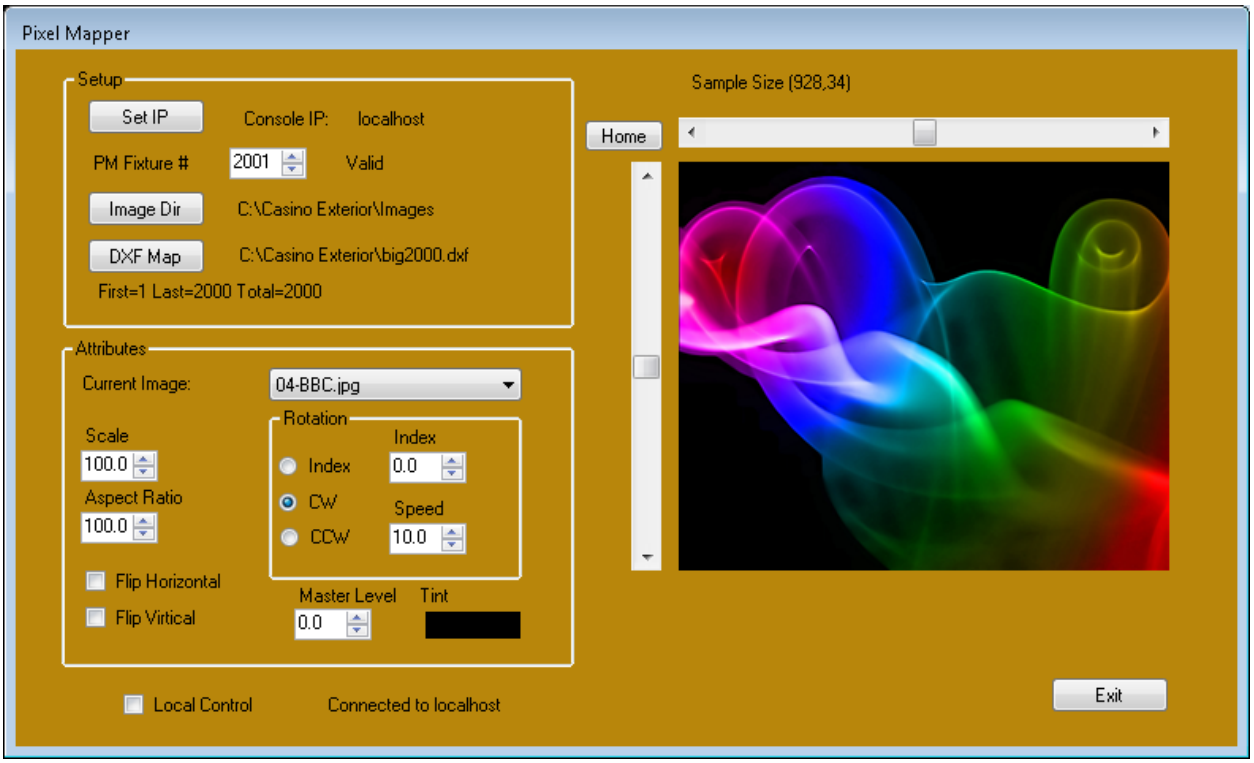
PIXEL MAPPER

Pixel Mapper is an OpenPalette application that is used to aid in controlling a huge number of colour mixing fixtures (typically RGB, but not necessarily). The problem with controlling a massive RGB rig is that simply crossfading the lights from one colour to another using cues or creating mathimatical based effects can get tedious and predictable very quickly. Working with the fixtures as an array of lights then overlaying graphics on that can produce much more organic or fluid effects and drastically reduces programming time.

To acheive this, a number of items must come into play:

- A large rig of colour mixing fixtures. Pixel Mapper deals with starndard geometric shapes (like 2D arrays) as well as obscure patterns with odd (non-linear) numbering sequences.
- A Palette show file with the rig of fixtures patched (addition AutoCAD tools are available to aid in patching huge RGB rigs).
- The “Pixel Mapper” fixture must be added to your show file (found in Generic section of the Fixture Library). It does not consume or need DMX output patching. It is only used to control the external Pixel Mapper application.
- A DXF map of the rig geometry showing where fixture numbers appear in 2D space.
- A selection of images as source material (*.jpg, *.bmp, *.tiff, *.gif). Only static images are supported.
- The Pixel Mapper application itself which reads the DXF map, manipulates the images and feeds the complex colour data to the Palette efficiently which in turn will control the rig:



Once you have the console connected to the Pixel Mapper application, you control it by tweaking the attributes of the PixelMapper fixture which is just another fixture number alongside your real RGB fixtures.

