

pv – A Tool for Color Palette Visualization

Irwin Sobel
Hewlett-Packard Laboratories
Palo Alto, CA

Abstract

We present a tool for interactively viewing the palette (3D color histogram) of an RGB image in TIFF format. The histogram is embedded in a wire-frame RGB cube. The use of a volume-renderer (**Volpack**²) allows us to display much fine detail not apparent using previous display methods. In addition internal histogram structure is revealed via user selection of one of 63 iso-surfaces of the distribution.

The tool allows interactive selection and modification of colors in the histogram and associated image. 3D histogram structure is made apparent via rotation of the RGB cube and rapid real-time re-rendering. There is also a stereo version *pvs* which works with **CrystalEyes** shutter-glasses by Stereographics Inc.⁸ and a compatible HP graphics card and monitor.

Introduction

There have been many visualizations created of 3D histograms of color images. A few recent examples are given in the references³⁻⁷. We have been working for some time to better visualize these objects. The use of a volume-renderer (**Volpack**³) allows us to display much fine detail not apparent using previous display methods. In addition internal histogram structure is revealed via user selection of one of 63 iso-surfaces of the distribution.

We call such a histogram the color **palette** of an image, since it is a quantitative representation of the colors present in the image. For example if the image were a painting, it would tell precisely how much of each distinguishable color was contained in the painting; thereby quantifying the concept of the artist's palette.

This document describes an interactive palette visualization tool that we have named *pv* (palette viewer). *pv* currently runs in the **HP-UX 9.07 X11R4/Motif1.2** environment on HP700 series workstations. It uses the **itcl**¹ scripting language which is an extension of **Tcl/Tk**¹. The current version is **itcl2.1** which includes an embedded, modified **Tcl7.5/Tk4.1**. A 24-bit graphics interface is not strictly required but is strongly recommended. If it is not present the color quality and display speeds will be severely degraded.

There is also a stereo version *pvs* which works with **CrystalEyes** shutter-glasses by Stereographics Inc.⁸ and a compatible HP graphics card and monitor.

A **palette** in *pv* consists of a 64x64x64 array of histogram **buckets**. Each bucket is addressed by a set of 18-bit **rgb** coordinates. To compute the palette of a 24-bit RGB color image, the pixels are scanned sequentially. For each pixel, an **rgb** bucket address is computed using the high-order 6-bits of its 8-bit RGB components. The bucket addressed by the **rgb** coordinates contains a counter which is incremented. At the end of the scan every bucket contains a count of all image pixels having its address. We refer to this count as bucket **occupancy**.

Description

pv accepts TIFF format images. It searches for files having .tif suffixes, computes an image palette and stores it if necessary, and produces an interactive display of the palette and the image as shown in Figure 1. Figure 1a is the "Picture" window which contains the image. Figure 1b is the "Palette" window which contains the palette. The palette is displayed in a colored wire-frame cube showing the boundaries of the **rgb** space. In the following description we use boldface and <> brackets to denote interactive controls such as buttons and sliders - e.g. the <**Reset**> button. Some buttons change name depending upon their state - e.g. <**Loop/Stop**>.

The Volume Cursor

In the palette view there is a 3D white crosshair volume cursor which is initially set at the **rgb** bucket position corresponding to the RGB color of the upper left corner pixel of the image - in this case a light gray background color. A typical RGB color value of the cursor position is also displayed numerically and as a color patch in the horizontal strip below the palette. In the case shown RGB = (160,175,167). There is also a 4th number - in this case 49 - listed in the strip. In general it has a value in the range [1:63] and is a monotonic function of the bucket occupancy which we call the **occupancy-group**. It is explained below.

The <**CursorOff**> button to the right of the horizontal cursor display strip, can be used to turn off the white crosshair display if it is obscuring some feature(s) of interest in the palette.

The Motion Controls

The row of buttons below the RGB display strip are for controlling the view of the palette. From left to right in Figure 1b, their names and functions are:

<2X/.5X>	enable/disable 2X magnification of the palette
<Loop/Stop>	rotate continuously about a line parallel to screen-y at the cube center / stop rotating
<CW>	rotate (or step if not Looping) CW as viewed from above
<CCW>	rotate (or step if not Looping) CCW as viewed from above
<RotX90>	rotate the cube 90 degrees about a line parallel to screen-x at the cube center.
<Reset>	reset the view to the initial one with neutral color axis(main diagonal) vertical

The State Toggles

Below the motion control buttons are a row of buttons which toggle the current state of the viewer. Their names correspond to boolean state variables. When the corresponding variable is set, the button's indicator is red. From left to right in Figure 1b, their meanings are:

