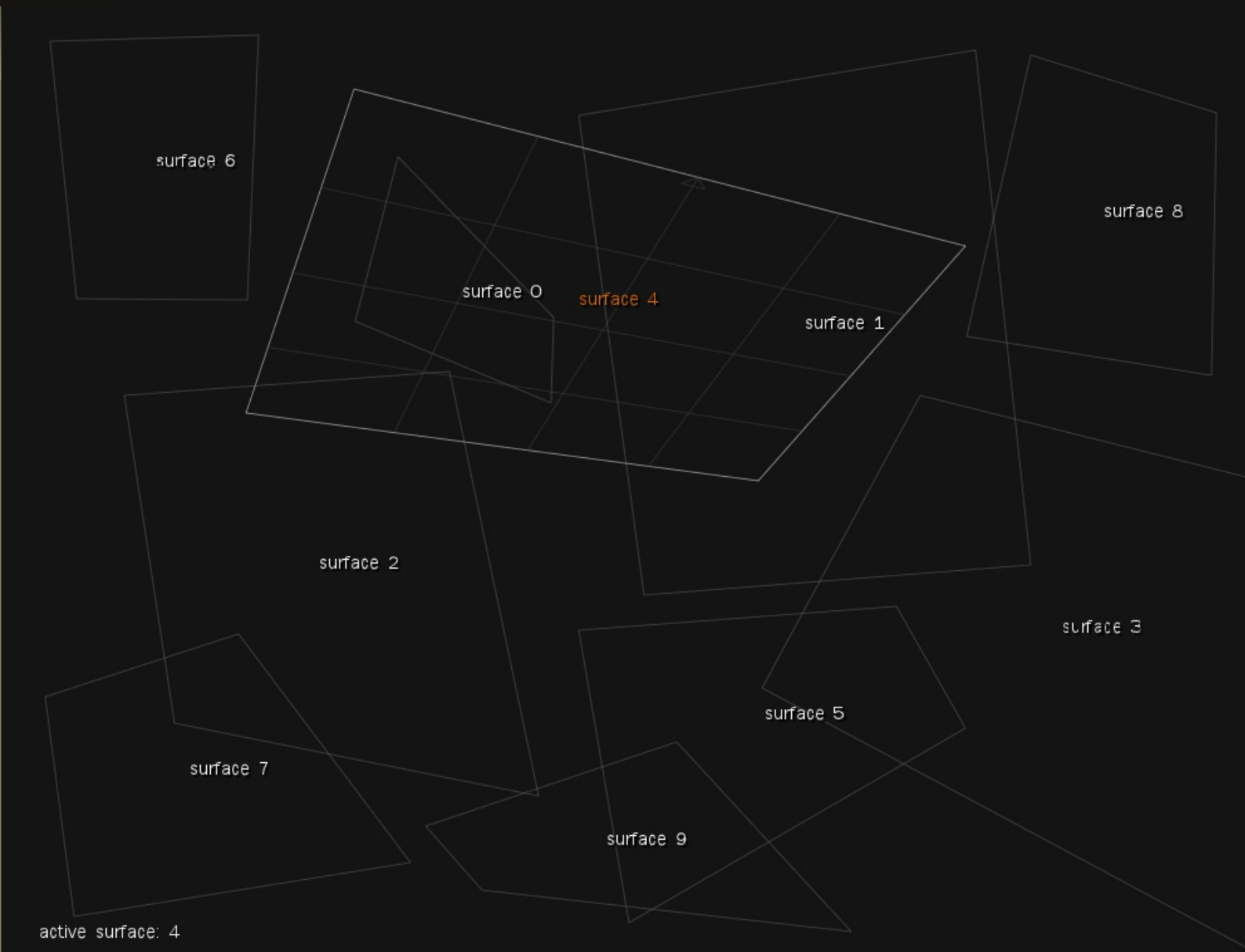


LPMT'S SSSM

Little Projection-Mapping Tool Super-Sexy Secret Manual

lpmt

little projection mapping tool

LPMT'S SSSM
Little Projection-Mapping Tool Super-Sexy Secret Manual
LPMT version 1.4

Introduction

We have no idea about how you got to this super-sexy secret manual for Little-Projection Mapping Tool version 1.4. Very likely you have been given the super-secret-incredibly-well-protected download link by one of our disgustingly unfaithful employees, upset because this week we ran out of olives for the Martini.

Anyway, now that you got hold of our well-kept-secret documentation, we feel obliged to warn you about the fact that this is an unfinished manual for an unfinished application, and that both are very likely to stay in this shiny unfinished state for quite a long time in the future. We have to inform you as well that LPMT (Little Projection-Mapping Tool) is an application mainly developed on Linux systems, it is not been widely tested yet, and it may do horrible things to you and your family, like transform your shiny new MacBook pro in a 1990's windows 95 desktop, install Android on your iPhone and iOS on your Android, change your fiancee's pregnancy test results or evaporate your bank account. You've been warned: YOU USE IT AT YOUR OWN RISK.
Ok? Let's go then...

What is Projection-Mapping?

Projection Mapping is the medium of the moment. Projection-mapping techniques allow you to paint real world using a simple video projector.
To do projection mapping you can usually choose between two main different approaches:
- - the so-called 3D approach implies the reconstruction of the 3d scene you have to project things on, measuring real-world objects and modeling them in a 3d software like Blender, Cinema4D or SketchUp. You can then UV map things on the virtual model, render them to video or images and project the results to the real world. You can even use adaptive techniques, using calibrated cameras and projectors to achieve an high degree of precision and accuracy. It's generally a work-intensive approach, that can give stunning results if properly done, but implies a good control of the environment you're working on and of the involved variables (projector model, point of view, projector location, etc.)
- - the so-called 2D approach relies generally on the perspective deformation of surfaces, mapped to real-world objects with keystone deformation of a quadrilateral surface by adjusting its corners. This setup process, although quite work-intensive itself, is usually more flexible and way simpler than the 3D reconstruction approach, allowing to be adapted on the fly to the real location without prior knowledge of the site and situation. Many projection-mapping applications are based on this somehow simpler concept, LPMT among them.

LPMT origins and concept

LPMT (Little Projection-Mapping Tool) is developed by HVA – Hermanitos Verdes Architetti, a small architectural practice based in Modena, Italy.
It was born as a small in-house tool we developed for a job we had to do, consisting in designing a little fair-booth. The project, as often happens to us poor architects, then vanished before completion leaving no trace, our customer too vanished forgetting to pay us for the work done, and we were left with this little half-finished piece of software.
As we were using LPMT as an occasion to learn a bit of C++ programming, and considering that at the moment, it was late 2010 then, there were not many available open-source solutions for video/projection-mapping, we decided to take lpmt development a little further, learning things by doing them, and see what we could achieve.
LPMT is the result of those messy efforts and of a development we made on a very irregular base during the last year and a half.

LPMT is released under GPL open-source license, it's coded in C++ and it's based on the powerful and wonderful OpenFrameworks (http://www.openframeworks.cc) framework for creative coding.
OpenFrameworks is a C++ library designed to assist the creative process and wraps together several commonly used libraries under a tidy interface: openGL for graphics, rtAudio for audio input and output, freeType for fonts, freeImage for image input and output, quicktime or Gstream for video playing and sequence grabbing.
OpenFrameworks is designed to be cross platform (PC, Mac, Linux, iPhone, Android) and cross compiler, so applications based on OF are supposed to be easily portable to any of these systems, but you must be aware that LPMT has been developed on a Linux (Ubuntu) environment and not widely tested on other systems.

Being based on OpenGL, lpmt graphics are rendered mainly on the GPU, this makes things rather fast and responsive, but not all system-gpu combinations have proven to be equally reliable. A good GPU on a modern system is highly suggested to run LPMT in a production environment, but we can't guarantee at all that LPMT is going to work on your system.

If you get LPMT running reliably on your system, then with a single projector connected to a laptop and the aid of LPMT, you can project elements to differently oriented surfaces and compensate projection distortion using deformable quads that can be adjusted directly on the screen by simply dragging their corners.
Each quad in LPMT is an independent object that can contain video, images, live-feed from a webcam, smoothly changing solid colors, a slideshow, etc.
For each quad you can as well configure the speed and looping mode of video, slides and colors transition speed, add a tint hue to any kind of content, adjust video volume and quad transparency.
A rather complex setup can be saved to xml file and reloaded later.

