threedit

Octave rendering and 3D object segmentation software.

---

without any form of guarantee

really! .. Since this is a collection of programs developed for a Licentiat thesis (Brian Lading; 'Face modelling - 3D object tracking and segmentation'), and initially not for the public, it will most likely NOT be plug and play.

To get all features working you'll need to do some modifications. Eg different imagebands are implemented (RGB,I,HSV,Edge,.. and all worked a long time ago), but it currently fails when using anything but default, RGB.

If you are fmiliar with octave/matlab, you'll need from one hour to one day of hacking. So run, read errors and iterate.

files

Download these files at `http://www.maths.lth.se/vision/downloads/`

threedit.tar.gz The main octave/matlab package, providing the 3d scene modelling and rendering rutines, the segmentation and the initialization rutines.

threedit_facedata.tar.gz Shape and texture data samples for face modelling. Raw and registered. Expression and ID data.

threedit_editing_programs.tar.gz Editors for texture coordinates, 3D shape editing and registrator programs.

dependencies

www.octave.org an open source matlab clone .. lots of extra packages, see eg. `octave.sourceforge.net/packages.html`

www.vtk.org (Only needed for the editing programs.) An open source visualization program. C++ programs with Python, Java and Tcl/Tk interfaces.

www.tcl.tk (Only for the editing programs.) Used for the GUI of the editing software.

compile '.c'-functions!

use 'mkoctfile filename.c' to compile the supplied c-functions. These functions are also supplied as '.m'files, locate them without the 'momo_'. If you do not compile and use these, you will be waiting for a looooonnng time. So Do Compile!

- momo_d_NormalVertexWorld_dp.cpp
- momo_normalAtPlane.cpp
- momo_normalAtVertexWorld.cpp
- momo_pixelVSpolyid_faster.cpp
- momo_world_points.cpp

.m sample code

```
ini_ini
ini_system

snapshot
showModelAndSampleOverlay

%% manual model placement
parameters = addRotation (parameters,[0 1 0],20);
parameters = addTranslation(parameters,-0.3,0.2,0);

snapshot
showModelAndSampleOverlay

%% construct aaaam .. explore variability in 2d image plane
aaaam.pose.rotX = linspace(-4,4,2);
aaaam.pose.rotY = linspace(-8,8,5);
aaaam.pose.rotZ = linspace(-4,4,2);
aaaam.pose.tX   = linspace(-0.1,0.1,4);
aaaam.pose.tY   = linspace(-0.1,0.1,4);
aaaam.pose.tZ   = linspace(-0.2,0.2,4);
aaaam.modes.id  = [-1,1];
aaaam.modes.exp = [];
aaaam.modes.texture  = [];
aaaam.light.position = [];
aam.additionalPertParameters = [];

aaaam_run

aam.c = zeros(aam.NCmodes,1);
aam_optimize

snapshot
showModelAndSampleOverlay


%% optimize using full 3d-system Jacobian
optimize_initialize
optimize_expression_and_position
%% optimize_all
snapshot
showModelAndSampleOverlay
```

threedit's whos who .. [structure.]

[mode.]

```
mode.addbackgroundnoise     mode.imageband_edge         mode.rotatesample
mode.allmessages            mode.ini                    mode.shadowblur
mode.blur                   mode.ini_system_call        mode.shadowmapping
mode.blurtexture            mode.jacobian_shadow        mode.silhuet
mode.debug                  mode.light_perturb_normal   mode.stablepoints
mode.debug_optimizer        mode.light_perturb_radius   mode.usetexture
mode.debug_optimizer_all    mode.light_perturb_std      mode.usetexture_rgbB
mode.debugall               mode.lightview              mode.usetexture_rgbG
mode.debugfigures           mode.model                  mode.usetexture_rgbR
mode.debugmessages          mode.normalize_image        mode.usetexture_setbg
mode.debugtimer             mode.picture                mode.usetexture_setbgB
mode.filename               mode.picture_h              mode.usetexture_setbgG
mode.fixscale               mode.picture_w              mode.usetexture_setbgR
mode.image_band             mode.remove_eyes
```

