Palette at use in Image Editors Softwares (Example of the *GIMP*)

Palettes

In the blog about color mixing, dated 2-15-2024 (link), the term palette was used to introduce how to handle a set of basic colors, like primary colors Red, Yellow and Blue.
The software program *MS Paint* was used and the colors of the palette applied through the use of the natural pencil tool.

A palette is a set of colors used on an artwork (by extension, used on any picture). Designers and creators use or choose a set from an existing artwork (like the main tints or shades), based on their own observation, or create a set from scratch to express something (fine art approach), or use an algorithm to generate a logical backend that an app or software can keep in memory for further user use.

- In a previous blog, I gave the reference of digitalsynopsis intending to discuss about how color palettes influence moods.

- A palette is also called a swatch, a term that can be used by software and apps programmers and digital painters. I have seen the term swatch on another blog, pixelblog, maintained by a game artist designer; here is the link: link
  The blogger (Slynyrd, @rayslynyrd) shares about his own experience how to make a palette. His approach is a little bit different than the fine art approach.

- Color palettes generators are proposed online:
  - https://icolorpalette.com/color-palette-generator
  - https://colors.muz.li/

- The fine art approach is drawn by the message and the impression that are wanted to be given. From a very starting point of scratch, the primary set of basic Red, Yellow and Blue can be used. And from this point, the mood is derived, or attained by gradual mix. This approach is directly inspired by how we make colors in reality. Manufacturers make colors by using pigments.

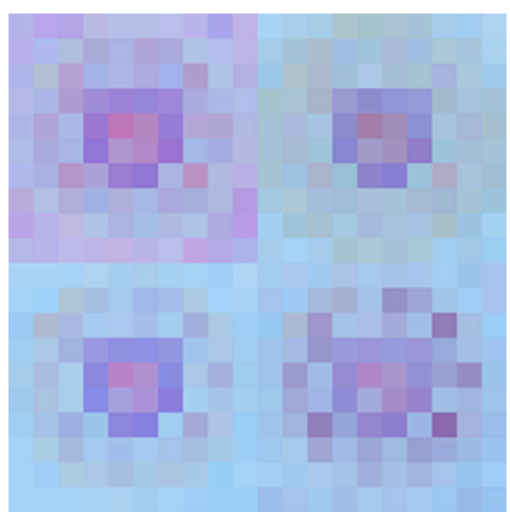
International Academy Projects gives some guidelines (link) how to make Red, Yellow and Blue pigments:

- Red pigment (Sienna, Umber, Cadmium red, Chrome red, Cobalt arsenate, Cobalt phosphate, Iron red, Red lakes and Madder, Quinacridone violet, Red lead and Vermilion):
  "*The pigment is selected from mined material, ground and then levigated and dried. The burnt form is prepared by heating the natural material";*
- Yellow pigment (Aureolin, Barium chromate, Cadmium yellow, Chrome yellow, Indian yellow, Iron yellows, Lead tin yellow, Naples yellow, Orpiment, Realgar, Turner's Yellow and Yellow lakes):
  "yellows are prepared by precipitation from an acid solution of a suitable salt";
- Blue pigment (Aerinite, Ultramarine, Azurite, Blue verditer, Cerulean blue, Cobalt blue, Egyptian blue, Indigo, Manganese blue, Maya blue, Phthalocyanine blue, Prussian blue, Smalt, Verdigris and Vivianite):
  Particles are found in copper carbonates materials for instance; *"Azurite is an important copper ore which occurs in many parts of the world."*

Artists can make their own color pigments. In addition to Red, Yellow and Blue, Green pigments, Black and Brown, pleochroic (Yellow to lavender to red pigments preparation), purple and violet and white pigments can be prepared.

I am not an artist, but below, I give an example on how palettes were used on mandala coloring by a mandalayaner amateur (learn more about digital mandalayaning).

- On the left snip, saturated R, Y, B blobs, and several blue shades, and adding warmness or light or saturation (mixing with yellow or red) has resulted in the other blobs shown. The color editor is *GIMP* (unlike, blog dated 2-15-2024, using *MS Paint*). *GIMP* is a raster image editor and photo manipulation software, of the family of softwares with functionalities like *Photoshop*.
  GIMP has tools like brushes and pencils, like *MS Paint*, and a smudge tool that was used to mix the colors (like below, left snip).
- On the right, a tetraptych mandala after the picture shown in section Moods was pixelized (each quarter has 10 x 10 pixels).



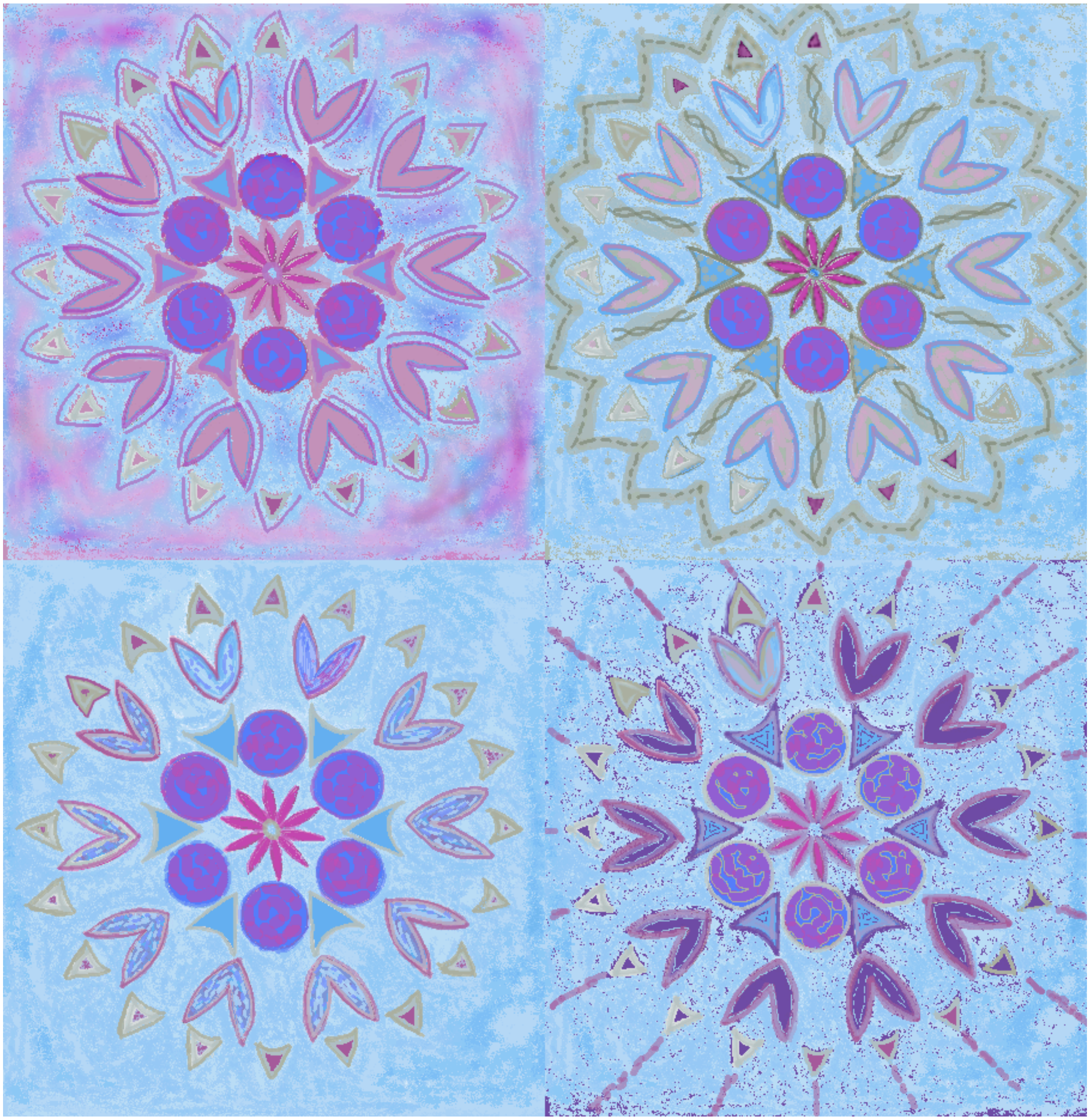
Moods

I used a palette and made four versions around a basis mandala. The design is nothing more than the mandala issued online and discussed in a previous blog (2024/02/2-9-24-website-updated-with-patterns).

I applied a digital coloring using the same color palette and ended up in the four versions tiled below.

What can we think? The intent was to get a different impression on each version. I made a difference of impression, but not easy to characterize as a single mood like happy or sad. I found a way to express these images with words (as it is not supposed to be) and used the parts of the day, as sets of impressions (for instance the morning is moment of freshness of the mind, not yet full of things, clear light -thinking to a summer or spring morning-).

So, for me the related moods for these versions (top left, clockwise) are evening (because it is deeper), noon (because it is still bright and heavier), night (because the overall impression is darker), morning (because it is fresh and light).



Additional Color Picking Methods

In a previous blog, I have explained how to perform color averaging (color mixing) by using an image scaling to one pixel (one by one), and how to pick up for checking a color attribute on a single pixel. In this paragraph, I present other methods than the basic eyedropping to get color attribute on an image.

Some additional color pick up methods are proposed in *GIMP*. Future blogs will talk about other functions. *artgraphic-db* has aimed to present material on the website, resulting from the application of image treatment routines present in the type of *GIMP* softwares (tools, filters…).

Color effects and manipulation in raster images editing softwares

Compared to *MS Paint*, *GIMP* has more tools for painting, like pencils and brushes. These tools are used locally, for applying changes on a few pixels and small regions of an image. In *GIMP*, image manipulation routines can be used to retouch images globally (on the entire picture).

