

Proposal for a standard 3D-LUT format based on PNG-files

Jan Fröhlich

Technical Director / CinePostproduction

Truelight



```
# Truelight Cube v2.0
# lutLength 101
# iDims 3
# oDims 3
# width 16 16 16
```

```
# InputLUT
0.000000 0.000000 0.000000
0.010000 0.010000 0.010000
```



IRIDAS

```
LUT_3D_SIZE 64
0.000000 0.000000 0.000000
0.000809 0.000000 0.000214
0.002701 0.000275 0.001678
0.004578 0.000320 0.001770
0.006439 0.000320 0.001770
0.008301 0.000931 0.001770
0.010147 0.001587 0.001770
0.012558 0.002243 0.001770
0.015030 0.002914 0.001770
0.017456 0.003571 0.002441
0.019883 0.004211 0.003174
0.023392 0.005493 0.002533
0.026978 0.006790 0.001801
0.032166 0.008072 0.001083
0.037446 0.009384 0.000366
0.044099 0.010101 0.000000
0.050340 0.010758 0.000000
0.058762 0.012039 0.000000
0.067033 0.013321 0.000000
```



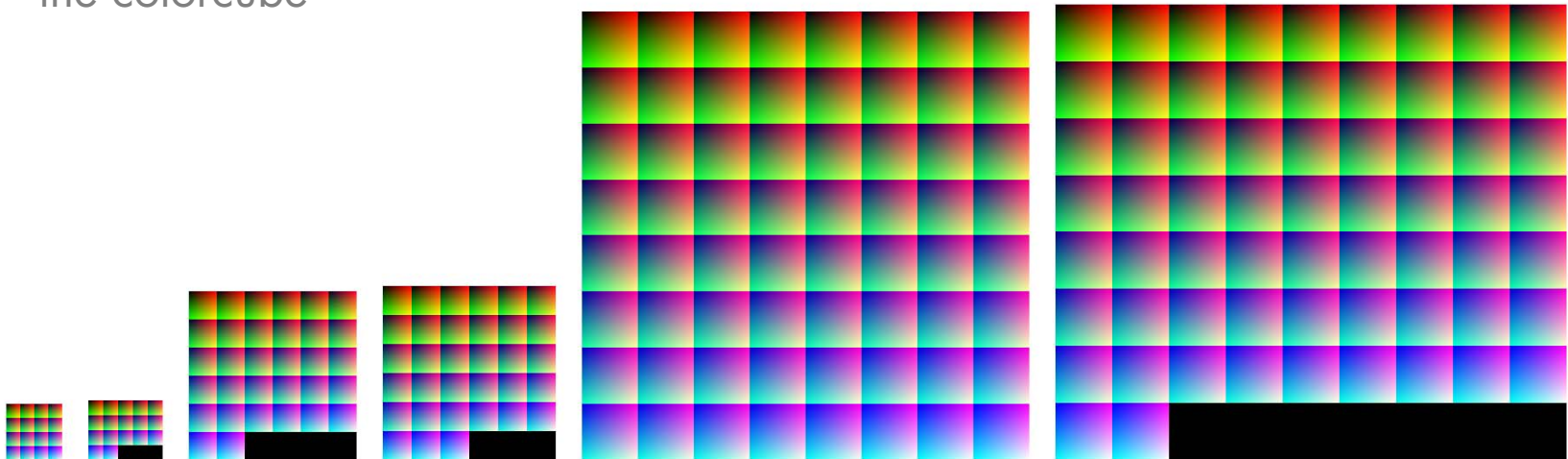
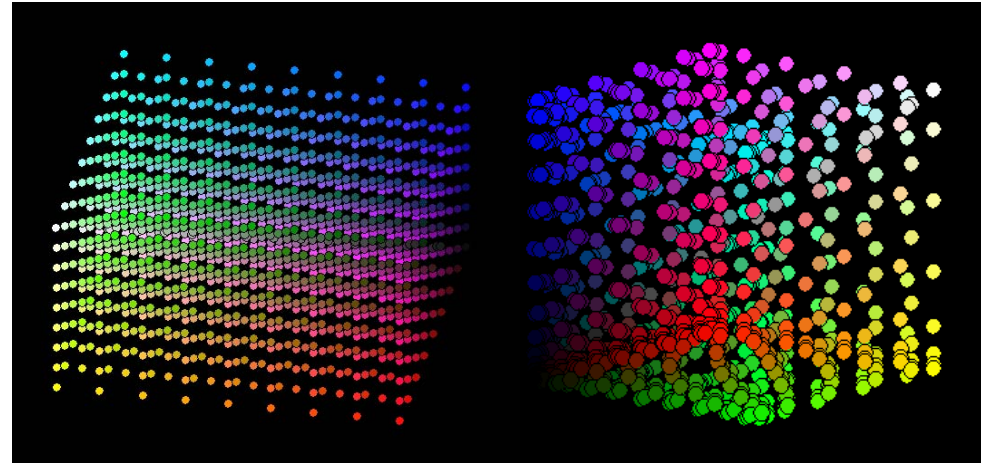
DVS

```
<LUT3D
name='64erSourceQubePreview_709'
N='17' BitDepth='16'>
  <values>
    0      0      0
    116    20     414
    114    186    965
    127    429    1717
    0      691    3187
    0      1022   5610
    0      921    9714
    1495   2055   15504
    2447   2520   21950
    3074   1992   27318
    3521   1167   31173
    3860   0      33890
    4044   0      35855
    4213   0      37069
    4299   0      37793
    4308   0      38258
```



