연구실 세미나 : 프로그래밍 교육 (OpenCV 3)

화면에 형상 그리기
Draw shapes on the screen

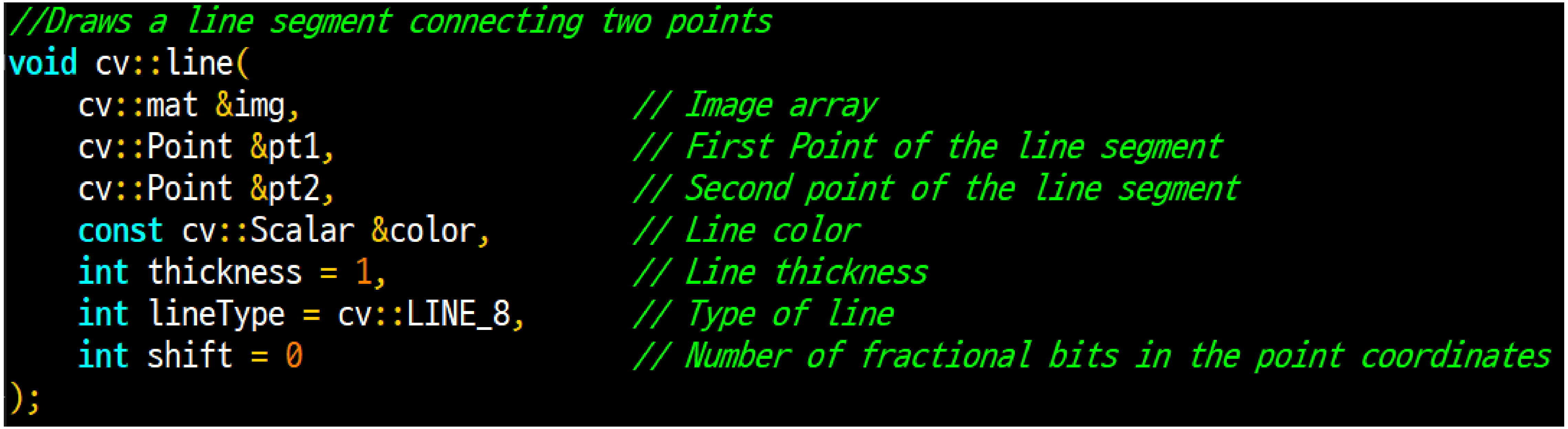
Amylose

DATE : 2019-08-13 TUE

연구실 세미나 : 프로그래밍 교육

Programming Material : OpenCV Draw shapes on the screen

- Drawing Functions

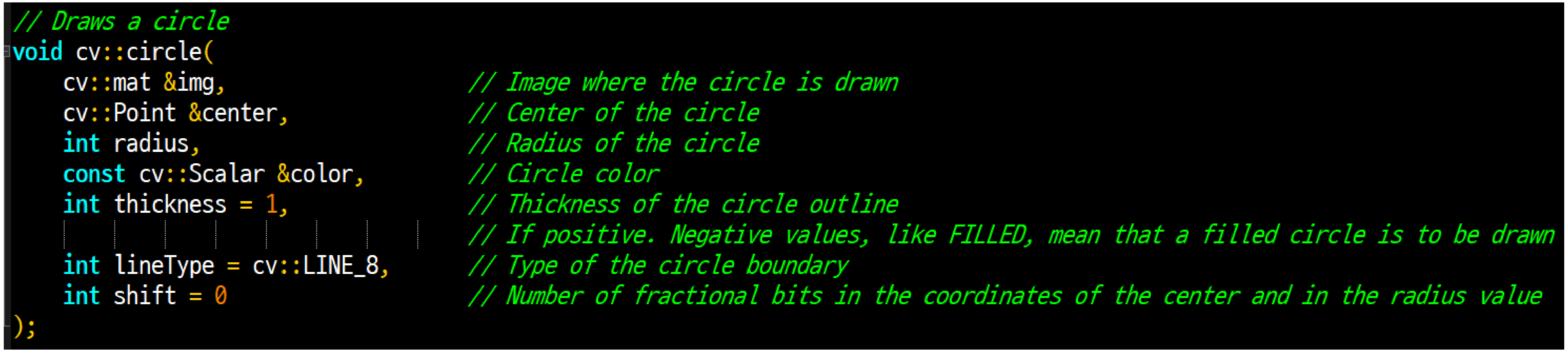


```
//Draws a line segment connecting two points
void cv::line(
    cv::mat &img,                   // Image array
    cv::Point &pt1,                 // First Point of the line segment
    cv::Point &pt2,                 // Second point of the line segment
    const cv::Scalar &color,        // Line color
    int thickness = 1,              // Line thickness
    int lineType = cv::LINE_8,      // Type of line
    int shift = 0                   // Number of fractional bits in the point coordinates
);
```