As an example, *GIMP version 2.10* has the following functionalities to deal with image colors:

1. *Adjusting color distribution, to correct predominant colors in digital photos*
   ➩ [Color Balance tool];
2. *Changing the color temperature of an image, depending of any source lighting parameterized as a temperature in Kelvin*
   ➩ [Color Temperature tool activates the related filter];
3. *Adjusting LCH Hue, Chroma and Lightness to use Hue-Chroma controls to modify saturation*
   ➩ [Hue-Chroma tool activates the related filter];
4. *Adjusting HSL Hue, Saturation and Lightness on a range of color weights*
   ➩ [Hue-Saturation tool activates the related filter];
5. *Focusing on saturation adjustment*
   ➩ [Saturation tool];
6. *Focusing on shadows and midtones adjustment*
   ➩ [Exposure tool activates the related filter];
7. *Correcting shadows and highlights separately*
   ➩ [Shadows-Highlights tool activates the related filter];
8. *Adjusting Brightness and Contrast easily, controlling bright colors differently from darker colors*
   ➩ [Brightness-Contrast tool];
9. *Adjusting color levels, using color distribution math characteristics; can make an image lighter or darker, changing contrast or correcting a predominant color cast*
   ➩ [Levels tool]
10. *Adjusting colors represented by curves on a chart; sophisticated tool for changing the color, brightness, contrast or transparency*
    ➩ [Curves tool];
    ➩ [Levels tool ⇒ Shadows and Highlights, Curves tool ⇒ tonal range]
11. *Making negatives, dark becoming bright, hue being set to complementary*
    ➩ [Invert tool];
    ➩ [Linear Invert tool];
12. *Adjusting brightness by inverting brightness value of an image and keeping Hue and Saturation*
    ➩ [Value Invert];

13. *Adjusting automatically colors*
    ➪ [Auto tool -Equalize, white balance, stretch contrast [and HSV], color enhance
    ➪ [with legacy-];
14. *Focusing on color components*
    ➪ [Components tool -Channel mixer, extract component; mono mixer; compose; decompose; recompose-];
15. *Converting to Grayscale*
    ➪ [Desaturate tool -Color to Gray, Desaturate, Mono mixer, Sepia-]
16. *Rearranging colors, less or more colors, grouping*
    ➪ [Map tool -rearrange colormap, set colormap, alien map, color exchange, rotate colors, gradient map, palette map, sample colorize-]
    ➪ [Tone mapping -Fattal et al. 2002, Mantiuk 2006, Reinhard 2005, Stress, Retinex-];
17. *Getting information and characterizing images with math parameters*
    ➪ [Info tool -Histogram, Border Average, colorcube analysis, smooth palette];
18. *Coloring to two colors* ➪ [Threshold tool]
19. *Grayscaling and monochromatic coloring* ➪ [Colorize tool]
20. *Reducing intelligently the number of colors* ➪ [Posterize tool]
21. *Converting colors to transparency* ➪ [Color to Alpha tool]
22. *Reducing color levels by channel* ➪ [Dither tool]
23. *Constraining ranges of colors* ➪ [RGB clip tool]
24. *Repairing like -find and fix pixels that my be unsafely bright for instance-* ➪ [Hot tool]

Transparency (alpha) because of computer
Channels RGB and alpha or HSL and alpha

https://ninedegreesbelow.com/photography/changing-saturation-using-lch-chroma.html
https://docs.darktable.org/usermanual/development/en/module-reference/processing-modules/shadows-and-highlights/

Color checking (color picking)

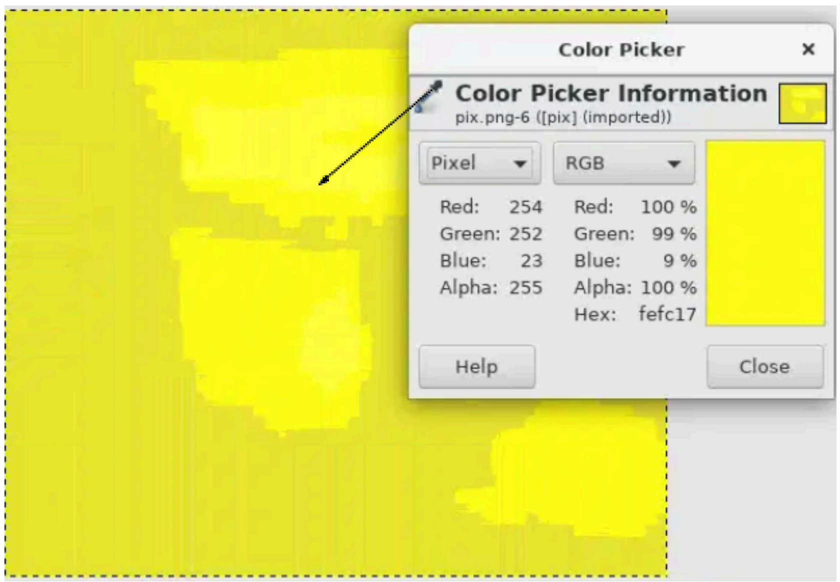
As seen in the previous paragraph, there is a lot of ways of changing colors or coloring. Sophisticated apps need more ways to get color information and they have diverse ways to pick up colors.
Gimp has the following additional features to pick up colors on images:

- Improved Usual Color picker;
- Sample Points Color Picking;
- Pointer Tool;
- Sample average Color Picking;

Just to remind that a basic eyedropper is usually used to set a color and eventually to copy the selection on a palette for further use. The palette can be accessed independently. As a result, several steps are sometimes needed to have the color of a pixel and the actual state of diverse settings (like the palette, a set of predetermined colors used on an artwork) at the same time.

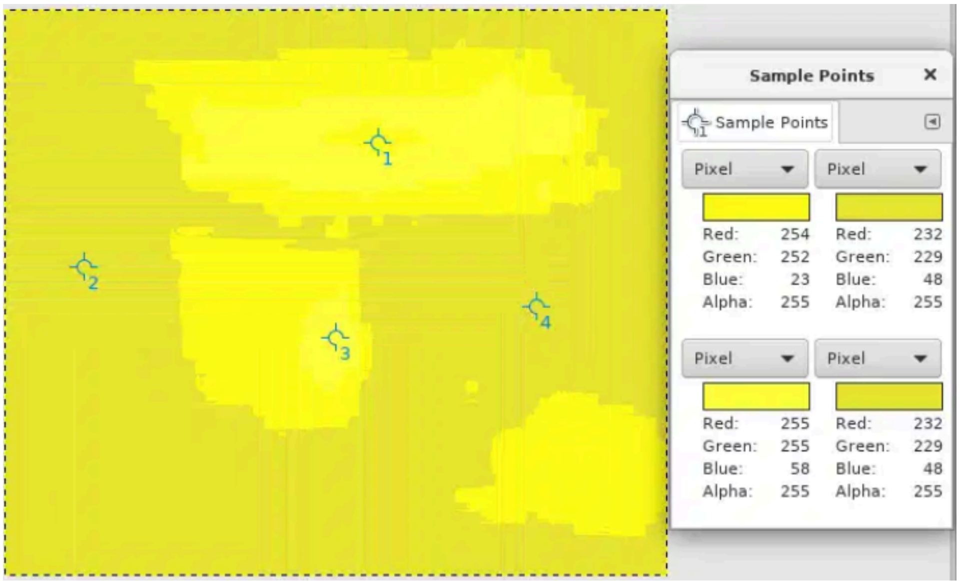
Improved Usual Color picker

When shift-clicking an eyedropper (preselection in the Tool Selection Dialog Box **Windows>Toolbox**) on an image, a pop-up is open to show the actual state of the “color picker” method, showing the value of the color attribute where the mouse was clicked.
As shown below, the pop up shows the color attribute in terms of coordinates in the selected color spaces (below RGB on a 0-255 scale and a 0%-100% scale, and hexadecimal color name here: #fefc17).



Sample Points Color Picking

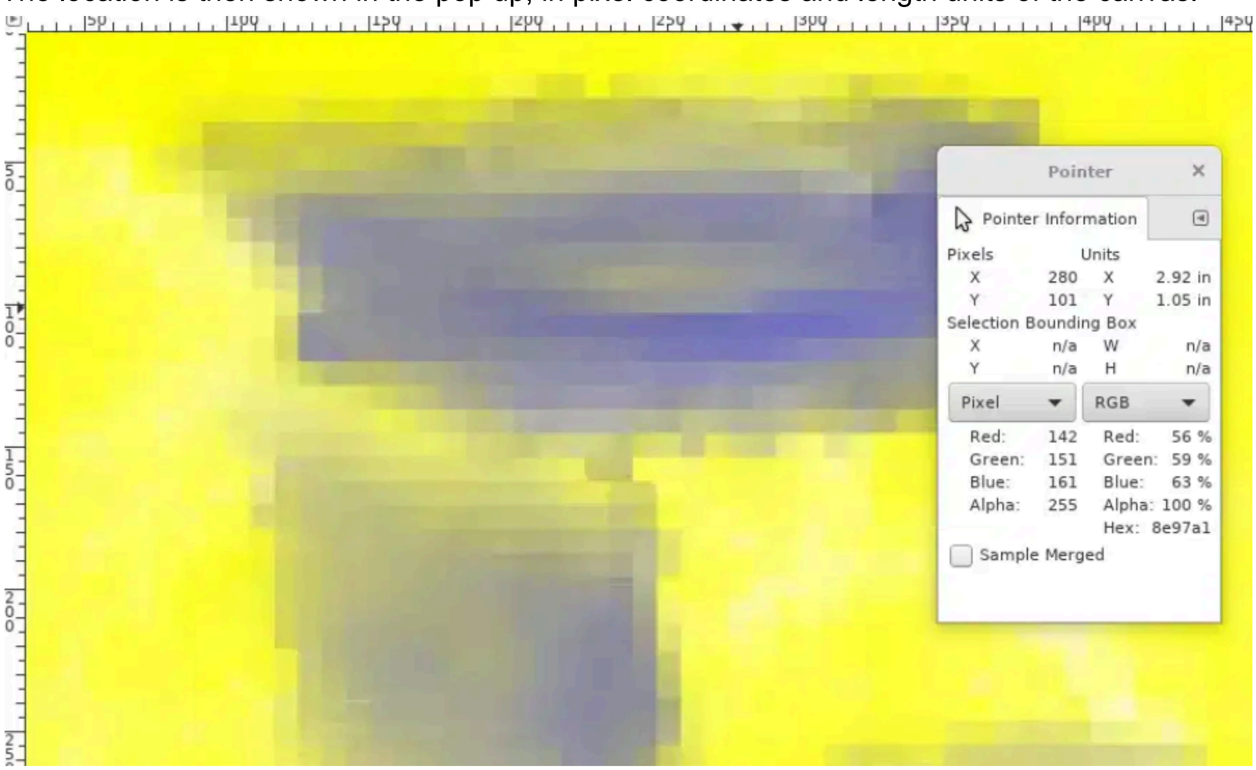
This method is accessible through the menu **Windows>Dockable Dialogs>Sample Points**.
As explained in the *GIMP* documentation, I had to click and drag to set the four sample points. For each point, I have Ctrl-clicked on the ruler and, keeping click, I have dragged the mouse onto the desired position. The snip shows the four points and the color attributes.