The „PNG-Square“-LUT

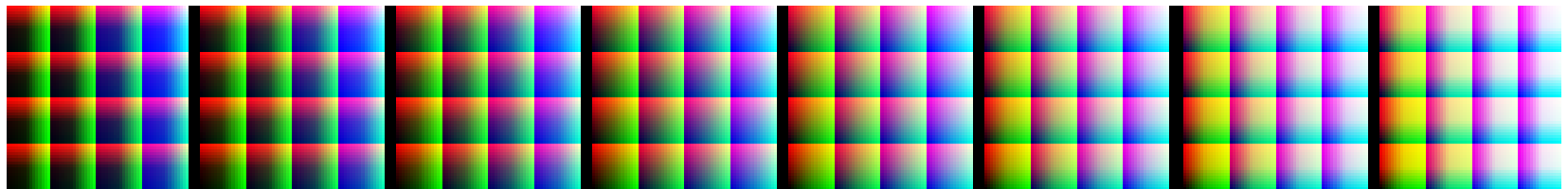
- Equally spaced vectorfield like other 3D-LUTs
- Any size
- 16 bit precision
- Complete slices through the colorcube



16x16x16, 17x17x17, 32x32x32, 33x33x33, 64x64x64, 65x65x65 3D-LUTs as „PNG-Square“

The „PNG-Square“-LUT

- Any software is able to render looks as 3D-LUTs
- Human readable
 - Graphics format
 - Complete slices through the colorcube
- Small in size - around 1/10 compared to text based formats due to PNG-compression → sequences for dynamic gradings possible
- PNG is royalty-free and widely supported

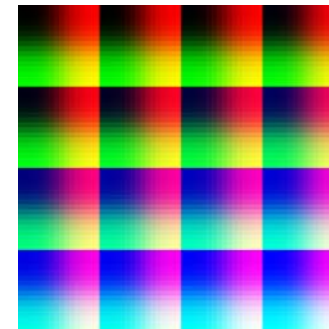
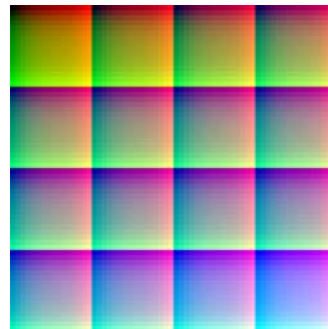
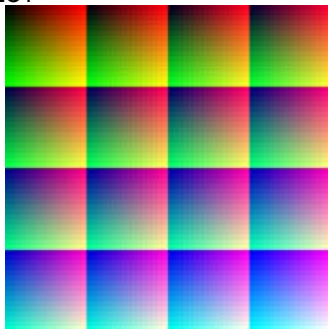


