Article

objectVisual3D: A standalone executable software for 3D visualization of objects

WenJun Zhang
School of Life Sciences, Sun Yat-sen University, Guangzhou, China
E-mail: zhwj@mail.sysu.edu.cn, wjzhang@iaees.org

Received 1 June 2023; *Accepted 20 June 2023*; *Published online 1 July 2023*; *Published 1 December 2024*

Abstract
A standalone executable software, objectVisual3D (version 1.0), for 3D visualization of objects, was developed in present study. The software uses the OBJ file of an object to generate its 3D graphics. Various parameters can be specified by users. In the generated 3D graphics window, users can right- or left-click mouse to zoom in or zoom out the 3D graphics, or mouse-drag the graphics to rotate the 3D graphics, or slide scrollbar to vertically or horizontally translate the 3D graphics. The 3D graphics can be saved as an image file (in BMP format). Both objectVisual3D and demonstration data files were given.

Keywords 3D visualization; objects; standalone software.

Selforganizology
ISSN 2410-0080
URL: http://www.iaees.org/publications/journals/selforganizology/online-version.asp
RSS: http://www.iaees.org/publications/journals/selforganizology /rss.xml
E-mail: selforganizology@iaees.org
Editor-in-Chief: WenJun Zhang
Publisher: International Academy of Ecology and Environmental Sciences

1 Introduction

Digital visualization is important for scientific research and applications (Narad et al., 2017). Various visualization software, e.g., the software for network visualization, were developed in the past (Zhang, 2007, 2021, 2023, 2024a-f). Some of them were developed as Java tools. In particular, I have presented a Java software for 3D visualization of objects and molecules based on JDK 1.1 and J2SDK 1.4.2 (Zhang, 2023). Java is powerful for developing platform independent tools and thus be widely used. However, the JRE is needed to support the software. Delphi is the development environment for standalone software and I have used it to develop some tools in recent days. In present study, a standalone executable software, objectVisual3D (version 1.0), for 3D visualization of objects was developed. The software uses the OBJ file of an object to generate its 3D graphics. Full Delphi codes, the software and demonstration data files are given.