➢ The function line draws the line segment between pt1 and pt2 points in the image. The line is clipped by the image boundaries. For non-antialiased lines with integer coordinates, the 8-connected or 4-connected Bresenham algorithm is used. Thick lines are drawn with rounding endings. Antialiased lines are drawn using Gaussian filtering.

Programming Material : OpenCV Draw shapes on the screen

● Drawing Functions



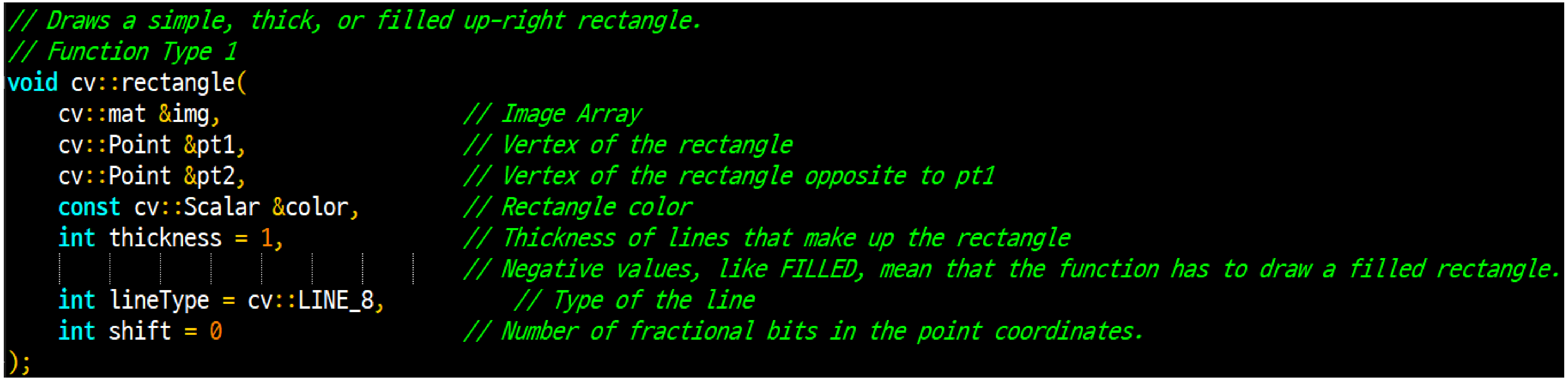
```
// Draws a circle
void cv::circle(
    cv::mat &img,                   // Image where the circle is drawn
    cv::Point &center,              // Center of the circle
    int radius,                     // Radius of the circle
    const cv::Scalar &color,        // Circle color
    int thickness = 1,              // Thickness of the circle outline
                                    // If positive. Negative values, like FILLED, mean that a filled circle is to be drawn
    int lineType = cv::LINE_8,      // Type of the circle boundary
    int shift = 0                   // Number of fractional bits in the coordinates of the center and in the radius value
);
```

➢ The function cv::circle draws a simple or filled circle with a given center and radius.