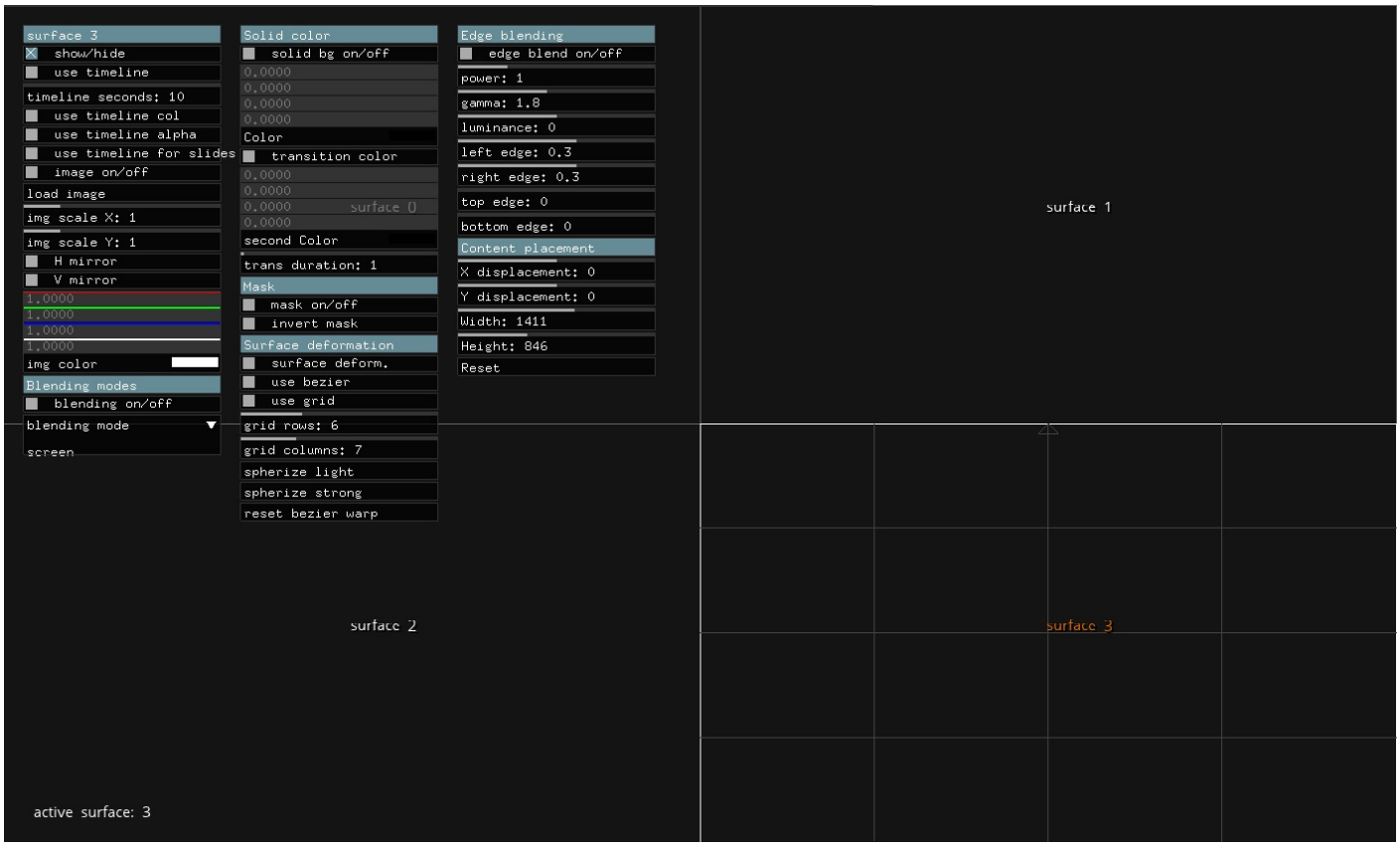
Where do I find LPMT? How do I install it?

LPMT is released as open-source code, and it's freely downloadable in source form from our website (http://www.hv-a.com/lpmt).
Development code tree is maintained at Github (https://github.com/).
In order to compile LPMT you will need a C++ development environment suited for your system, the OpenFrameworks code and quite a lot of OpenFrameworks addons.
We are probably going to detail lpmt compilation process in a dedicated section of this documentation somehow in the future, by now it's more or less a trial-error process.
LPMT is distributed in a pre-compiled form too, for Linux, Windows and MacOs platforms.
If pre-compiled versions for your system are available at this moment, you can always find them on the download section of LPMT website.
Generally speaking, applications developed with the aid of OF are not 'installed' in a traditional way on your system, they come as a compressed folder that you have to decompress somewhere on your system. Application itself is then launched directly from the root folder of the application. The application's folder structure must not be changed, or necessary assets may not be reachable by the app itself.

Basic LPMT use

When launched, LPMT presents you this interface:

As you can see, the GUI and the app itself are contained in a single window (the GUI will change in the future, a re-design work is going on).



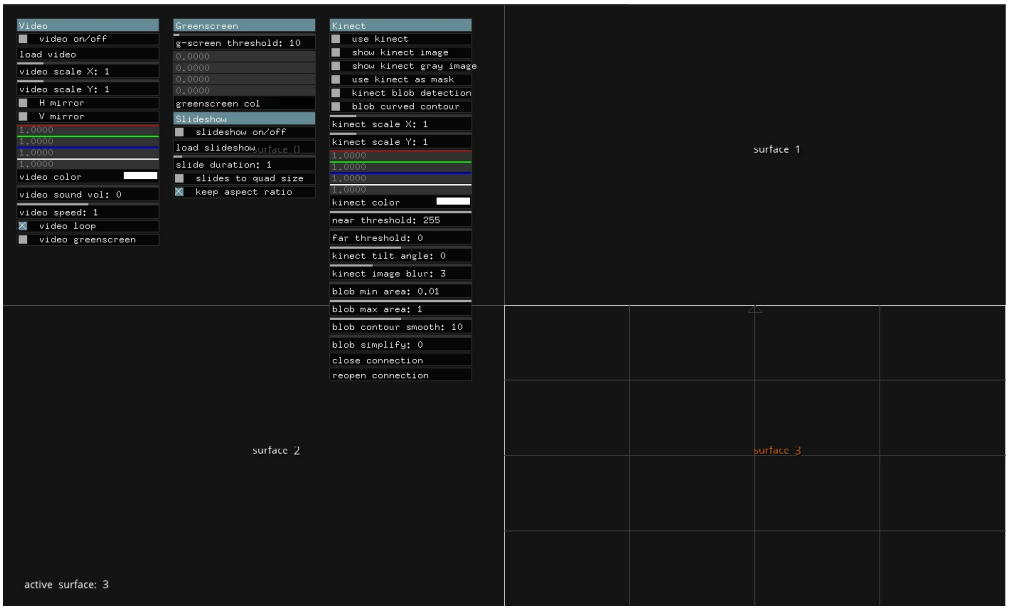
At the beginning you have 4 default projection-surfaces.
Projection surfaces in LPMT are used to show your content, they are independent quadrilateral objects that you can move and warp, adapting them to the real-world objects you must project on.
The GUI elements you are presented are always referred to the 'active' surface. The number of the active surface is written on the bottom-left of the screen.
Each change we do using the GUI is applied to the current active surface.
LPMT can be in different modes:

- content editing mode, in which for each surface you can edit parameters, load content, etc.
- surfaces setup mode, in which you move and warp the surfaces, draw masks, tweak deformations, etc.
- projection mode, in which magic happens

