

成像算法基础

显示,基本工具,和统计

自己的工具箱

- 显示
- 基本操作
- 分解,组合
- 直方图累积直方图
- 分块和计算平均值

Matplotlib Vs Opencv

```
plt.figure(num='test', figsize=(x, y))  
plt.imshow(image)  
plt.xticks([], plt.yticks([])) # 隐藏 X轴 和 Y轴的标记位置和labels  
plt.show()  
•
```

```
img = cv2.imread("kodim19.png")  
cv2.namedWindow('image',cv2.WINDOW_AUTOSIZE)  
cv2.startWindowThread()  
cv2.imshow("image", img)  
cv2.waitKey(0)  
cv2.destroyAllWindows()
```

<https://matplotlib.org/gallery/index.html>

figure

```
matplotlib.pyplot.figure(num=None, figsize=None, dpi=None, facecolor=None, edgecolor=None, frameon=True, FigureClass=<class 'matplotlib.figure.Figure'>, clear=False, **kwargs)
```

num : integer or string, optional, default: None

If not provided, a new figure will be created, and the figure number will be incremented. The figure object holds this number in a `number` attribute. If `num` is provided, and a figure with this id already exists, make it active, and returns a reference to it. If this figure does not exist, create it and returns it. If `num` is a string, the window title will be set to this figure's `num`.

figsize : (float, float), optional, default: None

width, height in inches. If not provided, defaults to `rcParams["figure.figsize"] = [6.4, 4.8] = [6.4, 4.8]`.

dpi : integer, optional, default: None

resolution of the figure. If not provided, defaults to `rcParams["figure.dpi"] = 100.0 = 100`.

Returns:

figure : `Figure`

The `Figure` instance returned will also be passed to `new_figure_manager` in the backends, which allows to hook custom `Figure` classes into the pyplot interface. Additional kwargs will be passed to the `Figure` init function

-



imshow

matplotlib.pyplot.imshow(*X*, *cmap*=None, *norm*=None, *aspect*=None, *interpolation*=None, *alpha*=None, *vmin*=None, *vmax*=None, *origin*=None, *extent*=None, *shape*=<deprecated parameter>, *filternorm*=1, *filterrad*=4.0, *imlim*=<deprecated parameter>, *resample*=None, *url*=None, *, *data*=None, **kwargs)
[\[source\]](#)

X : array-like or PIL image
The image data. Supported array shapes are:

- (M, N): an image with scalar data. The data is visualized using a colormap.
 - (M, N, 3): an image with RGB values (0-1 float or 0-255 int).
 - (M, N, 4): an image with RGBA values (0-1 float or 0-255 int), i.e. including transparency.
- The first two dimensions (M, N) define the rows and columns of the image.

Out-of-range RGB(A) values are clipped.

cmap : str or [Colormap](#), optional
The Colormap instance or registered colormap name used to map scalar data to colors. This parameter is ignored for RGB(A) data. Defaults to `rcParams["image.cmap"] = 'viridis'`.

interpolation : str, optional
The interpolation method used. If *None* `rcParams["image.interpolation"] = 'nearest'` is used, which defaults to 'nearest'.

Supported values are 'none', 'nearest', 'bilinear', 'bicubic', 'spline16', 'spline36', 'hanning', 'hamming', 'hermite', 'kaiser', 'quadric', 'catrom', 'gaussian', 'bessel', 'mitchell', 'sinc', 'lanczos'.

If *interpolation* is 'none', then no interpolation is performed on the Agg, ps, pdf and svg backends. Other backends will fall back to 'nearest'. Note that most SVG renders perform interpolation at rendering and that the default interpolation method they implement may differ.

See [Interpolations for imshow/matshow](#) for an overview of the supported interpolation methods.

Some interpolation methods require an additional radius parameter, which can be set by *filterrad*. Additionally, the antigrain image resize filter is controlled by the parameter *filternorm*.

alpha : scalar, optional
The alpha blending value, between 0 (transparent) and 1 (opaque). This parameter is ignored for RGBA input data.