연구실 세미나 : 프로그래밍 교육

Programming Material : OpenCV Draw shapes on the screen

● Drawing Functions



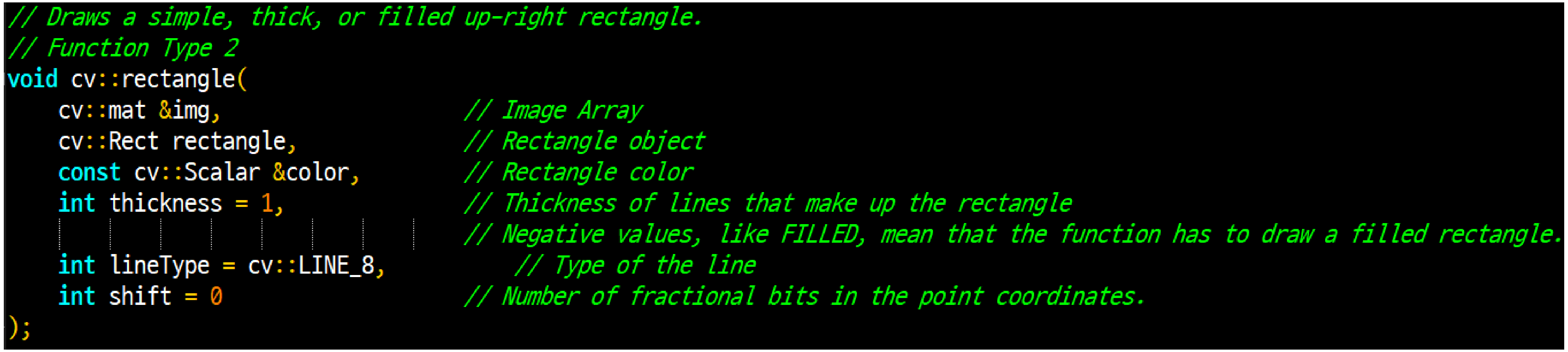
```
// Draws a simple, thick, or filled up-right rectangle.
// Function Type 1
void cv::rectangle(
    cv::mat &img,                   // Image Array
    cv::Point &pt1,                 // Vertex of the rectangle
    cv::Point &pt2,                 // Vertex of the rectangle opposite to pt1
    const cv::Scalar &color,        // Rectangle color
    int thickness = 1,              // Thickness of lines that make up the rectangle
                                    // Negative values, like FILLED, mean that the function has to draw a filled rectangle.
    int lineType = cv::LINE_8,          // Type of the line
    int shift = 0                   // Number of fractional bits in the point coordinates.
);
```

➢ The function cv::rectangle draws a rectangle outline or a filled rectangle whose two opposite corners are pt1 and pt2.

연구실 세미나 : 프로그래밍 교육

Programming Material : OpenCV Draw shapes on the screen

- Drawing Functions



```
// Draws a simple, thick, or filled up-right rectangle.
// Function Type 2
void cv::rectangle(
    cv::mat &img,                   // Image Array
    cv::Rect rectangle,             // Rectangle object
    const cv::Scalar &color,        // Rectangle color
    int thickness = 1,              // Thickness of lines that make up the rectangle
                                    // Negative values, like FILLED, mean that the function has to draw a filled rectangle.
    int lineType = cv::LINE_8,          // Type of the line
    int shift = 0                   // Number of fractional bits in the point coordinates.
);
```

➢ This is an overloaded member function, provided for convenience. It differs from the above function only in what argument(s) it accepts.

Programming Material : OpenCV Draw shapes on the screen

● Drawing Functions (con’t)

