

Handling Mobile High-Density Displays



☆

Your Rating: ☆☆☆☆☆ Results: ★★☆☆☆ 80 rates

With the iPhone 4, supporting images on mobile devices got more complicated. This is because of Apple's "Retina Display", which has approximately 4x as many pixels in a given space as previous displays (326dpi). One can still use standard-resolution images, but they come across as looking blurry or chunky for users accustomed to the sharper images the display can provide.

Since then, several other mobile vendors have also introduced devices with high-density displays.

Information on screen density can, in most browsers, be obtained using javascript's `window.devicePixelRatio` property, which is a floating value that indicates how much a regular image must be scaled up for the device's display. Its value is 1.0 for older, standard-resolution devices, but 2.0 for the iPhone 4. In addition, [various Nokia and Android devices](#) are known to return values of 1.5, 1.3, and 0.75.

In Texas State's implementation, we had an image-sizing utility written in Perl that would accept the desired dimensions for an image as URL parameters. Thus, we had a method that would wait for a **dom:loaded** event (which fires after the DOM is built, but before images are loaded), and would rewrite the SRC attribute of any images with a **txst-multiresolution-image** class to request a higher resolution based on the current **devicePixelRatio**:

```
// resize multiresolution images as needed
Event.observe( document, "dom:loaded", function() {
  if ( window.devicePixelRatio ) {
    $$('.txst-multiresolution-image').each( function( item ) {
      var src = item.src;
      src.match(/width=(\d+)/);
      var originalWidth = RegExp.$1;
      src.match(/height=(\d+)/);
      var originalHeight = RegExp.$1;

      src = src.replace(/width=(\d+)/, "width=" + originalWidth * window.devicePixelRatio );
      src = src.replace(/height=(\d+)/, "height=" + originalHeight * window.devicePixelRatio );
      item.src = src;
      item.width = originalWidth;
      item.height = originalHeight;
    });
  }
});
```

Note: this code relies on the Prototype Javascript library, and would probably need to be tweaked to work with our jQuery stuff. Additionally, it's important to note that the item's dimensions are still required to be the original values so that the image doesn't get larger on the page, only more detailed.

See <http://www.theatreanddance.txstate.edu/> for an example of our image gallery in action. (It also uses swiping gestures to scroll through images, though I was never quite happy with how smooth we were able to make them.)

Because the Magnolia imaging library relies on predefined image variations, we can't use this exact approach, which relies on being able to specify arbitrary dimensions for an image as part of the request. It should be possible to adapt to a range of variations without too much trouble though.