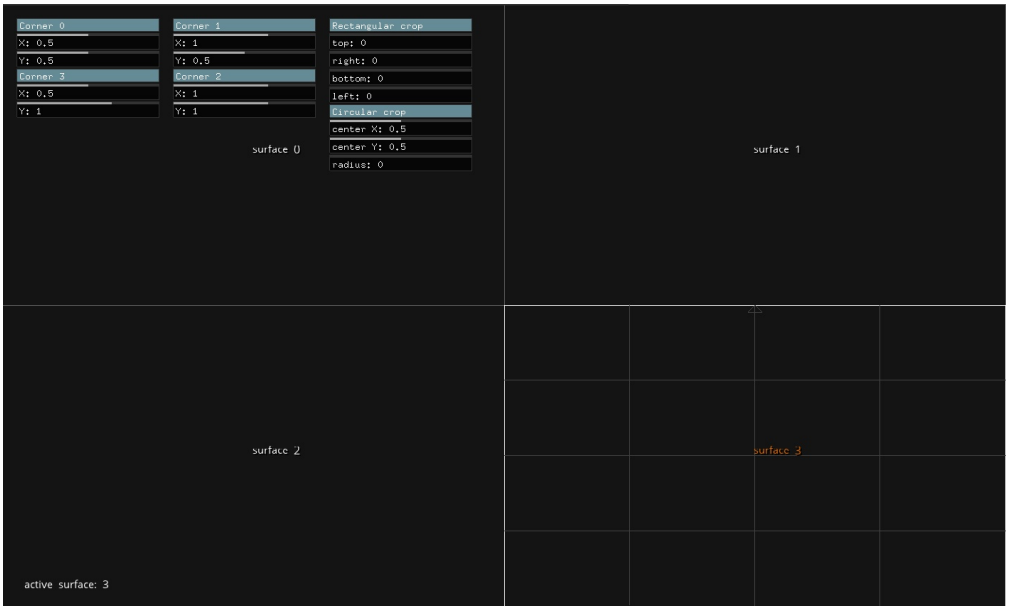
At start LPMT will be in parameters setup mode, the GUI is visible, surfaces have a frame and the screen background is grey.
For each surface the GUI is divided in several pages, you can pass from one page to the other with a combination of keys on your keyboard: 'z', 'x' and 'c' keys or 'F1', 'F2' and 'F3'

pressing key '1' you are presented with a general-settings gui page, on which you can turn individual surfaces on-off.
We will see later in detail the parameters you can change for each surface and how to fill them with images, video clips or other contents.

each LPMT surface can be controlled using a multi-page gui. activating the second gui page with 'x' hotkey, you can control video, cameras and, if present, a Kinect device.Video and Camera feed can use a greenscreen/chromakey effect

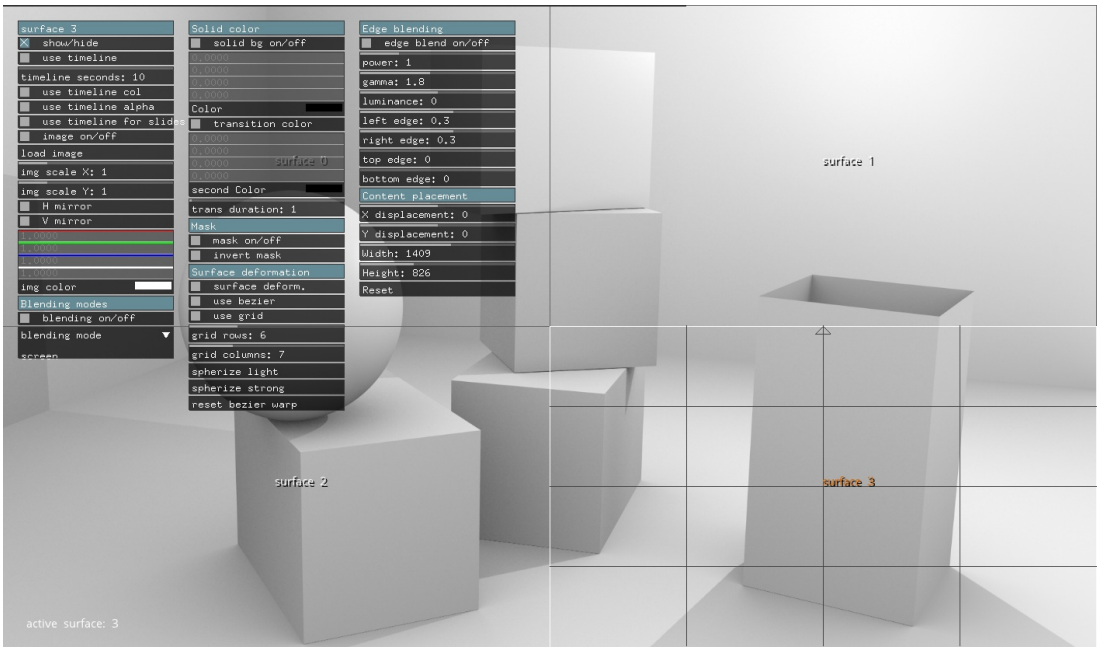


On the third gui page, activated with 'c' hotkey, you can fine-tune surface's corners position and edit rectangular and circular cropping mask



Preliminary setup

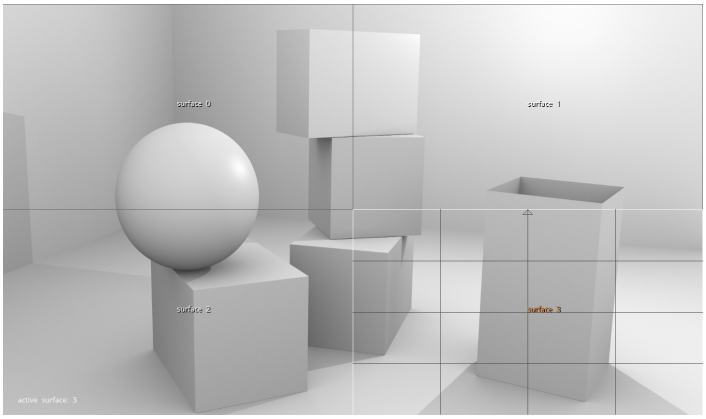
The 1.4 version of LPMT is a single-window application, the gui is presented on the same window used for projection, this is going to change in the future as we are working on a re-design that will separate gui and rendering window, but for the moment the more practical way of working on projection setup is by cloning the same desktop to your computer monitor and to projector output, looking at monitor for gui setup and at projected image for surface warping and adjusting.
If you need to make a quick setup to test your system you can load an image from your disk and use it as background in LPMT.
In order to load the background image press 'shift' + 'W' on the keyboard, a system dialog should appear, allowing you to choose the image file.
With 'w' key (lowercase), if you have a configured webcam attached, it will be used to take a screenshot that will become your background image.

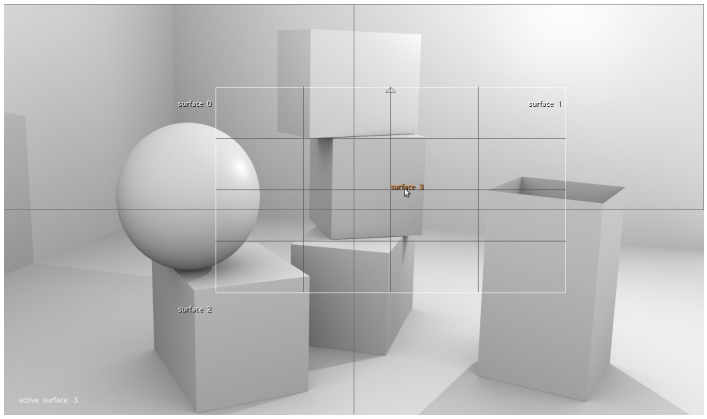


Surface-setup mode

Pressing 'g' key, you toggle gui on-off and switch back and forth to surfaces-setup mode.
When gui is turned off, you are in surface-setup mode.