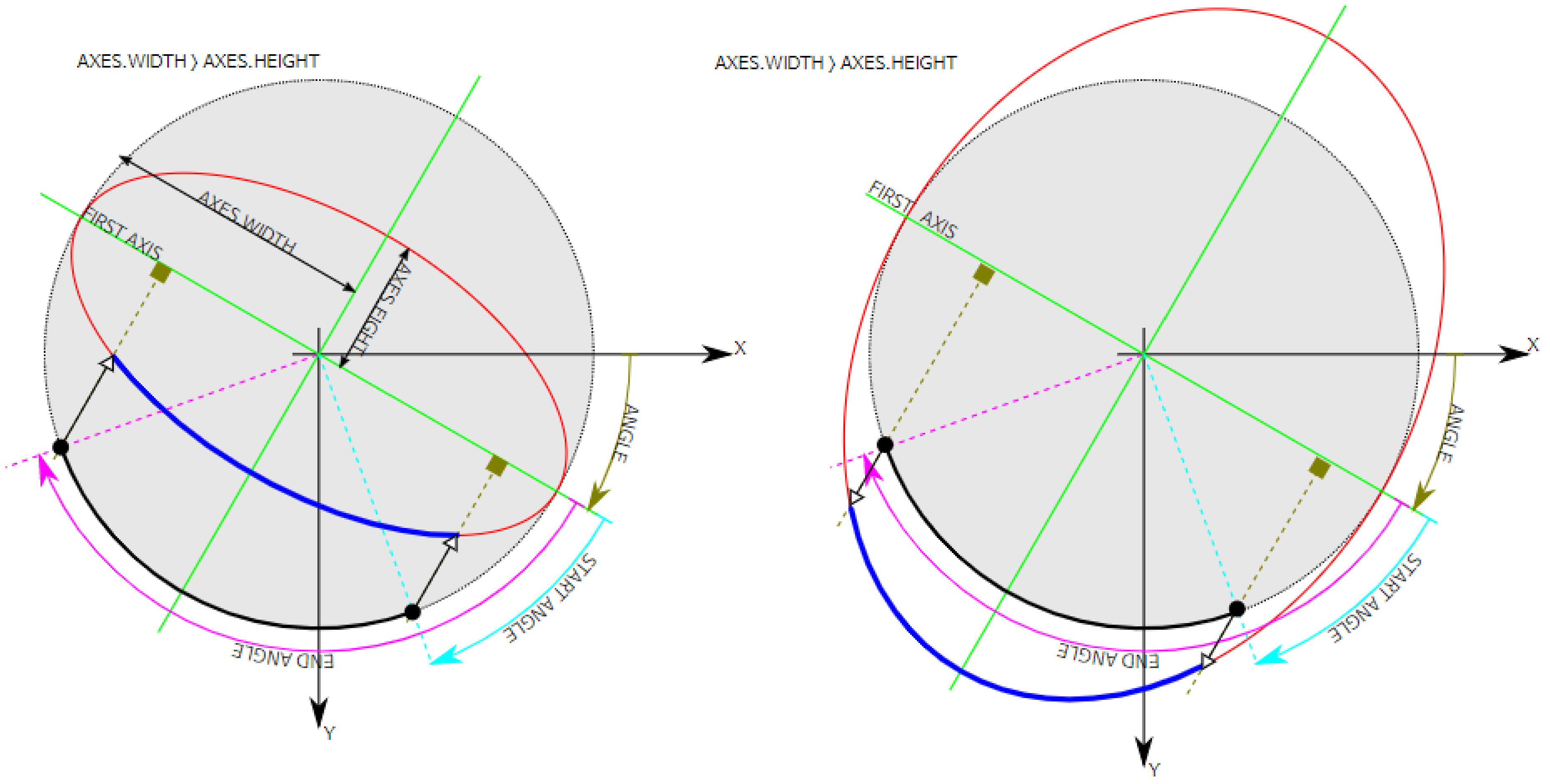
```
// Draws a simple or thick elliptic arc or fills an ellipse sector.
// Function Type 1
void cv::ellipse(
    cv::mat &img,                   // Image
    cv::Point position,             // Center of the ellipse
    cv::Size axes,                  // Half of the size of the ellipse main axes
    double angle,                   // Ellipse rotation angle in degrees
    double startAngle,              // Starting angle of the elliptic arc in degrees
    double endAngle,                // Ending angle of the elliptic arc in degrees
    const cv::Scalar &color,        // Ellipse color
    int thickness = 1,              // Thickness of the ellipse arc outline
                                    // If positive. Otherwise, this indicates that a filled ellipse sector is to be drawn
    int linetype = cv::LINE_8,      // Type of the ellipse boundary
    int shift = 0                   // Number of fractional bits in the coordinates of the center and values of axes
);
```

➢ The function cv::ellipse with more parameters draws an ellipse outline, a filled ellipse, an elliptic arc, or a filled ellipse sector. The drawing code uses general parametric form. A piecewise-linear curve is used to approximate the elliptic arc boundary.