<DepthCue>	enable/disable depth cueing in the palette view
<Fill>	enable/disable 3D filling for the selection
<Fine>	enable/disable fine (single bucket) color selection
<XFM>	enable/disable color space Transforms
<ShowSelect>	enable/disable showing only selected palette buckets

The Occupancy-group Slider

Bucket occupancy counts depend upon the size of an image and can take on an extremely large range of values. Thus we use a 32-bit counter to hold them. Also the occupancy counts for a given image tend to be sparsely distributed over their range. For example the digitized photograph shown in Figure 1a contains 174064 pixels which sort into 11715 palette buckets. If each of the occupied buckets were equally represented in the image, it would have $174064/11715 = 14.858216$ pixels on the average. There would be only 2 occupancy values (14 and 15) for all the buckets. - i.e. 1661

buckets with occupancy 14, and 10054 buckets with occupancy 15. This is an extreme case and is rarely found in photographic imagery. The image in Figure 1a is more typical. There are 289 distinct occupancy values for this palette. Of the 289 occupancy values, 131 are unique; that is, only a single bucket has the given occupancy value.

In order to create an interactive slider which can peel away less occupied buckets from the palette view in a useful manner, we had to resort to not using occupancy, but a more compact function of occupancy which we call **occupancy-group**. We defined this by sorting buckets on occupancy (lowest first), and defining 63 **occupancy-groups** such that each step, progressing from group 1 up the occupancy-group scale, removes roughly a constant fraction of buckets-left, with the constraint that all buckets with a given occupancy belong to the same group.

Below the state toggles lies the **<Occupancy-group Slider>** - the slider at bottom of the Palette window that runs from "opaque" on the left to "transparent" on the right. It is used to set an occupancy-group threshold T (31 in Figure 1b). The volume renderer will treat as transparent all buckets having

$$\text{occupancy-group} < T \quad (1)$$

T can take on values in the range [1:63]. Figure 1c shows The Palette window with $T = 1$. This is the complete palette and should be contrasted with the $T = 31$ iso-surface shown in Figure 1b. Notice how the dominant color clusters are apparent in Figure 1b.

The Selection Mechanism

The selection mechanism is designed to control which palette buckets are selected when the user points and clicks with the mouse over the palette or the image. It is modified by the **<Fill>** and **<Fine>** state toggles.

The **<Fine>** state toggle is meaningful only when **<Fill>** is off. In this case it modifies the range of colors selected for highlighting. The default (**<Fine>** off) is for pixels in the designated palette bucket and its 6 **3D-face-neighbor** buckets to be highlighted. The **3D-face-neighbor** buckets are those immediately adjacent to a bucket in the **R,G,B** palette directions respectively. If **<Fine>** is on, then only pixels in the designated bucket are selected. In what follows we refer to the effect of **<Fine>** by the "(s)" in the phrase "selected palette bucket(s)".

In the **<Fill>** off (default) state, bucket selection works as follows:

1. If the user clicks the **<left>** mouse button over a non-empty position in the **palette**, the white crosshair will move to that location, and all pixels in the **image** contained in the selected palette bucket(s) will be highlighted.

2. If the user clicks the **<left>** mouse button over any position in the **image**, the white palette crosshair will move to the color at that position, however **no** pixels will be highlighted in the **image**.
3. If the user clicks the **<middle>** mouse button over any position in the **image**, the white palette crosshair will move to the palette bucket containing the color at that image position, and all pixels in image belonging to selected palette bucket(s) will be highlighted. If the user **drags** the mouse over the **image** before releasing the button, the selection process will be repeated for all locations covered.

If the **<Fill>** state is true (toggle indicator is red), then the bucket designated in cases 1 and 3 above is used as a seed in a 3D fill algorithm. The effect is to expand the set of buckets selected to all those that are connected to the seed inside the currently selected palette iso-surface.

Successive selections using the mouse are cumulative. Thus more palette buckets and image pixels get selected at each relevant mouse click. The selection memory can be reset via the **<UnSelect>** button in lower right corner of the Picture window.

The Highlighting Mechanism and The Fade Slider

The highlighting mechanism was designed to solve the problem of how to highlight selected pixels in a color image with minimum color distortion to the pixels and their image context. The solution chosen was to fade all non-selected pixels into a neutral background. The amount of fading (in this case 50%) is interactively controlled via the **<Fade Slider>** at bottom of the Picture window. It ranges from 0% (no fading) to 100% whereby all unselected pixels are faded to the background color. The current background color (in this case black) is displayed in the square to the right of the **<Fade Slider>**. This can be interactively modified by clicking on it as follows:

<left> mouse button makes it darker (until black)

<right> mouse button makes it lighter (until white)

Any modifications of the fade background are also used to reset the background of the palette view.