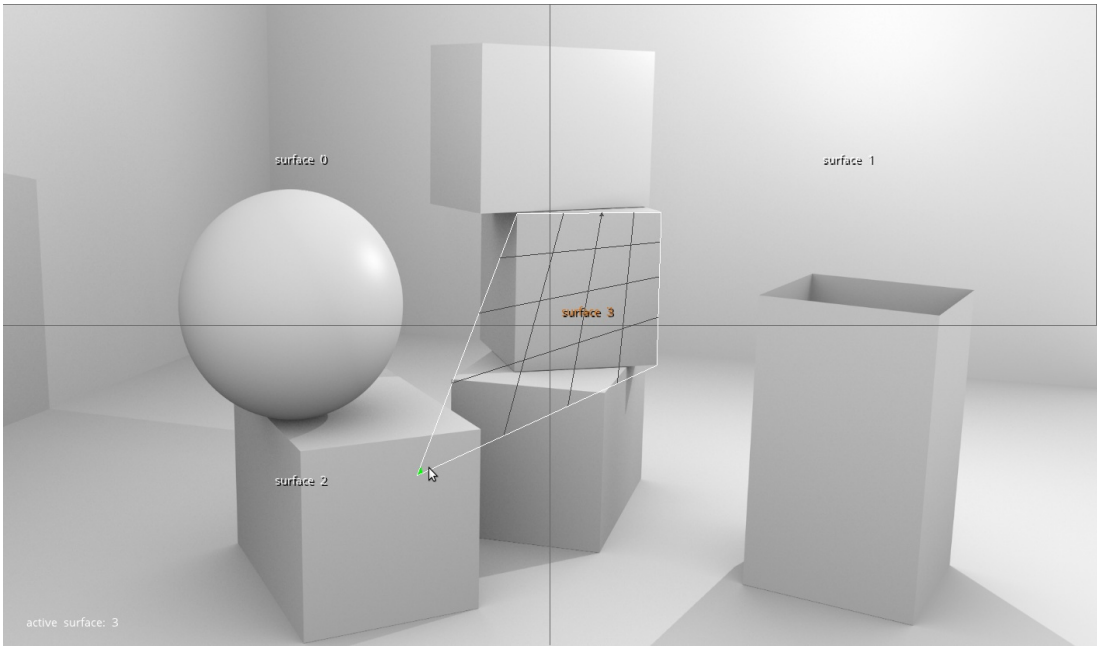
In this mode you can move the active surface, apply the perspective warping, draw masks and apply bezier or grid warping.
The active surface has a reference grid and its name appears in orange, to move it, just grab it with the left mouse button near to its center and drag it on the screen.





If you move the mouse pointer near to one of active surface's corners, a little green triangle will appear, you can then grab it with the left mouse-button and drag it on the screen. You will see the surface deform according to the perspective. You can grab each corner, the active one will always be indicated by a little green triangle.

active corner will show a little green triangle, corners can be dragged with left mouse-button to adjust surface

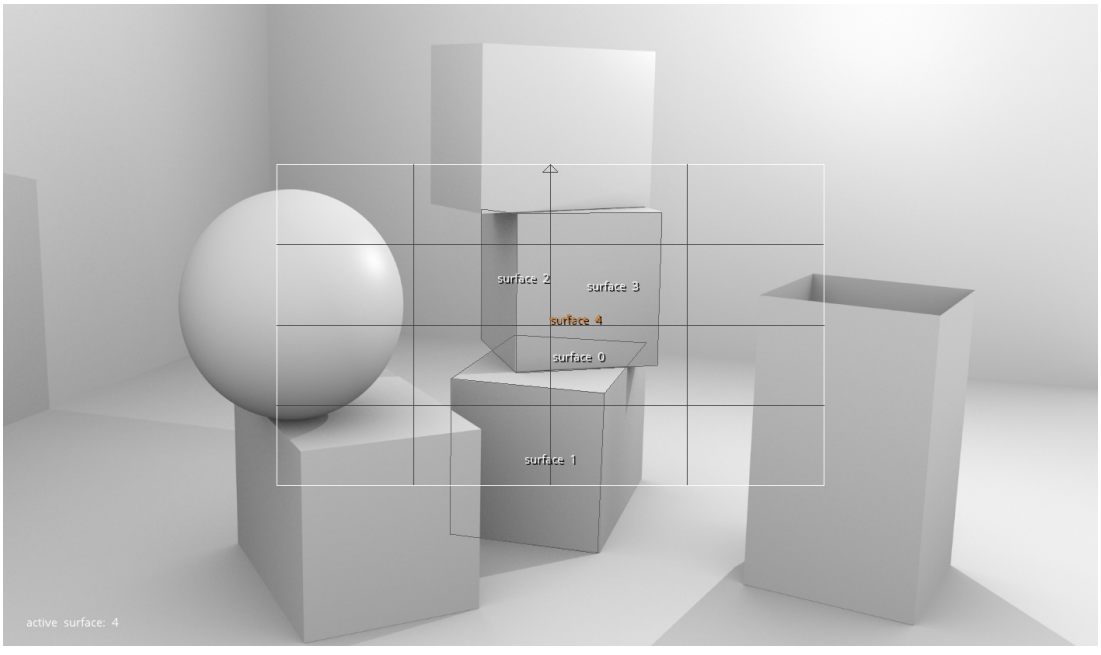


To change active surface you have two possibilities: you can double-click near the center of the surface you intend to activate or you can switch to previous-next surface using '<' and '>' keys on the keyboard. The double-click method only works when the gui is switched off.

When you drag and release one of the active surface's corners near to another surface's corner, it will snap to the its location, allowing to adjust surfaces on objects without leaving black gaps. Snapping can be turned off unchecking corresponding checkbox in general controls gui page (use number key '1' to get to general controls page).

The maximum number of surfaces in LPMT at present is pre-determined, you can have up to 36 surfaces.
Adding new surfaces to LPMT is just a matter of pressing 'a' key on your keyboard: a new surface will be added to the center of the screen and will be selected as current active surface.

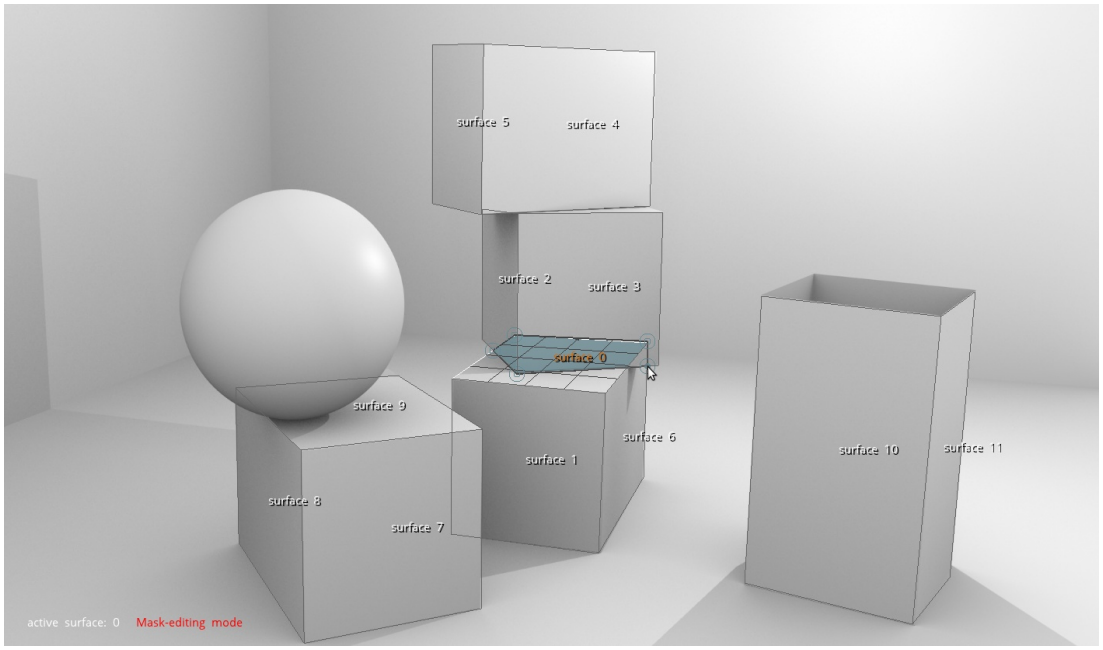
add a new surface with 'a' hotkey, the new surface will be immediately active and ready for editing



Mask-setup mode

When you are in surface-setup mode (with gui turned off) you can switch to two more tweaking modes:
- mask-setup mode
- deformation/warping setup mode

Turn off the gui with 'g' key, then press 'm' key on the keyboard, a red note on the bottom of the screen will confirm you you entered the mask-setup mode. You can then draw a mask on the active surface using left mouse button.
Masks are polygonal lines that can be used as negative masks to make vanish the corresponding parts of the surface, or as positive masks to reveal only the parts of the surface defined by the mask, hiding the rest.
The positive/negative state of the mask is defined on a surface base, clicking the corresponding toggle buttons on the gui.
In mask editing mode each mouse click will add a control point to the polygonal mask line.
For each surface only a single mask can be defined.
Masks can be edited by dragging and adjusting existing mask points. When mouse pointer approaches a mask point in mask editing mode, the point is shown as 'active' and can be dragged, if you press 'd' key on the keyboard, the currently active mask point is deleted. Pressing 'c' key in mask editing mode deletes the whole mask.
To exit mask editing mode press 'm' key again.
N.B. - You are not going to see the effect of masking on your surface until you turn masking on with the corresponding toggle button on the gui.