연구실 세미나 : 프로그래밍 교육

Programming Material : OpenCV Draw shapes on the screen

● Drawing Functions

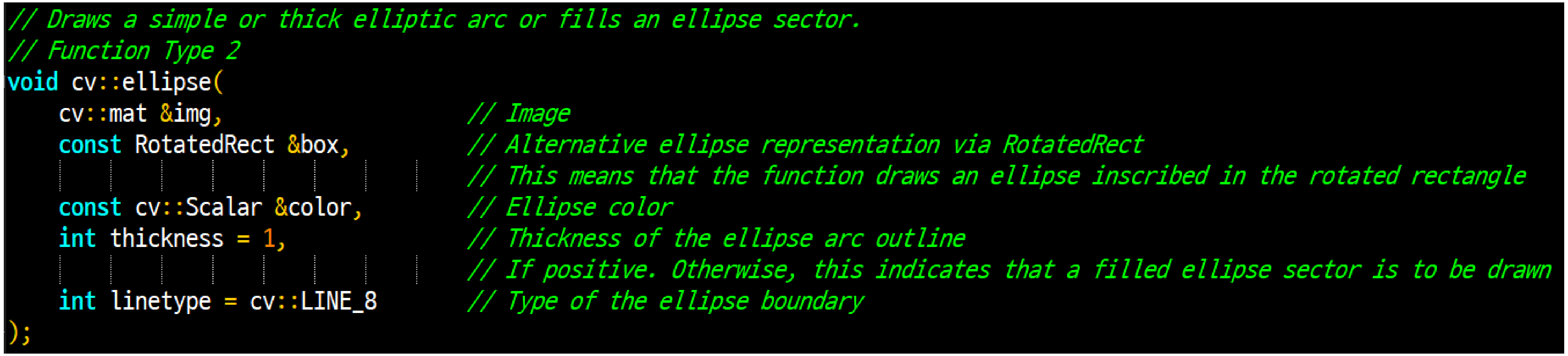


Parameters of Elliptic Arc

연구실 세미나 : 프로그래밍 교육

Programming Material : OpenCV Draw shapes on the screen

- Drawing Functions



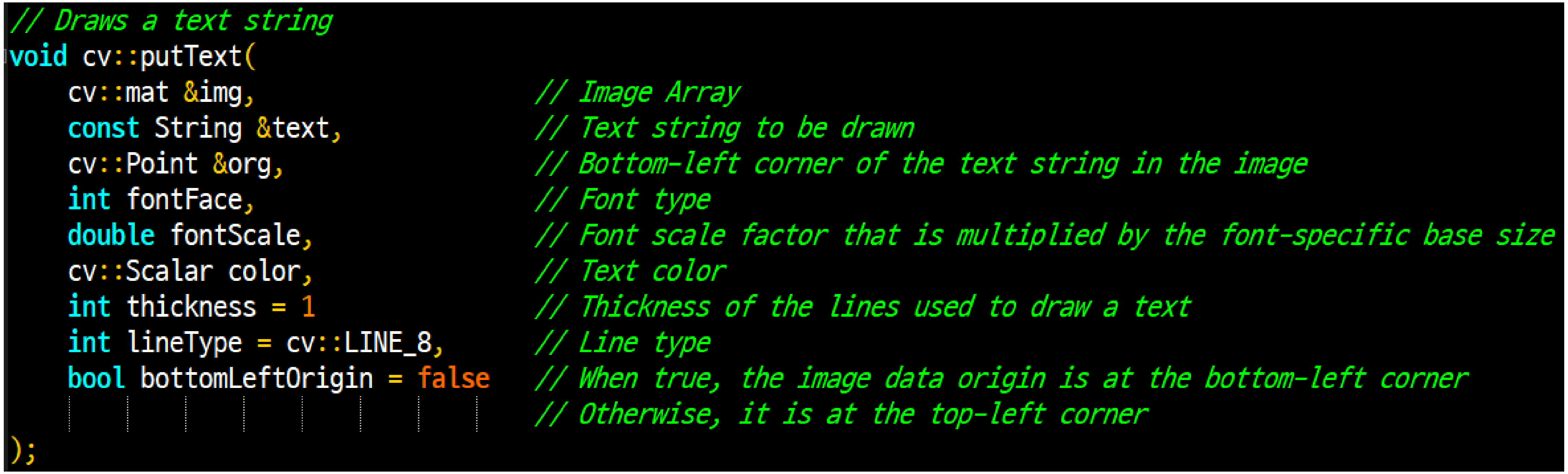
```
// Draws a simple or thick elliptic arc or fills an ellipse sector.
// Function Type 2
void cv::ellipse(
    cv::mat &img,                   // Image
    const RotatedRect &box,         // Alternative ellipse representation via RotatedRect
                                    // This means that the function draws an ellipse inscribed in the rotated rectangle
    const cv::Scalar &color,        // Ellipse color
    int thickness = 1,              // Thickness of the ellipse arc outline
                                    // If positive. Otherwise, this indicates that a filled ellipse sector is to be drawn
    int linetype = cv::LINE_8       // Type of the ellipse boundary
);
```

➢ This is an overloaded member function, provided for convenience. It differs from the above function only in what argument(s) it accepts.

