


Augment Photos With CSS3 matrix3d() Transforms

 www.useragentman.com/blog/2015/07/16/augment-photos-with-css3-matrix3d-transforms/

22:00 00

Jul 24,2015

The image above has dynamic HTML content super-imposed on top of a photo, styled using CSS 3D transforms to give the appearance that it is part of the picture. If you are familiar with 3D transforms, you can imagine it would be very difficult to figure out what combination of `translate3d()`, `scale3d()` `rotate3d()` functions will produce the pixel perfect effect needed. I knew it was possible to get exactly the shape needed using `matrix3d()`, but that it would be a chore to calculate the numbers to do this by hand. For example, for the above example, we need the following CSS to style the digital watch face.

```
transform: matrix3d(0.515078, -0.199333, 0, 0.00021,  
                   0.502268, 0.802615, 0, 0.00008,  
                   0, 0, 1, 0, 438, 240, 0, 1);  
  
transform-origin: 0 0 0;
```

Unfortunately, those numbers aren't the (x,y) coordinates for each of the corners of the transformed box that contains the date and time. They are the [values of a 4x4 matrix used in computer graphics to produce 3d effects](#).

If you have no idea how matrix arithmetic works and you don't have a degree in math, you may think you are out of luck here. However, I created a tool, [The Matrix Construction Set](#), to allow you to do this with a nice point and click GUI using HTML5 Drag and Drop. Take a look at this video to see how easy it can be:

A video posted by Zoltan "Du Lac" Hawryluk (@zoltandulac) on Jul 8, 2015 at 1:27pm PDT

[Generate your own matrix3d\(\) demos with The Matrix Construction Set](#)

The pink object represents the original HTML object without a transform applied, the purple one is the object transformed. You can transform the object using the red crosshairs on the corner of the purple object. As you do so, you will see the CSS above the grid change, giving you the correct `matrix3d()`. Click on the **Show Advanced Features** in the top nav and you'll be able to edit the HTML and CSS of the transformed objects so you insert your own images, fonts and text. When you are done, you can bookmark the page so you can share your proof-of-concept-creations with other developers and designers. Here are a few examples of some stuff I have created with the tool ([you can see the original, untransformed HTML by unchecking the Hide UI link in the tools main nav](#)):



This [Matrix Animated GIF](#) is placed as a background image in a DOM element. Along with the `matrix3d()` used to give the illusion it is in the screen, I used [a variation of the border-radius trick](#) to make HTML block the shape of a retro TV screen.



It is now possible to float text on top of 3D objects, like [JJ Abrams did on the TV show Fringe](#). The result is a look reminiscent of the [Constructivist Art Movement](#) (Photo Credit: [Hernán Piñera](#))