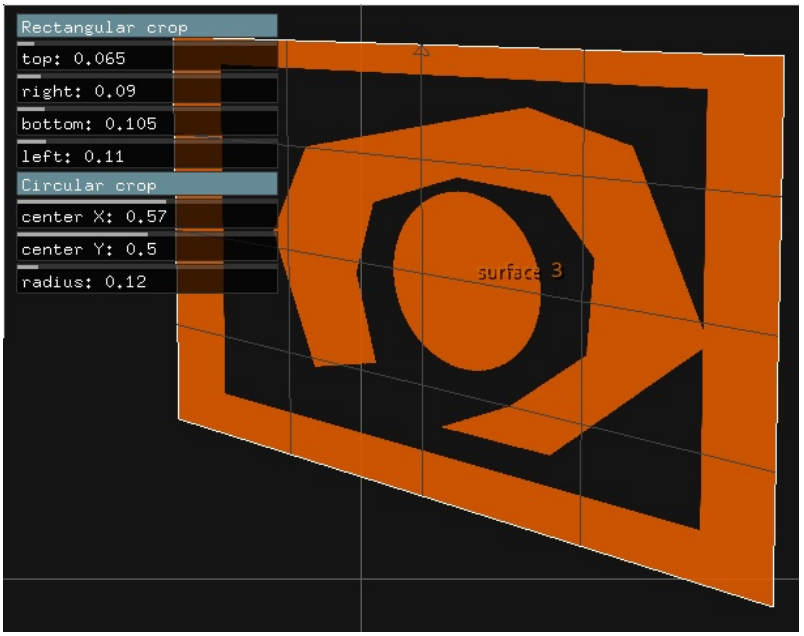


with 'm' hotkey you enter mask-editing mode, in mask-editing you can draw a polygonal mask with left mouse-button

A special kind of mask is a so-called "cropping mask", you have two kind of cropping-masks in LPMT:

- rectangular cropping
- circular cropping

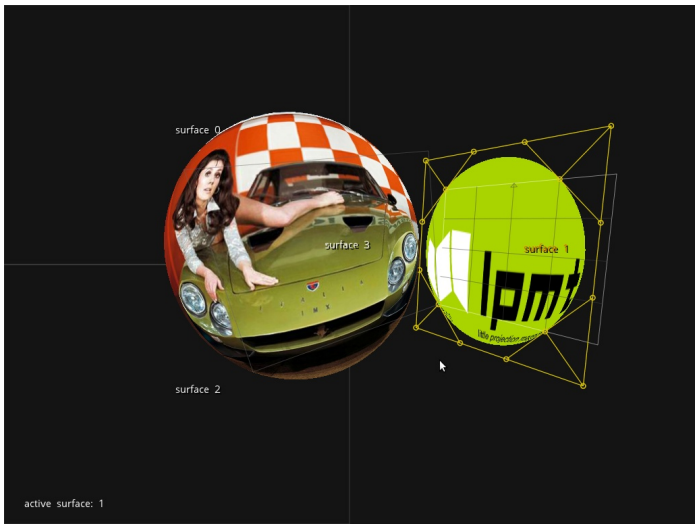
Cropping masks are controlled using sliders you find for each surface on the corresponding third page of the gui, activated with 'c' or 'F3' hotkeys on the keyboard.
For rectangular cropping, you can set the amount of cropping for the four sides of the surface, for circula cropping you can choose the radius of the circle and its center position in surface.
As for polygonal masks, cropping masks can be used as a negative or positive mask.
Cropping masks can be used togheter with hand-drawn polygonal mask, and all of them are controlled by mask toggle buttons on the first page of the gui.



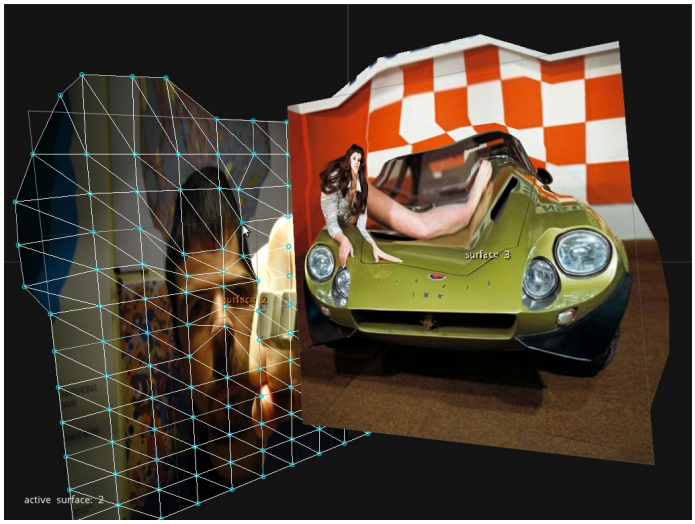
Deformation/Warping setup mode

Turn off the gui with 'g' key, then press the 'b' key on the keyboard, you will enter the deformation/warping setup mode. This mode is depending on the type of warping you have set for the active surface on the gui.
To adjust warping, you must have previously toggled the deformation button on on the gui, otherwise no control point will be displayed.
In LPMT, besides the surface perspective deformation, you can apply two different warping deformations to fine-tune your projections setup:

- bezier-surface deformation
- grid-warping deformation

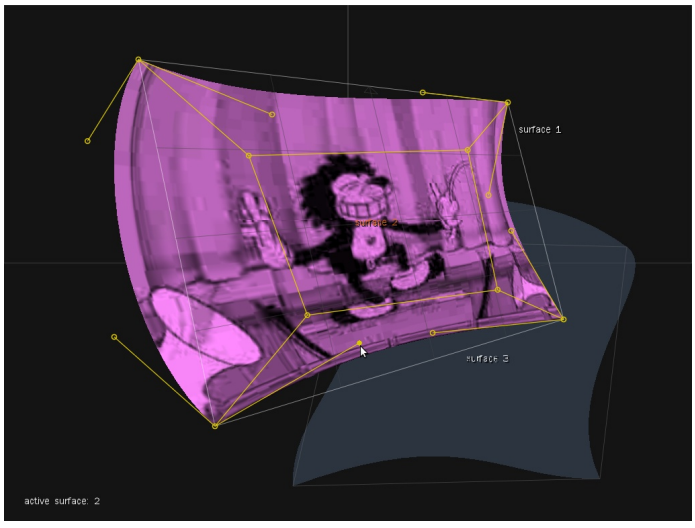


With bezier-surface deformation, you can adapt your surface to curved real-world surfaces, and correctly project images, for example, on rounded walls or emispheres.
When you select bezier deformation on the gui and enter warping setup mode as described above, you will be presented with a yellow structure defined by 16 control points that can be dragged on screen and adjusted to achieve the desired surface deformation.
On the gui you can select two pre-set bezier deformations that you can use to approach an hemispheric projection. This is not a geometrically correct hemispheric projection, but you could usually be happy with it.
Another gui button allows you to reset the bezier deformation.



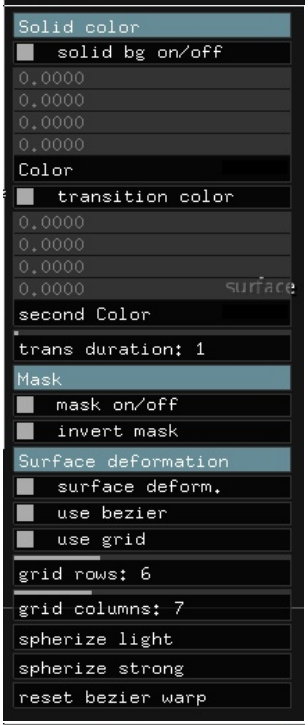
With grid-warping deformation, you can fine-tune your projection on surfaces that present local variations in their geometry, by defining a grid of NxM points and then, entering warping setup mode with 'b' key, drag each point individually on the screen to deform the projected content of the surface.
The density of the grid defaults to 6x7 points, but you can change it by using the corresponding sliders on the gui.
Be aware that if you change grid density after applying any change to grid points' position, the grid will be reset to an unwarped state.