연구실 세미나 : 프로그래밍 교육

Programming Material : OpenCV Draw shapes on the screen

● Drawing Functions



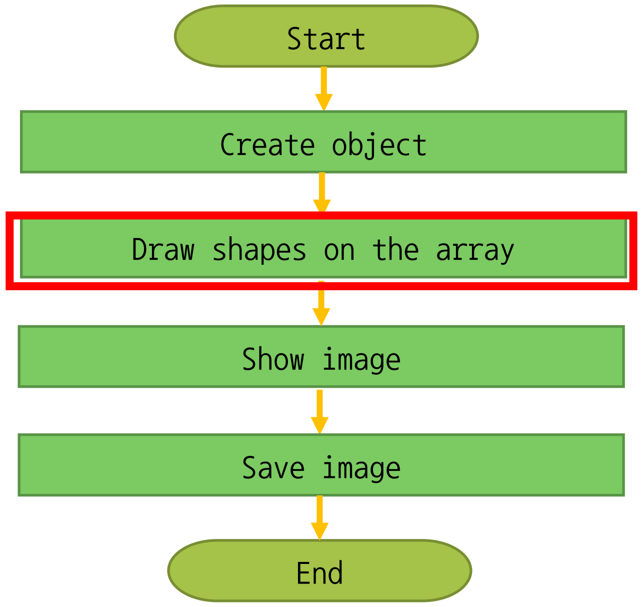
```
// Draws a text string
void cv::putText(
    cv::mat &img,                    // Image Array
    const String &text,              // Text string to be drawn
    cv::Point &org,                  // Bottom-left corner of the text string in the image
    int fontFace,                    // Font type
    double fontScale,                // Font scale factor that is multiplied by the font-specific base size
    cv::Scalar color,                // Text color
    int thickness = 1                // Thickness of the lines used to draw a text
    int lineType = cv::LINE_8,       // Line type
    bool bottomLeftOrigin = false    // When true, the image data origin is at the bottom-left corner
                                     // Otherwise, it is at the top-left corner
);
```

➢ The function cv::putText renders the specified text string in the image. Symbols that cannot be rendered using the specified font are replaced by question marks. See getTextSize for a text rendering code example.