Pointer Tool

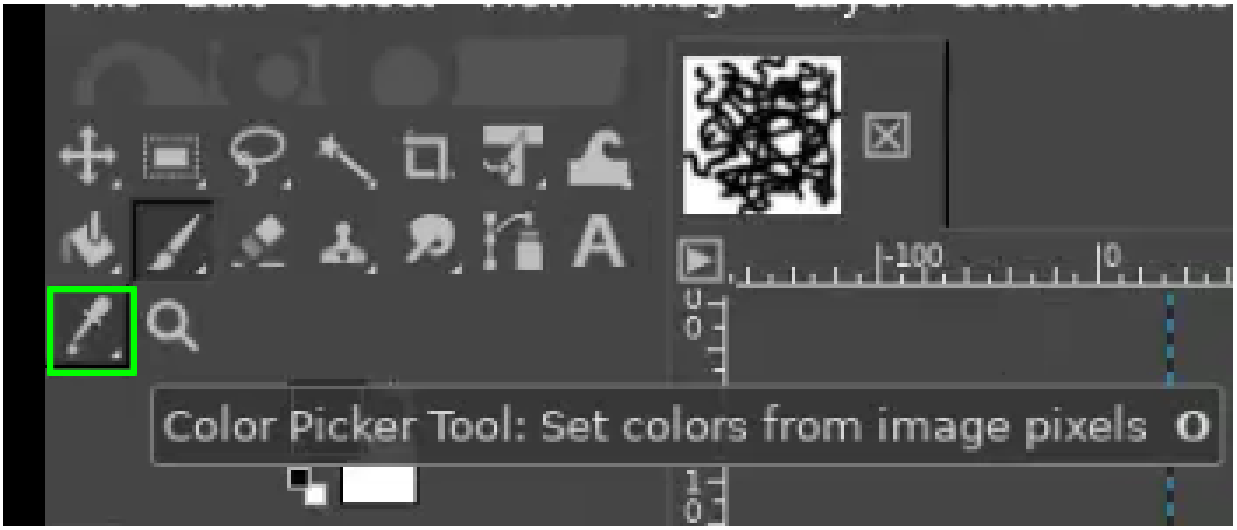
I wanted to have information to be sure about the location of the eyedropper where I got the color attribute. The pointer is accessible through the menu **Windows>Dockable Dialogs>Pointer**

The location is then shown in the pop up, in pixel coordinates and length units of the canvas.



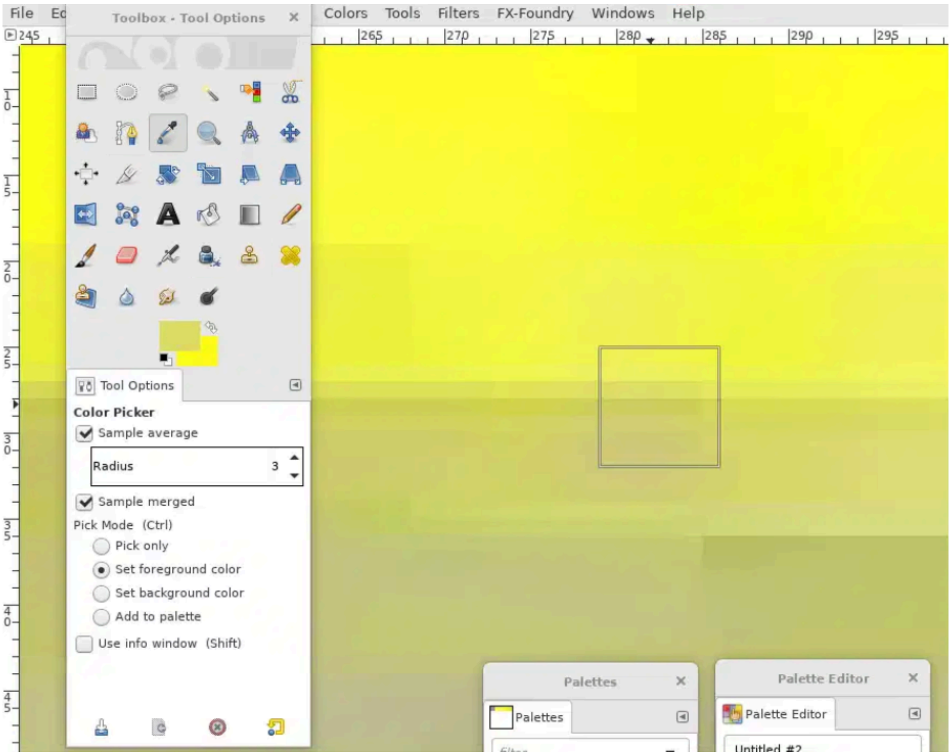
Sample average Color Picking

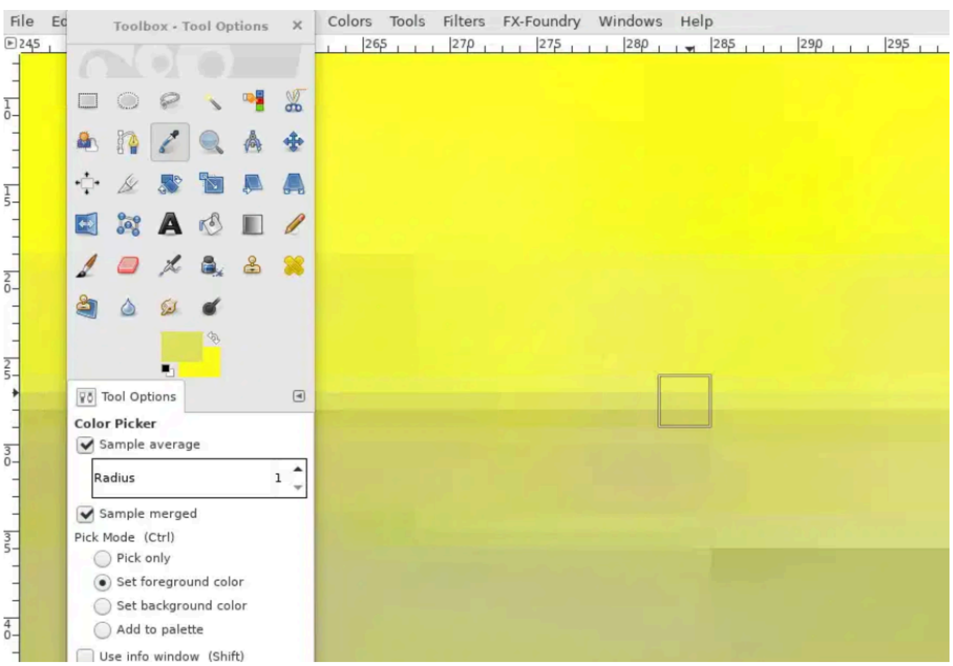
The sample average is an option of the usual eyedropper (accessible in the Toolbox).



Above, widget of the eyedropper in the toolbox

After selecting the eyedropper, the menu opens and shows the options. Then, the sample average can be toggled. First, I choose the size of the sample to be 3 px by 3 px (parameter "radius"). Second, my choice was to set back to one by one pixel (like the standard eyedropper finally !).



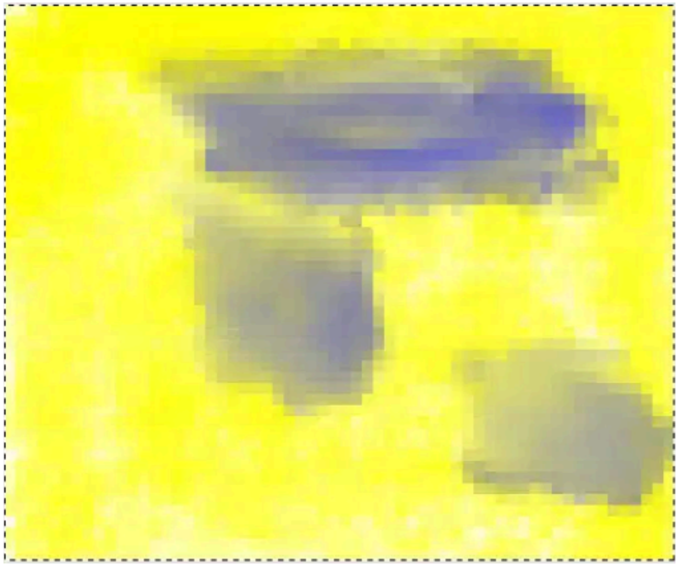


How to get Information of an Image

This section focuses on **InfoTool** (item #17) of the *GIMP (see the previous paragraph Color effects and manipulation in raster images editing softwares)*. This is about a type of information that is not perceptual and needs a software to compute characteristics of the digital image.

In *GIMP* this type of tool is accessed through **Colors>Info Tool**

Considering a given image as input, to be analyzed, like this picture:



I wanted to produce the following outputs: size of the image, number of colors, and information about how RGB quantities have been used.

Number of Unique colors

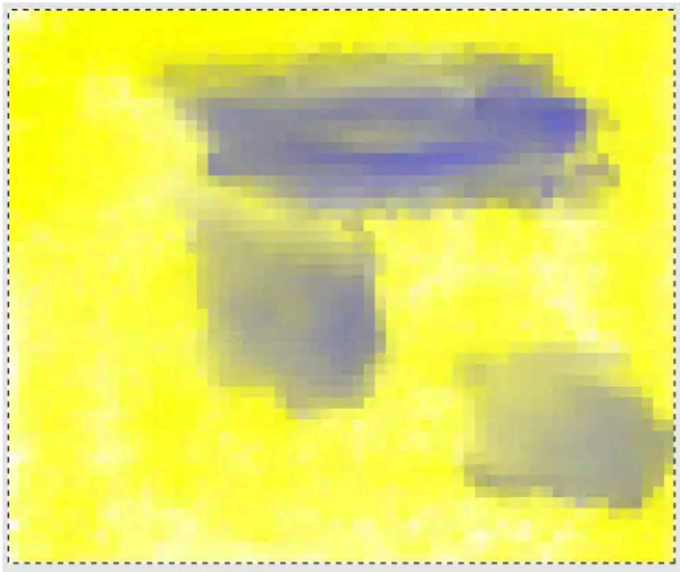
GIMP makes a difference, or better said, talks about unique colors instead of, simply, colors.
I used the tool Color Cube Analysis, (through **Colors>Info Tool>Color cube analysis)**
It gives Image dimensions and number of unique colors.

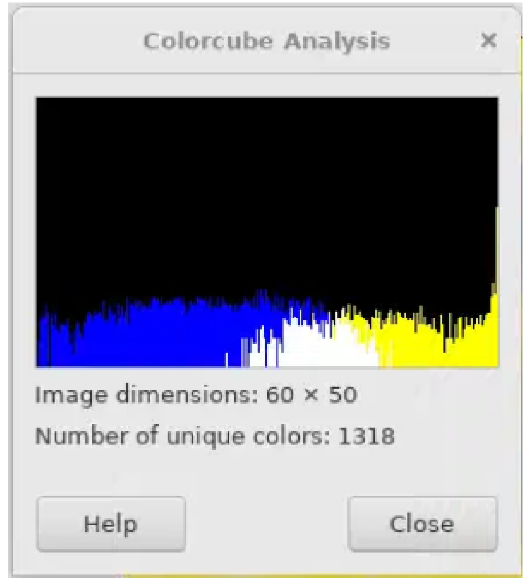
Mean, Standard deviation and median

The tool Histogram, accessible through **Colors>Info Tool>Histogram** allows to compute Mean, Standard deviation and median, which are statistical characteristics, of the sample made of the set of the pixels of the image.

Below, illustrations show the pop up outputs for both tools (color cube analysis and Histogram)

Color cube analysis tool: Output/Info 1





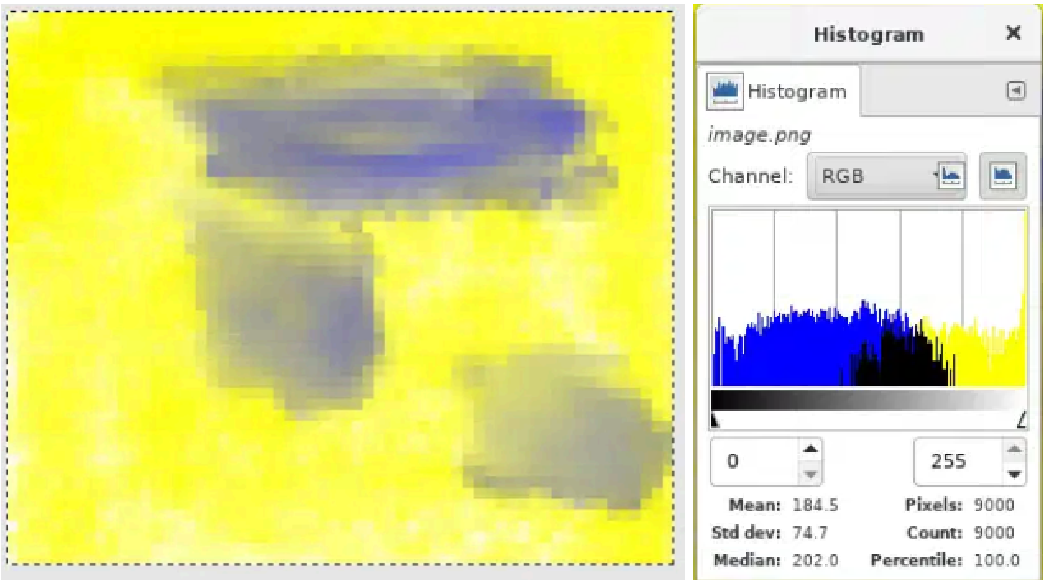
Above, screenshots of the picture analyzed on the left, and right,the pop up message given by the colocube analysis tool:

- Image dimensions are 60 by 50 pixels;
- Number of Unique colors is 1318;

A graph shows an histogram (also shown by the histogram tool). Height of the columns are related to the parameters used when rearranging a smooth palette based on the colors of the picture). My best guess is that map and Info tools call the same routine, and need to compute an histogram on the image sample.

Histogram Tool: Output/Info 2

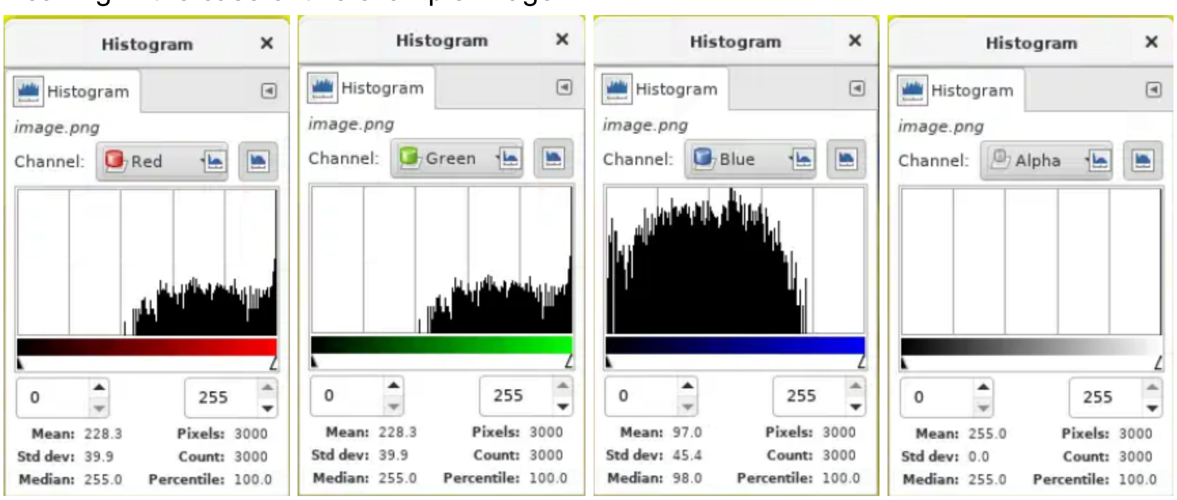
Analyzing the same image, I use the Histogram Tool to compute statistical information about the sample of pixels in the image.
Just to remind that this image has 60 x 50 pixels (=3000 pixels).



Above, screenshots of the picture analyzed on the left, and right,the pop up message given by the histogram tool:

- The sample is 9000 (60 by 50 pixels for the image, in three layers RGB 3x3000);
- The three colors Red and Green and Blue are scaled between 0 and 255 and can be considered as a single sample, mean is 184.5, standard deviation is 74.7 and median (as many pixels less than R or G or B of value 202, as above 202)

The pop up has options and histograms for each layer can be plotted (the sample is now 3000), as shown below. Because I made the picture with yellow and blue, red and green layers have the same properties.
The GIMP uses also the term channel for layer; the alpha channel for transparency has no meaning in the case of this example image.



Above, from left to right, statistical findings for each channel Red, Green, Blue, Alpha (transparency): Mean, Standard Deviation and Median
The pop ups read like the previous for the whole set.

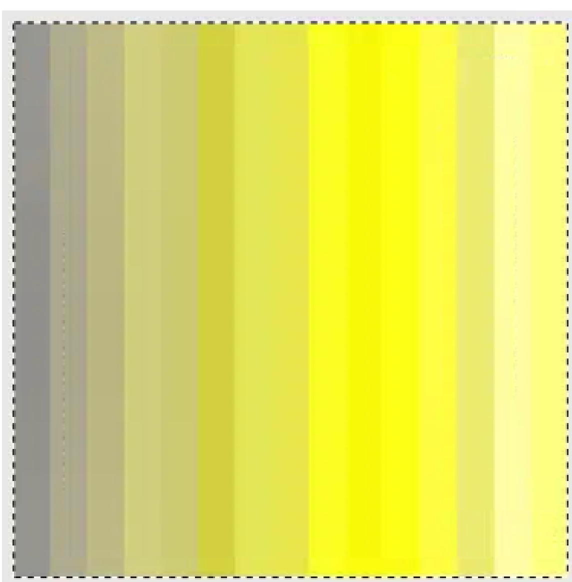
Statisticians use deciles, percentiles, quartiles, quintiles to talk about the way to split a sample represented by an histogram. The percentiles are type of quantiles helping to split a population in 100 ranks.
Above, the area between left and right cursors covers the overall range of color values, resulting in considering all the 100 ranks, including the 100th (percentile 100).
This is not far to say that 100% of the pixels are considered to compute the mean and standard deviation shown.

Smooth Palette Tool (reducing the palette)

This section introduces a tool present in advanced image editing software. As an example, I use the *GIMP*, in which a tool to analyze the arrangement of colors of an image is proposed. In *GIMP*, the tool is named “Smooth Palette”
The outline of this paragraph has been to:-

1. Show the smooth palette menu and the input parameters as well as how a colorcube analysis outputs its information (I really use the colorcube analysis on the smooth palette picture, not the original picture);
2. Show a couple of examples how the use of a smooth palette rearranges the colors of an image;

Smooth Palette: Input; output

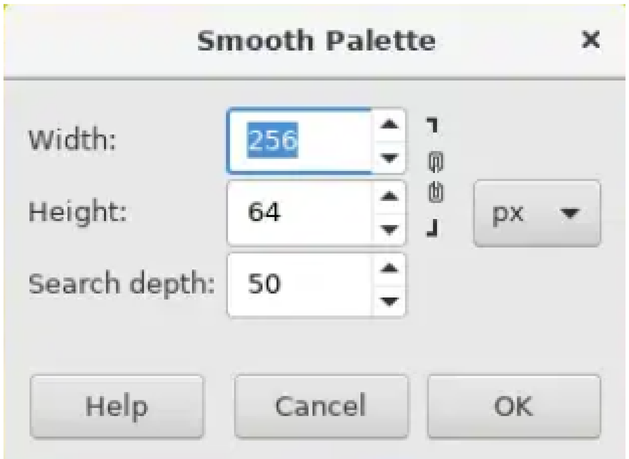


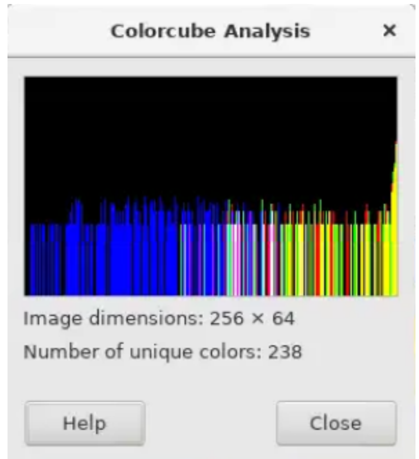
A typical image output snip of the **Smooth palette tool**, generated with parameters width=15; height=15, search depth=1;

The original output of 15 px by 15 px is further scaled to a larger size.

Picking up colors of the bands serves to put them in a user palette.