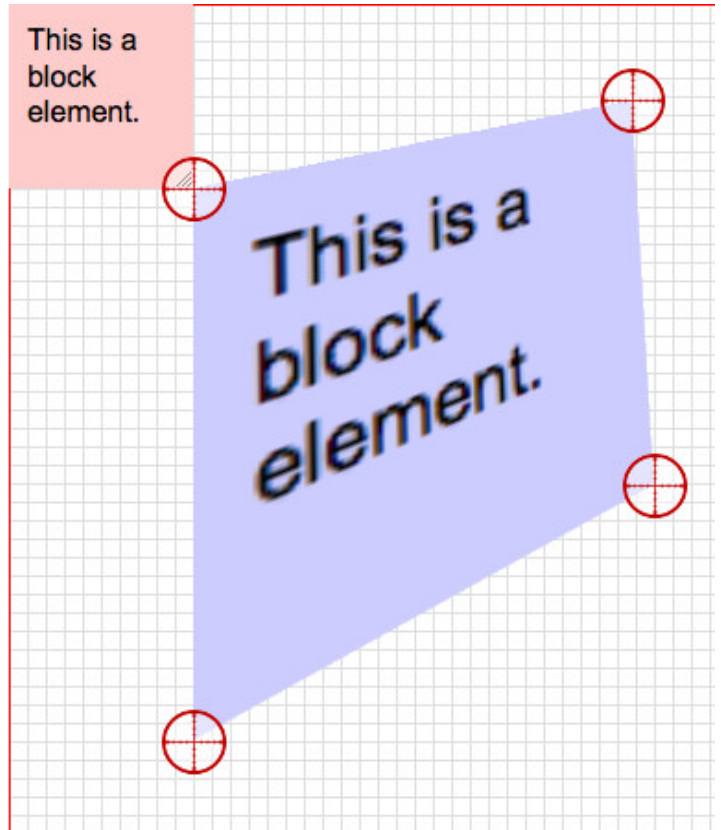
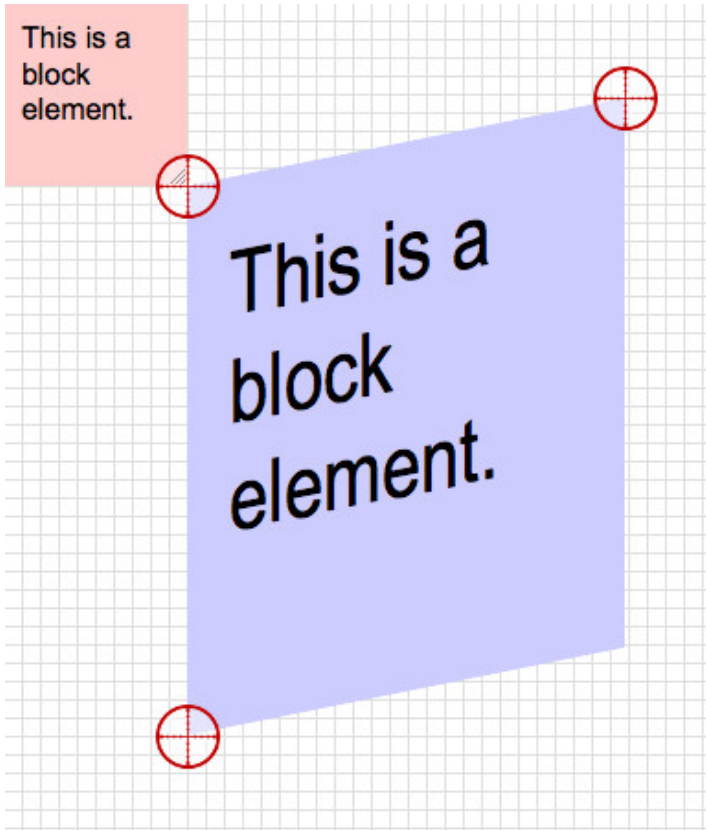


While we are talking about Constructivism, here is a animated version of a potential album cover of my band, [Conflicting Plaid](#) spoofing *Knigi* by [Alexander Rodchenko](#).

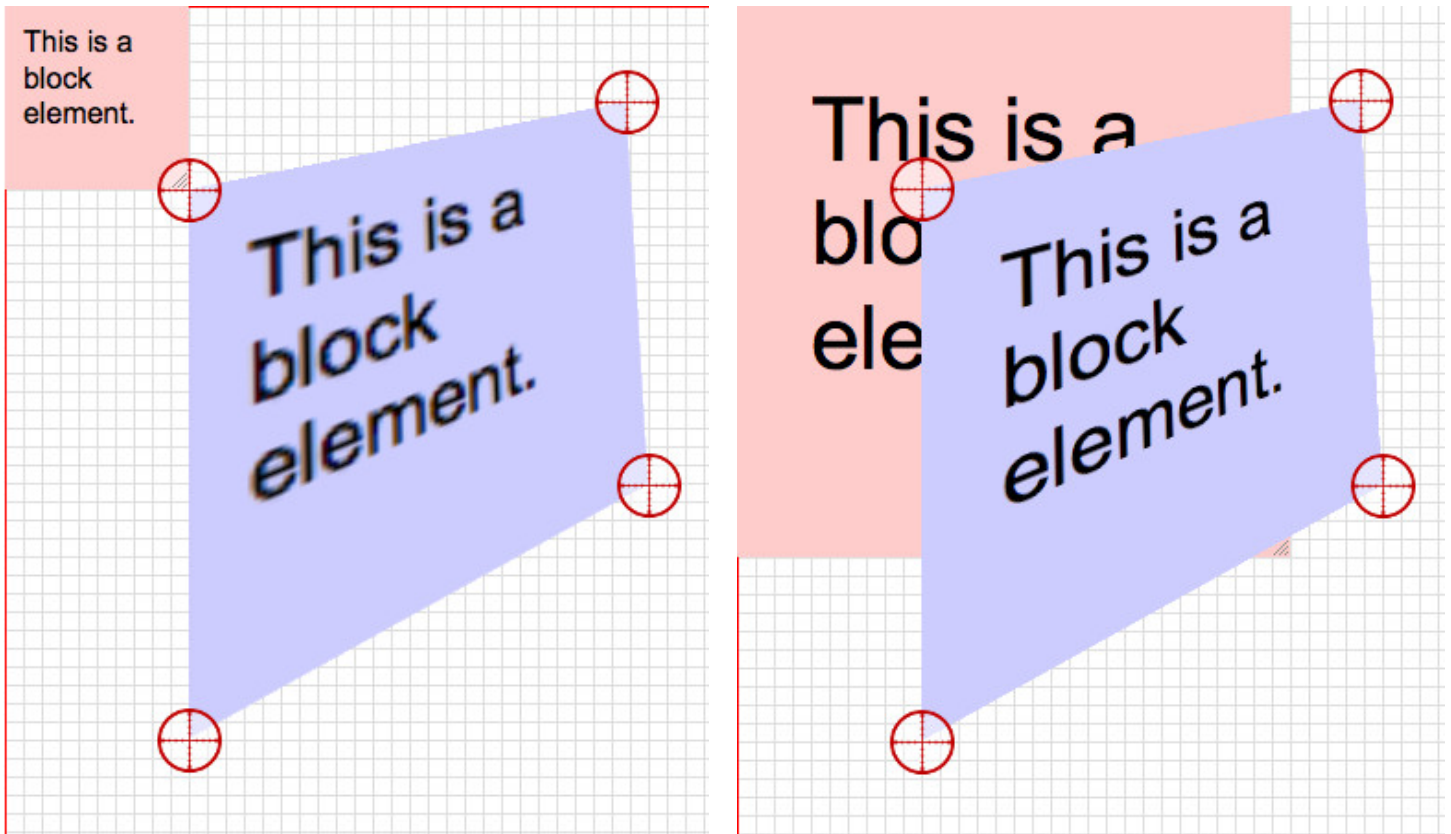
Difference Between `matrix()` and `matrix3d()`

