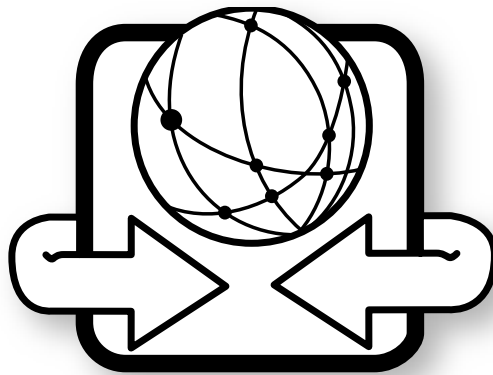


INTERNET & HDRI

How to show true HDR images on a website



Christian Bloch
Siggraph 2008

INTERNET & HDRI

How to show true HDR images on a website

Preface

Internet community galleries are one of the main reasons for the growing popularity of HDRI among photographers, yet none of the images on Flickr can be considered as real high dynamic range image. They are just tonemapped representations, more or less artful 8-bit snapshots of a larger dataset.

True HDRIs are kept in **floating point space** and allow **interactive exposure changes**.



Tapping through different exposures of an HDR image in Safari

This begs the question, how can we embed a true HDR on a website?

Enabling Technology

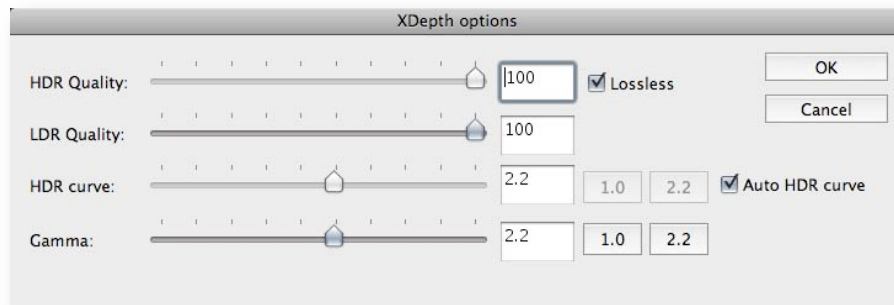
We need two things: an efficient file format that compresses HDR content to the size of a JPEG, and we need a matching viewer that gives us exposure controls and that can be embedded in a website. There are 4 qualifying file formats, but only two of them have a viewer:

1. **XDepth**, where the viewer is an ActiveX component for Internet Explorer.
Advantage: great compression ratio, visually high quality, compatible with JPEG-viewers.
Disadvantage: viewer needs to be installed on user's side, limited to Windows Vista/XP.
2. **FJPG**, where the viewer is a Java applet.
Advantage: Java applet is cross-platform compatible and runs without installation.
Disadvantage: Heavy compression artifacts, incompatible with other JPEG-viewers.

None of these two options can be considered ideal.

Using XDepth

The good news is: Creating an XDepth file is really as easy as it gets. You just download the file format plugin from www.xdepth.com, available for Photoshop or Lightwave. Now you can just save XDepth images with these programs, the standard extension is .xdp. Rename these files into .jpg right away. Since they are compatible formats, any basic image viewer can now open the image, even if it can't deal with HDR data. That's because all the data that exceeds the range a JPEG can hold is hidden in the metadata.



Saving an XDepth image in Photoshop

The bad news is: Web deployment is still in its infancy. Not only is the ActiveX component limited to the browser family that keeps giving web developers the worst headaches, even its installation on the user side is a bumpy road that involves digging deep into IE's Security settings. You can get a taste of it at www.xdepth.com/gallery.php. If you managed to install the ActiveX Component, you will find a web developer's guide in the installation directory on your hard disk. The XDepth developers are still working on a better web deployment method, possibly based on Flash, so my personal advice would be to wait.

In the meantime, we can use FJPG as an alternative, even though it has lower image quality.

What is FJPG?

FJPG stands for Floating Point JPEG and was developed by Helmut Dersch. Dersch is well known in the panorama community for his revolutionary PanoTools package, which is still to this day the foundation for most panorama stitching software. His viewer applet for FJPG actually a fully featured panorama viewer, with hotspots, scripting access, screen mask overlays and all that good stuff.

The format itself is a pretty simple hack, following the same RGBE encoding strategy as the Radiance format. This means a 4th channel E carries an exponent, that is used to propel the RGB image values into high dynamic range regions. The problem is, this exponent is always shared by an 8 by 8 pixel block. High contrast corners or small bright details will therefore show significant artifacts. In a technical sense, FJPG is a just Radiance file with JPEG compression applied, an unholy marriage that gives image engineers the creeps.