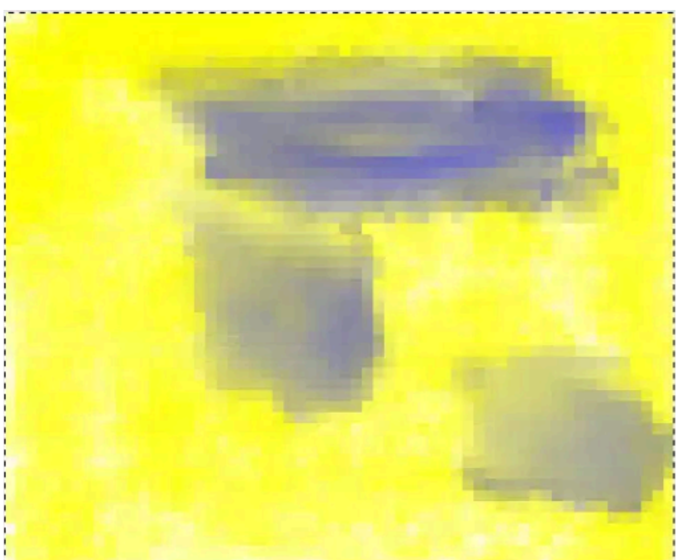
The Smooth Palette tool produces an image output, showing bands of uniform colors. These bands have a width and a height. The number of bands depends on the number of colors requirement. How the smooth palette meets a number of colors requirement is clearer when the search depth is set to a value of one (meaning, one color per color band). The height can be seen as a weight factor to produce another output image size, and consequently, to give another aspect to the output picture.





Above, left, smooth palette tool inputs: the setting of the height allows to give another look to the height of the columns of the histogram.

My first smooth palette use



Original picture used for the smooth palette use.

Same image shown in How to get Information of an Image paragraph

I show two examples of rearranged colors for the original picture shown above. In *GIMP*, I followed the following step-by-step:

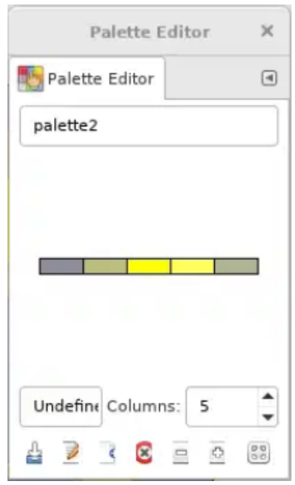
- First step: deriving a smooth palette from a set of given colors (the colors in the image); **Colors>Info>Smooth Palette**
- Second step: creating the palette and making it current;

Windows>Palettes>; *create the palette; add colors eyedropping the colors on the output image from the previous step*

- Third step: the new palette, being selected, and said “current”, is used in background of the mapping (rearrangement of the colors)
  Colors>Map>Smooth Palette

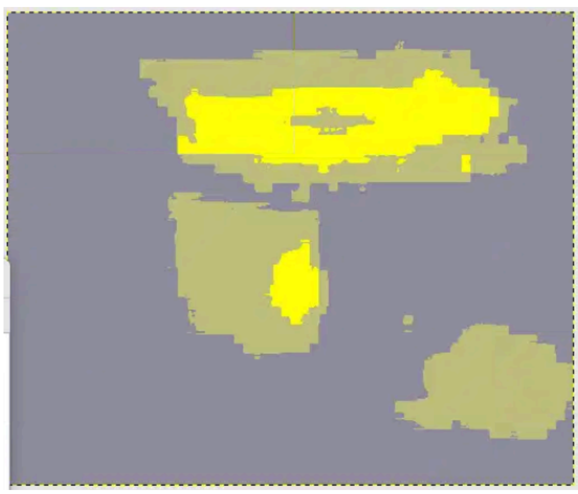
- 5 colors palette requirement (the rearranged picture shown on the left, based on the palette shown on the right)

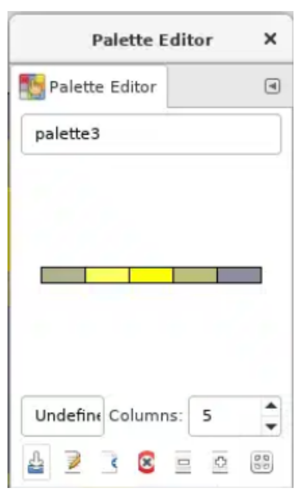
5 colors -> 3 regions (number of unique colors is three)



- Inverted version: 5 colors palette requirement (the rearranged picture shown on the left, based on the palette (inverted order) shown on the right)

5 colors -> 3 regions (number of unique colors is three)





As explained after in Why the number of unique colors does not meet necessarily the number of desired colors requirement? paragraph and in mapping on smooth palettes>5 levels: palette 1 paragraph, I give some clues about why.

Why the number of unique colors does not meet necessarily the number of desired colors requirement?

The previous paragraph raises the point that an optimum of color arrangement exists in a picture. This optimum of rearrangement depends on the two features:-

- The number of preexisting colors present in the picture and the related issue of having a maximum number of colors in the palette (256 possible rearranged colors as shown above); Thus, the test case imagined in this section to have a reduced number of preexisting colors in the original picture; As a result, I found out that even if the number of unique colors is small (in the original picture), the input requirement about the desired number of rearranged colors is not necessarily met;
- The way of how the colors of the smooth palette are organized in the color space that they are supposed to cover; I focused on monochromatic situations to illustrate that simple distributions in Brightness can lead to a mismatch;

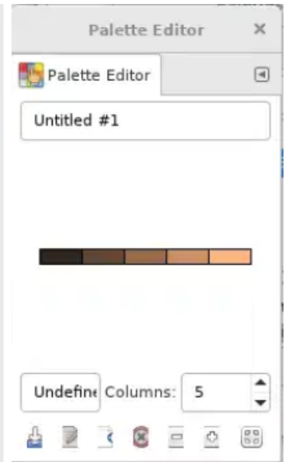
Below, I show the three cases, and how simple considerations on the brightness distribution explain the mismatch between the number of desired colors and the actual result.
In the three cases, a monochromatic palette of five shades has an equidistribution in brightness according to the following values 100, 80, 60, 40, 20 (Values in the HSV color space). The purpose is really to arrange the colors of the original picture on this new palette, say, I really want the five desired colors on the resulting picture.

Note that the 5 values above allow to rearrange a preset of given colors in the 4 following intervals [100, 80], [80, 60], [60, 40], [40, 20].

- First case: the colors of the original picture (having values 80, 60, 40, 20, 0) do not match correctly the desired palette because the lightest shade has only a value of 80. There is no shade in the interval [80, 100] in the original picture of this case; the resulting picture shows four shades, not five shades anymore;
- Second case: the colors of the original picture (having now values 100, 80, 60, 40, 20) do match correctly. All the shades can find a distinct interval; the resulting picture shows five shades;
- Third case: an alternate case of case 2, to check that the “smooth palette” distribution routine is mainly based on brightness (value) consideration; However, I found out that another variable is probably used in the non-monochromatic case in order to find the correct intervals for the resulting color distribution;

A simplified test case 1: screwed match

The colors of the original picture -left- (having values 80, 60, 40, 20, 0) do not match correctly the desired palette because the lightest shade has only a value of 80. There is no shade in the interval [80, 100] in the original picture of this case; the resulting picture -right- shows four shades, not five shades anymore.
The last snip is a snip of the palette editor showing the desired palette.

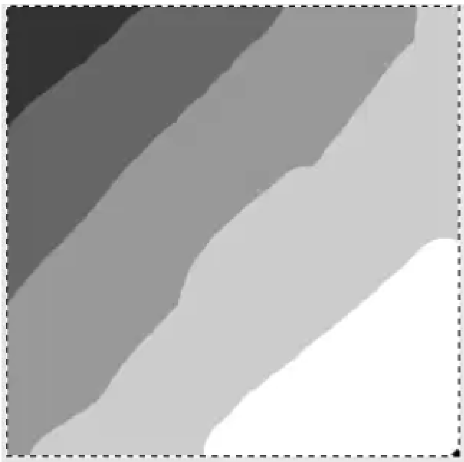


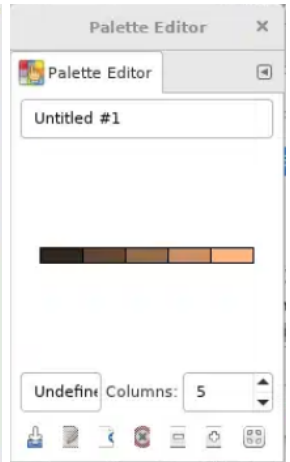
Schematic distribution of values:

Original picture:	80, 60, 40, 20, 0
Palette:	100, 80, 60, 40, 20
Resulting picture:	80, 60, 40, 20

A simplified test case 2: good match

The colors of the original picture -left, below- (having now values 100, 80, 60, 40, 20) do match correctly. All the shades can find a distinct interval; the resulting picture -right- shows five shades;
This new original picture is complementary of the previous case (*GIMP* tool **Color>Invert** or **Color>Value Invert** -because the picture is monochromatic-).
The last snip is a snip of the palette editor showing the desired palette.



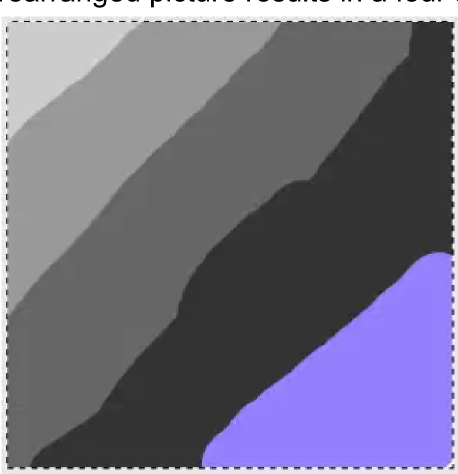


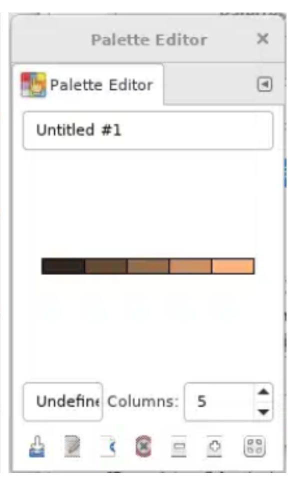
Schematic distribution of values:

Original picture:	100, 80, 60, 40, 20
Palette:	100, 80, 60, 40, 20
Resulting picture:	100, 80, 60, 40, 20

A simplified test case 3: good match?

An alternate case of case 2, to check that the “smooth palette” distribution routine is mainly based on brightness (value) consideration; However, I found out that another variable is probably used in the non-monochromatic case in order to find the correct intervals for the resulting color distribution.
The new color on the original picture has a value of 100 like the good match case. However, the rearranged picture results in a four shades version and not in the five shades version.