2 Software and User Guide

2.1 Software

The standalone executable software, objectVisual3D, was developed using Delphi (Fig. 1), based on the previous Java software (Zhang, 2023). The following are the full Delphi codes of the software:

```
unit Unit1;

interface

uses
  Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms, Dialogs, StdCtrls;

type
  TForm1 = class(TForm)
    Label1: TLabel;
    Label2: TLabel;
    Label3: TLabel;
    Label4: TLabel;
    Label5: TLabel;
    Button1: TButton;
    Button2: TButton;
    Edit1: TEdit;
    Edit2: TEdit;
    Edit3: TEdit;
    Edit4: TEdit;
    Edit5: TEdit;
    procedure Button2Click(Sender: TObject);
    procedure Button1Click(Sender: TObject);
  private
    { Private declarations }
  public
    { Public declarations }
  end;

var
  Form1: TForm1;
  scale,dila: single;
  maxscroll: Integer;
  scrollverpos, scrollhorpos: integer;

implementation

uses Unit2;

{$R *.dfm}

procedure TForm1.Button2Click(Sender: TObject);
begin
Application.Terminate;
```

```
end;

procedure TForm1.Button1Click(Sender: TObject);
begin
scale:=strtofloat(edit1.Text);
dila:=strtofloat(edit2.Text);
maxscroll:=strtoint(edit3.Text);
form2.width:=strtoint(edit4.Text);
form2.height:=strtoint(edit5.Text);
form2.scrollbar1.max:=maxscroll;
form2.scrollbar2.max:=maxscroll;
scrollverpos:=0;
scrollhorpos:=0;
form1.visible:=false;
form2.visible:=true;
Application.BringToFront;
end;

end.

unit Unit2;

interface

uses
  Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms, Dialogs, StdCtrls, math, ExtCtrls, ExtDlgs;

type
  TForm2 = class(TForm)
    SavePictureDialog1: TSavePictureDialog;
    Button1: TButton;
    OpenDialog1: TOpenDialog;
    PaintBox1: TPaintBox;
    ScrollBar1: TScrollBar;
    ScrollBar2: TScrollBar;
    procedure Button1Click(Sender: TObject);
    procedure PaintBox1MouseUp(Sender: TObject; Button: TMouseButton; Shift: TShiftState; X, Y: Integer);
    procedure FormShow(Sender: TObject);
    procedure FormResize(Sender: TObject);
    procedure PaintBox1DragOver(Sender, Source: TObject; X, Y: Integer; State: TDragState; var Accept: Boolean);
    procedure PaintBox1DragDrop(Sender, Source: TObject; X, Y: Integer);
    procedure FormClose(Sender: TObject; var Action: TCloseAction);
    procedure ScrollBar1Change(Sender: TObject);
    procedure ScrollBar2Change(Sender: TObject);
  private
```

```
    { Private declarations }
  public
    { Public declarations }
  end;

var
    Form2: TForm2;
    xx, xy, xz, xo, yx, yy, yz, yo, zx, zy, zz, zo: single;
    axx, axy, axz, axo, ayx, ayy, ayz, ayo, azx, azy, azz, azo: single;
    txx, txy, txz, txo, tyx, tyy, tyz, tyo, tzx, tzy, tzz, tzo: single;
    xmin, xmax, ymin, ymax, zmin, zmax: single;
    xfa, xfac, xfac0: single;
    nvert, ncon, maxcon, maxvert: integer;
    vert: array of single;
    tvert, con: array of integer;
    prevx, prevy: integer;
    transformed: boolean;
    filename: string;
    res: array of single;
    mres: integer;
    difx, dify, poshor, posver, scrollverpos, scrollhorpos: integer;

implementation

uses Unit1;

{$R *.dfm}

//Mat3D

procedure Mat3D();
begin
xx:=1.0;
yy:=1.0;
zz:=1.0;
end;

procedure amatMat3D();
begin
axx:=1.0;
ayy:=1.0;
azz:=1.0;
end;

procedure tmatMat3D();
```

```
begin
txx:=1.0;
tyy:=1.0;
tzz:=1.0;
end;

procedure scalef(f: single);
begin
xx:=xx*f;
xy:=xy*f;
xz:=xz*f;
xo:=xo*f;
yx:=yx*f;
yy:=yy*f;
yz:=yz*f;
yo:=yo*f;
zx:=zx*f;
zy:=zy*f;
zz:=zz*f;
zo:=zo*f;
end;

