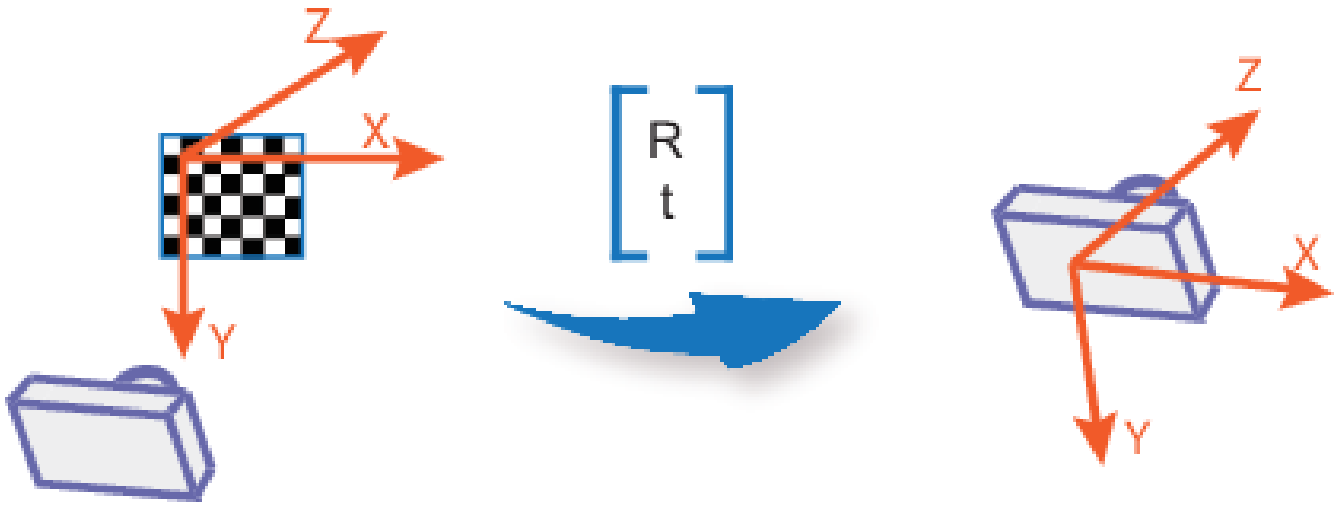
Camera Calibration

Sung Soo Hwang

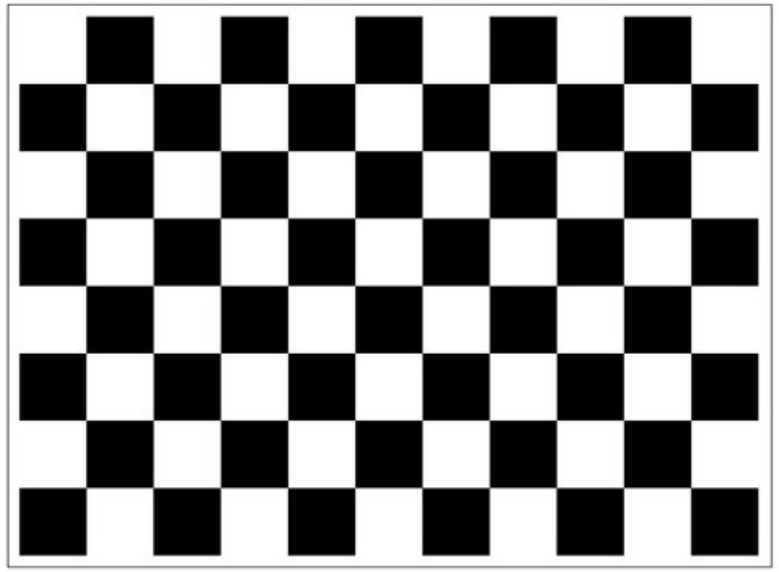
Camera Calibration

- Camera parameters
  - Intrinsic parameter
    - Principal point(u0, v0), focal length, skew parameter, pixel size(sx, sy)
  - Extrinsic parameter
    - Rotation & translation of a camera center