80, 60, 40, 20, 0 ⇒ HSB=(250,50,100), 80, 60, 40, 20

Schematic distribution of values:

Original picture (shades): 5 shades	100, 80, 60, 40, 20
Palette: 5 shades	100, 80, 60, 40, 20
Resulting picture: 4 shades	80, 60, 40, 20

I give additional insight in another paragraph: Complete examples of mapping on smooth palettes>5 levels: palette 1.

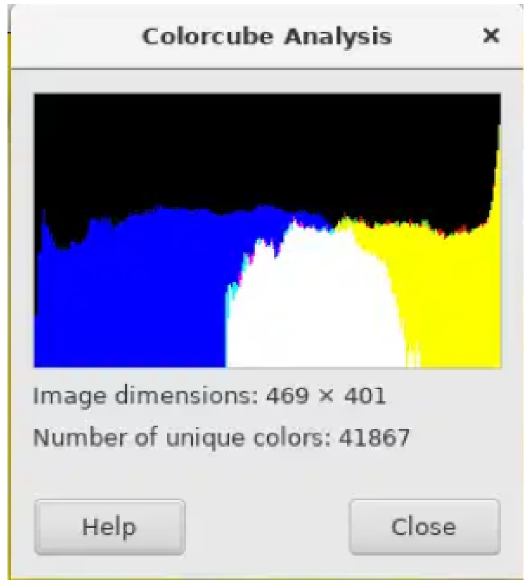
Complete examples of mapping on smooth palettes

In *GIMP,* I have followed a step-by-step mapping on a palette that was created as a user palette, in three steps:

- First step: deriving a smooth palette from a set of given colors (the colors in the original image);
  Colors>Info>Smooth Palette
- Second step: creating the palette and making it current;
  Windows>Palettes>; *create the palette; add colors eyedropping the colors on the output image from the previous step*
- Third step: the new palette, being selected, and said “current”, is used in background of the mapping (rearrangement of the colors)
  Colors>Map>Smooth Palette

Original Image

Colorcube Analysis Tool
In *GIMP*: **Colors> Info> Colorcube Analysis**
Image Dimensions: 469 x 401 pixels
Number of Unique colors is 41867



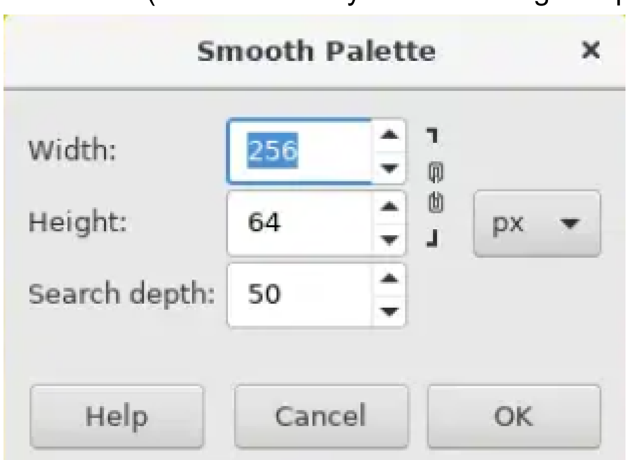
The 41867 unique colors can be reduced to a lower number of colors, in order to work on a specific mood, eventually.

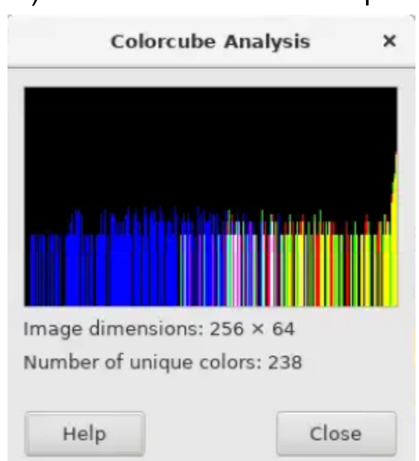
Smooth Palette Tool

The same methodology and framework as the previous paragraphs.

Input; output (256 -> 238)

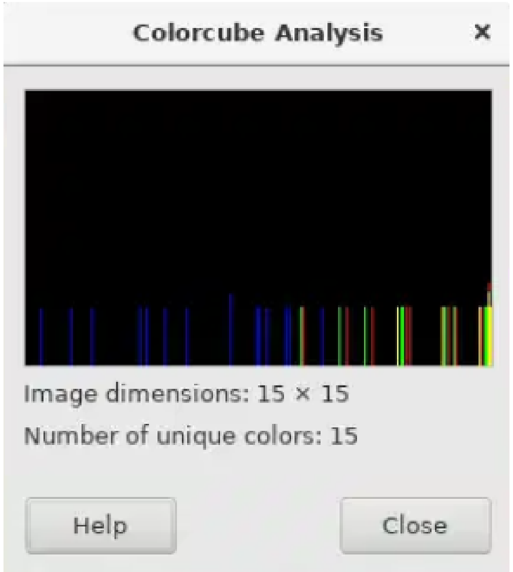
From the set of 41867 colors of the original picture, it is desired to create a palette of 256 colors. As a result (colorcube analysis of the image output) 238 colors define the required palette.

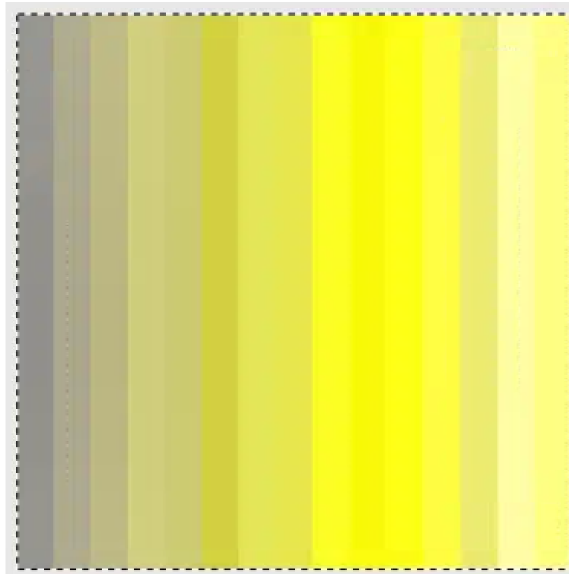




15 colors palette example

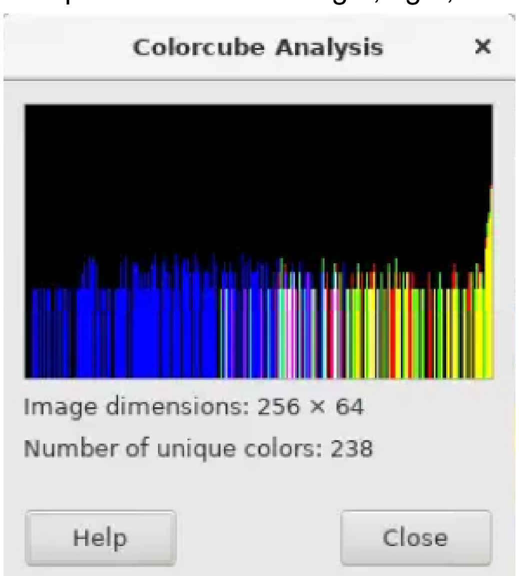
From the set of 41867 colors of the original picture, it is desired to create a palette of 15 colors. As a result (colorcube analysis of the image output) 15 colors define the required palette.

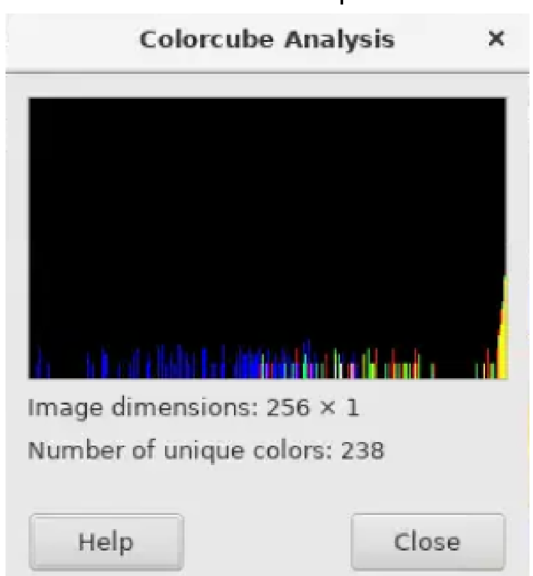




Additional: height of columns in color cube analysis

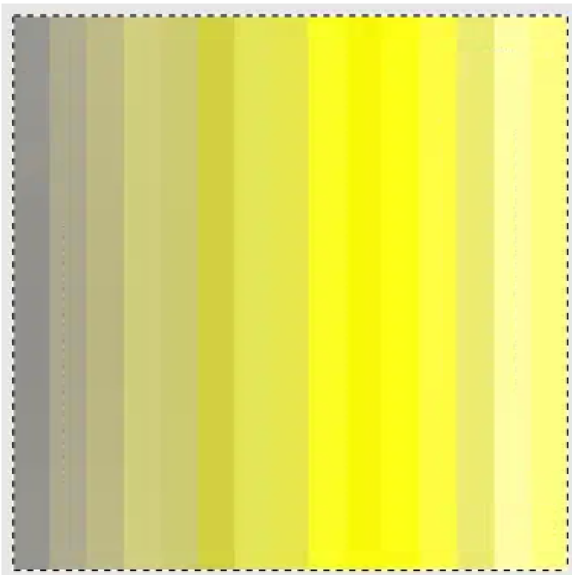
This small paragraph to show the effect of the height parameter: left, the columns are based on a 64 pixels color band height; right, the columns are based on a 1 pixel color band height;