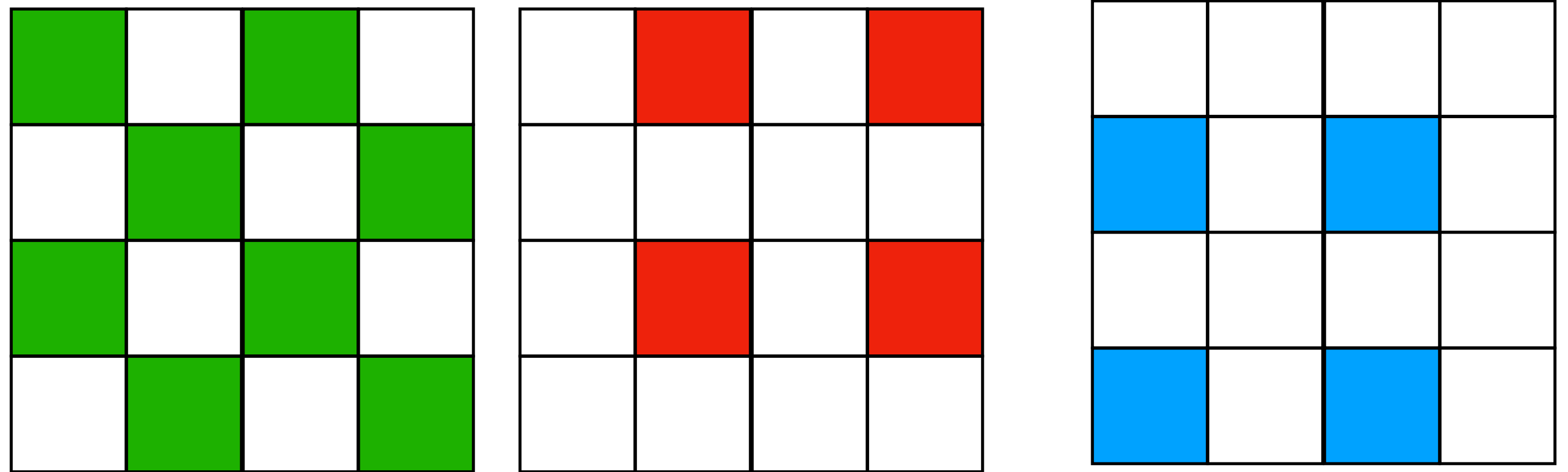
vmin, vmax : scalar, optional
When using scalar data and no explicit *norm*, *vmin* and *vmax* define the data range that the colormap covers. By default, the colormap covers the complete value range of the supplied data. *vmin*, *vmax* are ignored if the *norm* parameter is used.