Typical artifacts due to FJPG encoding method

Generating an FJPG

The original command line utility for this file conversion is PTConvert. Download it from <http://webuser.hs-furtwangen.de/~dersch/hdr/ptconvert.zip>. You can read the included documentation for in-depth information on usage. The basic workflow is this:

1. Downsize your spherical HDR panorama to be no bigger than 3200 by 1600 pixels in Photoshop, Picturenaut, Photomatix etc.
2. Save as Radiance format in the ptconvert folder.
3. Open a command line prompt, cd into the ptconvert folder and run PTConvert.exe with the basic arguments:

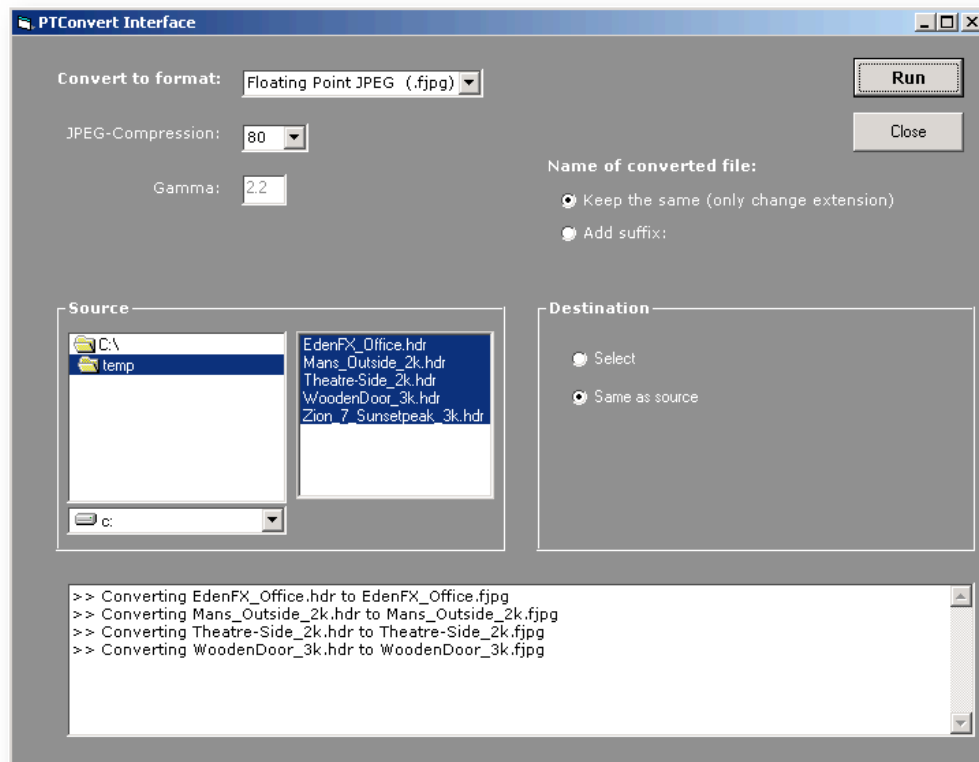
```
ptconvert.exe Inputfile [-o Outputfile] [-q quality]
```

For example, this is the command to convert the file Zion_Sunset.hdr:

```
ptconvert.exe Zion_Sunset.hdr -o zion_sunset.fjpg -q 80
```

The resulting image will be back in the ptconvert folder.

If you have a strong aversion against command line hacking, you can also use the GUI-based version. It is available for free from www.hdrsoft.com/download/ptconvertint.html



Usage is straight forward and should be self-explanatory. Just select a batch of Radiance files, set compression ratio (80 is usually good), and hit Run.

Note that even though PTConvert runs on Windows only, it works fine on a Mac equipped with Boot Camp, VMware Fusion or Parallels. (This is what this screenshot was taken with.)

Embedding an FJPG in a website

You need the PTViewer 3.1.2 Java Applet from <http://webuser.hs-furtwangen.de/~dersch/>. The download is halfway down the page (make sure to grab version **3.1.2!**). There are several example pages in that section; download them all because you can learn a lot from looking at the source code from these pages. The Readme file for 3.1.2 explains all the HDR-specific parameters, but it is really an addendum to the [Basic PTViewer Documentation](#).