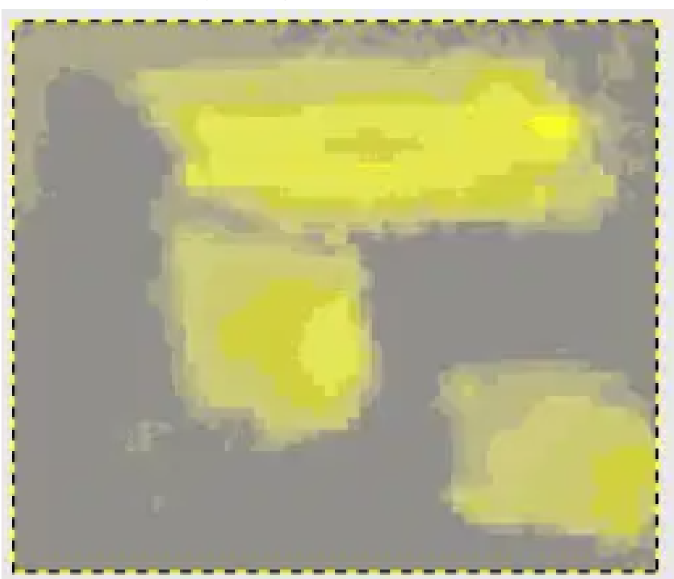


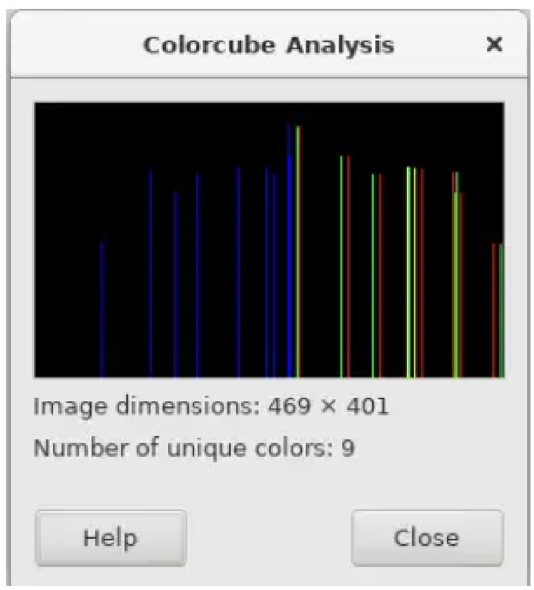
15 levels smooth palette

From the set of 41867 colors of the original picture, the *GIMP* has made a smooth palette of 15 colors.



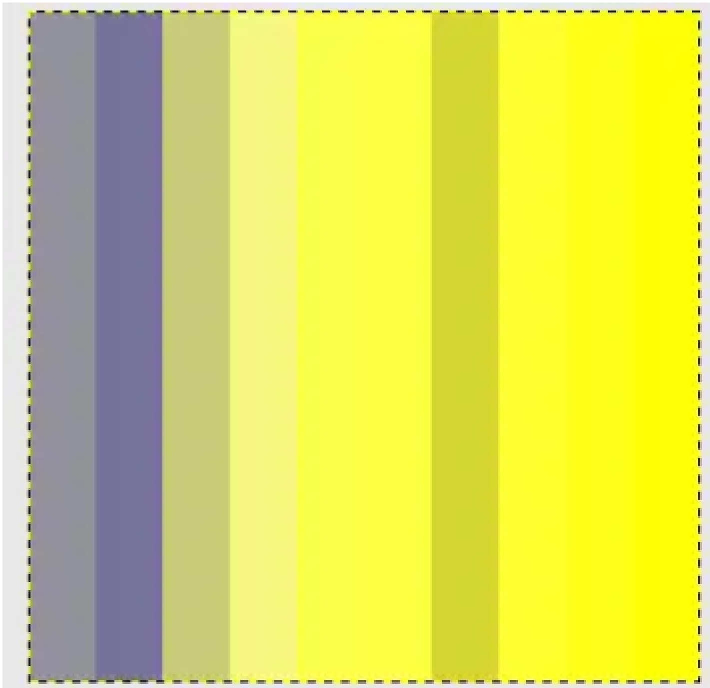
The rearrangement of the 41867 colors of the original colors onto the 15 levels palette has resulted in a 9 unique colors image.
15 colors ➪ 9 regions (number of unique colors is nine)



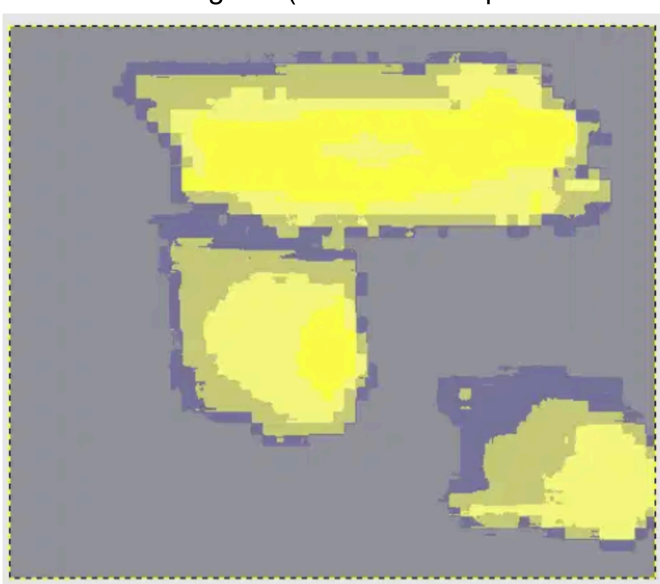


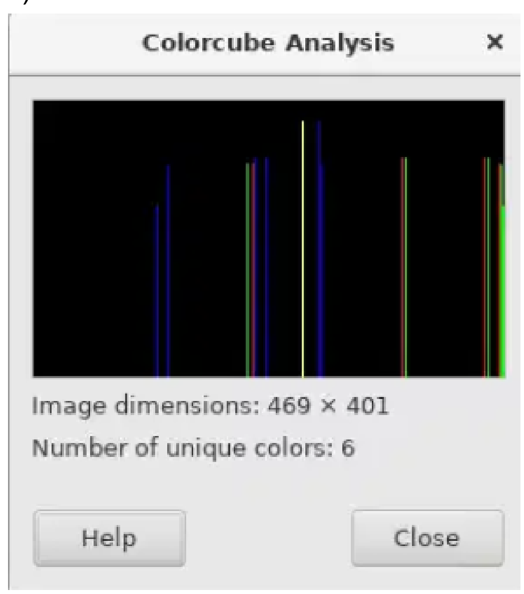
10 levels smooth palette

From the set of 41867 colors of the original picture, the *GIMP* has made a smooth palette of 10 colors.



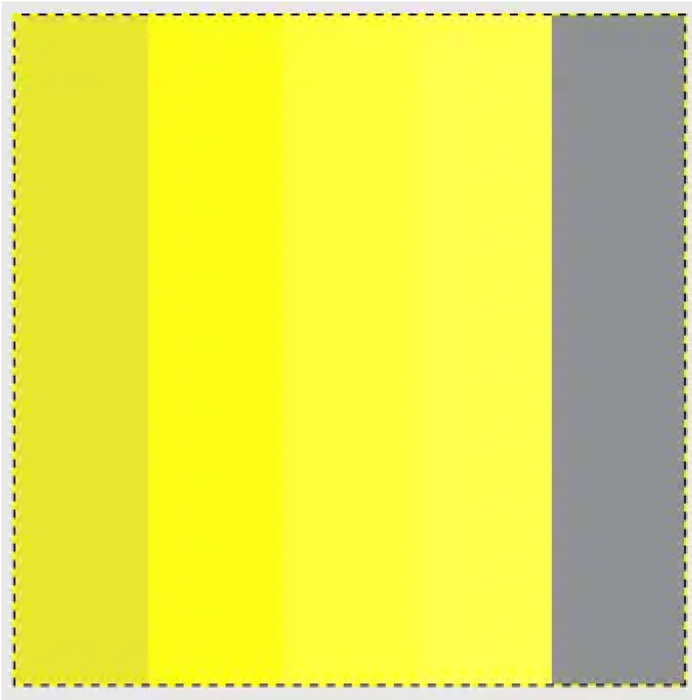
The rearrangement of the 41867 colors of the original colors onto the 10 levels palette has resulted in a 6 unique colors image.
10 colors ➪ 6 regions (number of unique colors is six)



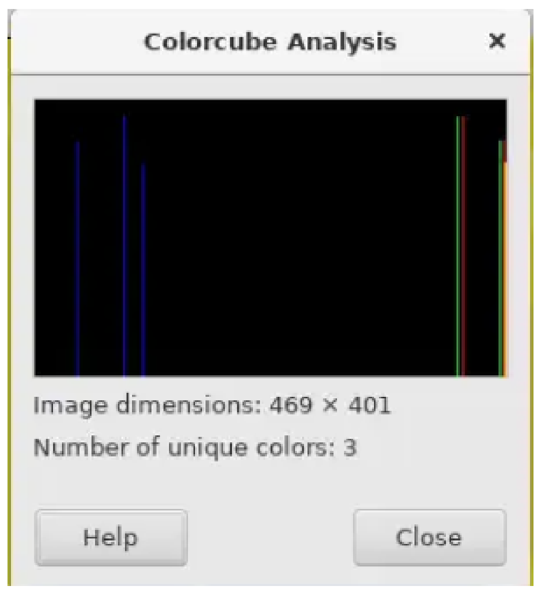


5 levels: palette 1 and Standard Deviations

From the set of 41867 colors of the original picture, the *GIMP* has made a smooth palette of 5 colors.



5 colors ➪ 3 regions (number of unique colors is three)
The rearrangement of the 41867 colors of the original colors onto the 5 levels palette has resulted in a 3 unique colors image.



Based on this example, I give my explanation why or how the routine rearranges three regions (and not six).

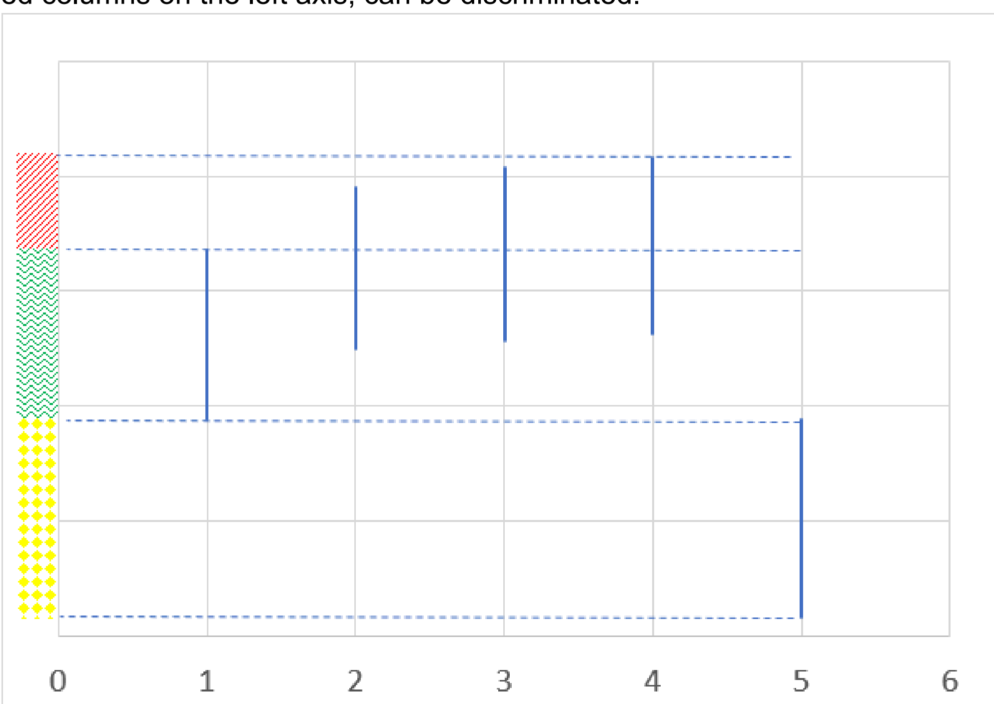
The 5 colors of the palette have the coordinates summarized in the following table. I compute a standard deviation reflecting a color range covered by each color of the subset (the smooth palette). Doing this, I assume that the routine works by using a parameter serving to attribute a color (of the set of 41867 colors of the original picture) to a color of the subset of the 5 colors.