N.B. the image is not faded in Figure 1a in spite of the fade slider indicating 50% fade. The program initializes with no fade. However, as soon as a selection is made, the indicated (50%) fade will be executed.

The Save Button

The **Picture** window has a **<Save>** button which allows the user to save the image at any time in file **<fname>.tif**. A pop-up dialog allows the user to choose **<fname>**.

Transform Mode and Controls

The **Palette** window has a **<XFM>** button. It enables **transform mode** in which the user may specify a color transform in terms of hue, chromaticity, lightness, and contrast changes. The transform is applied to the selected pixels and the result is immediately displayed both in the Palette and Picture views. The **<XFM>** button is only functional if there are some colors selected; if not it just beeps and reports

"RGB2XFM nothing selected (nothing done)"

in the terminal window from which it was started. Figure 2 shows the result of

- setting the **<Occupancy-group Slider>** to the smallest value that leaves the reds as an isolated cluster in the Palette window ($T = 10$)
- selecting most of the flowers by clicking on **<Fill>** and then clicking on the red blob in the Palette window with the **<left>** mouse button, clicking on **<Fill>** to turn it off
- setting the **<Occupancy-group Slider>** to a lower value ($T = 4$) to include some more reds in the Palette window, and then repeatedly clicking over the red cluster in different views to pick up more reds surrounding the original cluster, while avoiding the reds of the woman's lips
- clicking on **<ShowSelect>**, **<CursorOff>** and **<XFM>**. The Palette viewpoint snaps to the neutral axis above the white point to show the chromaticity projection (Figure 2a), and the "Change Color" transform control window (Figure 2b) appears.
- changing the hue angle by a 50 degree rotation (see "Hue Increment" in the "Change Color" window of Figure 2b). The result appears in the Picture window 50% faded to highlight the selection (Figure 2c)
- clicking on the left end of the fade slider under the Picture window with the **<middle>** mouse button to undo the fade and see the result in the normal image context (Figure 2d).

Diagonal Von Kries Type Transform

The "Von-Kries" type scaling transform is a diagonal matrix with per-channel scale factors (R_s, G_s, B_s). This allows previewing addition or removal of color-casts in images. The procedure is as follows:

- the user must make the usual selection of colors of interest (**COI**)

- the last point selected in the image of the **COI** with the **<middle>** button click will be singled out as the "color to be changed" (**CTBC**).
- The user must then click on the **<XFM>** button, and use the transform controls to change the selected area into the color desired. The original RGBH value of the **CTBC** is stored in the "Change Color" transform control window as the "BaseColor" display. As color changes are made, the updated **CTBC** is stored in the "Palette" window RGBH display.
- clicking on the **<VonKries>** button causes (**Rs,Gs,Bs**) scaling to be applied to the whole image. The scales are ratios of the new/old values for each color channel of the **CTBC**.
- Clicking again on the **<VonKries>** button will restore the old selection and allow for interactive modification of the scaling transform.

Figure 3 shows the effect of clicking on the **<VonKries>** button after all the operations above. The purple shade that the original flower sample was mapped into now gives an overall cast to the image. Note that the color of the flowers is essentially the same in Figure 3 as in Figure 2d. However the viewer's perception of the color is affected by the color cast of the image, and thus the flowers in Figure 2d will appear redder to most viewers.

Early Observations

Some early observations that we have made with **pv** are:

- Scaling an image changes its palette. Scaling up tends to give smoother palette shapes, while scaling down tends to make palette shapes more ragged and disconnected. This is consistent with the fact that scaling up via interpolation yields more pixels per bucket and creates intermediate colors at color boundaries - the net effect being to spread out and fill in the palette. Scaling down by interpolation also adds intermediate colors, but reduces the pixels per bucket to the point where bucket quantization effects interfere with palette occupancy-group thresholding, giving a ragged effect.
- Passive reflective surfaces have banana-shaped palettes roughly parallel to the lightness axis:
 - hue from body reflectance (matte component)
 - varying lightness from surface orientation and specular component

Peng et.al.⁵ have confirmed this in a computer simulation of the physics of image formation.

- Palettes of images whose color range has been reduced by quantizing, show up easily. So do those from images that have been contrast stretched.
- Many conjectures can be easily checked... e.g. that skin-tone diversity is 1-dimensional is valid. Mainly skin lightness, and to a lesser extent chroma, varies along a curve in RGB space, across racial groups.