procedure scalefff(f: single; f1: single; f2: single);
begin
xx:=xx*f;
xy:=xy*f;
xz:=xz*f;
xo:=xo*f;
yx:=yx*f1;
yy:=yy*f1;
yz:=yz*f1;
yo:=yo*f1;
zx:=zx*f2;
zy:=zy*f2;
zz:=zz*f2;
zo:=zo*f2;
end;

procedure translate(f: single; f1: single; f2: single);
begin
xo:=xo+f;
yo:=yo+f1;
zo:=zo+f2;
end;
```

```
procedure amattranslate(f: single; f1: single; f2: single);
begin
axo:=axo+f;
ayo:=ayo+f1;
azo:=azo+f2;
end;

procedure xrot(d: single);
var
    d1,d2: single;
    f, f1, f2, f3, f4, f5, f6, f7: single;
begin
d:=d*0.017453292500000002;
d1:=cos(d);
d2:=sin(d);
f:=yx*d1+zx*d2;
f1:=yy*d1+zy*d2;
f2:=yz*d1+zz*d2;
f3:=yo*d1+zo*d2;
f4:=zx*d1-yx*d2;
f5:=zy*d1-yy*d2;
f6:=zz*d1-yz*d2;
f7:=zo*d1-yo*d2;
yo:=f3;
yx:=f;
yy:=f1;
yz:=f2;
zo:=f7;
zx:=f4;
zy:=f5;
zz:=f6;
end;

procedure yrot(d: single);
var
    d1,d2: single;
    f, f1, f2, f3, f4, f5, f6, f7: single;
begin
d:=d*0.017453292500000002;
d1:=cos(d);
d2:=sin(d);
f:=xx*d1+zx*d2;
f1:=xy*d1+zy*d2;
f2:=xz*d1+zz*d2;
f3:=xo*d1+zo*d2;
```

```
f4:=zx*d1-xx*d2;
f5:=zy*d1-xy*d2;
f6:=zz*d1-xz*d2;
f7:=zo*d1-xo*d2;
xo:=f3;
xx:=f;
xy:=f1;
xz:=f2;
zo:=f7;
zx:=f4;
zy:=f5;
zz:=f6;
end;

procedure zrot(d: single);
var
    d1,d2: single;
    f, f1, f2, f3, f4, f5, f6, f7: single;
begin
d:=d*0.017453292500000002;
d1:=cos(d);
d2:=sin(d);
f:=yx*d1+xx*d2;
f1:=yy*d1+xy*d2;
f2:=yz*d1+xz*d2;
f3:=yo*d1+xo*d2;
f4:=xx*d1-yx*d2;
f5:=xy*d1-yy*d2;
f6:=xz*d1-yz*d2;
f7:=xo*d1-yo*d2;
yo:=f3;
yx:=f;
yy:=f1;
yz:=f2;
xo:=f7;
xx:=f4;
xy:=f5;
xz:=f6;
end;

procedure amatxrot(d: single);
var
    d1,d2: single;
    f, f1, f2, f3, f4, f5, f6, f7: single;
begin
```

```
d:=d*0.017453292500000002;
d1:=cos(d);
d2:=sin(d);
f:=ayx*d1+azx*d2;
f1:=ayy*d1+azy*d2;
f2:=ayz*d1+azz*d2;
f3:=ayo*d1+azo*d2;
f4:=azx*d1-ayx*d2;
f5:=azy*d1-ayy*d2;
f6:=azz*d1-ayz*d2;
f7:=azo*d1-ayo*d2;
ayo:=f3;
ayx:=f;
ayy:=f1;
ayz:=f2;
azo:=f7;
azx:=f4;
azy:=f5;
azz:=f6;
end;

procedure amatyrot(d: single);
var
    d1,d2: single;
    f, f1, f2, f3, f4, f5, f6, f7: single;
begin
d:=d*0.017453292500000002;
d1:=cos(d);
d2:=sin(d);
f:=axx*d1+azx*d2;
f1:=axy*d1+azy*d2;
f2:=axz*d1+azz*d2;
f3:=axo*d1+azo*d2;
f4:=azx*d1-axx*d2;
f5:=azy*d1-axy*d2;
f6:=azz*d1-axz*d2;
f7:=azo*d1-axo*d2;
axo:=f3;
axx:=f;
axy:=f1;
axz:=f2;
azo:=f7;
azx:=f4;
azy:=f5;
azz:=f6;
```

```
end;

procedure tmatxrot(d: single);
var
    d1,d2: single;
    f, f1, f2, f3, f4, f5, f6, f7: single;
begin
d:=d*0.017453292500000002;
d1:=cos(d);
d2:=sin(d);
f:=tyx*d1+tzx*d2;
f1:=tyy*d1+tzy*d2;
f2:=tyz*d1+tzz*d2;
f3:=tyo*d1+tzo*d2;
f4:=tzx*d1-tyx*d2;
f5:=tzy*d1-tyy*d2;
f6:=tzz*d1-tyz*d2;
f7:=tzo*d1-tyo*d2;
tyo:=f3;
tyx:=f;
tyy:=f1;
tyz:=f2;
tzo:=f7;
tzx:=f4;
tzy:=f5;
tzz:=f6;
end;

procedure tmatyrot(d: single);
var
    d1,d2: single;
    f, f1, f2, f3, f4, f5, f6, f7: single;
begin
d:=d*0.017453292500000002;
d1:=cos(d);
d2:=sin(d);
f:=txx*d1+tzx*d2;
f1:=txy*d1+tzy*d2;
f2:=txz*d1+tzz*d2;
f3:=txo*d1+tzo*d2;
f4:=tzx*d1-txx*d2;
f5:=tzy*d1-txy*d2;
f6:=tzz*d1-txz*d2;
f7:=tzo*d1-txo*d2;
txo:=f3;
```

```
txx:=f;
txy:=f1;
txz:=f2;
tzo:=f7;
tzx:=f4;
tzy:=f5;
tzz:=f6;
end;

procedure mult(xx1: single; xy1: single; xz1: single; xo1: single; yx1: single; yy1: single; yz1: single; yo1: single; zx1: single; zy1: single; zz1: single; zo1: single);
var
    f, f1, f2, f3, f4, f5, f6, f7, f8, f9, f10, f11: single;
begin
f:=xx*xx1+yx*xy1+zx*xz1;
f1:=xy*xx1+yy*xy1+zy*xz1;
f2:=xz*xx1+yz*xy1+zz*xz1;
f3:=xo*xx1+yo*xy1+zo*xz1+xo1;
f4:=xx*yx1+yx*yy1+zx*yz1;
f5:=xy*yx1+yy*yy1+zy*yz1;
f6:=xz*yx1+yz*yy1+zz*yz1;