Unlike the simpler example (paragraph Why the number of unique colors does not meet necessarily the number of desired colors requirement?), it is not enough to scale a difference in value (in value only).
So, let us compute a "*coverage*" parameter based on the standard deviations of RG and B components.
Example: SD_{color} = f(sig_R, sig_G, sig_B), a function invented to produce the coverage factor SD_{color}, depending on standard deviations of RG and B components (like the ones shown in paragraph
To avoid some confusion, I do not give the resulting numerical values, but only an illustration (plot below) showing a possible coverage. The main point is that an overlap between colors avoids the dispatch on five colors (having same coverage in fact).

Table of (H,S,L,R,G,B) components of colors 1, 2, …, 5 of the smooth palette. sigR (G or B) is a typical standard deviation of the component R (Red) on this picture (Tool **Colors>Info Tool>Histogram).**

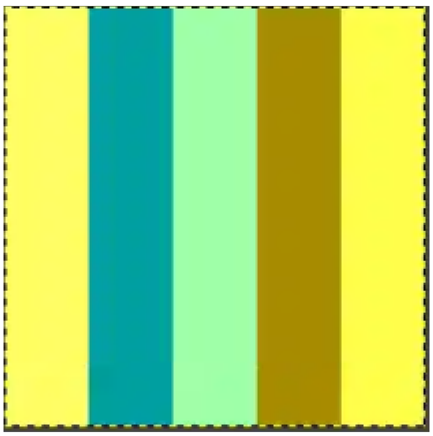
color	H	S	L	R	G	B	sig_R	sig_G	sig_B
1	60	80	55	232	230	48	23.7	23.7	26.7
2	60	99	54	254	253	23	23.7	23.7	26.7
3	60	100	62	255	255	59	23.7	23.7	26.7
4	60	100	65	255	255	78	23.7	23.7	26.7
5	228	2	57	143	144	148	23.7	23.7	26.7

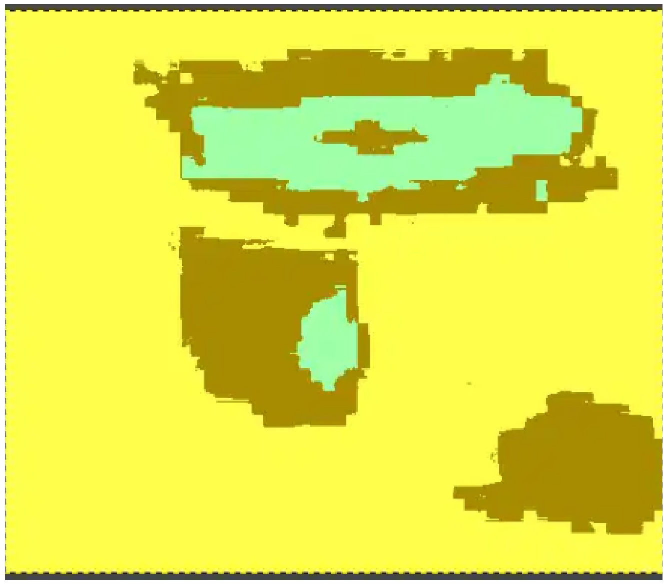
Plot of the coverage for each color 1, 2, 3, 4, 5 of the smooth palette. Three regions, depicted by patterned columns on the left axis, can be discriminated.



5 levels: palette 2

5 colors ➪ 3 regions (number of unique colors is three)





Other examples of Mapping on palettes

The same basis image used in the section paragraph serves to illustrate mappings on smooth palettes derived from colors present in an image.


original image (reminder)

As a second reminder, the mapping on a palette, and especially a user palette, is made in three steps:

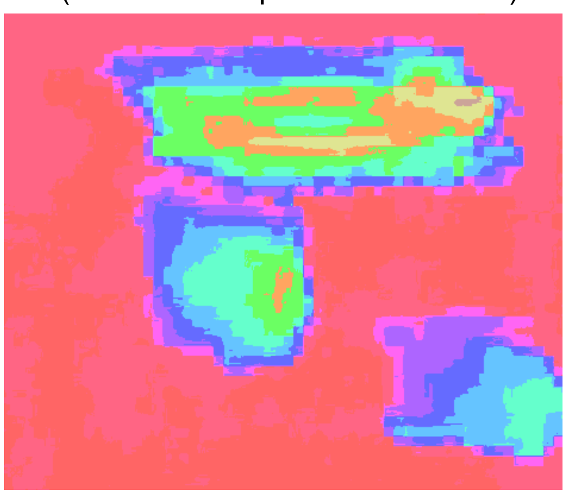
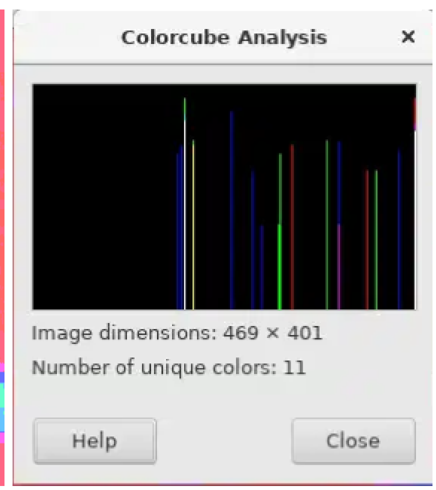
- First step: deriving a smooth palette from a set of given colors (the colors in the image); **Colors>Info>Smooth Palette**
- Second step: creating the palette and making it current; **Windows>Palettes>;** *create the palette; add colors eyedropping the colors on the output image from the previous step*
- Third step: the new palette, being selected, and said “current”, is used in background of the mapping (rearrangement of the colors) **Colors>Map>Smooth Palette**

In the three following examples, the match between the number of colors desired and the resulting “Number of unique colors” is not met because the arrangement of the colors has not a good combination (for similar reasons given in previous paragraphs).

Example 1: 18 colors palette

This example does not follow the step-by-step. In fact, the palette used is a preexisting palette (**Windows>Palette**).

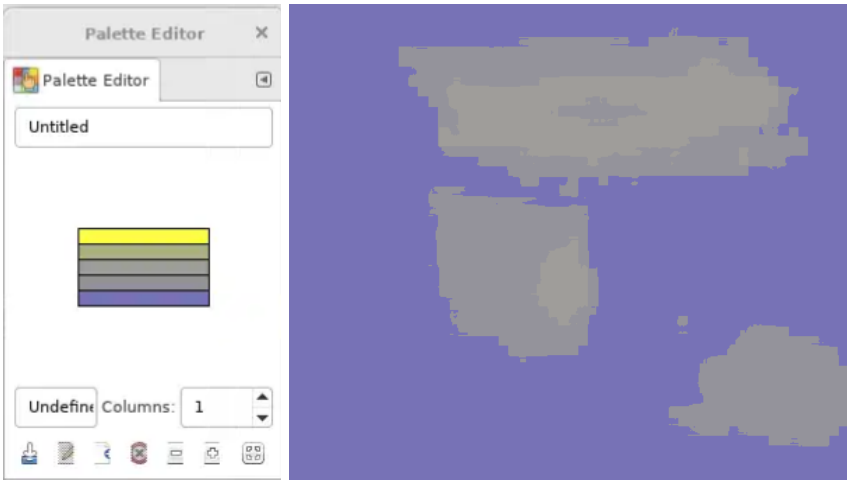
18 colors ➪ 11 regions (number of unique colors is eleven)

Example 2: 5 colors palette

In this example the step-by-step is followed. The smooth palette used is just an alternative subset in all the possible subsets of the 238 preexisting colors in the image (paragraph Input; output (256 -> 238)).

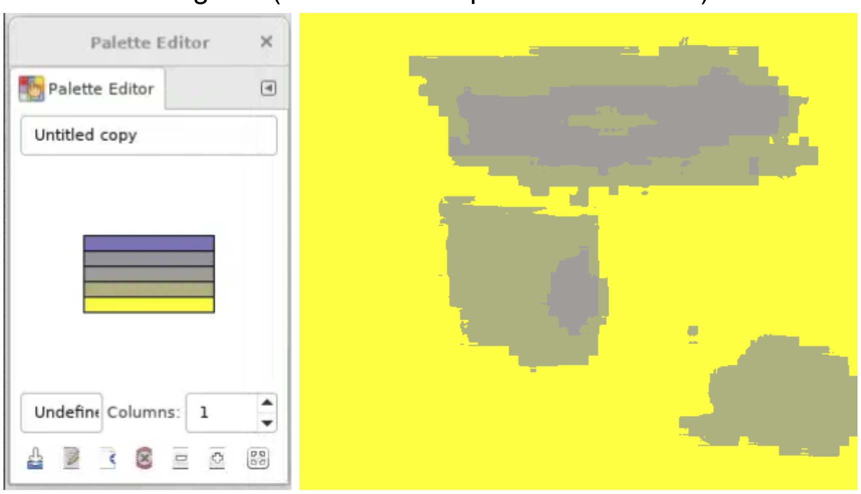
5 colors ➪ 3 regions (number of unique colors is three)



Example 3: another 5 colors palette

In this example the step-by-step is followed. The smooth palette used is the same as the previous example, in the opposite order.

5 colors ➪ 3 regions (number of unique colors is three)



CONCLUSION

This article has been presenting some aspects of the use of the *GIMP* smooth palette tool. The small number of color levels has some limitations because of the inherent constraint of choosing a subset within a larger set, resulting in a loose control of valuing desired areas of a picture.

Another article will talk about other related tools, following the same idea of valuing a desired part of an image.