연구실 세미나 : 프로그래밍 교육

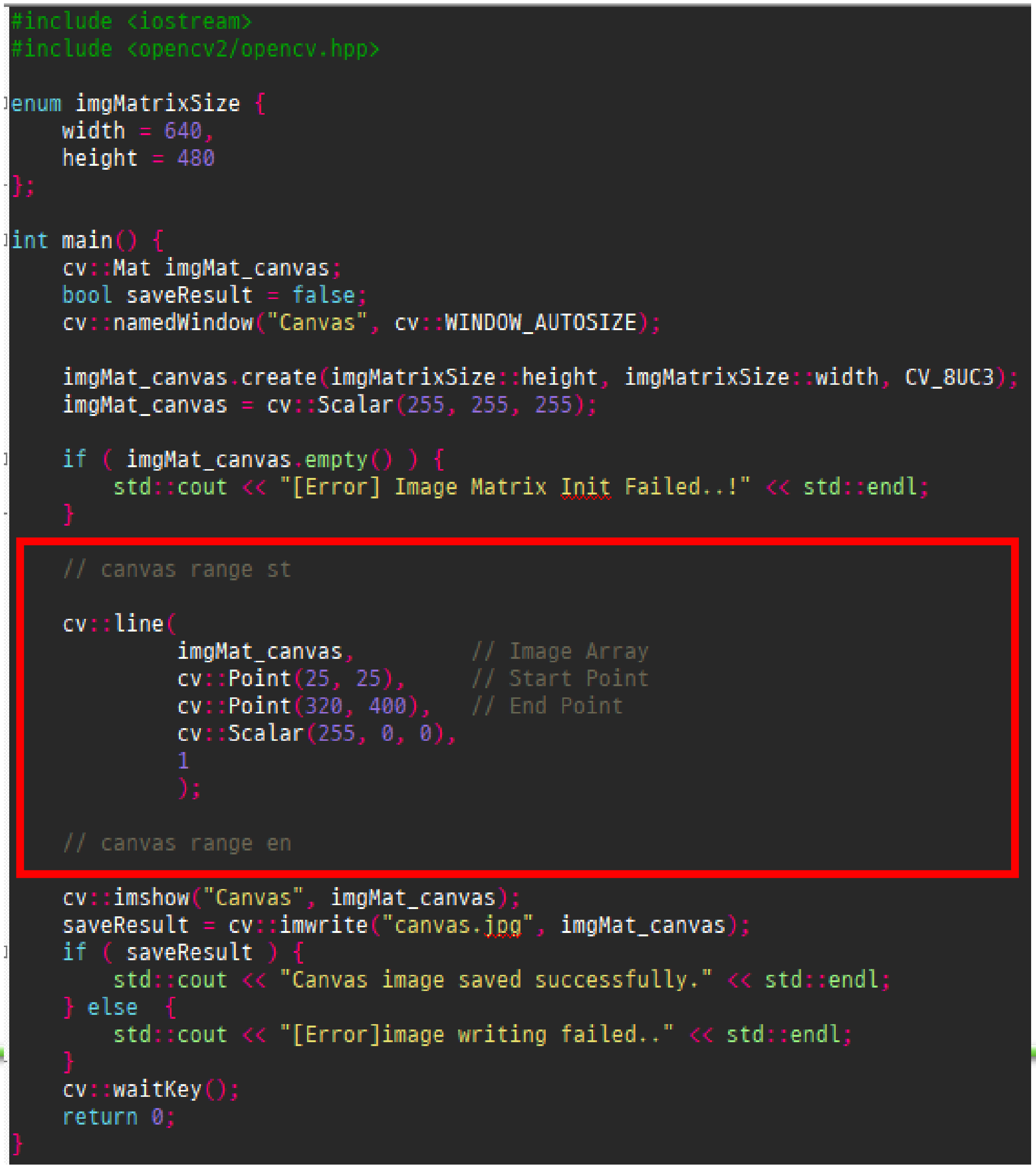
Programming Material : OpenCV Draw shapes on the screen

- Examples of use : Drawing Function
  - Programming Environment
    - Used GCC C++, and OpenCV only. (in Windows)
    - Load image file and converse color model.
- Flow Chart