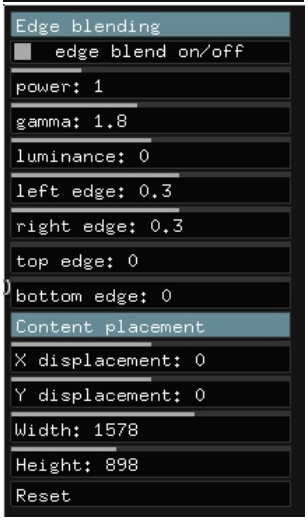
Please note that bezier-warping and grid-warping can't be applied togheter at the same time to the same surface, if you check both bezier and grid warping toggles on the gui, only bezier deformation will be applied.



Content-editing mode

The content of surfaces is mainly controlled using the gui. If it is not visible, the gui can be switched-on with 'g' hotkey. Usually when the gui is toggled on, the first gui page is displayed. Remember you can cycle through gui pages with 'z', 'x', and 'c' or 'F1', 'F2' and 'F3' hotkeys.

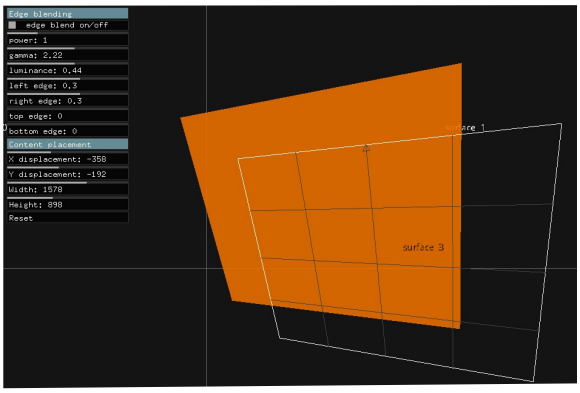




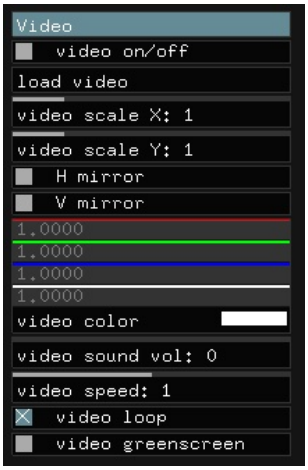
<u>first gui page:</u>
gui pages are organized in columns, in left column you find the name of the active surface, then a toggle control to show or hide the whole surface. When a surface is created the display control is on by default.
The following controls are used to control the timeline and are general controls, they have a global effect, not restricted to the current active surface. Timeline use will be discussed in detail on another section of this manual.
Below timeline controls, you can find the image section, with a toggle button to show/hide the image in active surface, follow by a button to load image file from disk. Pressing 'load image' button a system dialog should open, allowing you to choose an image file to be loaded into your surface.
The image scale can be adjusted with the following sliders on X and Y axis.
Below you can find two toggle buttons to mirror image on horizontal and vertical axis, and a set of color sliders that affect the hue and alpha of displayed image.
With color sliders you can generally adjust the color channels and the alpha channel of surfaces content.
After image section you can find a couple of controls that allow to choose and activate a blending mode for your surface. Blending modes in this version of LPMT are somehow limited and basic and you should not expect them to behave like in photoshop. Blending modes control how your surface will mix with other underlying surfaces, and must not be confused with edge-blending that will be discussed later.

On central gui column you have controls for giving a solid color background to active surface, as usual you can control RGB and alpha channels. Solid color can be static or you can set a smooth cyclic transition between two available colors, you can set transition speed with the 'trans duration slider' that will set the cycle duration in seconds.
After solid-color controls, you have the toggle buttons to control masking effect. Masks use and mask-editing have already been discussed in previous chapter.
The other controls on this column are related to bezier and grid warping that have been already treated as well in a previous section of this manual.

The right column on first gui page allows you to control the edge-blending effect for active surface.
Edge-blending is a useful smoothing effect applied to the borders of your surfaces that allow you to blend togheter seamlessly the output of several projectors. With LPMT and cheap home-theater projectors you can easily achieve a rather complex projection setup, comparable to expensive dedicated commercial solutions.
The edge blending in LPMT is based on GLSL shaders, relying on the gpu (some cheaper gpu may have problems with edge-blending in LPMT) and the concept is based on a work by Paul Bourke.
For a description of the involved concepts and parameters we address you to original Paul Bourke's paper that you can find on his site at following link:
http://paulbourke.net/texture_colour/edgeblend/
Be aware that at present in LPMT masks can't be used on a surface that has edge-blending effect.



The last controls of this column allow you to displace surface content relative to surface's frame, and to tweak its dimensions. This can be useful in many different situations. Displaced content will always be moved retaining a correct perspective deformation. A Reset button allow you to clear content displacement.



second gui page:

To activate second gui page use 'x' hotkey. On second gui page you find controls for loading and controling, video, webcams' live-feed and even Kinect sensors.

The first column controls allow you to load video-clips to your active surface and to tweak the appearance as we have seen in case of static images on the previous chapter.

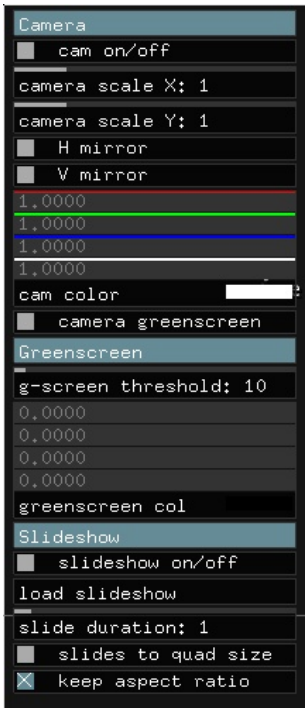
LPMT relies on OpenFrameworks structure for video and cameras, in OF, depending on your platform, video is managed through Apple Quicktime (MacOs and WIndows) or Gstreamer (Linux). Every video-clip viewable through thanks to this libraries should be usable in LPMT as video source. Obviously you should have the corresponding libraries and codecs installed on your system.

As a general note, you are encouraged not to use too high resolutions for your video clips (a HD clip doesn't make much sense if the resolution of your projector is 800x600 pixels), and to avoid cpu-intensive codecs as h264.

Besides scale, mirror and hue controls, that behave the same way as with static images, for video you can control as well the sound volume, and the speed. Speed can be positive or negative, and the displayed value can be intended as 'normal speed' x 'value', so a value of '1' means video forward at normal speed, a value of '2' means video forward at double speed, a value of '-1' will give you a backward playing video at normal speed and so on.

The two remaining toggle buttons in video section of the gui allow you to control video loop mode and greenscreen/chromakey mode for your video-clip.

Video clips in LPMT are normally playing in an endless loop, if you want your video to stop after completion, then you must un-check the 'video loop' checkbox.



On the second column of the gui, if a recognized webcam is attached to your system, you can find controls to enable/disable the cam as source for active surface. As in the case of images and video-clips, you have controls for RGB and alpha channels, horizontal and vertical mirror and scale, and a toggle button to enable greenscreen/chromakey effect. If more than one camera is available on your system, you will be able to choose the camera from a drop-down list.

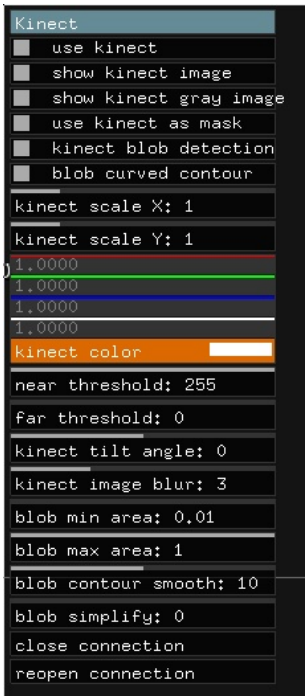
The following controls allow you to tweak the greenscreen/chromakey effect (as we have already seen you must enable it separately for video and camera in the corresponding sections of the gui).

You can set the threshold of the effect and the base color, default greenscreen color is black.

The threshold and key-color settings are the same for video and camera.

Last section of this column groups controls for slideshows.

In LPMT slideshows are folder of images that are presented in a endless loop sequence in a projection surface. Clicking 'slideshow button' you can choose a directory containing images that will be used as slides. By default the order of the slides is linked to image files names, and thay will be shown in alphabetical orders. Slides can be controlled as well by the timeline, as we eill see on the timeline chapter. You can set slide changing interval in seconds with the 'slide duration' slider. Two checkboxes allow you to adapt slides to surface size, and choose if they must fill surface respecting image aspect ratio or not.

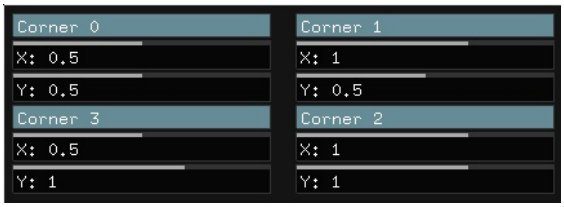


The third column of second gui page will be shown only if a Kinect sensor is attached and available on your system. At present only MacOs and Linux versions of LPMT are released as kinect-enabled.

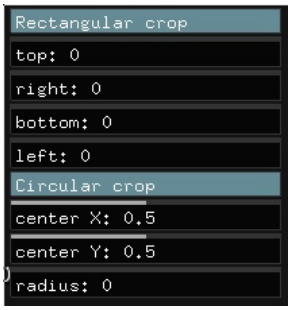
If a Kinect is connected and recognized, you will be presented with several controls. The two first checkboxes allow you to use kinect as source for active surface, and to choose if you want the resulting image displayed. In LPMT kinect is mainly used as a source for masking other content and to project dinamically on people or moving objects. LPMT does not use kinect in order to re-construct the 3D geometry of projection space.

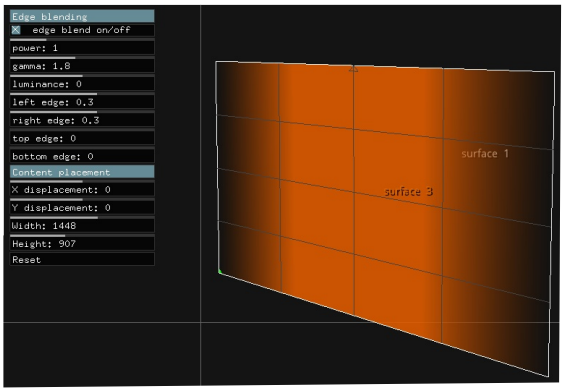
If you check 'show kinect gray image' the active surface will show the grayscale depth image from kinect sensor, otherwise it will display a 1 bit image of the portion of space contained in chosen depth range. You can set the depth range with 'near threshold' and 'far threshold' sliders.

If you check 'use kinect as mask' kinect source will be used as dynamic mask for the content of surface. 'kinect blob detection' will use a vectorial simplified version of image contours. The other controls allow you to further tweak kinect image, to change sensor's tilt angle controlling kinect motor and to close/re-open connection with sensor.



third gui page:
Activating third gui page with 'c' or 'F3' hotkeys, you will have access to sliders you can use to fine-tune position of surface corners and to controls for rectangular and circular cropping. We have already seen how to use them in the masking section of the manual.



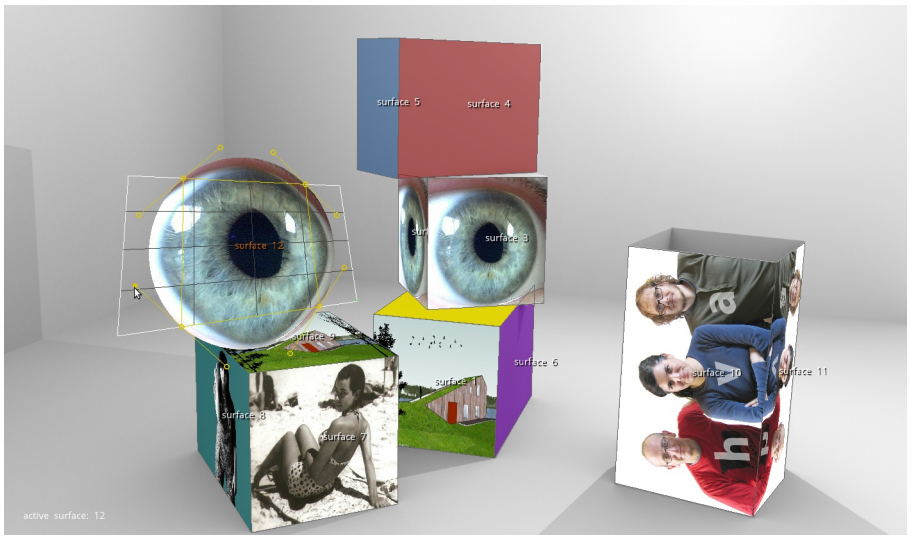


on the left: edge-blending allows to superimpose and seamlessly blend surfaces from different LPMT istances driving different projectors

on the right: a Kinect sensor can be used to enable dynamic masking of content and project on moving objects or people

Projection mode

Once you are satisfied with your setup you are ready for projection and you can enter Projection mode.
press 'spacebar' on your keyboard and you'll see background turn to black, gui and surfaces' frames disappear.



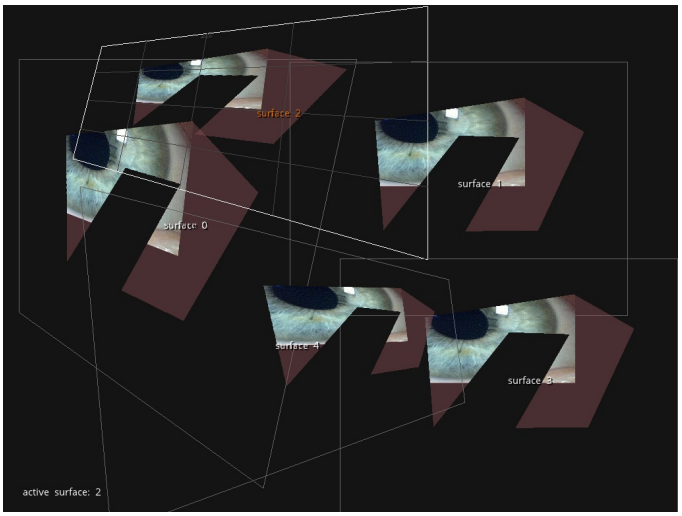
your finished stage setup in editing and projection modes.
you can switch to and from projection mode with 'spacebar'

There are several other hotkeys you may find useful to control your projection:
'o' key will turn all surfaces off, this can be useful if you want your projection to begin at a precise moment.
'p' key will turn the projection on again (play)
'r' key will re-sync all dynamic content (video, slideshows) to initial state.

Using the above keys for example you can: put LPMT in projection mode, turn surfaces off, put videos and slideshows to 'zero' moment and only when you and your audience are ready for the show, make magic start hitting 'p' key on your keyboard.

Saving and re-loading setup

Your stage setup can be saved at any moment with 's' hotkey. setup will be recorded in a file named '_lpmt_settings.xml' inside 'data' subfolder in LPMT folder structure.
Settings file can be reloaded with 'l' hotkey. Surfaces' position and content is saved in settings file, the file itself is a xml text file, with a very simple structure, and can be edited or changed with any text editor.
If you want to have more settings files, you can just copy and rename _lpmt_settings.xml.

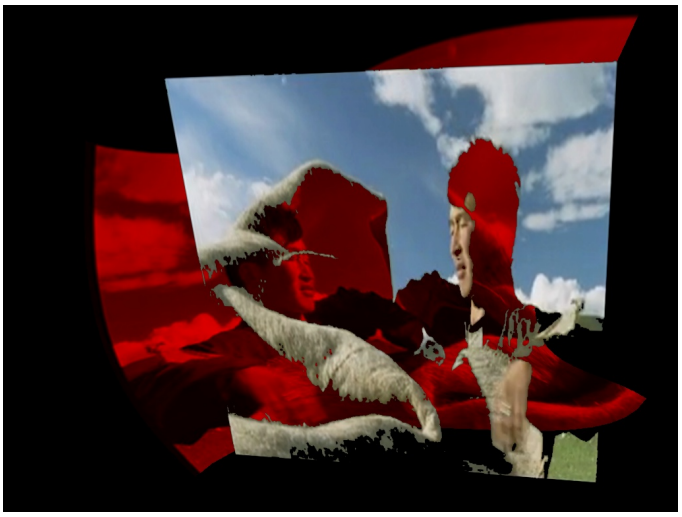


Copy settings between surfaces

A simple way of populating a surface that can be very useful in many cases, is to copy settings from another surface.
In LPMT this can be easily done by activating source surface, hit 'ctrl+c' on the keyboard, activate target surface by double-clicking it or using '<' and '>' keys, or even create a new empty surface with 'a' key, and then paste settings from source surface with 'ctrl+v' key combination. Source surface will remain the same until a new 'ctrl+c', so same settings can be applied to several surfaces in a sequence in a very quick and effective way.

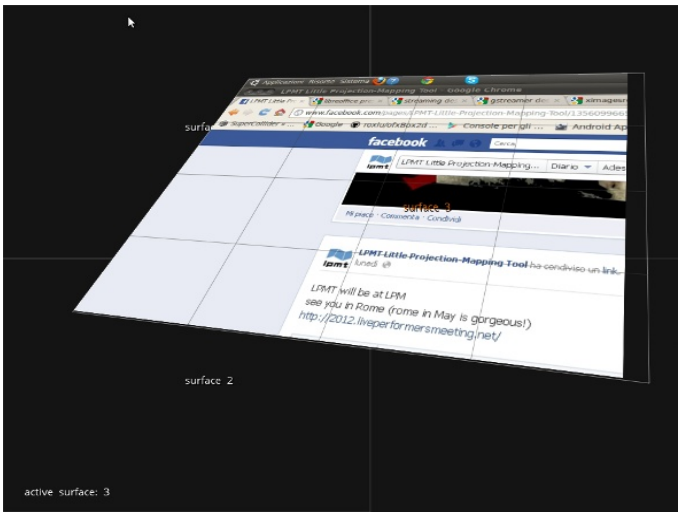
Cameras and virtual-cameras in LPMT

In LPMT you can use webcams as source input for your projection surfaces. Both USB and Firewire cameras are supposed to work on LPMT, as long as they are recognized on your system.
In order for LPMT to use a camera, you must enable it in a configuration xml file named 'config.xml' located in 'data' subfolder.
config.xml contains a <CAMERAS> tag, in which you can put multiple camera sections in following form:



above: with v4l2loopback virtual video devices, you can pipe live video content to LPMT surfaces, for example using Gstreamer.

below: desktop to LPMT with this gstreamer pipe 'gst-launch ximagesrc startx=0 starty=0 endx=640 endy=480 ! video/x-raw-rgb,framerate=30/1 ! ffmpegcolorspace ! v4l2sink device=/dev/video0'




```
<CAMERAS>
<CAMERA>
        <WIDTH>640</WIDTH>
        <HEIGHT>480</HEIGHT>
        <ID>0</ID>
</CAMERA>
<CAMERA>
        <WIDTH>640</WIDTH>
        <HEIGHT>480</HEIGHT>
        <ID>1</ID>
</CAMERA>
</CAMERAS>
```

In the above example, LPMT is expecting two cameras to be available and recognized on your system and will try to use them, if they can be both correctly initialized, the camera section of the gui will present you a dropdown list for each surface with available cameras you can use as content source.
Be aware that your cameras section in config.xml file should match the number of cameras currently available on your system, otherwise you may face crashes of LPMT.

On Linux systems a very interesting use of cameras is with v4l2loopback kernel module.
This module can create multiple V4L2 loopback 'virtual video devices'. Applications able to write to this kind of device can them be used to 'pipe' content to LPMT, that can be used as a very effective live-video projection tool.
v4l2loopback module can be downloaded from github (https://github.com/umlaeute/v4l2loopback) and is supposed to be possible to compile it on most recent kernels. Starting with Natty, it is also installable on Ubuntu (and other Debian-based distributions) through package-management system with package 'v4l2loopback-dkms'.
Recent versions of Gstreamer provide a v4l2sink module

that allow you to build Gstreamer pipes that output content to LPMT surfaces.
Once you have compiled v4l2looback module or installed a pre-compiled version for your kernel, you should load the v4l2loopback module as root :

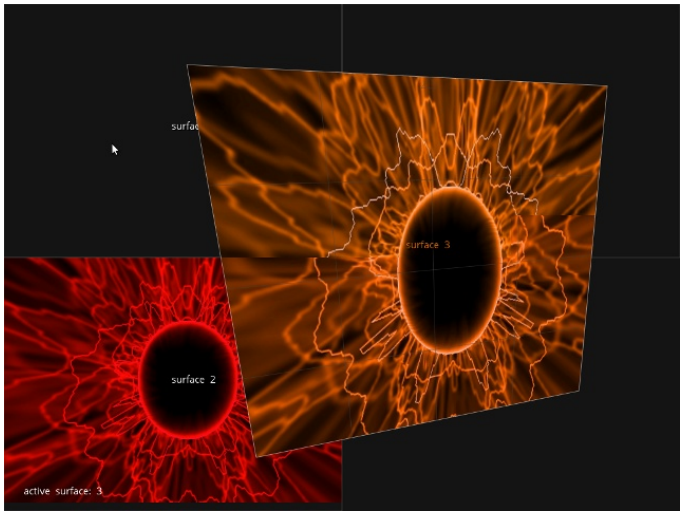
modprobe v4l2loopback

using sudo use:
$ sudo modprobe v4l2loopback

this will create an additional video-device, e.g. /dev/video0 (the number
depends on whether you already had video devices on your system), which can be
fed by various programs. Multiple virtual devices can be created using:

$ sudo modprobe v4l2loopback devices=n
where 'n' is the number of video devices you want to create.

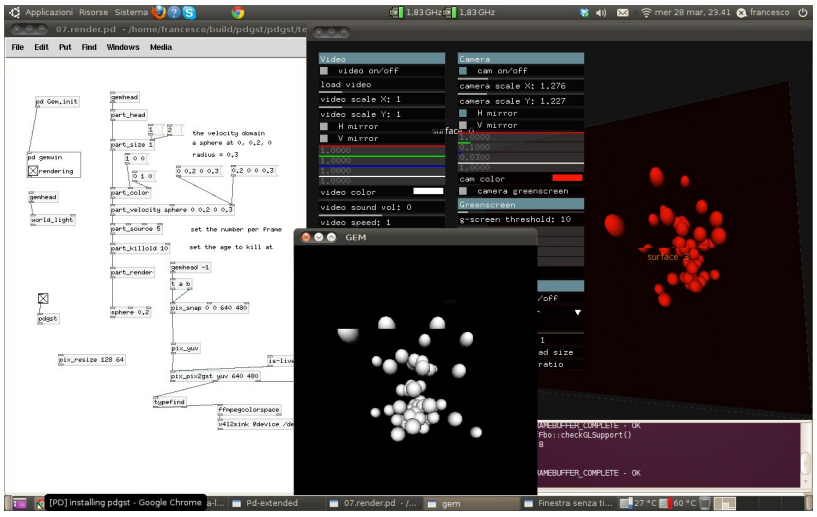
you can also specify the device IDs manually; e.g.
$sudo modprobe v4l2loopback video_nr=3,4,7
will create 3 devices named '/dev/video3', '/dev/video4' and '/dev/video7'.



above: visualizing pulseaudio output with 'gst-launch-0.10 pulsesrc device=alsa_output.pci-0000_00_04.0.analog-stereo.monitor ! queue ! audioconvert ! libvisual_infinite ! video/x-raw-rgb,width=640,height=360,framerate=25/1 ! ffmpegcolorspace ! queue ! v4l2sink device=/dev/video0'

below: puredata wth GEM sending live-graphics to LPMT

v4l2loopback feeders could be:
- GStreamer: using the "v4l2sink" element
- Gem(>=0.93) using the "recordV4L2" plugin
but most programs capable of writing/streaming to a v4l2 device should work.



Virtual video devices created with v4l2loopback module, provide a reliable alternative on Linux to a mac-only technology as Syphon frame-serving. Although not based on the gpu as Syphon, the video loopback devices have proven to be fast and versatile and open many interesting scenarios to LPMT linux users for feeding live content to projector.
Puredata or even other OpenFramewors applications can easily output content to v4l2 loopback devices.