MANUAL FOR SHADOW ART

This is a very short manual for *ShadowArt: Creator and Editor*. We include the Windows installer. The software has been developed and tested on (Intel x86) machines with 64bit winxp and nVIDIA graphics card and 4GB RAM. Expect glitches when using other graphics cards or lower than 2GB RAM.

Comments:

- This demo tool for parallel light source is a research prototype. The program has memory leaks leading to crashes after several edits.
- In the submitted version, we put a cap on maximum resolution settings to minimize memory issues. Results presented in the paper, which were computed at higher resolutions, can be slightly different.

RUNNING THE PROGRAM

Step 0: Installation. Run the installer and follow the instructions. Once installation is complete, please ensure that the directory "IncludedData" has been created in the folder containing ShadowArt.exe. This directory contains all the included images/meshes.

Step 1A: Select options. (a) You can select between *Preset*, x-y-z, $0-45-90$, and $0-60-120$. Preset selects the projection we consider to be the most interesting for the selected example. You can leave this to default to start with. (b) We allow coarseness

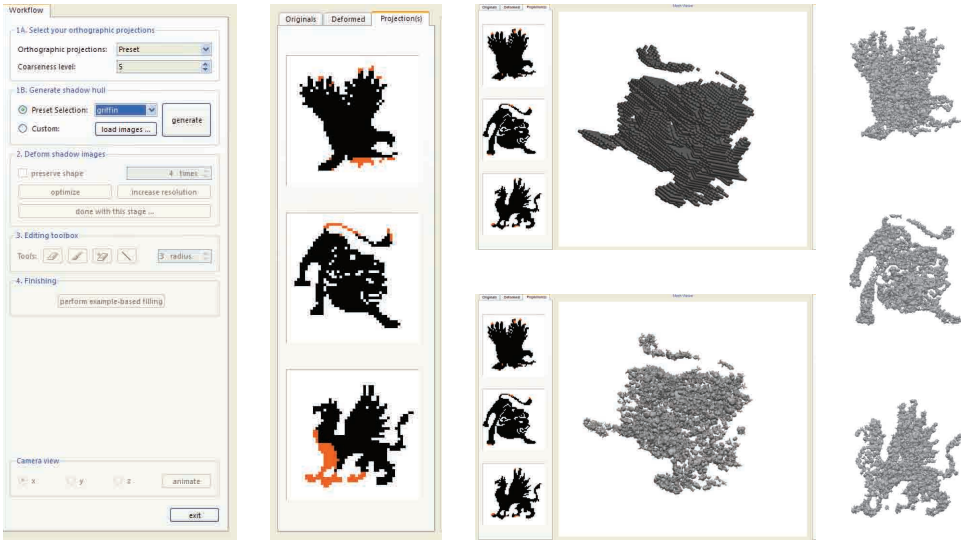

FIGURE 1. (Left) Panel for directional lights, (left-center) input shadows, (right-center) shadow hull after a few optimization steps, example based filling, (right) viewed along three projection directions.

level between 3 and 8. One can also increase the resolution later in step 2. As a multi-resolution strategy, start with a coarser grid and then optimize, refine, optimize, refine, etc.

Step 1B: Generate. We support two modes:

- Preset Selection: It is easiest to load a preset example from the pull-down menu. You can override the default projections for any preset example using "Orthographic projections" from 1A.
- Custom: You can also specify a set of three images in a desired configuration. Since we do not include our GPU based initial positioning tool (see supplementary video), we suggest that you adhere to the following format: Image sizes should be 250×250 including a white boundary 25 pixels wide. As a rule of thumb, images with very few black pixels are not suited for low resolution (like the Einstein sequence). Images are used to automatically generate meshes (if not already present), and stored in the same directory. If desired, you can provide semantic hints by performing some topological mesh surgery. If the initial shadow hull is empty, then the program behaves badly.

Press *generate* and the shadow hull should appear. Tabbed left panels are used to show a comparison between original and deformed images along with the current shadows and their inconsistencies.

Step 2: Deform shadow images. Specify the number of iterations and hit *optimize*. You can check the "preserve shape" flag to indicate higher stiffness for the image mesh. Switching between Originals/Deformed tabs help to detect the deformations, which are often subtle. You can also increase resolution (up to a coarseness level of 3).

Interaction with shadow volume is as follows: left button for rotating, right button for panning, scroll for zooming. Also there is a camera view panel in the lower right to switch to exact views, or to animate between views.

Once happy with the optimization results (a few pixel may remain red due to resolution limits, or deformation energy exceeding a threshold bound) move to the next stage.

Step 3: Edit toolbox. We support remove, add, erode, and ray shoot. All the tools use the specified radius. Remove balls/rays work only if the full solids can be removed (see paper for details). Erode and shoot rays are good tools for fast but interesting removal.

Step 4: Finishing. You can fill the shadow hull using four popular graphics models. The density of filling is kept fixed for the demo, again for memory (leak) issues. The program may crash if the resolution is too high, or the shadow volume have few active cells.

Uninstall. Please go to "Control Panel → Add or Remove Programs" to uninstall the program.

Final words. We hope that this application helps in better judging the algorithm steps, its performance, and limitations. We include all the examples presented in the paper, along with a few extras. Have fun with shadows!