Camera Calibration

- The most popular way of camera calibration is to use chessboard



- You should know the actual size of rectangles in a chessboard

Camera Calibration

- Example code

```
#include <opencv/cv.h>
#include <opencv/highgui.h>
#include<opencv2/opencv.hpp>
#include<iostream>

using namespace cv;
using namespace std;

int main(){
    // open vidoe webcam
    VideoCapture cap = VideoCapture(0);
    int successes = 0;

    int numBoards = 70; // total num of corners
    int numCornersHor = 10; // num of horizon corners
    int numCornersVer = 7; // num of vertical corners
    int Rect_size = 20; // length of one side of the rectangle

    int numSquares = (numCornersHor - 1) * (numCornersVer - 1);
    Size board_sz = Size(numCornersHor, numCornersVer);

    // Container
    vector<vector<Point3f> > object_points;
    vector<vector<Point2f> > image_points;
    vector<Point2f> corners;
    vector<Point3f> obj;
```

Camera Calibration

- Example code

```
for (int i = 0; i < numCornersHor; i++){
      for (int j = 0; j < numCornersVer; j++){
          obj.push_back(Point3f(i*Rect_size, j*Rect_size, 0.0f));
      }


  }
  Mat img;
  Mat gray;
  cap >> img;
  cout << "Image size:"<<img.size() << endl;
  while (successes < numBoards)
  {
      cap >> img;
      cvtColor(img, gray, CV_RGB2GRAY);
      if (img.empty()) break; // end of video stream
      if (waitKey(1) == 27) break; // stop capturing by pressing ESC


      // Finds the positions of internal corners of the chessboard.
      bool found = findChessboardCorners( gray, // Source chessboard view. It must be an 8-bit grayscale or color image.
                                          board_sz, // Number of inner corners per a chessboard row and column
                                          corners, // Output array of detected corners.
                                          CALIB_CB_ADAPTIVE_THRESH
                                          // Various operation flags that can be zero or a combination
                                          // Use adaptive thresholding to convert the image to black and white, rather than a fixed threshold level
                                          );
```

Camera Calibration

- Example code

```
if (found == 1){
        // Refines the corner locations.
        cornerSubPix(gray, // Input single-channel, 8-bit or float image.
                corners, // Initial coordinates of the input corners and refined coordinates provided for output.
                Size(11, 11), // Half of the side length of the search window
                Size(-1, -1), // Half of the size of the dead region in the middle of the search. zone over which the summation in the formula below is not done

                // The class defining termination criteria for iterative algorithms
                TermCriteria(CV_TERMCRIT_EPS | CV_TERMCRIT_ITER, // The type of termination criteria, one of TermCriteria::Type
                        30, // The maximum number of iterations or elements to compute.
                        0.1 // The desired accuracy or change in parameters at which the iterative algorithm stops.
                        )
                );
        // Renders the detected chessboard corners.
        drawChessboardCorners(  img, // Destination image. It must be an 8-bit color image.
                        board_sz, // Number of inner corners per a chessboard row and column
                        corners, // Array of detected corners, the output of findChessboardCorners.
                        found // Parameter indicating whether the complete board was found or not
                            // The return value of findChessboardCorners should be passed here.
                    );
        image_points.push_back(corners);
        object_points.push_back(obj);
        printf("Snap stored!\n");
        successes++;
}
```