f7:=xo*yx1+yo*yy1+zo*yz1+yo1;
f8:=xx*zx1+yx*zy1+zx*zz1;
f9:=xy*zx1+yy*zy1+zy*zz1;
f10:=xz*zx1+yz*zy1+zz*zz1;
f11:=xo*zx1+yo*zy1+zo*zz1+zo1;
xx:=f;
xy:=f1;
xz:=f2;
xo:=f3;
yx:=f4;
yy:=f5;
yz:=f6;
yo:=f7;
zx:=f8;
zy:=f9;
zz:=f10;
zo:=f11;
end;

procedure amatmult(xx1: single; xy1: single; xz1: single; xo1: single; yx1: single; yy1: single; yz1: single; yo1: single; zx1: single; zy1: single; zz1: single; zo1: single);
var
    f, f1, f2, f3, f4, f5, f6, f7, f8, f9, f10, f11: single;
begin
```

```
f:=axx*xx1+ayx*xy1+azx*xz1;
f1:=axy*xx1+ayy*xy1+azy*xz1;
f2:=axz*xx1+ayz*xy1+azz*xz1;
f3:=axo*xx1+ayo*xy1+azo*xz1+xo1;
f4:=axx*yx1+ayx*yy1+azx*yz1;
f5:=axy*yx1+ayy*yy1+azy*yz1;
f6:=axz*yx1+ayz*yy1+azz*yz1;
f7:=axo*yx1+ayo*yy1+azo*yz1+yo1;
f8:=axx*zx1+ayx*zy1+azx*zz1;
f9:=axy*zx1+ayy*zy1+azy*zz1;
f10:=axz*zx1+ayz*zy1+azz*zz1;
f11:=axo*zx1+ayo*zy1+azo*zz1+zo1;
axx:=f;
axy:=f1;
axz:=f2;
axo:=f3;
ayx:=f4;
ayy:=f5;
ayz:=f6;
ayo:=f7;
azx:=f8;
azy:=f9;
azz:=f10;
azo:=f11;
end;

procedure transmat(var af: array of single; var ai: array of integer; nvert: integer);
var
    j: integer;
    f, f1, f2, f3, f4, f5, f6, f7, f8, f9, f10, f11, f12, f13, f14: single;
begin
f:=xx;
f1:=xy;
f2:=xz;
f3:=xo;
f4:=yx;
f5:=yy;
f6:=yz;
f7:=yo;
f8:=zx;
f9:=zy;
f10:=zz;
f11:=zo;
j:=nvert*3;
while(j>=0) do
```

```
begin
f12:=af[j];
f13:=af[j+1];
f14:=af[j+2];
ai[j]:=floor(f12*f+f13*f1+f14*f2+f3);
ai[j+1]:=floor(f12*f4+f13*f5+f14*f6+f7);
ai[j+2]:=floor(f12*f8+f13*f9+f14*f10+f11);
j:=j-3;
end;
end;

procedure units();
begin
xo:=0.0;
xx:=1.0;
xy:=0.0;
xz:=0.0;
yo:=0.0;
yx:=0.0;
yy:=1.0;
yz:=0.0;
zo:=0.0;
zx:=0.0;
zy:=0.0;
zz:=1.0;
end;

procedure tmatunits();
begin
txo:=0.0;
txx:=1.0;
txy:=0.0;
txz:=0.0;
tyo:=0.0;
tyx:=0.0;
tyy:=1.0;
tyz:=0.0;
tzo:=0.0;
tzx:=0.0;
tzy:=0.0;
tzz:=1.0;
end;

function toString(): String;
begin
```

```
result:='['+floattostr(xo)+','+floattostr(xx)+','+floattostr(xy)+','+floattostr(xz)+';'+floattostr(yo)+','+floattostr(yx)+','+floattostr(yy)
+','+floattostr(yz)+';'+floattostr(zo)+','+floattostr(zx)+','+floattostr(zy)+','+floattostr(zz)+']';
end;

procedure Model3D();
begin
Mat3D();
xrot(20.0);
yrot(30.0);
end;

procedure addVert(f: single; f1: single; f2: single);
var
    i, j: integer;
    af: array of single;
begin
i:=nvert;
if(length(vert)=0) then
begin
maxvert:=100000;
setLength(vert,maxvert*6);
end
else
begin
setLength(af,maxvert*6);
for j:=0 to length(vert)-1 do
af[j]:=vert[j];
end;
i:=i*3;
vert[i]:=f;
vert[i+1]:=f1;
vert[i+2]:=f2;
af:=nil;
nvert:=nvert+1;
end;

procedure add(i: integer; j: integer);
var
    k, l: integer;
begin
k:=ncon;
if((i>=nvert) or (j>=nvert)) then
exit;
if(length(con)=0) then
begin
```

```
maxcon:=100000;
setLength(con,maxcon);
end
else
begin
end;
if(i>j) then
begin
l:=i;
i:=j;
j:=l;
end;
con[k]:=(i shl 16) or j;
ncon:=k+1;
end;

procedure transform();
begin
if((transformed=true) or (nvert=0)) then
exit;
if(length(tvert)<nvert*6) then
setLength(tvert,nvert*6);
transmat(vert,tvert,nvert);
transformed:=true;
end;

procedure swap(var ai: array of integer; i: integer; j: integer);
var
    k: integer;
begin
k:=ai[i];
ai[i]:=ai[j];
ai[j]:=k;
end;

procedure quickSort(var ai: array of integer; i: integer; j: integer);
var
    k, l, i1: integer;
begin
k:=i;
l:=j;
if(j>i) then
begin
i1:=ai[floor((i+j)/2)];
while(k<=l) do
```

```
begin
while((k<j) and (ai[k]<i1)) do
k:=k+1;
while((l>i) and (ai[l]>i1)) do
l:=l-1;
if(k<=l) then
begin
swap(ai,k,l);
k:=k+1;
l:=l-1;
end;
end;
if(i<l) then