History and Acknowledgement

Milon Mackey of HP-Labs, who had been working with the **Volpack**² volume-renderer from Stanford, was able to take my histogram data, format it, and display it in 24-bit volume-rendered color. Moreover the renderer was fast enough to allow for the possibility of an interactive 3D orientation capability. Milon also showed me **vprdemo** - an interactive X11/Motif-based program, provided with the the **Volpack** package. **vprdemo** came with source code in both **C** and the **Tcl/Tk** scripting language. We started with **vprdemo**, added color histogram input and many interactive tools over a period of about 7 months, and arrived at **pv**.

References

1. itcl [Incr Tcl], <http://www.tcltk.com/itcl>, based on the Tcl scripting language, and the Tk GUI toolkit, <http://www.tclconsortium.org/>
2. Philippe Lacroute and Marc Levoy, The VolPack Volume Rendering Library, <http://www-graphics.stanford.edu/software/volpack>, last update: 4 January 1995, volpack@graphics.stanford.edu
3. Carol L. Novak, Steven A. Shafer, Estimating Scene Properties from Color Histograms, <ftp://reports-archive.adm.cs.cmu.edu/1992/CMU-CS-92-212.ps>, CMU-CS-92-212, November 1992 (30 pages), Available from CMU/CSD
4. Johan Lammens, Constancy in a Computational Model of Color Perception and Color Naming., <http://www.cs.buffalo.edu/pub/colornaming/diss/subsection2.11.1.2.html>, <http://www.cs.buffalo.edu/pub/colornaming/diss/diss.html>, PhD Thesis, 1994, CSD SUNY Buffalo, Buffalo, NY.
5. Liang Peng, Eric P. F. Lafortune, Donald P. Greenberg, Irwin Sobel, Use of Computer Graphic Simulation to Explain Color Histogram Structure, *Proc. Fifth Color Imaging Conference: Color Science, Systems, and Applications (CIC97)*, Scottsdale, AZ, Nov, 1997, IS&T, 7003 Kilworth Lane, Springfield, VA 22151
6. Giordano B. Beretta, Graphical User Interface for Controlling Color Gamut Clipping, U.S. Patent 5,416,890, May 16, 1995, assigned to Xerox Corp., Stamford, Conn.
7. Giordano B. Beretta, Graphical User Interface for Editing a Palette of Colors, U.S. Patent Office Statutory Invention

Registration SIR-H1506, Dec. 5, 1995, assigned to Xerox Corp., Rochester, N.Y.

8. StereoGraphics Inc., 2171 E. Francisco Blvd., San Rafael, CA 94901, U.S.A., sales@stereographics.com, <http://www.stereographics.com>