Here is just quick tutorial to get you started:

If you're using a WYSIWYG website editor, you need to find a spot where you can enter some pure HTML code. For our example, let's make a page from scratch starting out with a very basic skeleton of a website:

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN">
<html>
  <head>
    <title>Zion Sunset</title>
  </head>
  <body bgcolor="#585858">
    <div style="border: 1px solid black; padding: 5px; width: fit-content; margin: 0 auto 10px auto;">
      Our content goes in here.
    </div>
  </body>
</html>
```

Place ptviewer.jar and your FJPG in the same folder as this HTML file. Subfolders are fine as well, if you like your content to be structured. Now this would be the code snippet for embedding PTViewer in our page:

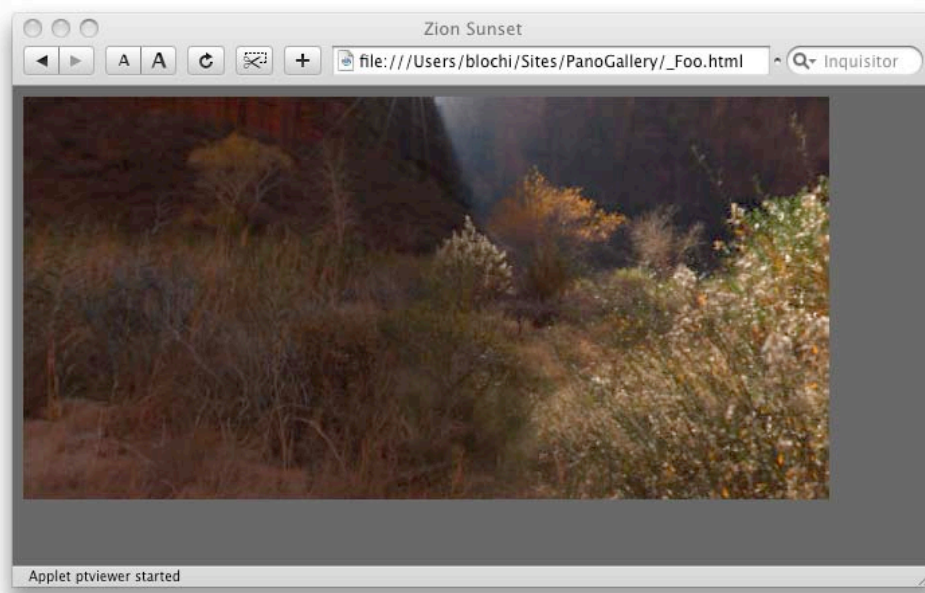
```
<applet name="ptviewer" archive="ptviewer.jar" code="ptviewer.class"
  mayscript="false" width="600" height="300">
  <param name="file" value="media/zion_sunset.fjpg">
  <param name="exposure" value="300">
  <param name="gamma" value="1.6">
  <param name="autorange" value="30">
  <param name="quality" value="2">
</applet>
```

Pretty straightforward. The basic parameters are:

- **width , height** - size of the viewer window.
- **file** - path to your FJPG.
- **exposure** - sets the default exposure, found values between 300 to 800 useful.
- **gamma** - post adjustment to get the HDR out of the linear space. I prefer 1.4 to 1.6 for keeping it catchy and contrasty.

- **autorange** - regulates how aggressive the auto exposure works while panning in the panorama, similar to the area of exposure metering in a camera viewfinder. Smaller values result spot metering, exposing the entire view according to the brightness of the pixels in the center of the view. Larger values result in more broad metering and smooth exposure changes while panning.
- **quality** - Sets the interpolation mode, values range from 0 to 3. 0 will always use Nearest Neighbor interpolation and is very fast, 3 will always use Bilinear interpolation and is of much better quality. 2 uses NN for panning and Bilinear while standing still.

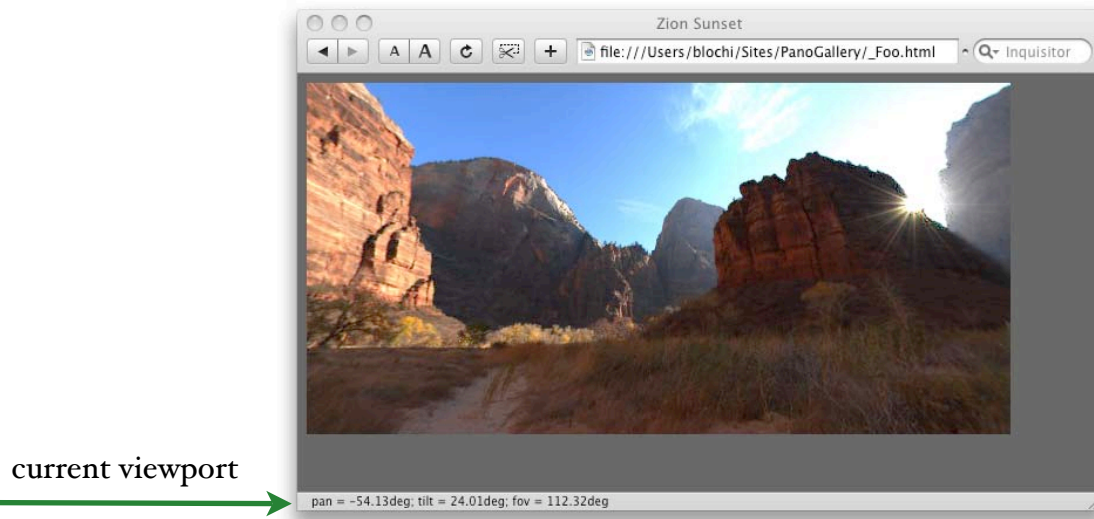
For now, our website looks like this:



It works, but the initial viewpoint isn't very exciting. There is also no indication that this actually is an interactive panorama, so it makes a lot of sense to include some instructions for the user. Standard usage is:

- Drag mouse to pan
- Zoom with [+] and [-] keys
- Change exposure with [b] and [d] keys (for **b**righter and **d**arker).

Use these controls in to find a better viewpoint and and then hit [v] to display the current viewport parameters in the status bar.



Now we can steal these values to configure the default viewport by adding some parameters to the HTML file:

```
<param name=      "pan" value="-54">
<param name=      "tilt" value="24">
<param name=      "fov" value="112">

<param name=      "fovmin" value="30">
<param name=      "fovmax" value="130">
```

I have also added limits for the field of view, set these according to the resolution of your panorama. There is a ton of other parameters that you can use, all described in detail in the documentation on <http://webuser.hs-furtwangen.de/~dersch/PTVJ/doc.html>.

Here is a selection of parameters I found most useful:

```
<param name=      "maxarray" value="140000">
```

A somewhat mysterious value, that causes Browsers to assign more memory for the Java Applet. This speeds up the initialization of the viewport, or makes it even possible in the first place when you have a large image. Essentially, this will set the maximum image size you can pipe through, the value shown should be good for images up to 3200 by 1600 pixels.

```
<param name=      "bgcolor" value="585858">
```

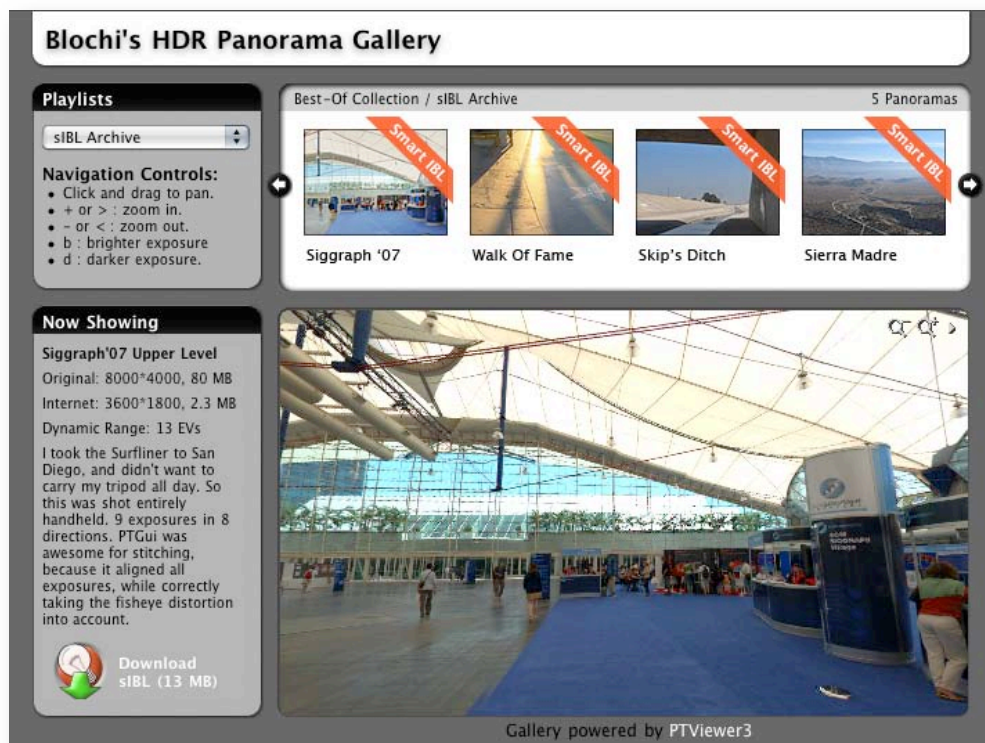
Defines the background color, shown while the panorama is still loading. Set it to the same color as the background of your HTML page to make display more seamless.

```
<param name="cursor" value="move">
```

Changes the cursor while the users hovers over the panorama, giving the user an additional clue that he can pan on this panorama.

That's about it, happy embedding!

To see some more examples, check out my HDR Panogallery, which happens to also serve as the download central for Smart IBL lighting sets: www.hdrilabs.com/gallery/panogallery.html



Also, you can easily see your own FJPGs without any coding. Just sign up in the HDRI Community forum on www.HDRILabs.com and attach an FJPG to your post. The forum software automatically takes care of linking it to PTViewer and embedding it in your post. Pretty neat, huh?

Christian Bloch, EdenFX