quickSort(ai,i,l);
if(k<j) then
quickSort(ai,k,j);
end;
end;

procedure compress();
var
    i, l, j, k, i1: integer;
    ai: array of integer;
begin
i:=ncon;
setLength(ai,length(con));
for j:=0 to length(con)-1 do
ai[j]:=con[j];
quickSort(con,0,ncon-1);
j:=0;
k:=-1;
for l:=0 to i-1 do
begin
i1:=ai[l];
if(k<>i1) then
begin
ai[j]:=i1;
j:=j+1;
end;
k:=i1;
end;
for k:=0 to length(ai)-1 do
con[k]:=ai[k];
ai:=nil;
ncon:=j;
```

```
end;

procedure findBB();
var
    f, f1, f2, f3, f4, f5, f6, f7, f8: single;
    i, j: integer;
    af: array of single;
begin
if(nvert<=0) then
exit;
setLength(af,length(vert));
for j:=0 to length(vert)-1 do
af[j]:=vert[j];
f:=af[0];
f1:=f;
f2:=af[1];
f3:=f2;
f4:=af[2];
f5:=f4;
i:=nvert*3;
while(true) do
begin
i:=i-3;
if(i<=0) then
break;
f6:=af[i];
if(f6<f) then
f:=f6;
if(f6>f1) then
f1:=f6;
f7:=af[i+1];
if(f7<f2) then
f2:=f7;
if(f7>f3) then
f3:=f7;
f8:=af[i+2];
if(f8<f4) then
f4:=f8;
if(f8>f5) then
f5:=f8;
end;
af:=nil;
xmax:=f1;
xmin:=f;
ymax:=f3;
```

```
ymin:=f2;
zmax:=f5;
zmin:=f4;
end;

procedure modelpaint();
var
    i, j, k, l, i2, j3, j4, k2, l3: integer;
    ai2, ai3: array of integer;
    gr: array of tcolor;
begin
if((length(vert)=0) or (nvert=0)) then
exit;
transform();
if(length(gr)=0) then
begin
setLength(gr,16);
for i:=0 to 15 do
begin
k:=floor(170*(1.0-power(i/15.0,2.3)));
gr[i]:=TColor(RGB(k,k,k));
end;
end;
j:=0;
l:=ncon;
setLength(ai2,length(con));
setLength(ai3,length(tvert));
for i:=0 to length(con)-1 do
ai2[i]:=con[i];
for i:=0 to length(tvert)-1 do
ai3[i]:=tvert[i];
if((l<=0) or (nvert<=0)) then
exit;
for i2:=0 to l-1 do
begin
k2:=ai2[i2];
j3:=((k2 shr 16) and $ffff)*3;
l3:=(k2 and $ffff)*3;
j4:=ai3[j3+2]+ai3[l3+2];
if(j4<0) then
j4:=0;
if(j4>15) then
j4:=15;
if(j4<>j) then
begin
```

```
j:=j4;
form2.PaintBox1.Canvas.Pen.color:=gr[j4];
end;
form2.paintbox1.Canvas.Brush.Style:=bsclear;
form2.paintbox1.Canvas.MoveTo(ai3[j3],ai3[j3+1]);
form2.paintbox1.Canvas.LineTo(ai3[l3],ai3[l3+1]);
end;
ai2:=nil;
ai3:=nil;
end;

procedure appletpaint();
begin
units();
translate(-(xmin+xmax)/2.0,-(ymin+ymax)/2.0,-(zmin+zmax)/2.0);
mult(axx,axy,axz,axo,ayx,ayy,ayz,ayo,azx,azy,azz,azo);
scalefff(xfac,-xfac,(16.0*xfac)/form2.width);
translate(form2.width/2.0,form2.height/2.0,8.0);
transformed:=false;
form2.paintbox1.repaint;
modelpaint();
end;

procedure strval(s: string);
var
    str: TStringList;
    i: integer;
begin
if((pos('v',s)<>0) or (pos('V',s)<>0)) then
begin
s:=stringreplace(s,'v',' ',[rfReplaceAll]);
s:=stringreplace(s,'V',' ',[rfReplaceAll]);
end;
if((pos('f',s)<>0) or (pos('F',s)<>0)) then
begin
s:=stringreplace(s,'f',' ',[rfReplaceAll]);
s:=stringreplace(s,'F',' ',[rfReplaceAll]);
end;
if((pos('o',s)<>0) or (pos('O',s)<>0)) then
begin
s:=stringreplace(s,'o',' ',[rfReplaceAll]);
s:=stringreplace(s,'O',' ',[rfReplaceAll]);
end;
if((pos('l',s)<>0) or (pos('L',s)<>0)) then
begin
```

```
s:=stringreplace(s,'l',' ',[rfReplaceAll]);
s:=stringreplace(s,'L',' ',[rfReplaceAll]);
end;
s:=trim(s);
str:=TStringList.Create;
try
str.Delimiter:=' ';
str.DelimitedText:=s;
mres:=str.count;
setLength(res,mres);
for i:=0 to mres-1 do
res[i]:=strtofloat(str[i]);
finally
str.Free;
end;
end;

procedure run();
var
    f, f1, f2, f3, f4, sig: single;
    i, j, n: integer;
    Fil: textfile;
    lab: array of integer;
    s: string;
begin
amatMat3D();
tmatMat3D();
amatxrot(20.0);
amatyrot(20.0);
Model3D();
maxcon:=0;
ncon:=0;
maxvert:=0;
nvert:=0;
AssignFile(Fil,filename);
Reset(Fil);
n:=0;
while not Eof(Fil) do
begin
Readln(Fil,s);
n:=n+1;
end;
CloseFile(Fil);
setLength(lab,n);
AssignFile(Fil,filename);
```

```
Reset(Fil);
i:=0;
while not Eof(Fil) do
begin
Readln(Fil,s);
s:=trim(s);
lab[i]:=0;
if(((pos('v',s)=1) or (pos('V',s)=1)) and (pos(' ',s)=2)) then
lab[i]:=1;
if(((pos('f',s)=1) or (pos('F',s)=1)) and (pos(' ',s)=2)) then
lab[i]:=2;
if(((pos('f',s)=1) and (pos('o',s)=2)) or ((pos('F',s)=1) and (pos('O',s)=2))) then
lab[i]:=3;
if(((pos('l',s)=1) or (pos('L',s)=1)) and (pos(' ',s)=2)) then
lab[i]:=4;
i:=i+1;
end;
CloseFile(Fil);
AssignFile(Fil,filename);
Reset(Fil);
i:=0;
while not Eof(Fil) do
begin
Readln(Fil,s);
if(lab[i]=1) then
begin
strval(s);
addVert(res[0],res[1],res[2]);
end
else if((lab[i]=2) or (lab[i]=3) or (lab[i]=4)) then
begin
strval(s);
for j:=0 to mres-2 do
add(floor(res[j]-1),floor(res[j+1]-1));
add(floor(res[0]-1),floor(res[mres-1]-1));
end;
i:=i+1;
end;
CloseFile(Fil);
lab:=nil;
res:=nil;
findBB();
compress();
f:=xmax-xmin;
f1:=ymax-ymin;
```

```
f2:=zmax-zmin;
if(f1>f) then
f:=f1;
if(f2>f) then
f:=f2;
f3:=form2.width/f;
f4:=form2.height/f;
if(f3<f4) then sig:=f3
else sig:=f4;
xfac:=0.7*sig*scale;
xfac0:=xfac;
xfa:=xfac/xfac0;
appletpaint();
end;

procedure TForm2.Button1Click(Sender: TObject);
var
    str: string;
    bmp: TBitMap;
    R: TRect;
begin
scrollbar1.visible:=false;
scrollbar2.visible:=false;
bmp:=TBitmap.Create;
bmp.Width:=PaintBox1.Width;
bmp.Height:=PaintBox1.Height;
R:=Rect(Paintbox1.Left,Paintbox1.Top,Paintbox1.Width,Paintbox1.Height);
bmp.Canvas.CopyRect(R,PaintBox1.Canvas,R);
if savePictureDialog1.execute then
filename:=savePictureDialog1.filename;
str:=ExtractFileExt(filename);
bmp.SaveToFile(filename);
bmp.Free;
scrollbar1.visible:=true;
scrollbar2.visible:=true;
form2.PaintBox1.refresh;
end;

procedure TForm2.PaintBox1MouseUp(Sender: TObject; Button: TMouseButton; Shift: TShiftState; X, Y: Integer);
begin
xfa:=xfac/xfac0;
prevx:=floor(X*xfa);
prevy:=floor(Y*xfa);
if((Button=mbright) or (Button=mbmiddle)) then
if((1.0-dila)>0.0) then
```

```
xfac:=xfac-xfac*dila
else
xfac:=0.1;
appletpaint();
end;

procedure TForm2.PaintBox1DragDrop(Sender, Source: TObject; X, Y: Integer);
begin
xfa:=xfac/xfac0;
prevx:=floor(X*xfa);
prevy:=floor(Y*xfa);
xfac:=xfac+xfac*dila;
appletpaint();
end;

procedure TForm2.PaintBox1DragOver(Sender, Source: TObject; X, Y: Integer; State: TDragState; var Accept: Boolean);
var
    i, j: integer;
    f, f1: single;
begin
i:=floor(X*xfa);
j:=floor(Y*xfa);
tmatunits();
f:=(prevy-j)*(360.0/(form2.width*xfa));
f1:=(i-prevx)*(360.0/(form2.height*xfa));
tmatxrot(f);
tmatyrot(f1);
amatmult(txx,txy,txz,txo,tyx,tyy,tyz,tyo,tzx,tzy,tzz,tzo);
appletpaint();
prevx:=i;
prevy:=j;
end;

procedure TForm2.FormShow(Sender: TObject);
begin
if opendialog1.execute then
filename:=OpenDialog1.filename;
run();
end;

procedure TForm2.FormResize(Sender: TObject);
begin
form2.PaintBox1.Repaint;
appletpaint();
end;
```

```
procedure TForm2.FormClose(Sender: TObject; var Action: TCloseAction);
begin
vert:=nil;
tvert:=nil;
con:=nil;
Application.Terminate;
end;

procedure TForm2.ScrollBar1Change(Sender: TObject);
begin
poshor:=ScrollBar1.position;
difx:=poshor-scrollhorpos;
scrollhorpos:=poshor;
amattranslate(-difx,0.0,0.0);
appletpaint();
end;

procedure TForm2.ScrollBar2Change(Sender: TObject);
begin
posver:=ScrollBar2.position;
dify:=posver-scrollverpos;
scrollverpos:=posver;
amattranslate(0.0,dify,0.0);
appletpaint();
end;

end.
```

2.2 User guide

The data files for 3D visualization of objects are OBJ files. OBJ is a 3D model file format (*.obj). It is a type of text files, which can be opened with a text editor (e.g., notepad). Almost all 3D software supports OBJ files.

Double-click objectVisual3D.exe to run the software, accept or imput parameters, and choose a data file (Figs 2 and 3), the window for 3D graphics will be generated. In the generated 3D graphics window, users may right- or left-click mouse to zoom in or zoom out the 3D graphics, or mouse-drag the graphics to rotate the 3D graphics, or slide scrollbar to vertically or horizontally translate the 3D graphics. Finally, the 3D graphics can be saved as an image file (in BMP format) at its current appearance.

Users may download free OBJ files from internet resources for using the present software. There are numerous such files on the internet.

The software and demo data files (demo data were introduced from JDK 1.1) are included in the package: http://www.iaees.org/publications/journals/selforganizology/articles/2024-11(3-4)/e-suppl/Zhang-Supplementary-Material.rar

Users may occasionally examine and download available higher versions of the software in this package.

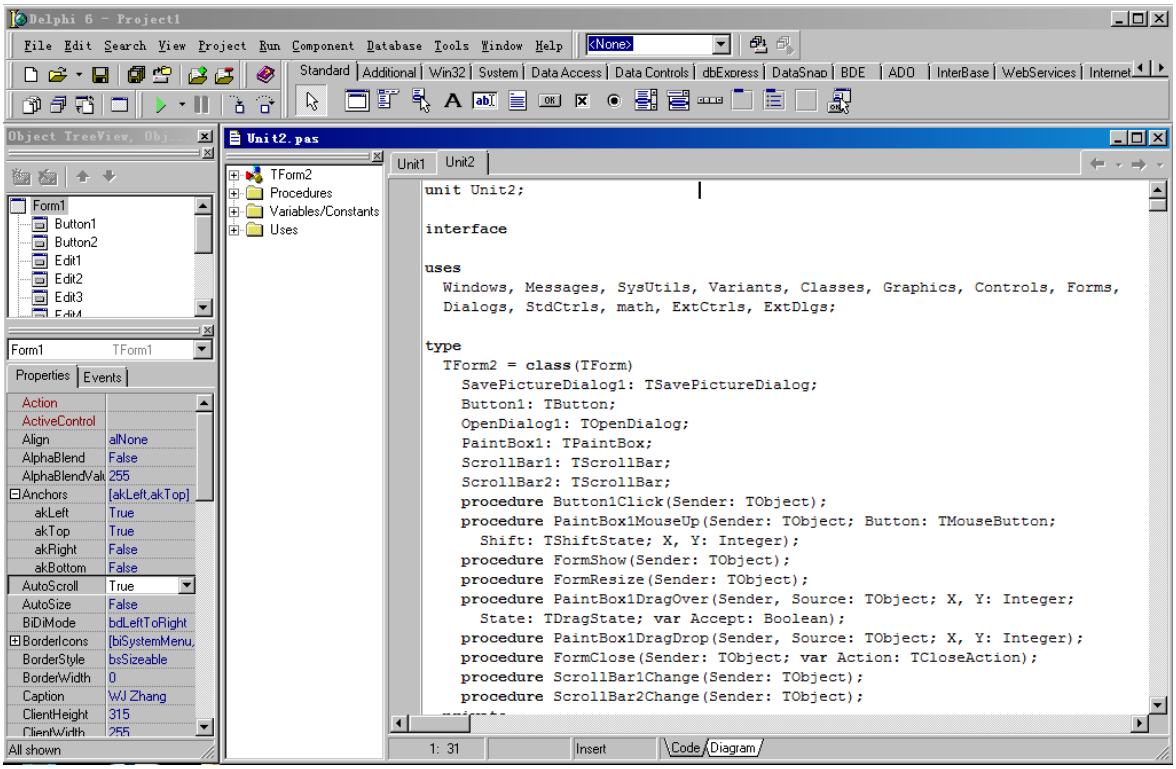

Fig. 1 The Delphi development environment of objectVisual3D.

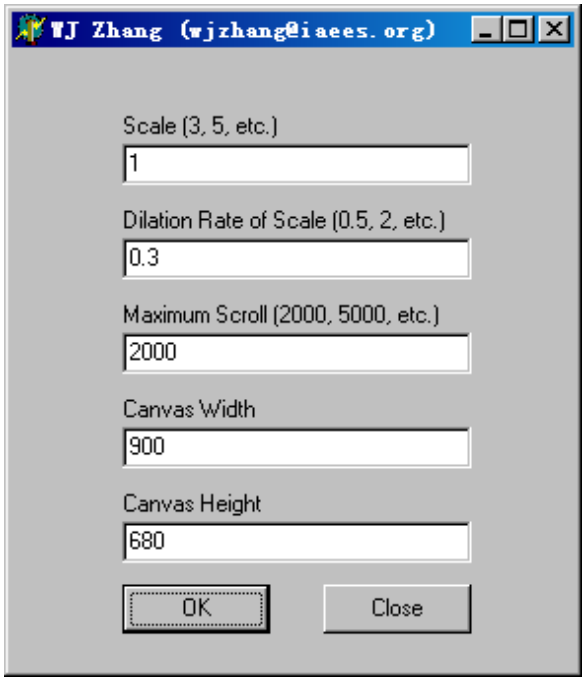

Fig. 2 The parameter input interface of objectVisual3D.

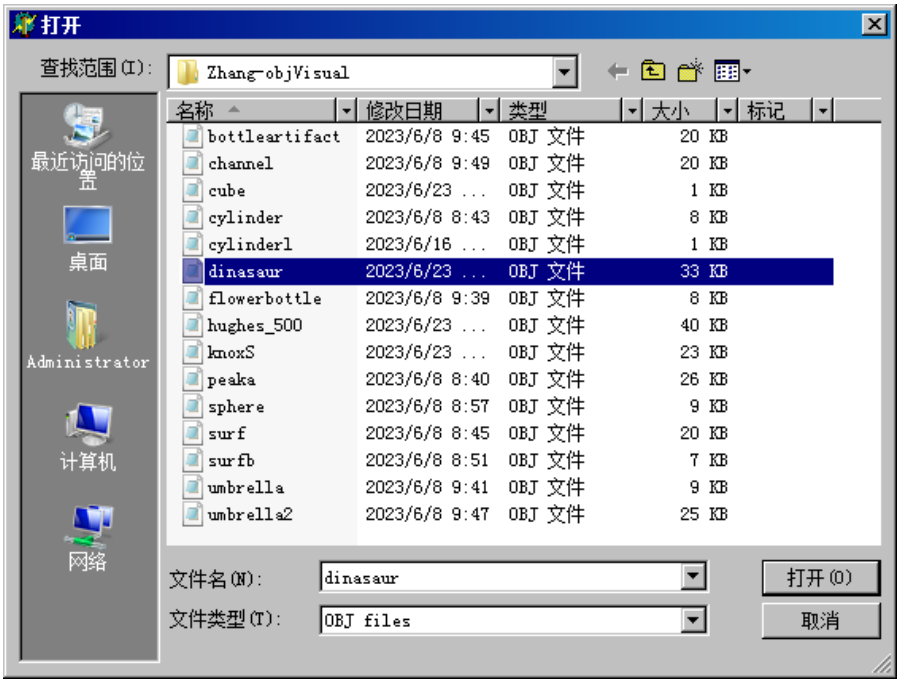

Fig. 3 The file open dialog of objectVisual3D.

3 Example Demonstration

The OBJ data for demonstration were from JDK 1.1 (Zhang, 2023). Some of the generated 3D graphics are indicated in Figs 4 and 5.