[model.]

```
model.X                 model.outline_vertices  model.vertices
model.err               model.polyN             model.vp_ang
model.face              model.poly_back         model.vp_indx
model.image             model.polys             model.vp_poly
model.n_world           model.ssm               model.world
model.nmodes            model.tcoords
model.outline_polys     model.texture
```

[image.]

```
image.AreaAlpha       image.height          image.rgb             image.textureB
image.AreaBeta        image.imageB          image.sampleB         image.textureBorg
image.AreaGamma       image.imageG          image.sampleBorg      image.textureG
image.AreaTriangles   image.imageI          image.sampleG         image.textureGorg
image.bary            image.imageR          image.sampleGorg      image.textureR
image.errorB          image.phong           image.sampleI         image.textureRorg
image.errorG          image.phong_shadow    image.sampleIorg      image.width
image.errorI          image.pixel           image.sampleR
image.errorR          image.polyId          image.sampleRorg
```

[optimizer.]

```
optimizer.J_condition_factor     optimizer.lambda_ini
optimizer.all_parameters         optimizer.lambda_min
optimizer.blur_errorImage        optimizer.lambda_newton_max
optimizer.blur_from              optimizer.lambda_newton_min
```

```
optimizer.blur_jacobian          optimizer.max_iter
optimizer.blur_step              optimizer.max_step
optimizer.blur_to                optimizer.minimum_improvement
optimizer.force_lambda_newton    optimizer.optimize_subset
optimizer.ignore_improvement     optimizer.parameter_constraints
optimizer.improvement
```

[**indx.**]

```
indx.active_pid     indx.active_vid_v1  indx.active_vid_v3
indx.active_shadow  indx.active_vid_v2  indx.active_wh
```

[**shadow.**]

```
shadow.depth   shadow.image   shadow.polyId
```

[**der.**]

```
der.Itexture    der.R          der.areaG       der.nvw         der.xw
der.L           der.V          der.areaT       der.tcoords
der.LightImage  der.alpha      der.beta        der.texture
der.Mm          der.areaA      der.gamma       der.vi
der.N           der.areaB      der.nplane      der.vw
```

[**aaaam.**]

```
aaaam.force_cmodes      aaaam.light             aaaam.min_shapemodes
aaaam.force_shapemodes  aaaam.min_cmodes        aaaam.modes
aaaam.pose
```

[**aam.**]

```
aam.Cc2p                    aam.additionalPertParameters  aam.modelImage0
aam.I0                      aam.background                aam.modelWorld
aam.IG                      aam.bary                      aam.modelWorld0
aam.IG0                     aam.baryActive                aam.p0
aam.IG_aligned              aam.blur                      aam.p0_texture
aam.Image                   aam.c                         aam.phiC
aam.ImageBary               aam.combined                  aam.phiT
aam.ImagePixel              aam.delBorder                 aam.phiX
aam.ImagePolyID             aam.h                         aam.pids
aam.IsampleB                aam.image_band                aam.pidsActive
aam.IsampleG                aam.indxActiveWH              aam.pix
aam.IsampleR                aam.iter_MAX                  aam.removeBorderPixels
aam.JacR                    aam.lambdaC                   aam.sample
aam.LMx                     aam.lambdaT                   aam.sampleB
aam.ModelPolys              aam.lambdaX                   aam.sampleE
```

```
aam.NCmodes
aam.NShapeModes
aam.NTextureModes
aam.Pc2p
aam.Pexplored
aam.Pexplored_texture
aam.R
aam.Rc2p
aam.Tdelaunay
aam.Ws
aam.landmarks
aam.landmarks0
aam.landmarksGridResolution
aam.landmarks_aligned
aam.landmarks_aligned_mean
aam.meanT
aam.meanX
aam.mode
aam.mode_construct_data
aam.modelImage
aam.sampleG
aam.sampleR
aam.silhuet
aam.tex
aam.vid1
aam.vid2
aam.vid3
aam.w
aam.warp_order
```

single matrices

```
Mc ; camera matrix
Mm ; model view matrix
Jac; Jacobian matrix
```