

Blog svg to pixel

In this blog, SVG and PIXEL, which are two types of computer graphics and image displays, have a talk about their characteristics, features, pros and cons.

Who is not comfortable with painting with *GIMP* or other digital color pixel editor could be interested in the SVG world.

SVG, Scalable Vector Graphics, is a way of doing graphics or images on computers, on the basis of objects defined by properties and attributes, mainly geometric. Typically allowing scaling the image, and generally zooming without loss of continuity.

Link to the blog article: [link](#)

Discussion

[artgraphic-db]: Hi !

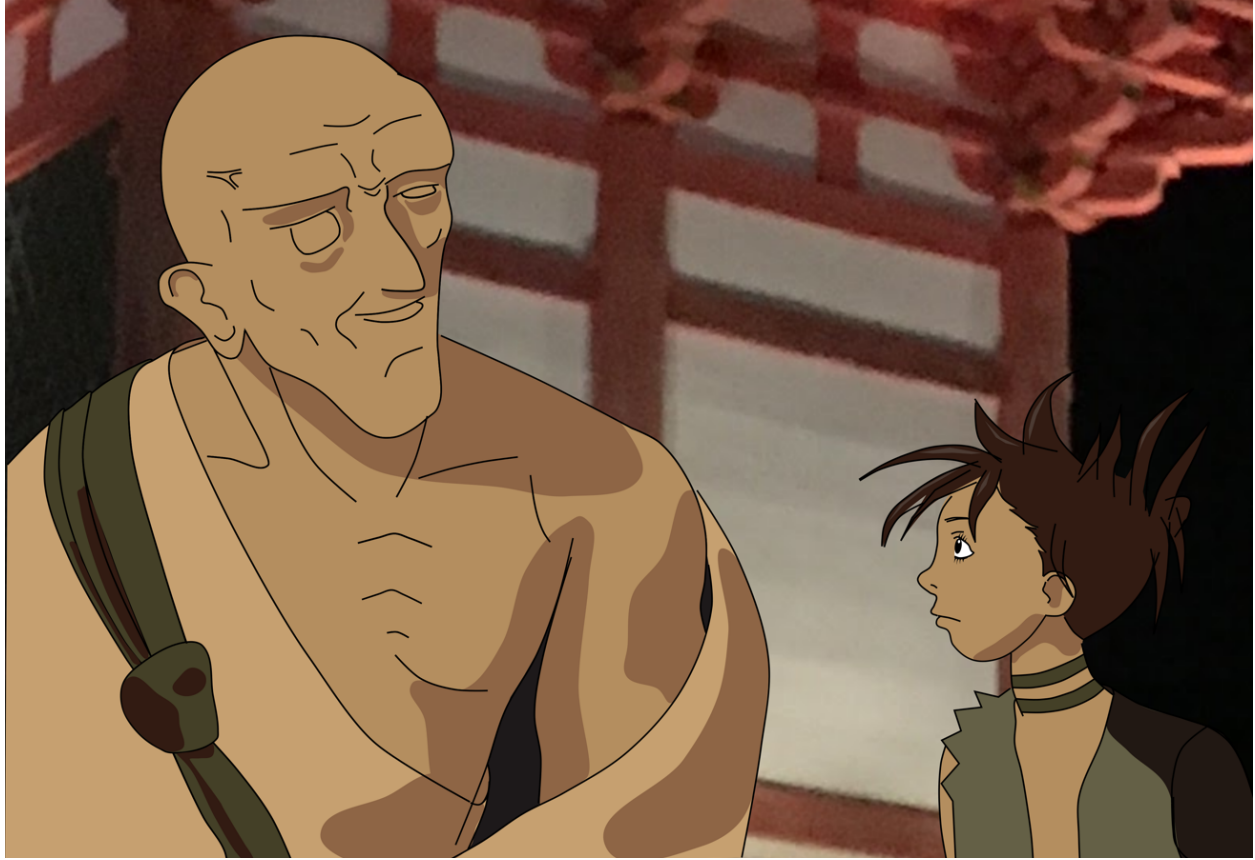
SVG: Hello folks.

PIXEL: Hi encounters.

PIXEL: This picture, this “dororo” and biwamaru, is it made of pixels or SVGs ? I want the GOAT exercise on that !

¹([artgraphic-db] PIXEL maunders ironically Dororo in “dororo”)

²([artgraphic-db] the GOAT exercise has become a famous hoax from GIMP about some on-coming intelligent scripting capability, but it is just a simple color invert filter, in fact, nothing to see with SVG graphics objects, and, in fact, the GOAT exercise CANNOT be done on this Dororo and Biwamaru image (the link), which is an SVG graphic, and GIMP does not handle SVGs. However it can be done on the shortcut display below, which IS a raster image).