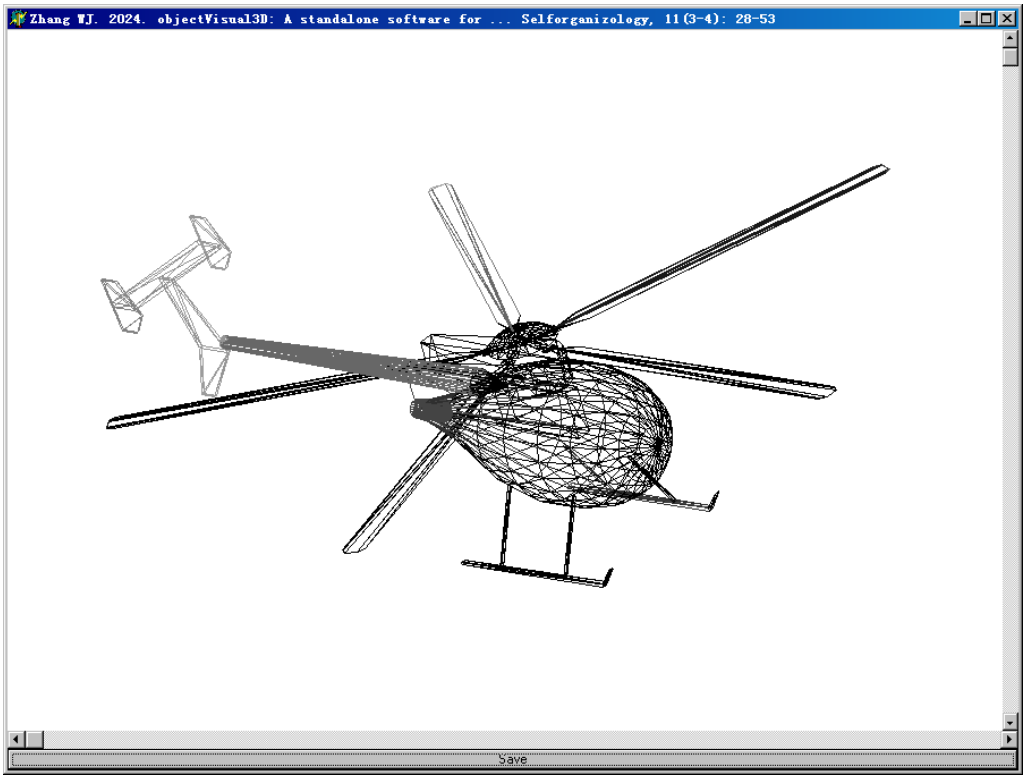

Fig. 4 The 3D graphics of a helicopter.

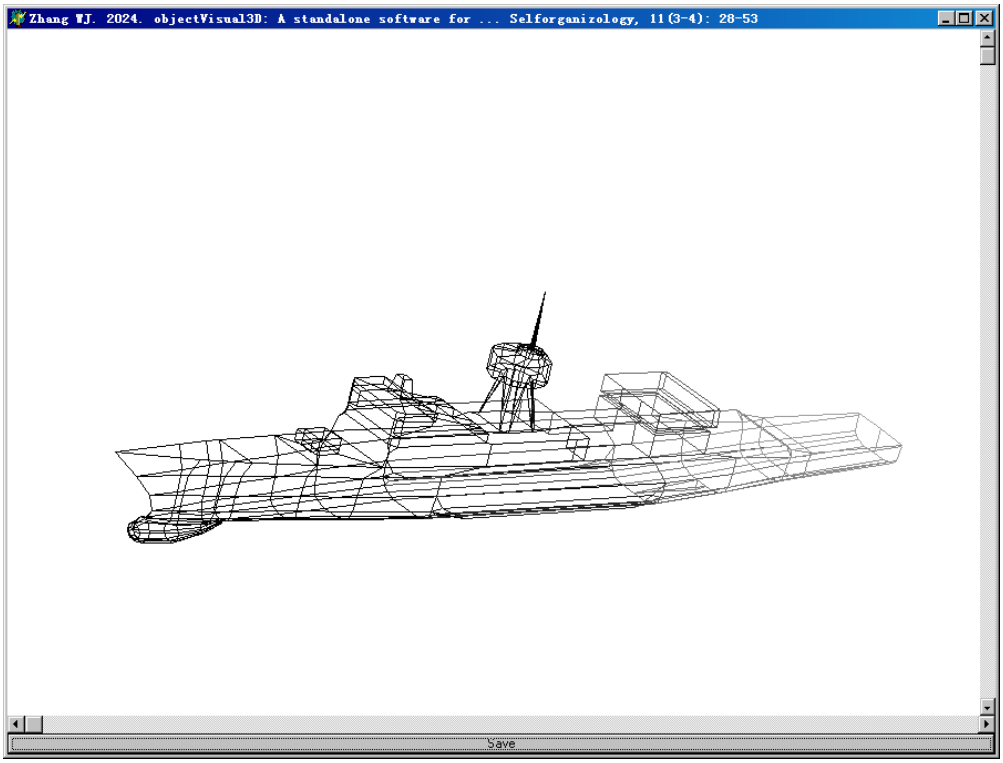

Fig. 5 The 3D graphics of a battleship.

References

Narad P, Upadhyaya KC, Som A. 2017. Reconstruction, visualization and explorative analysis of human pluripotency network. Network Biology, 7(3): 57-75

Zhang WJ. 2007. Computer inference of network of ecological interactions from sampling data. Environmental Monitoring and Assessment, 124: 253–261

Zhang WJ. 2021. A web tool for generating user-interface interactive networks. Network Biology, 11(4): 247-262

Zhang WJ. 2023. 3D visualization of objects and molecules: An integrative Java software. Computational Ecology and Software, 13(4): 81-108

Zhang WJ. 2024a. A Matlab software for visualizing user-interface interactive networks. Network Biology, 14(1): 13-19

Zhang WJ. 2024b. A standalone executable software for network visualization. Network Pharmacology, 9(1-2): 1-10

Zhang WJ. 2024c. An executable Java software for visualizing networks. Network Biology, 14(1): 1-12

Zhang WJ. 2024d. netGen 3.0: The executable Java software for network visualization. Selforganizology, 11(1-2): 1-27

Zhang WJ. 2024e. Several digital timers and clocks for desktop computers. Computational Ecology and Software, 14(1): 1-13

Zhang WJ. 2024f. Two image viewers: A projector and a screen saver. Network Pharmacology, 9(3-4): 11-23