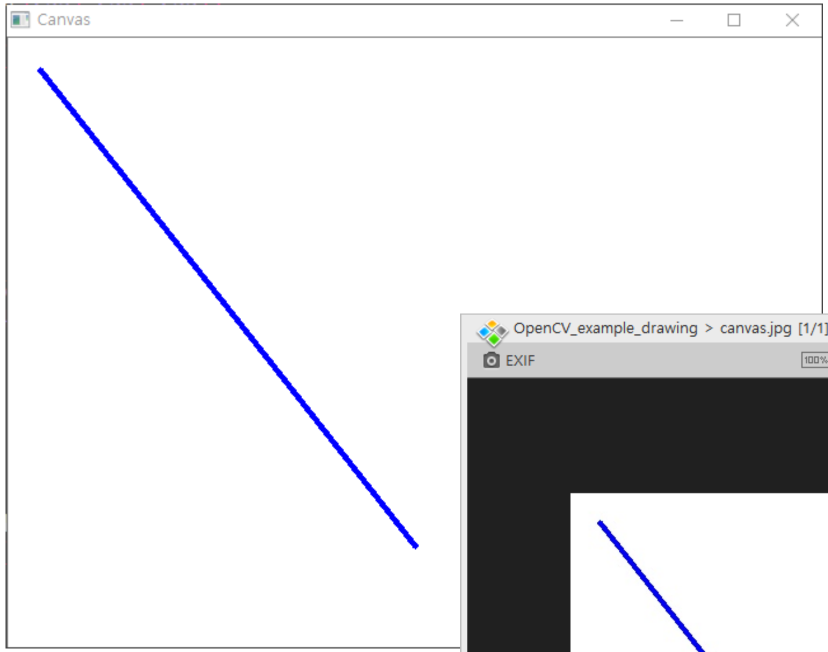


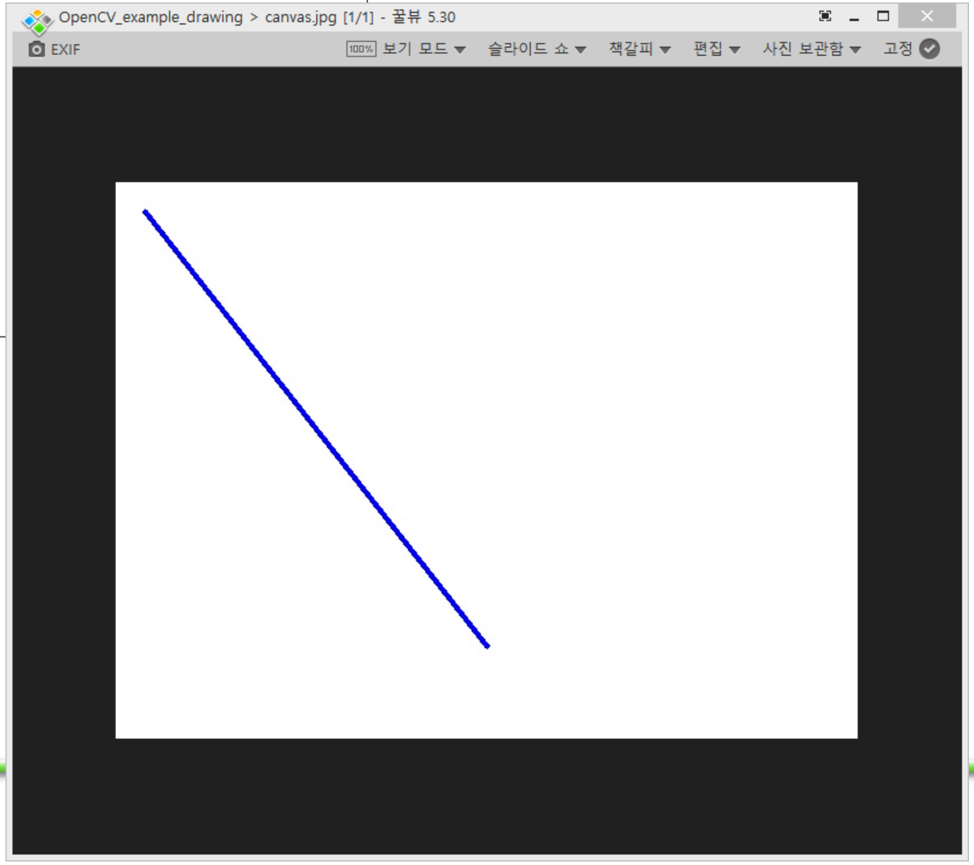
연구실 세미나 : 프로그래밍 교육

Programming Material : OpenCV #2 : Color Model Conversion & Color Filter



```cpp
#include <iostream>
#include <opencv2/opencv.hpp>

enum imgMatrixSize {
    width = 640,
    height = 480
};

int main() {
    cv::Mat imgMat_canvas;
    bool saveResult = false;
    cv::namedWindow("Canvas", cv::WINDOW_AUTOSIZE);

    imgMat_canvas.create(imgMatrixSize::height, imgMatrixSize::width, CV_8UC3);
    imgMat_canvas = cv::Scalar(255, 255, 255);

    if ( imgMat_canvas.empty() ) {
        std::cout << "[Error] Image Matrix Init Failed..!" << std::endl;
    }

    // canvas range st

    cv::line(
             imgMat_canvas,          // Image Array
             cv::Point(25, 25),      // Start Point
             cv::Point(320, 400),    // End Point
             cv::Scalar(255, 0, 0),
             1
             );

    // canvas range en

    cv::imshow("Canvas", imgMat_canvas);
    saveResult = cv::imwrite("canvas.jpg", imgMat_canvas);
    if ( saveResult ) {
        std::cout << "Canvas image saved successfully." << std::endl;
    } else  {
        std::cout << "[Error]image writing failed.." << std::endl;
    }
    cv::waitKey();
    return 0;
}
```




OpenCV_example_drawing > canvas.jpg [1/1] - 꿀뷰 5.30
EXIF
100%
보기 모드
슬라이드 쇼
책갈피
편집
사진 보관함
고정