-

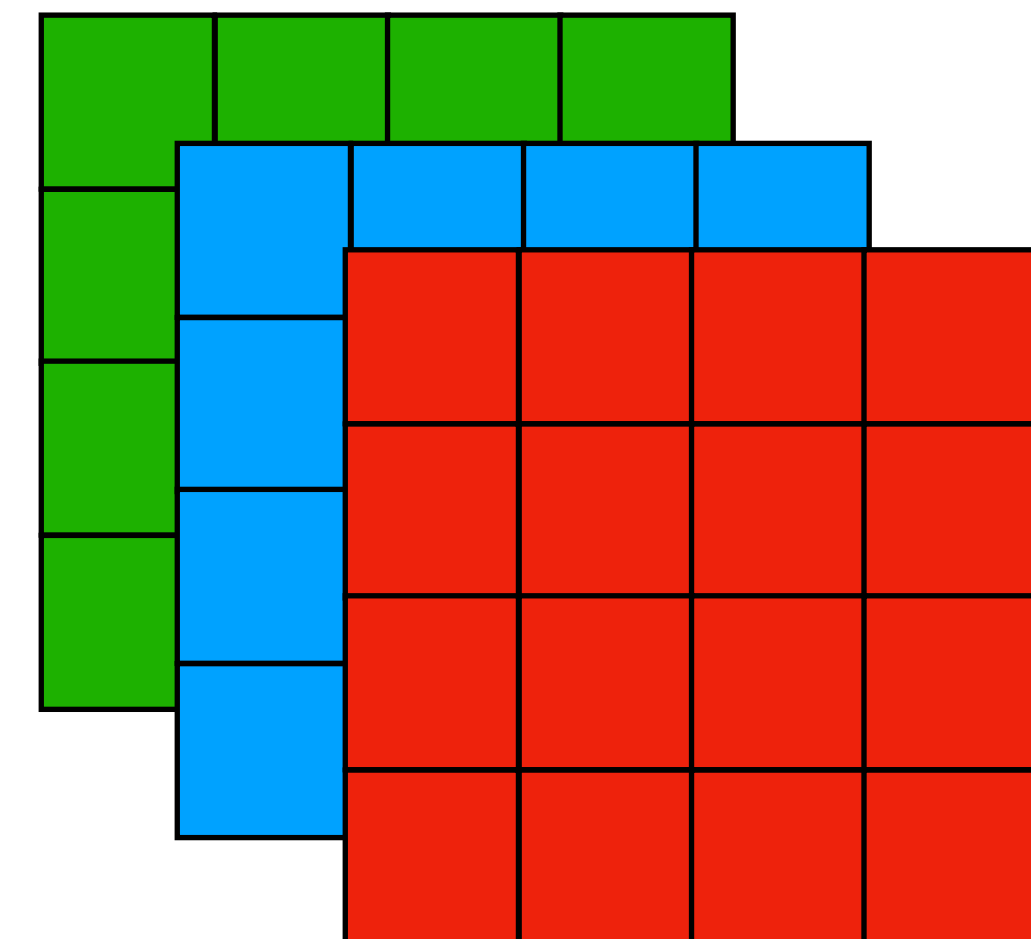


黑白和RGB显示

- 伪彩色
- 更接近原始图像信息



- 真彩色
 - 更接近人眼的感观
 - 需要进行demosaic,容易受到demosaic算法的影响



3D显示

`ax.plot_surface(X, Y, Z, rstride=1, cstride=1, cmap='rainbow')`

`ax.plot_wireframe(X, Y, Z, rstride=10, cstride=10)`

rstride, cstride : int

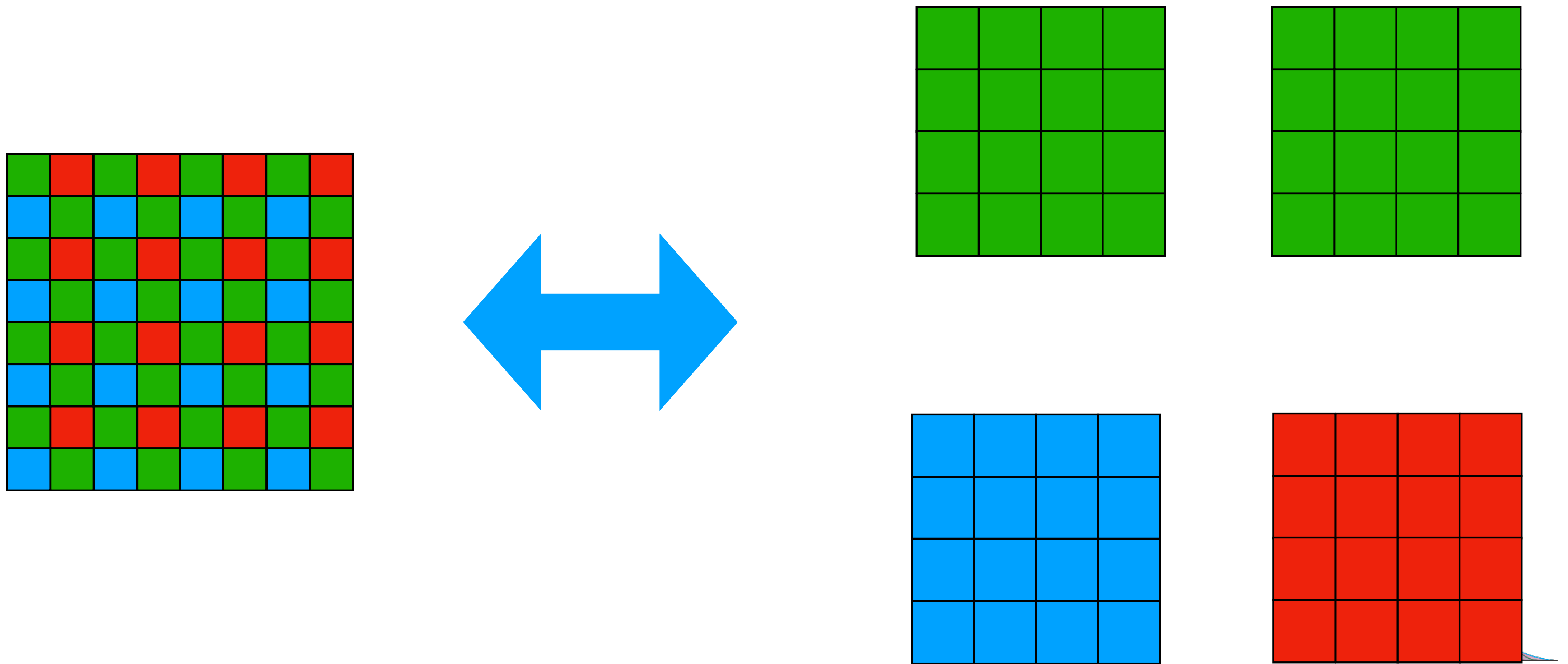
Downsampling stride in each direction. These arguments are mutually exclusive with *rcount* and *ccount*. If only one of *rstride* or *cstride* is set, the other defaults to 10.

'classic' mode uses a default of `rstride = cstride = 10` instead of the new default of `rcount = ccount = 50`.

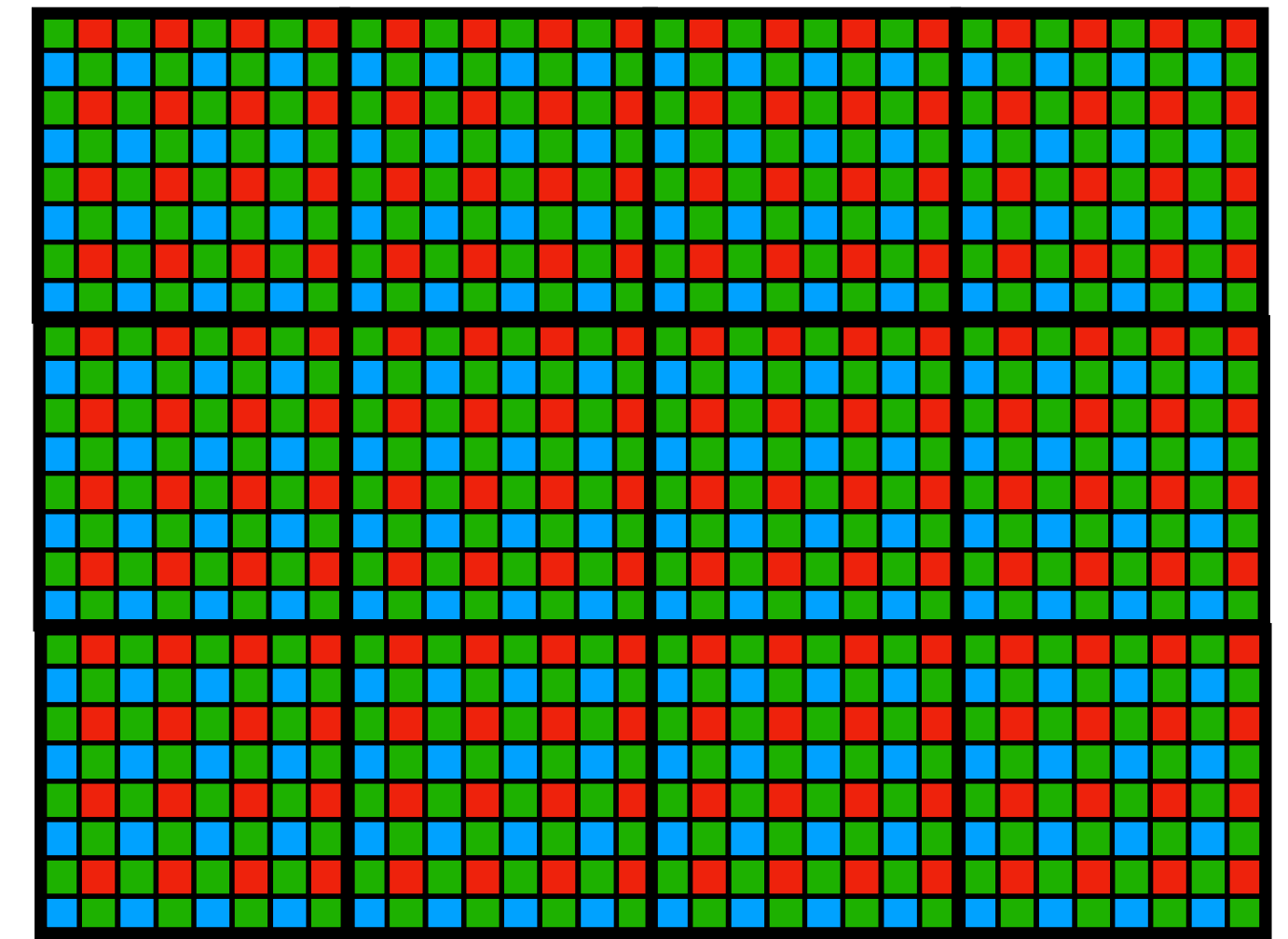
cmap : Colormap

Colormap of the surface patches.

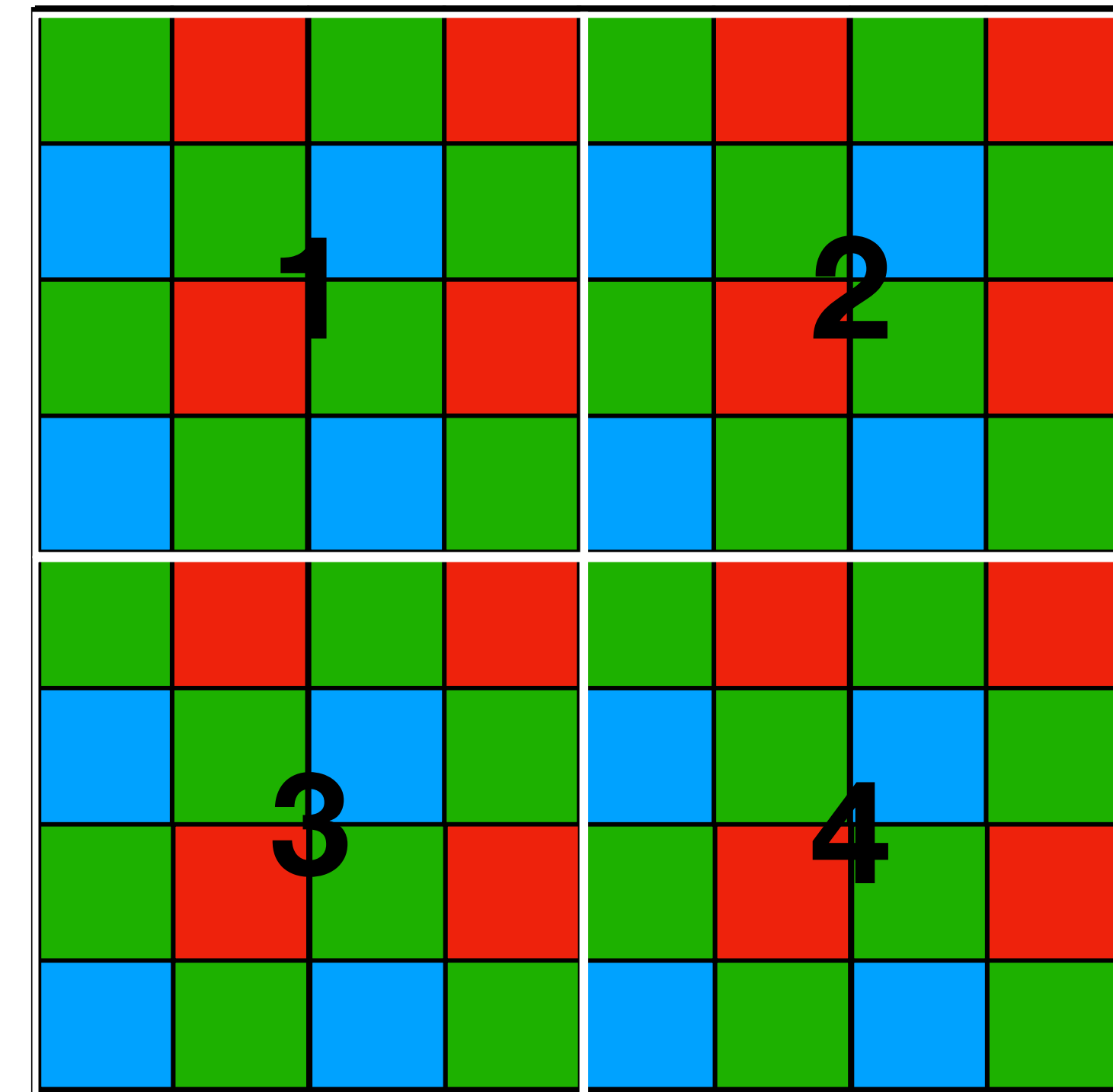
raw图像的分解和组合



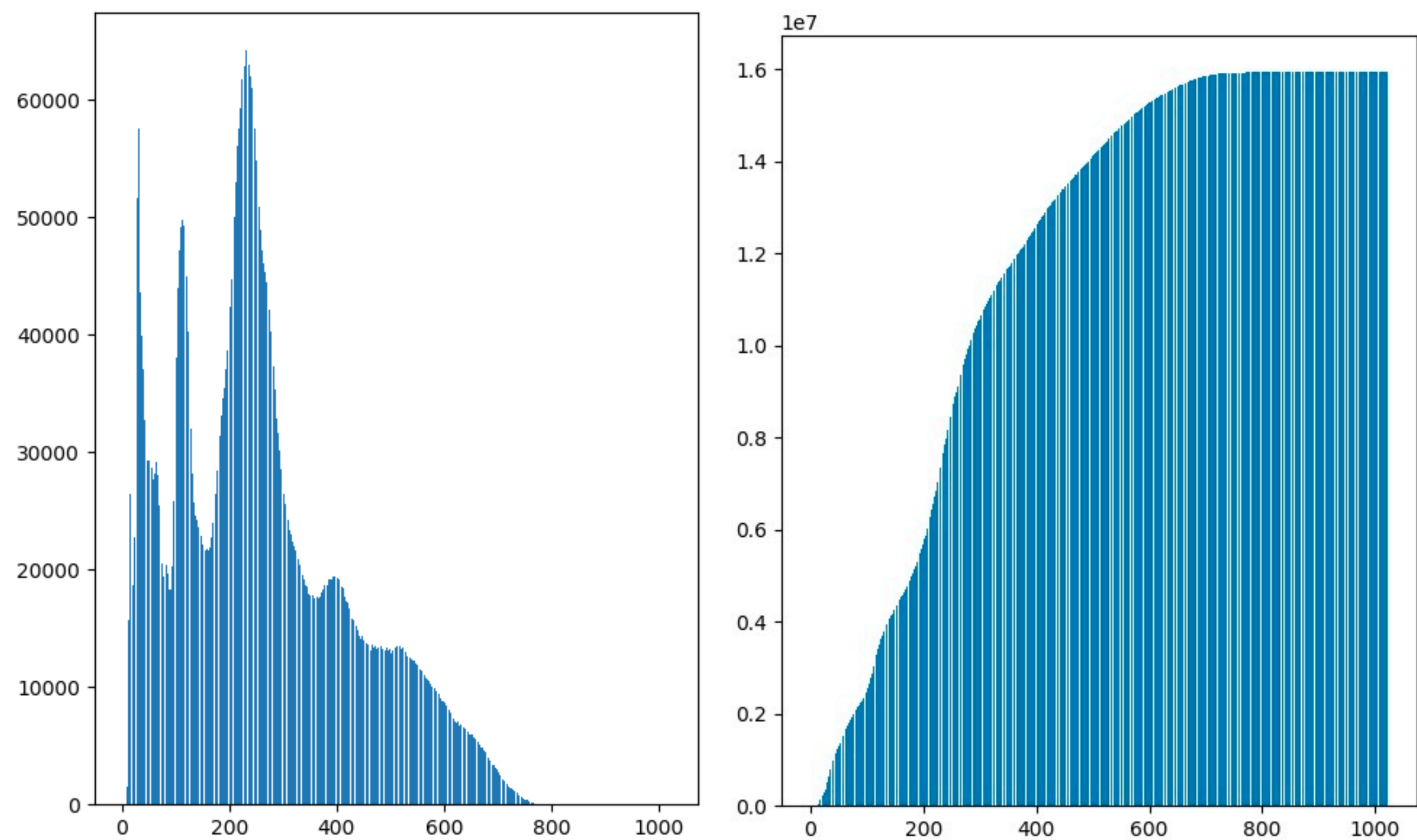
分块



- 目的:
- 1 反应局部特征
- 2 减少运算量



直方图和累积直方图



np.histogram

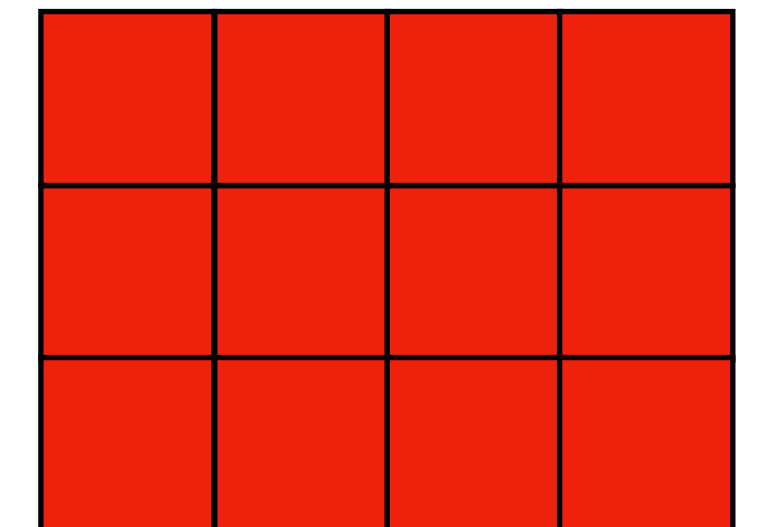
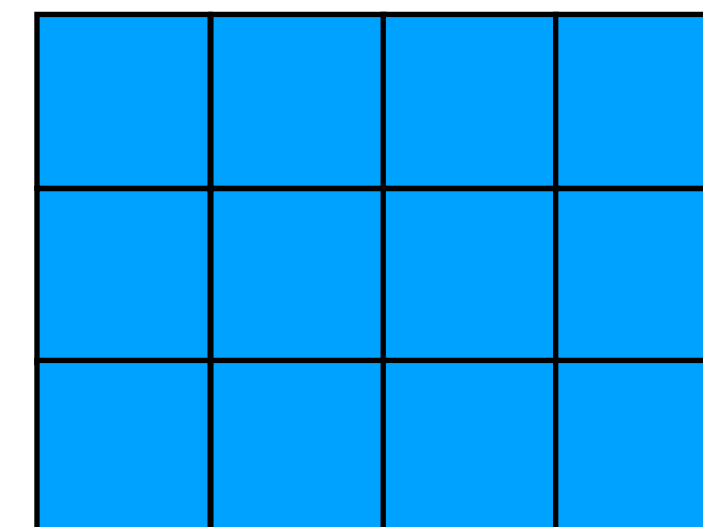
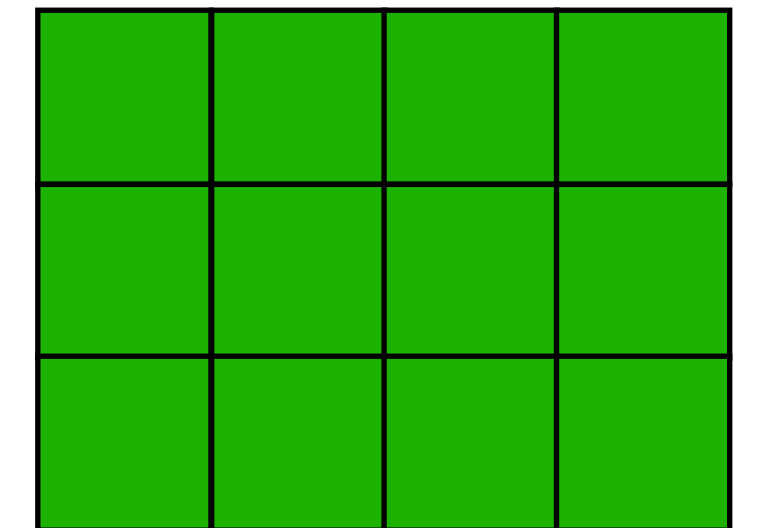
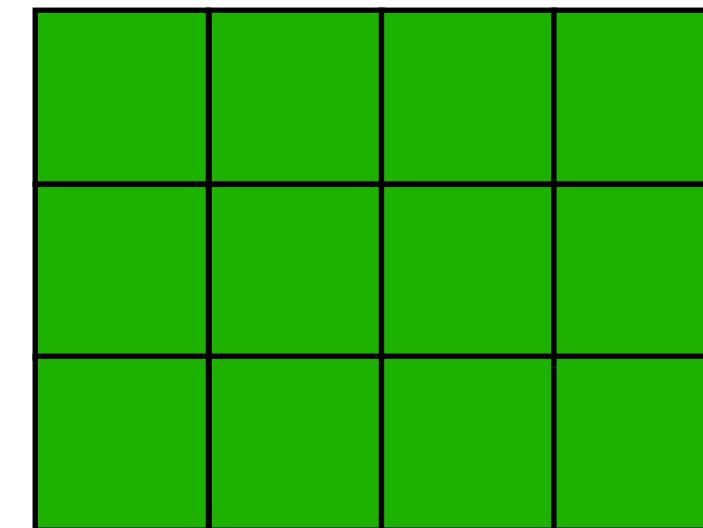
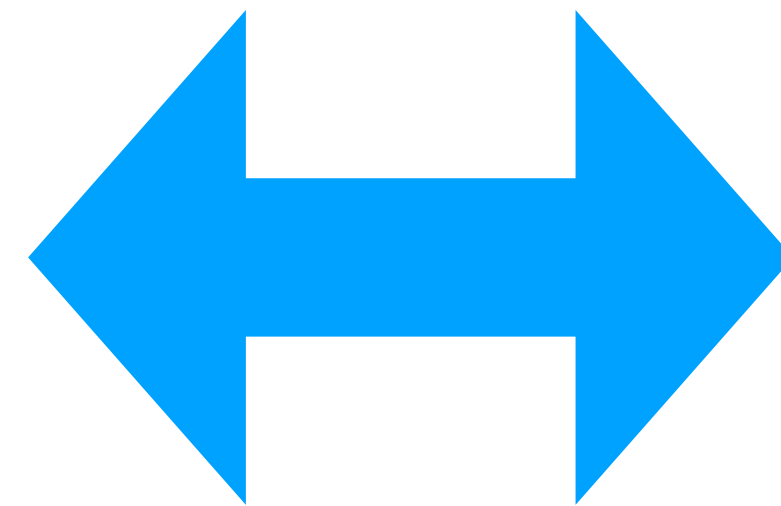
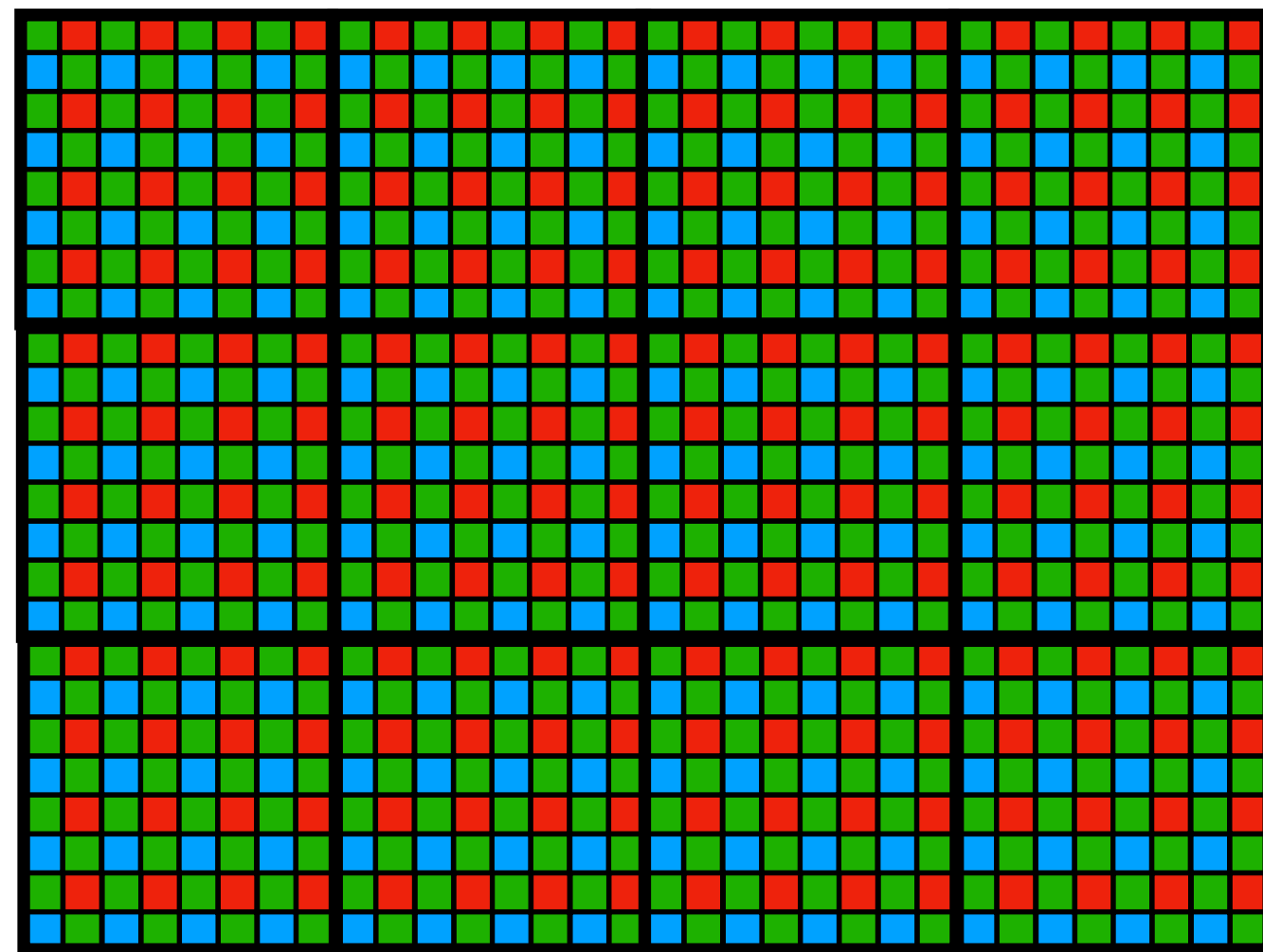
`numpy.histogram()` 函数将输入数组和 `bin` 作为两个参数。 `bin` 数组中的连续元素用作每个 `bin` 的边界

```
matplotlib.pyplot.bar(x, height, width=0.8, bottom=None, *, align='center', data=None, **kwargs)
```

```
import numpy as np
```

```
a = np.array([22,87,5,43,56,73,55,54,11,20,51,5,79,31,27]) ]
np.histogram(a,bins = [0,20,40,60,80,100])
hist,bins = np.histogram(a,bins = [0,20,40,60,80,100])
print hist
print bins
plt.bar(bins, hist)
```

平均值



统计