Camera Calibration

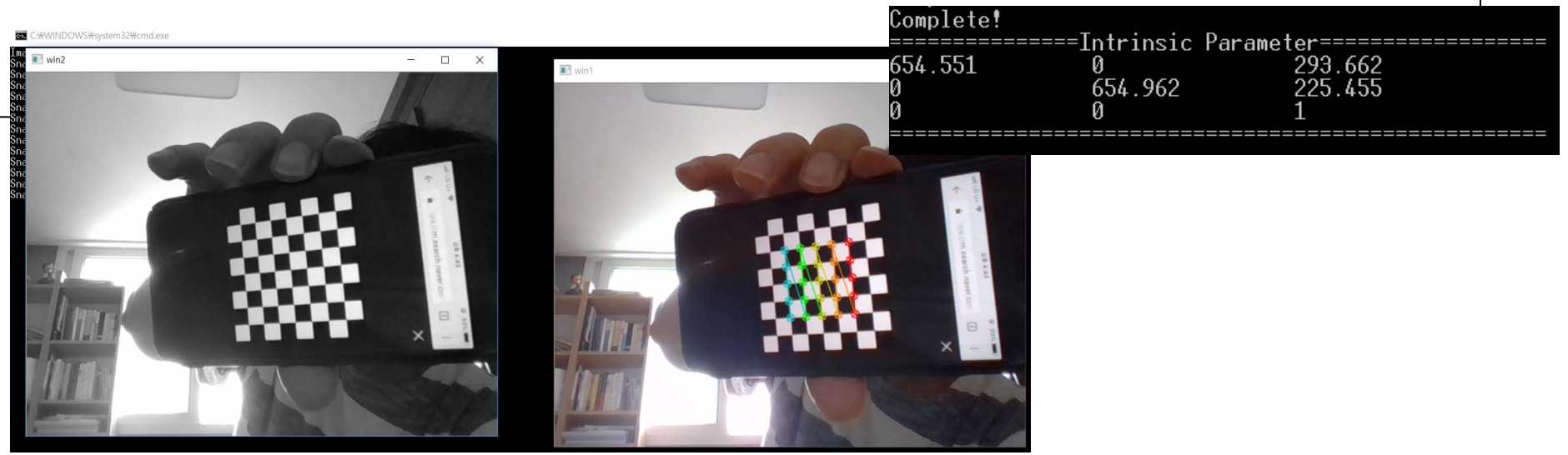
- Example code

```
imshow("win1", img);
        imshow("win2", gray);
        waitKey(1000);
    }


    cout << "Complete!" << "\n";

    Mat intrinsic = Mat(3, 3, CV_32FC1);
    Mat distCoeffs;
    vector<Mat> rvecs;
    vector<Mat> tvecs;

    intrinsic.ptr<float>(0)[0] = 1;
    intrinsic.ptr<float>(1)[1] = 1;

    // Finds the camera intrinsic and extrinsic parameters from several views of a calibration pattern
    calibrateCamera( object_points, // The outer vector contains as many elements as the number of the pattern views.
                image_points, // A vector of vectors of the 2D image points.
                img.size(), // Size of the image
                intrinsic, // Intrinsic camera matrix
                distCoeffs, // Lens distortion coefficients. These coefficients will be explained in a future post.
                rvecs, // Rotation specified as a 3×1 vector.
                    // The direction of the vector specifies the axis of rotation and the magnitude of the vector specifies the angle of rotation.
                tvecs // 3×1 Translation vector.
                );
```

Camera Calibration

- Example code

```
    Mat imageUndistorted;
    // Print result
    cout << "==================Intrinsic Parameter====================" << "\n";
    for (int i = 0; i < intrinsic.rows; i++){
        for (int j = 0; j < intrinsic.cols; j++) {
            cout << intrinsic.at<double>(i, j)<<"\t\t";
        }
        cout << endl;
    }
    cout << "=========================================================" << "\n";
    cap.release();
    waitKey();
    return 0;
}
```



Camera Calibration

- Camera calibration by GML Toolbox[only window os]

- How to use

1. Create a project by entering the number of chessboards to be used for calibration and the number of horizontal, vertical, and actual length of each chessboard.
2. Click ‘Add Image’ to upload a picture of the chessboard from various angles.
3. Press the ‘Detect All’ button to detect the chessboard portion of each photo and calculate the calibration matrix value (undetected photos are shown in orange)
4. If a chessboard is detected in more than a certain number of pictures, the ‘Calibration’ button is clickable
5. Press the ‘Calibration’ button and enter ‘Result’ to check the result.

Camera Calibration

- Camera calibration by GML Toolbox