Note that the Matrix Construction Set has been around for a while for doing 2D `matrix()` transforms. Over the years, [Federico Brigante](#), [Gerard Delmàs](#) and others have requested a 3D version since the **2D `matrix()` function only allows you to manipulate three corners** of a rectangle, while **3D ones allow manipulation of all four**.

The one drawback is with `matrix3d()` (and 3D transforms in general) is that a bitmap of the original, un-transformed object is transformed, including fonts. 2D transforms will transform typography's vector data, resulting in crisper results:



This can be worked around by making the original object (and typography) larger and then transforming it to a smaller object. Here is the same example as above with the `width`, `height` and `font-size` increased by 300%:



Fallbacks

Note also that **IE9 and Opera Mini do not support 3D transforms** (however, IE9 *does support 2D transforms* using `-ms-transform`). In order to handle these browsers correctly, you should either ensure that the original untransformed object is positioned someplace that makes sense without the transform applied (this can be done in the tool by dragging the red object around, which adjusts the object's `transform-origin`). You can also use [Modernizr](#) to give alternate styles to browsers that don't support 3D transforms if you want to get fancy.

I know some of my readers might have expected a polyfill for older IE, since I've done some previous work supporting 2D transforms [with](#) and [without Javascript](#). There isn't one that I know of, and I don't think it can be done accurately, if at all.

Other Caveats

1. The Matrix Construction Set currently only works on Desktop web browsers, due to mobile browsers lacking support of [the HTML5 Drag and Drop API](#). There is a [polyfill that translates drag and drop events to touch events](#), but unfortunately [it doesn't yet support offsetX/Y and layerX/Y](#), which are needed for the pixel-precision this tool needs. I may take a stab at adding this feature in the polyfill and updating the Matrix Construction Set with it in a future release.
2. The Matrix Construction Set is not supported in IE10, even though it does support 3D transforms. The code it produces, however, will work in IE10 and above.

3. The tool only allows to insert a small subset of HTML in the transformed object, due to the dangers of XSS attacks by allowing arbitrary HTML. The list of HTML tags supported can be seen in the [Sanatize.js recipe that I use](#) to restrict the HTML devs can use in the tool. If you want the tool to support more, please feel free to [grab the source of the Matrix Construction Set](#) to add the support you wish on your own web server.

Further Reading

Shout Outs

Although this tool was built using libraries I have created over the years ([config](#), [visibleIf](#) and [DragDropHelpers](#)), it would not be as cool as it is without the following free and open source projects: