OpenCV – Histogram 다루기

김성영교수

금오공과대학교

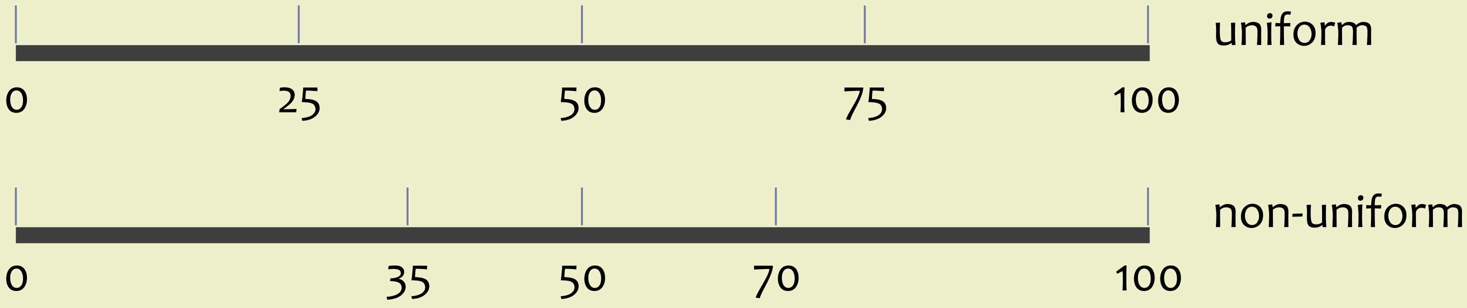
컴퓨터공학과

학습 내용

- calcHist( ) function
- Gray-level Histogram 생성 및 출력
- Color Histogram 생성 및 출력

calcHist( ) function

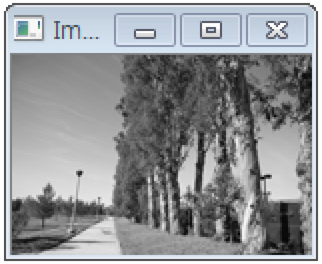
```
void calcHist(
       const Mat*     images,   // Source arrays
       int            nimages,  // Number of source images
       const int*     channels, // the channel used
       InputArray     mask,     // Optional mask
       OutputArray    hist,     // Output histogram
       int            dims,     // dimension
       const int*     histSize, // number of bins
       const float**  ranges,   // pixel value range
       bool           uniform=true,
       bool           accumulate=false
)
```

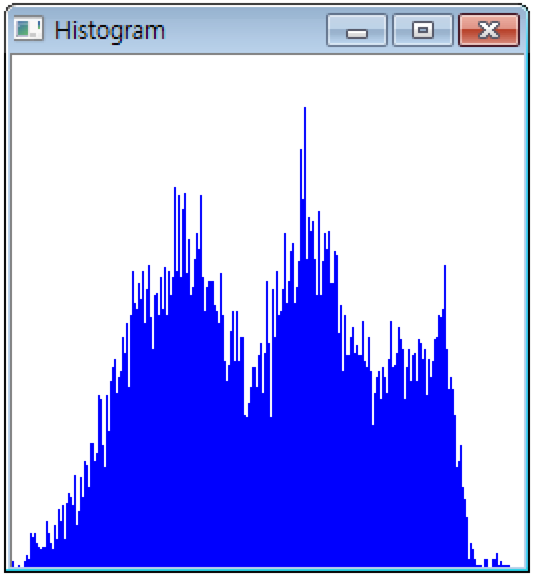


example

```
cvtColor( src, hsv, CV_BGR2HSV );

int hbins = 30, sbins = 32;
int histSize[] = {hbins, sbins};

float hranges[] = { 0, 179 };
float sranges[] = { 0, 255 };
const float* ranges[] = { hranges, sranges };

MatND hist;
int channels[] = {0, 1};

calcHist( &hsv, 1, channels, Mat(),
          hist, 2, histSize, ranges,
          true, false );
```

Gray-level Histogram 생성 및 출력

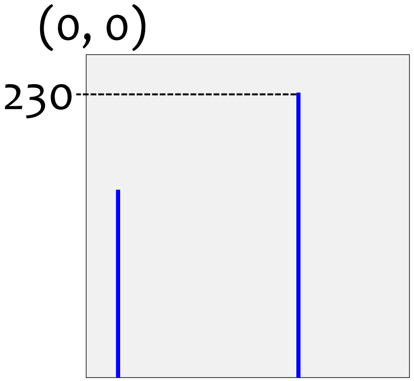




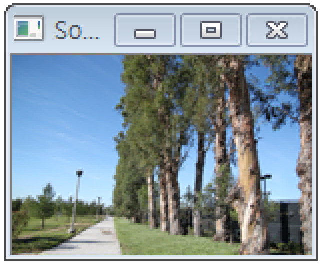
```
#include "opencv2/core/core.hpp"
#include "opencv2/imgproc/imgproc.hpp"
#include "opencv2/highgui/highgui.hpp"

using namespace cv;

MatND getHistogram( const Mat &image, int nbins=256 );
Mat createHistImage( const MatND &hist, int nbins=256 );

int main(void) {
   Mat image = imread( "input.bmp", 0 );
   if( image.data == NULL ) return -1;

   // get histogram
   MatND hist = getHistogram( image );

   // create histogram image
   Mat histImg = createHistImage( hist );

   // Display the images
   namedWindow( "Image" );
   namedWindow( "Histogram" );
   imshow( "Image", image );
   imshow( "Histogram", histImg );

   waitKey();

   return 0;
}
```

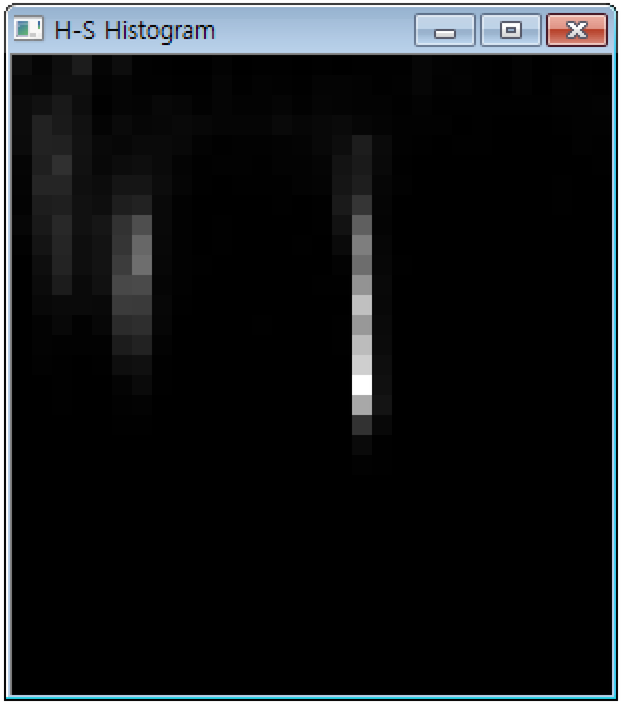
```
MatND getHistogram( const Mat &image, int nbins )
{
   int histSize[1] = { nbins };

   float hranges[2] = { 0, 255 };
   const float* ranges[1] = { hranges };

   cv::MatND hist;
   int channels[1] = { 0 };

   // Compute histogram
   cv::calcHist( &image,
      1,        // histogram of 1 image only
      channels, // the channel used
      cv::Mat(),// no mask is used
      hist,     // the resulting histogram
      1,        // it is a 1D histogram
      histSize, // number of bins
      ranges    // pixel value range
   );

   return hist;
}
```

```
Mat createHistImage( const MatND &hist, int nbins ) {
   double maxVal = 0;
   double minVal = 0;
   minMaxLoc( hist, &minVal, &maxVal, 0, 0 );

   // Image on which to display histogram
   Mat histImg( nbins, nbins,
            CV_8UC3, Scalar::all(255) );

   // set highest point at 90% of nbins
   int hpt = static_cast<int>( 0.9*nbins );

   // Draw vertical line for each bin
   Scalar color = Scalar( 255, 0, 0, 0 ); // Blue
   for( int h = 0; h < nbins; h++ ) {
      float binVal = hist.at<float>( h );
      int intensity = static_cast<int>( binVal*hpt/maxVal );
      line( histImg, Point(h, nbins),
            Point(h, nbins-1-intensity),
            color, 1 );
   }

   return histImg;
}
```



Color Histogram 생성 및 출력





```
#include "opencv2/core/core.hpp"
#include "opencv2/imgproc/imgproc.hpp"
#include "opencv2/highgui/highgui.hpp"

using namespace cv;

MatND getHistogram( const Mat &image, int hbins=30, int sbins=32 );
Mat createHistImage( const MatND &hist, int hbins=30, int sbins=32 );

int main(void) {
   Mat image = imread( "input.bmp", 0 );
   if( image.data == NULL ) return -1;

   Mat hsv;
   cvtColor( image, hsv, CV_BGR2HSV );

   // get histogram
   MatND hist = getHistogram( hsv );

   // create histogram image
   Mat histImg = createHistImage( hist );

   // Display the images
   . . .

   waitKey();

   return 0;
}
```

```
MatND getHistogram( const Mat &image, int hbins, int sbins )
{
   int histSize[] = { hbins, sbins };

   // hue varies from 0 to 179
   float hranges[] = { 0, 179 };
   // saturation varies from 0 to 255
   float sranges[] = { 0, 255 };
   const float* ranges[] = { hranges, sranges };

   MatND hist;
   // compute the histogram from the 0-th and 1-st channels
   int channels[] = { 0, 1 };

   calcHist( &image, 1, channels, Mat(),
        hist, 2, histSize, ranges,
        true, false );

   return hist;
}
```

```
Mat createHistImage( const MatND &hist, int hbins, int sbins )
{
   double maxVal = 0;
   minMaxLoc( hist, 0, &maxVal, 0, 0 );
   int scale = 10;
   Mat histImg = Mat::zeros( sbins*scale, hbins*scale, CV_8U );
   for( int h = 0; h < hbins; h++ )
      for( int s = 0; s < sbins; s++ )
      {
         float binVal = hist.at<float>( h, s );
         int intensity = cvRound( binVal*255/maxVal );
         rectangle( histImg,
            Point(h*scale, s*scale),
            Point( (h+1)*scale-1, (s+1)*scale-1),
            Scalar(intensity),
            CV_FILLED
         );
   }
   return histImg;
}
```

Reference

- R. Laganière, **OpenCV2 Computer Vision: Application Programming Cookbook,** PACKT Publishing, 2011
- G. Bradski and A. Kaebler, **Learning OpenCV: Computer Vision with the OpenCV Library**, O'REILLY, 2008
- http://docs.opencv.org