Figure 1a. The Picture Window

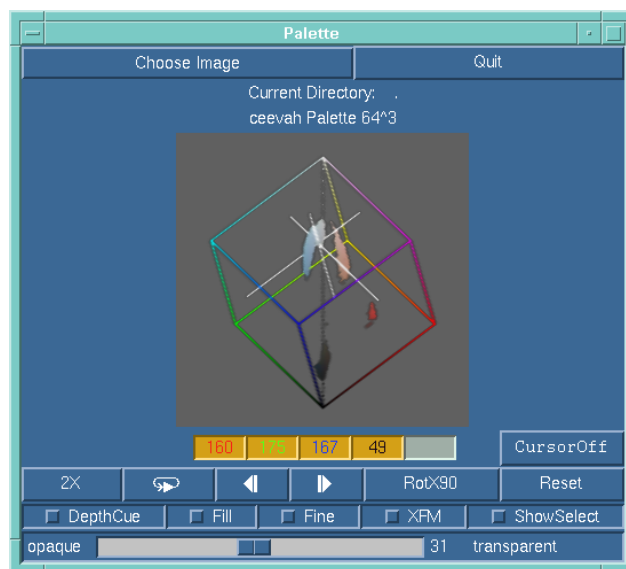


Figure 1b. The Palette Window

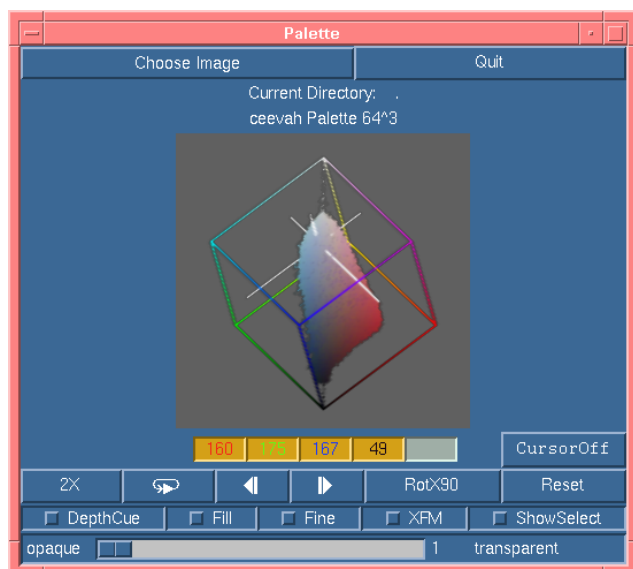


Figure 1c. The Palette Window (complete palette)

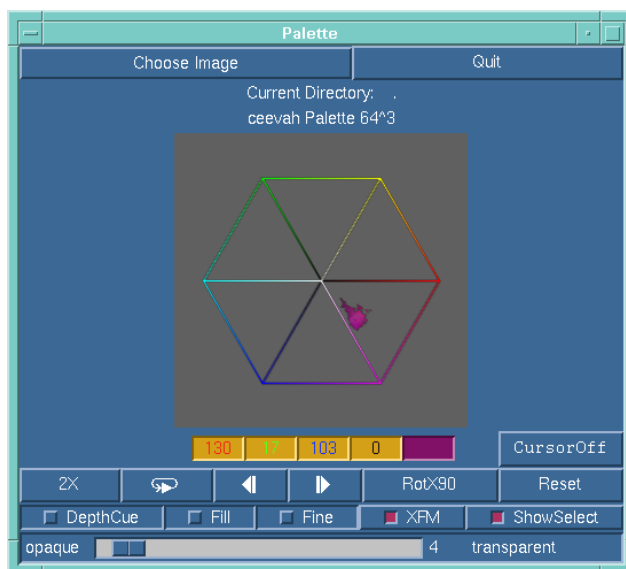


Figure 2a. The Palette Window (chromaticity view)



Figure 2b. The Change Color Window

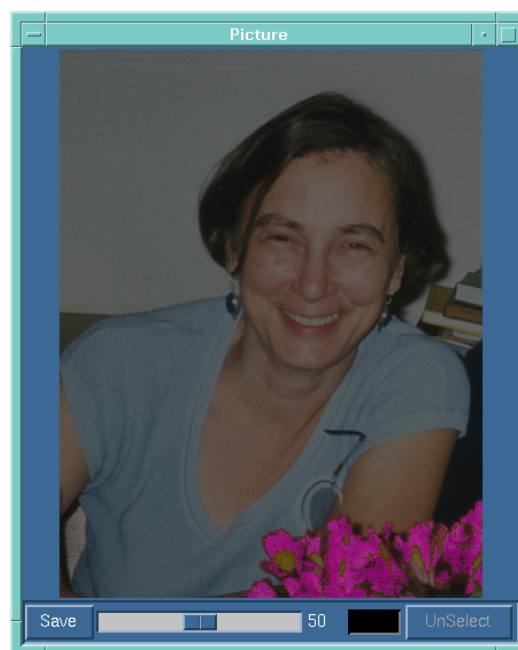


Figure 2c. The (XFM'd) Picture Window



Figure 2d. The (XFM'd) Picture Window (no fade)

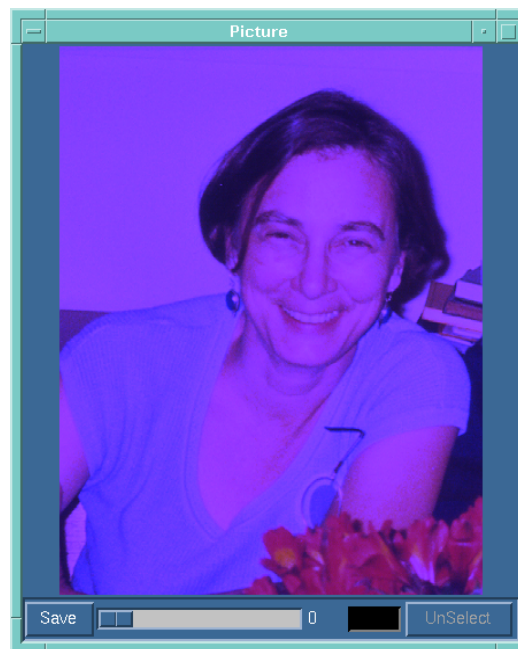


Figure 3. After Von Kries Transform