Link to the SVG image editor inkscape (v0.92) example file (more layers) - Dororo and Biwamaru (variant image): [link](#)

SVG: I do not have GOAT !

([artgraphic-db] true, SVG does not have GOAT, which comes with GIMP, but DOES HAVE a color invert filter).

[artgraphic-db]: SVG, PIXEL, artgraphic-db is very pleased to present a discussion about pros and cons, differences and similarities between you both, SVG and PIXEL, thus, let us start the discussion with the question of who, or maybe, what, you are and what are your main features or characteristics.

SVG, PIXEL: thank you for the question (lol), and thank you, artgraphic-db, for this opportunity...

SVG: if I do not have the privilege of years, I would be pleased to introduce myself as the lucky child of other members of the *Vector Graphics* family; thinking to *postscript* (and PDF), and *.DWG*. I usually go by *SVG* which is an acronym for *Scalable Vector Graphics* to emphasize on my ability to scale on a wide range of zoom (*inkscape*, vector graphics editor, for instance, can zoom up to 26000% my graphics, and still being able to show the native shapes of the geometries).

PIXEL: I am now the very famous smallest constitutive element of displayed images (computer imaging concept), as well as the smallest element of device displays and printing techniques...

([artgraphic-db]: printers compose images by applying dots of ink)

SVG: the usual inevitable “*computer imaging concept*” that is being served systematically...

PIXEL: sorry for that, but nobody will go through without this “*concept*”, and the reality of display constraints on electronic devices...

SVG: LEDs technologies definitely imposed the PIXELs everywhere, but, I have found and used many turn-arounds to give an impression of continuity and quality to my graphics and make users forget about ugly renders of arrays of squares...

PIXEL: I make images by putting together these arrays of squares, I make bitmaps!

And, ugly renders do not exist with enough image resolution; Image editing apps cannot trick their max zooming capability, and nobody is prevented from using an image resolution to avoid the “*ugly*” that you are talking about...

At the end of the day, you will have to convert to the PIXEL world...

([artgraphic-db]: PIXEL refers to a turn around for avoiding “steps” rendering in lines when drawing at a resolution, in fact too small. Illustration below.)

First, a canvas of 1920px by 1080px (the size of the screen for display) is used to draw a black line, which shows acceptable on the left, at a zoom level of 100% (dot for dot), but shows “steps”, one says “loss of quality” at a zoom ratio of 10:1.



Second, the suggestion of PIXEL is followed, and now a canvas of 19200px by 10800px (10 times the size of the screen for display) is used to draw a black line, which shows very acceptable on the left, and still acceptable at a zoom level of 1000% (ratio 10:1, this time, dot for dot) -no more steps-.

But, this is just a turn around, and what is wanted is to compare 100% of the first case to the 10% of the second case, and the 1000% of the first case to the 100% of the first case.



Finally, rescaling to the 1920px by 1080px canvas and showing at a 1000% zoom level on the snip below.



These snips conserve the real sizes as the image is displayed on the screen at the mentioned ratios.

This illusion is similar to working at a resolution 10 times bigger (say, 1000 dpi rather than 96 dpi) to keep a good “quality” at the desired zoom ratio of 10:1.

SVG: Look, I can keep my rendering if I stay in the digital world, I can have outputs for the internet, I can have outputs for electronic displays, browsing my images on the internet allows changing of scale, zooming, keeping a line a line or a circle, a circle, with no change to “broken lines” or “steps” rendering...

PIXEL: This is because your editors use an idea of “dynamic” canvas, that is not frozen at the beginning, unlike painting or retouching editors, but chosen at the step of image exporting.

SVG: “Exporting” to your world, PIXEL, is just an ability of my editors that is more a compliance than a benefit (*laughs*); converting my vectorial metadata to bitmaps is a disadvantage, but allows creating a raster image at any desired resolution (96 dpi, 300dpi, 1500 dpi).

([artgraphic-db]: here, SVG uses the wording “raster” image instead of bitmap, which PIXEL does not like; raster images can be exported in PNG format).

([artgraphic-db]: SVG refers to graphics objects as metadata^{see 6-25-2024 note}; graphics objects are made as vectors; for instance, in a vector graphics editor, a rectangle can have seven

components (25 25 200 200 "lime" 4 "pink"), which reads like `<rect x="25" y="25" width="200" height="200" fill="lime" stroke-width="4" stroke="pink" />` by the editor, and still comprehensive by an internet browser)

6-25-2024 note: SVG webpage says “Metadata: In accord with the W3C’s Semantic Web initiative, SVG allows authors to provide **metadata** about SVG content.”; “Metadata (or metainformation) is “data that provides information about other data”, but not the content of the data itself.../...” meaning that the graphics objects could be metadata of the data describing them in SVG language, or more precisely in XML language, I guess. More common metadata are, in the same representation space, information about other data, which could be, author name, address, ..., which have nothing to see with the graphics objects and would be also described in XML language.