The PixelMapper fixture has attributes like any moving light, is recorded in palettes, subs and effects like any other lights and adheres to Tools like Highlight, Colour Picker and Move In Black just like any moving light would. The difference is that it dynamic feeds colour data to the rest of your rig in real-time which is resolved with all other colour data that maybe have been recorded in cues, looks and FXs.

Setup

Set IP: Press this button to assign the IP address of the console. If PixelMapper is running on a separate machine, you will need the real IP address. Localhost or 127.0.0.1 will work if you're running this application on the same PC as the Palette software. Connection status is displayed below the Attributes group box.

PM Fixture #: Enter the PixelMapper fixture number in your show file. If it's not a valid fixture number (i.e., you gave it the fixture number of a Vari*Lite vs. a PixelMapper), it will print Not Valid next to the fixture number. The PixelMapper can be found in the Generic Section of the Palette fixture library.

Image Directory: Press this button to locate the directory where your images are stored. Having them numbered with a two-digit prefix is desirable so as you add images, they are not randomly re-ordered. They will appear in the list alpha-numerically. If you're going to do a lot of programming, adding a profile in the Palette patch and assigning it to the PixelMapper's Gobo attribute makes selecting images much easier.

DXF Map: Point to the DXF file that maps out your fixtures. The PixelMapper application will read this file and determine if the map is valid then resize the sample image and aspect ratio to appropriate dimensions to represent a suitable down-sampled version of your rig. The map must use contiguous numbering (but can start at any number – not necessarily #1). The DXF files tested were AutoCAD v14 DXFs. The only thing in the DXF should be TEXT attributes on Layer 0. The scale and units do not matter, but the geometry and relationship between the fixtures should be correct and to scale. Origin, font, size and style does not matter. The TEXT values should be nothing more than whole numbers representing fixtures patched in your show file. After the file is read, you can see the file statistics below and the aspect ratio of the image and its scroll bars to the right are changes as well as the Sample Size text above the image.

Attributes

If **Local Control** is not checked off at the bottom, none of these controls are editable with the mouse, but they will change to reflect the PixelMapper's current attributes in Palette, be those captured or recorded values. If you do check of **Local Control** changing values here with the mouse will capture those values in Palette.

Current Image: This combo box will change as you change the PM's Gobo attribute. When you choose the Image Directory (above), this combo is populated with all the valid image file names found in the directory – sorted alpha-numerically.

Scale: The default scale, 100%, fills the preview space to the right. Images can be scaled from 1% to 1000%. They scale from the centre of the Fixture map as defined in the DXF.

Aspect Ratio: The valid values for the aspect ratio are between 1% and 200%. The default is 100% and will fill the preview pane when the Scale is 100%.

Flip: The Flip values are self explanatory and are accessed via S3 and S4 under the Lens Attributes of the PixelMapper fixture.

Rotation: These attributes are found under the Gobo family of the PixelMapper fixture. Index goes from -180 degrees to 180 degrees and rotation goes from 0 RPM to 120 RPM. Rotation happens around the centre of the viewable image.

Master Level: If the level is zero, the RGBs in your rig will get their colour from their own colour attributes. (Their respective intensity is always controlled by their own intensity value.) Regular cue levels are alpha blended with the sampled colours as the Master Level increases. If the level is at zero, only cue levels are seen. If the level is 100%, only the PixelMappers image is seen. This allows you to fade from recorded looks to images on cue. Any pixel that is BLACK in the sampled image will allow cue values to shine through (alpha = 0).

Tint: The colour attributes of the PixelMapper fixture allows you to tint the image towards the selected colour. The more saturated your tint colour, the more effect it will have. If it's black or white or gray (unsaturated), it will have no effect. There is no local control for tinting; the parameters must be set in the Palette software.

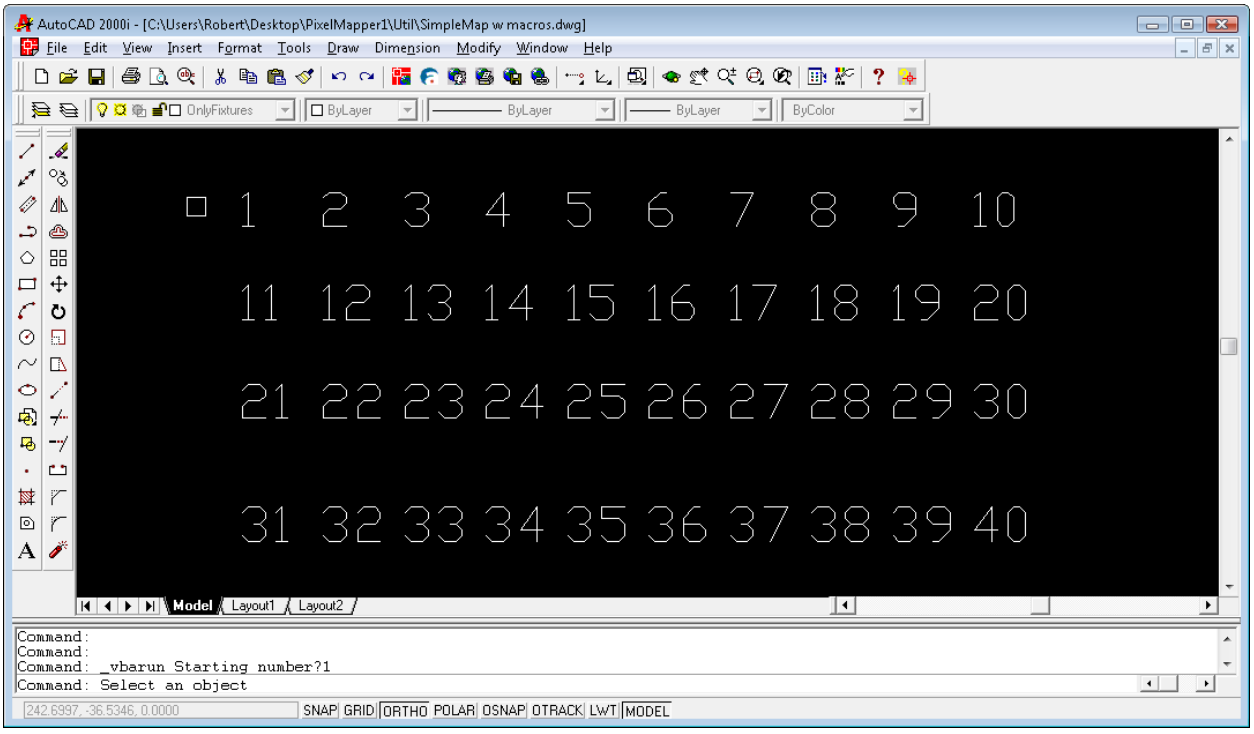
Preview Image

The preview pane on the right has scroll bars that represent Pan and Tilt on the PixelMapper fixture. Pressing the HOME button takes Pan and Tilt to 0,0. The relative lengths of the scroll bars match the aspect ratio of your pixel map (DXF). The extents are -100 and 100 degrees, but think of them as % of

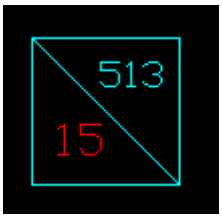
the screen. (i.e., a Pan of -100 will take the image just off the left hand side of the viewable area.) The images are displayed here in high resolution, but the dimensions of the actual down-sampled resolution can be read above the image.

AutoCAD tools to create the DXF Fixture Map

The AutoCAD macro language, Visual Basic for Applications (VBA), allows for repetitive manipulation of the entity database in an interactive way. Handy macros are embedded in the sample drawing "SimpleMap w macros.DWG" and can be accessed via the TOOL|MACROS menu. As described above, the only thing needed in the map are text entities representing the fixture numbers in your rig. Their relative spacing in 2D and the extents of their positions is all that is important. The text data should be whole numbers, starting at your first fixture and ending with your last fixture (contiguous – no skipped numbers). The easiest method for a simple grid of lights is to use the AutoCAD Array tool then use the VBA macro RenumberText(). It will ask for a starting number, then all you do is touch text entities one at a time and it will successively modify their values adding one each time:



If you want to manage both fixture numbers and DMX in one CAD file, there is a sample file called “RGB Block w Renumber Macro.DWG”. This is a very simple block which shows a fixture number (in red) and a DMX number (in cyan).



There is a macro called Renumber RGBFix(). It asks if you want to renumber the fixture (f) numbers or the DMX (d). When entering DMX numbers, you can enter absolute addresses (e.g., 513) or Universe.Offset (e.g., 2.1).

If you use this block, it cannot be read by the PixelMapper application (see above – only text entities are read). So – there is a utility macro called AddFixtureNums() that adds a simple text entity in the place of every RGB Block. You then have to DXF out those text entities (without the RGB blocks) so that they can be read. You can also use ExportPatch() which writes a comma separated file of the fixture patch which can be read by the application “ConsolePatch.exe”.