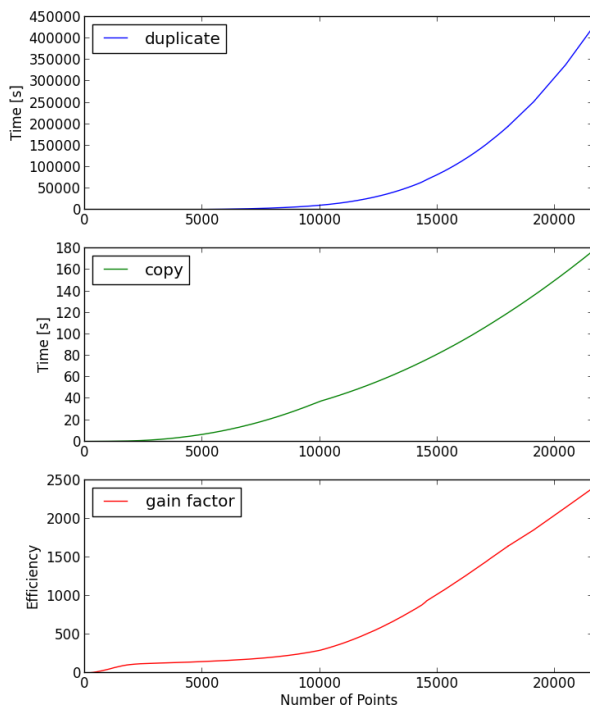
You can also use the copy method, as illustrated below. A huge thanks to Ester Ramos Carmona and Salvador Duarte Puertas at the Departamento de Física Teórica y del Cosmos of University of Granada for their analysis and plots of improving the loading times:

```
import bpy

with open("data.txt") as f:
    lis=[x.split() for x in f]
    cols=[x for x in zip(*lis)]

sample = 10000
x, y, z = cols[0][:sample], cols[1][:sample], cols[2][:sample]
scale = 100
bpy.ops.object.select_all(action='DESELECT')
galaxy = bpy.context.active_object

for i in range(len(x)):
    ob = galaxy.copy()
    ob.data = galaxy.data.copy()
    bpy.context.scene.objects.link(ob)
    galaxy.location.xyz=(float(x[i])*scale, float(y[i])*scale,
float(z[i])*scale)
bpy.context.scene.update()
```



Links:

<http://stackoverflow.com/questions/11110621/blender-performance-issues-with-generating-uv-spheres/>

Further examples and the publication that goes with this tutorial can be found at:

<http://www.cv.nrao.edu/~bkent/computing/kentPASP.html>