Example of Metadata in SVG language (NB it might not been accessible by the Vector Graphics Editor or Viewer):

```
<metadata
  id="metadata3756">
  <rdf:RDF>
    <cc:Work
      rdf:about="">
      <dc:format>image/svg+xml</dc:format>
      <dc:type
        rdf:resource="http://purl.org/dc/dcmitype/StillImage" />
      <dc:title></dc:title>
    </cc:Work>
  </rdf:RDF>
</metadata>
```

See a later blog for more information.

PIXEL (smiling): Again, nobody prevents from starting over with new images at the desired resolutions...

Image retouching softwares used to say 300ppi is a good resolution and worth using if drawing artists or digital artists want to print. But “printer manufacturers” can work up to 5760 x 1440 dpi and that has to be thought at the starting point of an artwork.

([artgraphic-db]: Commonly, the maximum DPI of a printer is restricted by the hardware it's using. For instance, while inkjet printers can have a resolution ranging from 300 to 720 DPI, laser printers can go anywhere from 600 to 2,400 DPI.)

SVG: I have a question for you, PIXEL, what can you do if the resolution of your image is not adapted at the end of the day, if you need a finer resolution or a different resolution?

PIXEL (*thinking*): ...

SVG: I would be pleased to help you or to revert you to a third party service, converting your pixels back to vector graphics objects; however, my vector graphics editors have capabilities to revert your grids of squares back to work, tracing to vector graphics and letting exporting finer images ([The Complete Guide To Using Trace Bitmap in Inkscape](#)).

Vectorization

In the sense of [artgraphic-db] vectorization is not only meant in the sense of arrays of X,Y,Z but also other types of arrays like for example:

```
<rect x="25" y="25" width="200" height="200" fill="lime" stroke-width="4" stroke="pink" />
```

which can be read as the following vector (25 25 200 200 "lime" 4 "pink") with coordinates x="25", y="25", width="200", height="200", fill="lime", stroke-width="4", and stroke="pink".

In the sense of Peter Selinger, who [wrote](#) about the program POTRACE, the process of converting a vector outline to a bitmap is called rendering. The converse process of turning bitmaps into outlines is called tracing. A vector outline describes an image via an algebraic description of its contours, typically in the form of Bezier curves.

This way of tracing images is also called vectorization.

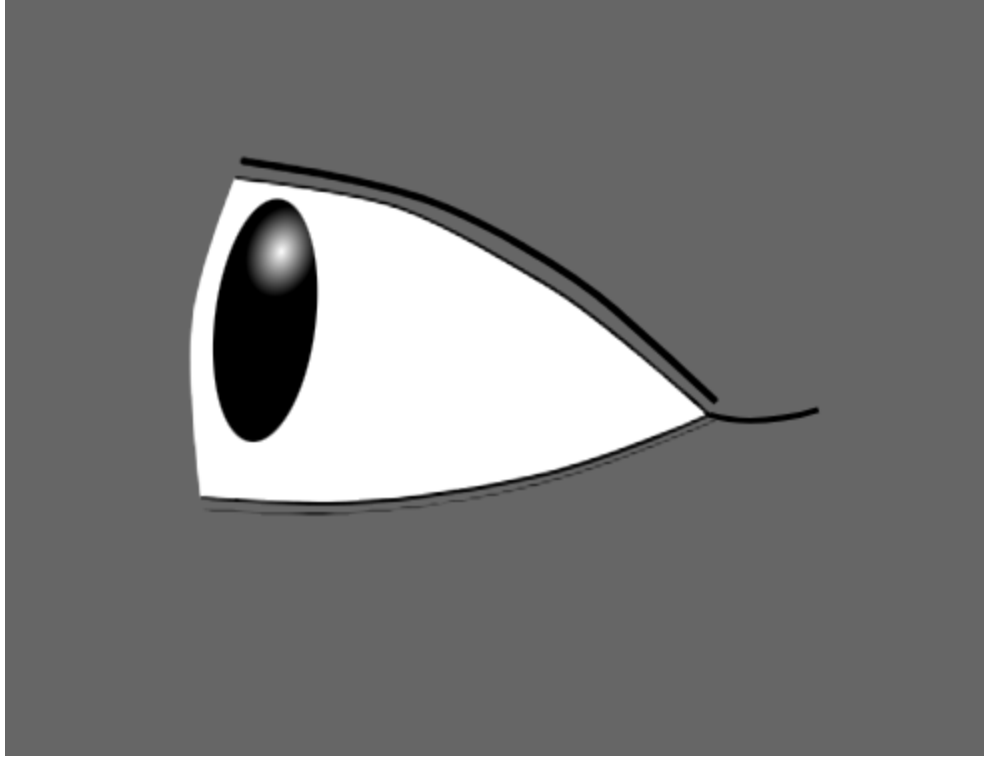
Tribute to Dororo

Link to the SVG image editor inkscape (v0.92) example file (more layers) - Dororo and Biwamaru (alternate image): [link](#)

This example shows an image worked out onto several layers, no bitmap tracing has been done, a first sketch has allowed to reproduce the proportions inspired by the cartoon series *Dororo*.