Sequence of 16x16x16 3D-LUTs as „PNG-Square“ for dynamic gradings

Source picture



3D-LUT

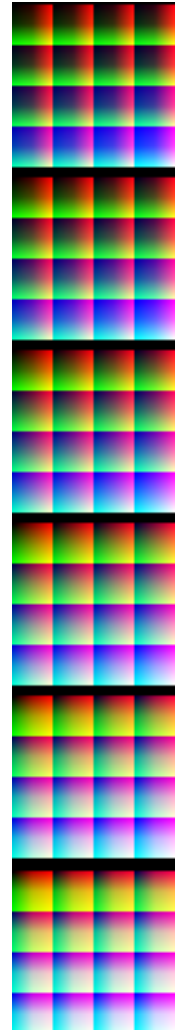


Resulting picture




```
%Initialisations:
Qube.Size = 64; %try out different values.
%CinePostproduction usually uses 64x64x64 3D-Luts for maximum accuracy.
Qube.Path = 'C: ';
Qube.Name = 'PNG_Square_Qube_TEST_64x64x64';
Qube.Data = uint16(repmat(0,[Qube.Size Qube.Size Qube.Size 3]));
for a=1:Qube.Size, for b=1:Qube.Size, for c=1:Qube.Size
Qube.Data(a,b,c,1)=(((a-1)/(Qube.Size-1)*65535));
Qube.Data(a,b,c,2)=(((b-1)/(Qube.Size-1)*65535));
Qube.Data(a,b,c,3)=(((c-1)/(Qube.Size-1)*65535));
end; end; end;

%Format the Qube:
disp('Writing PNG-Square Qube')
Width = Qube.Size* ceil(((Qube.Size^3)^(1/2))/Qube.Size)
Height = Qube.Size* ceil(((Qube.Size^3)^(1/2))/Qube.Size)
QubesPerLine=ceil(((Qube.Size^3)^(1/2))/Qube.Size)
for c=1:Qube.Size, for b=1:Qube.Size, for a=1:Qube.Size
Bild((rem(b-1,Qube.Size)+1+Qube.Size*floor((c-1)/QubesPerLine)),...
a+Qube.Size*rem((c-1),QubesPerLine),1) = Qube.Data(a,b,c,1);
Bild((rem(b-1,Qube.Size)+1+Qube.Size*floor((c-1)/QubesPerLine)),...
a+Qube.Size*rem((c-1),QubesPerLine),2) = Qube.Data(a,b,c,2);
Bild((rem(b-1,Qube.Size)+1+Qube.Size*floor((c-1)/QubesPerLine)),...
a+Qube.Size*rem((c-1),QubesPerLine),3) = Qube.Data(a,b,c,3);
end; end;
if Qube.Size>50, disp(strcat('Slice Number: ',num2str(c)));, end; end;
%Write the Qube to disk:
imwrite(Bild,strcat(Qube.Path,'\ ',Qube.Name, '.', 'png'),'BitDepth',16);
disp('PNG-Square Qube has been written')
```



Let's make it simple!
Cameras, Workflows, Fileformats ...

CinePostproduction: www.cinepostproduction.de

Jan Fröhlich jan.froehlich@cinemedia.de
+49 89 64 99 2823

PNG-Square LUT Specification: www.digitalintermediate.de



SI-2K camera picture courtesy of Pillefilm, Germany, Tape picture courtesy of Sony