An eye

Link to the SVG image editor *inkscape* (v0.92) example file broke down onto few layers - *the eye*: [link](#)

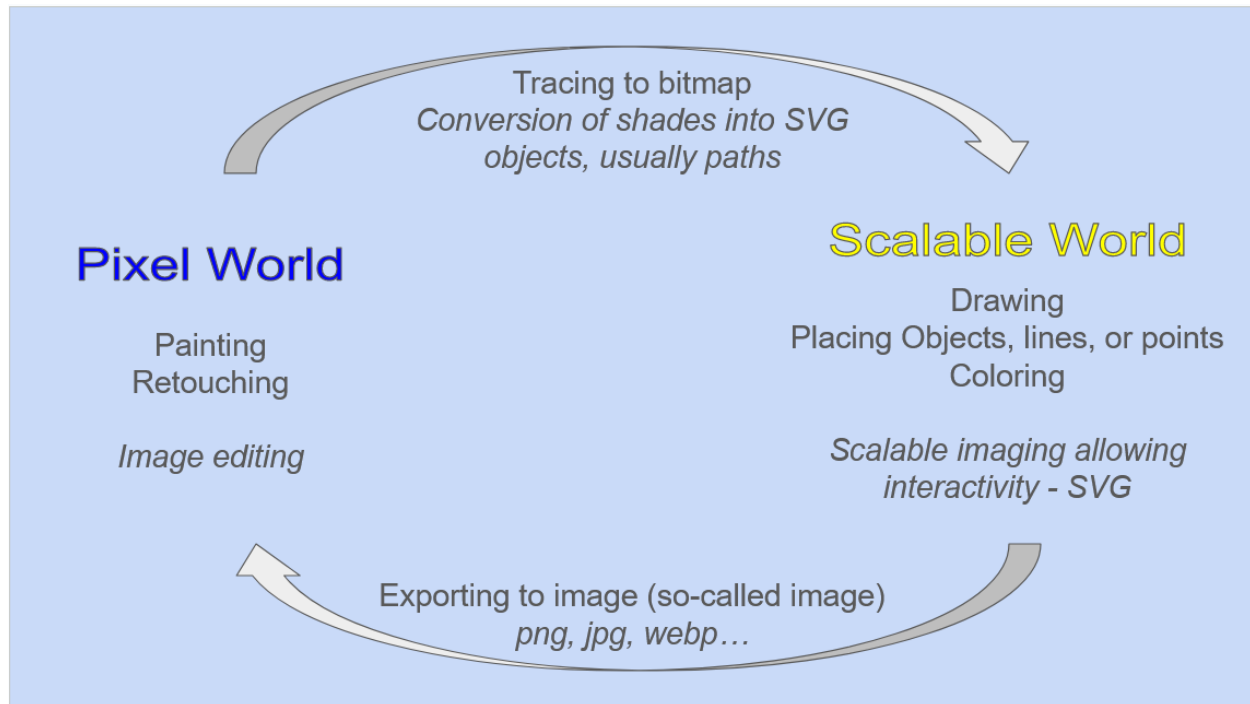


Example of an inkscape file: [link](#)

Flow Chart

Flow Chart showing connections from the Pixel world to the Scalable World and vice-versa.

Below, schematic of interrelation between PIXEL and SVG worlds.



[SVG, vector image format](#), [vector image graphics](#), [.dwg](#), [postscript](#), [Raster graphics](#), [Xfig](#), [Corel Draw](#), [Inkscape](#), [Illustrator](#), [WebP](#), [pixel](#), [image tracing](#), [trace bitmap](#), [Potrace](#), [Vector Graphics Editors](#), [Vectorization](#), [Image Convertors](#), [Image Convertor Potrace](#), [raster to vector](#)

Additional

A contributor has made videos about SVG (SVG is also a format to *Adobe Illustrator*, not only *Inkscape*):

Jennifer Maker about SVG files

<https://www.youtube.com/watch?v=fpHj8vVOTHE>

<https://www.youtube.com/watch?v=e9SMkaIRzLo>

<https://www.youtube.com/watch?v=nUbWr1BdEZ0>

Playing with Trace Bitmap (reverting raster images to Vector Graphics) with Inkscape

[The Complete Guide To Using Trace Bitmap in Inkscape](#)

[William Newman \(computer scientist\)](#)

Advocate of PIXEL

